Sistemas de Aprendizagem

Ricardo Ferreira[A82568] Hugo Faria[A81283]
Rodolfo Silva[A81716] Bruno Veloso[A78352]

Universidade do Minho, Departamento de Informática

Resumo

Quando pretendemos elaborar um sistema inteligente, revela-se essencial escolher um sistema de aprendizagem adequado aos objectivos a atingir. Com o objectivo de efectuar uma escolha ponderada, este trabalho pretende explorar três sistemas de aprendizagem existentes no mercado. No nosso caso iremos abordar a aprendizagem em *Case Based Reasoning*, *Artificial Neural Networks* e *Support Vector Machines*.

Keywords: Case Based Reasoning · Artificial Neural Networks · Support Vector Machines.

1 Introdução

Este artigo é o resultado de investigação e consolidação do funcionamento dos sistemas de aprendizagem por Raciocínio Baseado em Casos, Redes Neuronais Artificias e Máquinas de Vector de Suporte no âmbito da unidade curricular de Aprendizagem e Extracção de Conhecimento do perfil de especialização de Sistemas Inteligentes do Mestrado Integrado em Engenharia Informática.

Para cada um dos sistemas de aprendizagem, será realizada uma descrição geral do seu funcionamento e da sua capacidade de aprendizagem.

Além disso serão ainda apresentadas algumas ferramentas de desenvolvimento existentes para cada modelo e ainda alguns exemplos de casos reais na sua utilização.

2 Case Based Reasoning

2.1 Descrição

No nosso quotidiano deparamos-nos constantemente com problemas dos quais necessitam de uma solução. Muitas vezes, as soluções encontradas para resolver tais problemas são provenientes de situações ou experiências vividas e das quais adquirimos conhecimento suficiente que possa ser usado para a resolução destes problemas. A partir daqui iremos usar o nome em português para Case Base Reasoning que é Raciocínio Baseado em Casos e por sua vez a nomenclatura RBC.

O raciocínio baseado em casos baseia-se neste princípio em que, através de uma análise do problema é possível encontrar na base de casos um ou mais casos similares (que já tenham sido experienciados), adaptando-se a solução deste caso similar ao nosso problema de forma a resolvê-lo. Após ser resolvido, actualiza-se a base de casos para ser utilizado futuramente.

O sistema não só guarda soluções que tiveram sucesso mas também situações que falharam, de forma a poder aprender com os erros e evitar, no futuro, que tais erros se voltem a repetir.

Verificando um problema a resolver é possível passar ao uso de RBC, em que a descrição inicial do problema será considerado o caso. Para isto segue-se o modelo seguinte(1):

- **Recuperar** os casos que sejam parecidos com o caso que pretendemos resolver, estes casos podem ser considerados similares tendo em conta preferências heurísticas(o que nem sempre leva ao caso mais similar com o novo caso) ou através do cálculo de uma métrica;
- **Adaptação**, que verifica o conhecimento e informação obtida em casos semelhantes ao que queremos resolver, e tenta adaptar a solução ao nosso problema de forma a tentar resolver o mesmo;
- **Revisão e Reparação**, em que revisão é a parte em que é testada a solução proposta no nosso caso a solucionar, reparando se necessário a solução ao problema;

– **Aprender**, em que no decorrer deste processo é actualizada a base de casos com o nosso caso para ser utilizado futuramente caso o caso seja necessário.

Definição de Caso e tipos de representação

Um caso é um triplo `<D,S,J>` em que, D é a descrição do problema, S é a solução e J é a justificação para a solução do problema. São eventos que ocorreram no passado e que foram resolvidos. Existem dois tipos de representação de um caso(1):

– **Representação não hierarquizada:** não permite partição de casos e tem problemas em lidar com informação fraca (desconhecida, incompleta, etc);

– **Representação estruturada:** permite a partição dos casos em características conseguindo assim dividir a informação sobre o caso.

