

Agentes Aprendizes

Capítulo 5:

Costa, E. e Simões, A. (2008). Inteligência Artificial – Fundamentos e Aplicações, 2.^a edição, FCA.

Agentes Aprendizes

O significado de Aprender

- Aprender é uma característica essencial dos seres inteligentes
- É por isso que a área de **Aprendizagem Artificial (AA)** se tornou uma área central em Inteligência Artificial

Aprendizagem Artificial

A Aprendizagem Artificial tem 3 objetivos principais:

1. O desenvolvimento de teorias computacionais da aprendizagem
2. A implementação de sistemas com capacidade de aprender
3. A análise teórica e o desenvolvimento de algoritmos genéricos de aprendizagem

Aprendizagem Artificial

- Uma definição rigorosa de **aprendizagem** encontra as mesmas dificuldades que a tentativa de definir **inteligência**
- O verbo **aprender** vem do latim **apprehendâre**, que significa compreender, traduzindo
 - Adquirir conhecimento de
 - Instruir-se
 - Estudar
 - Ficar sabendo

Aprendizagem Artificial

- Sendo que o substantivo **aprendizagem** significa
 - Ação ou efeito de aprender
 - Aprendizado
 - Aquisição, mediante uma atividade de ensino, dos conhecimentos necessários sobre determinada profissão

Aprendizagem Artificial

- Estas definições correspondem à noção intuitiva que todos temos, de que aprender significa **adquirir novos conhecimentos** ou **reestruturar conhecimentos anteriores**
- Também está ligado ao conceito de **aquisição de novas capacidades** que, em princípio, nos permitem desempenhar melhor no futuro uma dada tarefa

Aprendizagem Artificial

- Como transportar esta capacidade para um agente artificial ?
- Ao longo do tempo foram várias as propostas de definição que os diferentes algoritmos de aprendizagem procuraram operacionalizar

Aprendizagem Artificial

- “Aprender denota mudanças num sistema que são adaptativas, no sentido de que permitem ao sistema realizar a mesma tarefa ou tarefas da mesma classe de modo mais eficiente e mais efectivo no futuro”
(Simon, 1983)
- Noção de melhoria no desempenho

Aprendizagem Artificial

- “Aprender significa realizar mudanças úteis no modo de funcionar da nossa mente”
(Minsky, 1985)
- Definição em torno do conceito geral de utilidade

Aprendizagem Artificial

- “Aprender consiste em construir e modificar representações do que está a ser vivenciado”
(Michalski, 1986)
- Define o conceito de aprender sem recorrer à noção de melhoria de desempenho

Aprendizagem Artificial

- “ ... A aprendizagem pela máquina preocupa-se com o desenvolvimento de programas de computador que são capazes de construir novo conhecimento ou de modificar de um modo útil ‘velho’ conhecimento, utilizando informação do ambiente ... ”
(Michalski e Kodratoff, 1990)
- Visão pragmática e operacional de aprendizagem pela máquina definida como área científica

Aprendizagem Artificial

- Todas estas definições remetem diretamente para um conhecimento representado por estruturas e, indiretamente, para mecanismos de alteração dessas estruturas
- Aprender aparece assim ligado ao conceito de conhecimento e da sua representação

Aprendizagem Artificial

- A definição de **linguagens de representação** que permitam a aquisição de conhecimento de forma eficiente e que tenham poder explicativo é uma tarefa de primordial importância para a aprendizagem
- A noção de experiência envolve quer o **agente**, quer o **ambiente** externo, do mesmo modo que o problema do **desempenho** terá que ser referido a uma dada **tarefa**

Aprendizagem Artificial

- Consideremos uma definição que tenta tornar mais claro o relacionamento entre estas quatro diferentes componentes:
- “Aprendizagem é o processo pelo qual um **agente** altera de modo automático as suas