Repositório de casos e seus diferentes modelos

Um Repositório de casos (também chamado de memória de casos ou base de casos) é onde estão guardados todos os casos experienciados até à data. RBC requer um acesso rápido a estes casos e para isso existem modelos de manuseamento destes no qual falaremos de três(1):

– **Modelo de memória dinâmica:** representa uma estrutura hierárquica na qual os casos estão organizados pelas propriedades que têm em comum criando assim episódios generalizados (os nodos destes podem ser outros episódios generalizados ou casos com características em comum). Os índices são termos que se referem a propriedades que distinguem os casos na estrutura, podendo indicar um episódio generalizado ou diretamente um caso, e são compostos por um par (atributo, valor)(1).

– **Modelo baseado em exemplos e categorias:** representa uma rede semântica que interliga categorias ou conjunto de casos similares. Neste modelo os casos são considerados exemplos. Para indexação neste modelo existem 3 tipos, que são(1):
  - tipo associativo, que faz ligação entre o descritor do problema e um exemplo ou categoria;
  - tipo protótipo, que faz ligação entre descritor do problema, uma categoria e dentro desta se liga também aos casos escolhidos;
  - tipo diferenças, que faz ligação entre o descritor do problema e o/s vizinho/s (exemplos) mais próximo/s.

– **Modelo não hierarquizado:** não existe qualquer tipo de indexação, os casos são guardados sequencialmente(1).

Vantagens e Desvantagens

Vantagens

A utilização de um sistema RBC tem várias vantagens :

- casos são o conhecimento, o que faz com que a manutenção e actualização do conhecimento seja feita automaticamente;

- este sistema permite obter soluções para problemas usando conhecimento que não está completamente definido, pouco estruturado ou desconhecido, porém não lida tão bem com estas situações (neste aspeto redes neuronais artificiais são melhores)

- Filtração de características pertinentes ao caso, isto é, permite que possamos utilizar aspectos ou características do problema que consideramos pertinentes para obter a sua solução;

- Aprendizagem com o erro, como guarda soluções falhadas impede também que se repitam os mesmo erros cometidos em casos semelhantes anteriores.

Desvantagens

Porém também tem as suas desvantagens, entre as quais destacam-se:

- Necessidade de um elevado nível de armazenamento para guardar todos os casos, o que leva a um elevado nível de investimento;

- Dificuldade de adaptar uma solução ao caso é grande e não há garantia de sucesso;

- Possibilidade de criação de soluções para os problemas sem a garantia que a solução seja a melhor solução possível

2.2 Capacidade de Aprendizagem

Como anteriormente explicado, RBC adquire conhecimento tendo por base experiências passadas.

O processo de aprendizagem tem alguns passos que devem ser considerados, começando pela informação que deve ser seleccionada do caso, a que deve ser retida, de que forma deve ser retida e como será guardada na base de casos. Completando estes passos, o sistema retém mais conhecimento, aumentando assim a sua base de casos.

Estes passos são, *seleção de informação, seleção de índices (indexação) e integração do caso na base de casos*(1):

- **Seleção de informação**: se o problema for resolvido baseando-se num caso passado, então é possível generalizar esta situação ou criar um caso novo. Se forem utilizados outros métodos para a resolução do caso (que não um caso passado), ou existiu intervenção do utilizador então deve ser adicionado um novo caso à base de casos(1);

- **Seleção de índices (indexação)**: para atualizar a base de casos é necessário existir indexação. Para isto são usados um conjunto de regras chamadas regras de indexação apresentadas de seguida. As descrições (descritores) do novo caso são tratados como índices, destes descritores escolhemos as mais relevantes, no final utilizam-se os descritores que consigam distinguir os casos dependendo do modelo utilizado(1);

- **Integração do caso na base de casos**: existem bases de casos não hierarquizados e outros tipos de organização não tão complexos. Se a base de casos for não hierarquizado é possível simplesmente adicionar o novo caso uma vez que se adiciona de forma sequencial, senão a estrutura tem de ser atualizada, baseando-se no índice do caso a inserir(1).

2.3 Ferramentas de Desenvolvimento

Para desenvolver metodologias de RBC existem três tipos de ferramentas.Temos as shells, os ambientes de desenvolvimento e frameworks(2).

As shells são geradores de aplicações com interface gráfica. Estas podem ser usadas por pessoas não ligadas ao mundo da programação mas a extensão ou integração de novos componentes no seu desenvolvimento não é possível uma vez que só servem para desenvolver aplicações de RBC num domínio específico. Ao contrário das shells, os ambientes de desenvolvimento permite uma implementação de metodologias de RBC num domínio que queiramos de forma a resolver o problema.

Muitas shells ainda se encontram no mercado. Apresentando de seguida dois exemplos:

AIAI CBR Shell

Tem características como: K vizinho mais próximo, múltiplos algoritmos de diagnóstico, os pesos podem ser definidos manualmente ou otimizados através de um algoritmo genérico(3).

myCBR

Desenvolvido pela "German Research Center for Artificial Intelligence" em parceria com "University of West London". O objetivo deste é minimizar o esforço para construir aplicações RBC que precisam de alta intensidade cognitiva para saber se casos são similares. Tem extensões para trabalhar com representações de casos orientados a objetos estruturados(2).

CASPIAN

É um ambiente de desenvolvimento que serve como uma componente RBC do sistema Wayland, desenvolvido pela Universidade Aberystwyth, em País de Gales. Tem uma interface de linha de comandos simples e que pode ser integrada com uma GUI caso preciso. Tem características como: usar o vizinho mais próximo e usa regras para adaptação dos casos(5).

jCOLIBRI

Uma das frameworks usadas em RBC é jCOLIBRI. Esta é uma framework orientada a objetos que facilita a construção de sistemas RBC, que foi desenvolvida por GAIA(2).

2.4 Soluções de Mercado

Em termos de aplicabilidade, existem alguns campos no qual o RBC é usado por forma a resolver problemas propostos como a área do diagnóstico, restauração, contratos, etc. Ferramentas ainda usadas temos:

DOCSIM
O DOCSIM, Prediagnosis Doctor Simulation Using Case-Based Techniques, é um software utilizado para pré-diagnósticos realizados por médicos. O objetivo é auxiliar esses médicos na recuperação de casos de tratamentos bem sucedidos anteriormente, guardando os novos casos para futuras utilizações(6).

CBR-Answers
É um sistema usado para suporte por telefone e por outras aplicações help-desk. Quando há um problema, este analisa documentos numa base FAQ de forma a encontrar um caso parecido e dá ao cliente em forma textual uma possível solução(6).

3 Artificial Neural Networks

3.1 Descrição

Uma das maiores ferramentas usadas em *Machine Learning* são as redes neuronais artificiais (RNA). RNA são modelos computacionais inspirados no cérebro humano que tentam replicar a maneira como um ser humano aprende/pensa. Estas costumam ser apresentadas como sistemas de "neurónios interligados" distribuídos em diferentes camadas(7).

RNA são modelos idealmente utilizados para encontrar padrões demasiado complexos, em que um ser humano não conseguiria extrair tal informação de forma a ensinar à máquina como o reconhecer. Existem então 3 camadas principais associadas ao modelo(7):

– **input**, que recebe a informação que processar;
– **output**, que nos dá e estabelece o resultado final;
– **camadas escondidas**, nem sempre utilizadas, que servem para transformar a informação recebida em input em algo que a camada output consiga utilizar e reconhecer.

Qual a função destas camadas escondidas?
Para responder a isto veremos o exemplo da RNA que diz se uma imagem tem presente um cão ou gato. Neste caso recebe no input a imagem. A primeira camada escondida pode analisar o brilho da imagem, a segunda pode tentar reconhecer a textura do pêlo, e por aí fora. Quando chega ao fim, junta todos os elementos descobertos em cada camada escondida e verifica em que caso é que estes elementos se encontram juntos (em cão ou gato)(7).

De forma a entender melhor RNA passaremos a abordar o que é uma rede neuronal artificial, introduzindo conceitos como "*Perceptron*", "*backpropagation*", entre outros, que serão explicados no decorrer do documento(8).

Perceptron

Em 1957, Frank Rosenblatt propôs um algoritmo para aprendizagem supervisionada de classificadores binários chamado "Perceptron". Há dois tipos de Perceptron, que são: Perceptron de uma só camada e o Perceptron de multi camadas, que tem maior capacidade de processamento (também chamadas redes neuronais) (7)(figura 1).

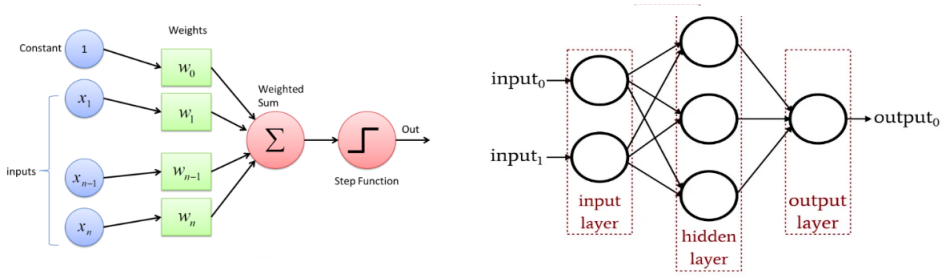

Figura 1. Perceptron de uma camada e perceptron de multi camadas

O Perceptron consiste em 4 partes(7):

- a todos os inputs multiplicar pelos pesos correspondentes
- somar todos os valores dados na primeira fase
- atribuir esta soma a uma função de activação que transforma a soma num valor entre 0 e 1 ou -1 e 1
- obter o resultado

Estas redes neuronais podem ter um valor "bias" que permite deslocar a curva da função de activação (step function da figura 1) para a esquerda ou direita conforme seja desejado obter um resultado mais próximo da realidade ou não. Esta função de ativação serve para indicar o output que deve ser dado, baseado nos cálculos feitos anteriormente(7).

Que tipos de redes neuronais existem?
Há vários tipos de redes neuronais possíveis de implementação, dependendo sempre da informação que queiramos treinar. Dois dos tipos mais comuns são(8):

- "*rede neuronal feedforward*", em que a informação só vai num sentido, do input para o output;
- "*rede neuronal recorrente*", em que a informação pode ir em múltiplas direcções existindo assim uma maior capacidade de aprendizagem, o que permite realizar tarefas mais complexas, tais como aprender caligrafia ou reconhecimento de linguagem.

Comparando RNA com RBC, é possível verificar que RNA têm melhor desempenho pois possui uma maior capacidade de lidar com casos que têm informação incompleta ou falta total de informação, contudo o utilizador não consegue justificar as decisões fornecidas pela rede.

3.2 Capacidade de Aprendizagem

Ao longo dos anos, as RNA têm vindo a ter mais uso. Isto deve-se ao facto de uma técnica chamada "backpropagation", utilizada em "redes neuronais feedforward" permitir uma aprendizagem supervisionada. "Backpropagation" é utilizado quando o resultado final não coincide com o resultado esperado, e consiste em corrigir erros que tenham sido feitos, para depois a rede neuronal continuar a sua tarefa de classificação sem constante interveniência do ser humano. Por exemplo, no caso da rede neuronal que vê se a foto era de cão ou gato, dar a foto de um cão e a rede neuronal dizer que era gato(8).

As redes neuronais artificiais têm três tipos de aprendizagem(9):

- **supervisionada**, que significa ter um conjunto de dados rotulados enquanto se treina um algoritmo, ou seja, cada exemplo no conjunto de dados de treino é marcado com a resposta que o algoritmo deve apresentar, em que no final este avalia com que precisão acertou ou não. Por exemplo, um conjunto de dados rotulados de fotos de animais, cada foto tinha um rótulo a dizer se

era cão, gato ou tartaruga e no fim o algoritmo calcula a precisão de acerto. Depois de ter treinado os dados, cria uma função de inferida em que vai permitir mapear novos exemplos (nunca antes vistos) por forma a aprender(9).

- **não supervisionada**, contrariamente ao tipo de aprendizagem supervisionada, o modelo recebe um conjunto de dados sem quaisquer instruções do que fazer com estes ou qual o output esperado. A rede neuronal tenta automaticamente encontrar uma estrutura nos dados em que seja possível extrair características e analisar a estrutura(9).

- **por reforço**, é uma área de *machine learning* preocupada com as ações dos agentes de software, maximizando a noção de recompensa, fazendo recuso ao método de tentativa e erro (se falha tem punição). Neste tipo de aprendizagem, os agentes tentam procurar um caminho ótimo para atingir um objectivo ou melhorar a performance numa determinada tarefa, em que conforme vão agindo em torno desta tarefa recebe uma recompensa. O foco principal é tentar prever o próximo passo de forma à recompensa final ser a melhor possível (utilizado no AI dos vídeo jogos).Para tomar decisões, o agente foca-se no feedback de aprendizagens passadas e na exploração de novas táticas que representem um melhor pagamento(9).

3.3 Ferramentas de Desenvolvimento

TensorFlow

É uma biblioteca *open source* de python desenvolvida pela Google, é bastante utilizada nas áreas de machine learning e neural networks(10).

Python

É uma linguagem de alto nivel, apresenta um grande numero de bibliotecas para Neural Networks, sendo atualmente uma das linguagens mais utilizadas ao nível de machine learning e data science(10).

Keras#

E também uma biblioteca de python. Relativamente ao TensorFlow apresenta como vantagem o fácil de ser mais acessível a utilizadores, permitindo mais rápida experimentação. Tem Pior performance do que TensorFlow(10).

3.4 Soluções de Mercado

Neural Designer

É uma aplicação de desktop que utiliza Neural Networks para análise de dados(10).

DeepArt

DeepArt é um programa que permite criar arte utilizando um algoritmo que redesenha uma imagem utilizando elementos estilísticos de outra(11).

Noise2noise

É um programa de restauração de imagens corruptas, desenvolvida pela Nvidia(12).

4 Support Vector Machines

4.1 Descrição

Por fim, existem as Máquinas de Vetor de Suporte ou *Support Vector Machines* (SVM), um conjunto de modelos correspondentes a algoritmos de aprendizagem supervisionada, desenvolvido por Vladimir Vapnik e os seus colegas no início dos anos 90, que utiliza a teoria de aprendizagem estatística **Vapnik-Chervonenkis** (VC Theory).

Este algoritmo consiste na generalização dos dados de treino, de forma a conseguir estabelecer planos de corte entre os vários *data points* do *training set*. Esta abordagem visa permitir a classificação de novos *inputs* que lhes sejam fornecidos perante a sua posição, comparando com os planos de corte previamente definidos.

As *Support Vector Machines* focam maioritariamente na resolução de problemas de classificação, que tendem a revelar-se uma classificação binária, *Yes or No*. Caso se tratem de problemas onde os *data points* têm N classes, estes podem ser interpretados como sucessivas classificações binárias.

Não existe um algoritmo entre RNA e SVM que seja melhor que o outro, uma vez que depende do caso a ser estudado. Posto isto, serão apresentadas algumas vantagens que cada algoritmo dispõe quando comparado com o outro.

RNA apresentam melhor performance quando:

- Existe um grande número de instâncias de treino.
- Estamos perante um problema com múltiplas classes.

Por outro lado, SVM têm as vantagens de:

- Melhor performance quando existe um número reduzido de dados de treino.
- Não sofre de problemas de sobreajuste.
- De fácil implementação e flexível.

4.2 Capacidade de Aprendizagem

Support Vector Classification

O presente objetivo do *Support Vector Classification* é desenvolver um algoritmo eficiente, em termos computacionais, capaz de gerar um bom hiperplano num espaço de representação induzido pelo *Kernel*.

Neste âmbito, vão ser brevemente abordados os diferentes algoritmos que permitem a criação dos modelos.

The Maximal Margin Classifier:

O MMC, *Maximal Margin Classifier*, foi o primeiro e mais simples modelo de SVM a ser introduzido. Funciona unicamente para dados que possam ser linearmente divididos. Por este motivo, são escassas as situações reais onde este seja aplicado. No entanto, é o algoritmo de mais fácil compreensão e apresenta diversas bases que são essenciais na perceção do SVM.

O *Maximal Margin Classifier* consiste em desenhar um hiperplano no espaço de representação dos dados induzido pelo *Kernel*, com auxílio dos vetores de suporte, de forma a separar as diferentes classes com a maior margem possível entre os dados e o plano.

Neste contexto, **margem** corresponde à distância entre os vetores de suporte e o plano, como pode ser visualizado na figura 2.

Os vetores de suporte correspondem a uma linha que interseta os pontos cujo resultado da função de cálculo do plano seja 1 (um). Assim sendo, estes pontos irão afetar o local onde o plano será colocado. Qualquer ponto que se encontre para lá desses vectores de suporte tem peso 0 (zero) no cálculo do plano.

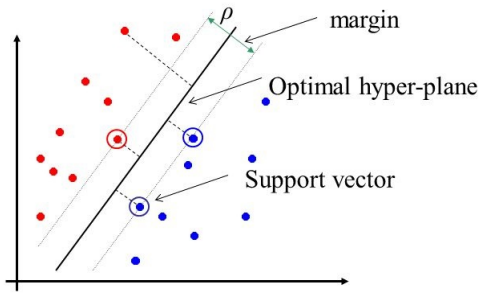

Figura 2. Representação gráfica de um algoritmo de *Maximal Margin Classifier*

Soft Margin Optimization:

Como previamente mencionado, o *Maximal Margin Classifier* não pode ser aplicado em casos reais. Consequentemente, numa tentativa de o adaptar a casos concretos, foi introduzida uma nova variável às funções de cálculo, C, que corresponde à tolerância da presença de erros na classificação, isto é, de entre os pontos do *training set*, quantos podem estar do lado errado do plano e, apesar disso, corresponderem a uma solução válida.
Neste caso, quanto menor o valor de C, menos erros poderão existir. Assim sendo, quando C = 0, verifica-se um SVM de *Maximal Margin*.

Por conseguinte, o desenvolvimento de um dado SVM para um *training set* específico irá consistir na execução do algoritmo para diferentes valores de C e observar para que valor de C a percentagem de erro em novos *inputs* será menor.

Kernel

Uma vez que as máquinas de aprendizagem linear são extremamente limitadas, a sua utilidade em aplicações do mundo real é reduzida. Esta inutilidade deve-se ao facto de o problema que se pretende resolver não poder ser, frequentemente, mapeado linearmente, dada a necessidade da avaliação de múltiplos atributos, a fim de se obter uma classificação.

Posto isto, foram criadas funções *Kernel* que projetam os dados num espaço de dimensão superior, de forma a aumentar o poder computacional das máquinas de aprendizagem linear e permitir que os dados sejam linearmente divisíveis. Como se pode observar na figura 3, os dados não podiam, originalmente, ser linearmente divisíveis e, por isso, aplicou-se uma função *Kernel* adequada, permitindo obter uma divisão entre os dados.

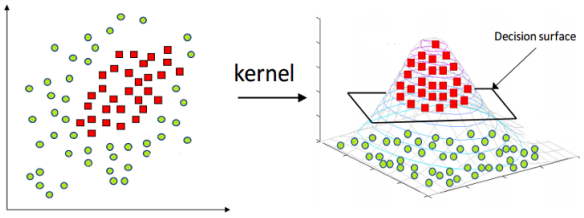

Figura 3. Exemplo da representação de um *data set* com a ajuda do *kernel*.

A inexistência de um *Kernel* perfeitamente compatível com todas as distribuições de dados conduziu à aplicação dos diversos *Kernels* existentes para resolução de um certo problema:

– Polynomial Kernel

- Linear Kernel
- Gaussian Kernel
- Laplace Kernel
- entre outros...

Posteriormente, é necessário escolher aquele que melhores resultados obteve perante os dados do nosso problema.

4.3 Ferramentas de Desenvolvimento

De seguida, vão ser descritas três das diversas plataformas que existem para desenvolvimento com *Support Vector Machines*.

WEKA
Waikato Environment for Knowledge Analysis, WEKA, é uma *framework* desenvolvida pela universidade de Waikato da Nova Zelândia que engloba diversos algoritmos utilizados em Mineração de Dados ou *Machine Learning*, incluindo as *Support Vector Machines* previamente retratadas, sem a necessidade de construir código, mas exclusivamente o *dataset* que se pretende testar nos diversos algoritmos disponíveis.

Scikit-learn
Scikit-learn é uma biblioteca de *Machine Learning* para desenvolvimento em *Python*, que envolve algoritmos de aprendizagem supervisionada, bem como algoritmos de aprendizagem sem supervisão, incluindo, ainda, algoritmos de *Support Vector Machines*.

MATLAB
Por fim, uma das plataformas de programação mais conhecidas de hoje em dia, MATLAB, permite também o desenvolvimento de aplicações com*Support Vector Machines*.

4.4 Soluções de Mercado

Reconhecimento de Imagens
Encontra-se, mais do que nunca, uma quantidade descomunal de imagens, tanto na internet, como em bases de dados especializadas. Como consequência, foram desenvolvidos algoritmos que possibilitassem a caracterização automática de uma imagem consoante o seu conteúdo.(14)

Por este motivo, as *Support Vector Machines* têm revelado bastante utilidade nesta área.

Ciências Geográficas e Ambientais
As *Support Vector Machines* são, também, usadas nas áreas das ciências ambientais, por exemplo, na elaboração de um modelo para classificar dados sobre poluição. Exemplificativamente, recorrendo ao *Radial Basis Function Kernel*, foi possível modelar os dados sobre a poluição de sedimentos do lago Leman.(15)

5 Conclusão

Com a execução deste artigo concluímos que cada um destes sistemas de aprendizagem, apesar de distintos, se unem no propósito final de solucionar problemas de forma inteligente e precisa. Todos estes sistemas que apresentamos são extremamente úteis, porém cada um tem pontos fortes e pontos fracos quando comparado com os restantes em diferentes situações.

Foi possível compreender o que Raciocínio Baseado em Casos já teve a sua importância no mundo de inteligência artificial porém na actualidade recente não é tão usado devido ao facto de precisar de algum investimento, de ser difícil obter conhecimento e de, apesar de descoberta a solução, não conseguir garantir que a solução é óptima.

Quanto a Redes Neuronais Artificiais, estas são cada vez mais usadas pois têm alto desempenho quando comparado com RBC, conseguem lidar melhor com casos de falta de informação e treinar uma rede de forma a que os resultados sejam o mais próximos da realidade possível.

No caso de Máquinas de Vector de Suporte estas apresentam cada vez mais aderência na resolução de problemas reais devido à sua fiabilidade e devido ao facto de em muitas situações serem superiores aos outros métodos de *Machine Learning*.

Bibliografia

[1] rbc-texto.pdf (blackboard)

[2] Raciocínio Baseado em Casos. (2016). DESAFIOS - Revista Interdisciplinar Da Universidade Federal Do Tocantins, 3(1), 89-94. https://doi.org/10.20873/uft.2359-3652.2016v3n1p89

[3] AIAI CBR Homepage, http://www.aiai.ed.ac.uk/project/cbr/CBRDistrib/. Last accessed 1 November 2020.

[4] Adedoyin, Adeyinka & Kapetanakis, Stelios & Petridis, Miltos & Panaousis, Emmanouil. (2016). Evaluating Case-Based Reasoning Knowledge Discovery in Fraud Detection. In 24th Proceedings of the ICCBR 2016 Workshops. Atlanta, Georgia, US.

[5] https://www.aber.ac.uk/~dcswww/Research/mbsg/cbrprojects/getting_caspian.shtml. Last accesed 1 November 2020

[6] RACIOCÍNIO BASEADO EM CASOS NA CONFECÇÃO DE TERMOS DE REFERÊNCIA PARA CONTRATAÇÕES PÚBLICAS : JULIÃO , Daniel.

[7] Simplilearn Homepage, What is Perceptron, https://www.simplilearn.com/what-is-perceptron-tutorial. Last accessed 2 November 2020.

[8] Digital Trends Homepage, What is an artificial neural network, https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/. Last accessed 2 November 2020.

[9] Nvidia Blog, What's the Difference Between Supervised, Unsupervised, Semi-Supervised and Reinforcement Learning, https://blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/. Last accessed 2 November 2020.

[10] Predictive Analytics Today, Top 27 Artificial Neural Network Software, https://www.predictiveanalyticstoday.com/top-artificial-neural-network-software/. Last accessed 2 November 2020.

[11] https://deepart.io/ Last accessed 2 November 2020.

[12] https://github.com/NVlabs/noise2noise Last accessed 2 November 2020.

[13] Burges, C.J. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998).

[14] Nello Cristianini, John Shawe-Taylor, "An Introduction to Support Vector Machines and other kernel-based learning methods", Cambridge University Press, 2000.

[15] Gilardi, N. Kanevski, Mikhail Maignan, Michel Mayoraz, E.. (1999). Environmental And Pollution Spatial Data Classification With Support Vector Machines And Geostatistics.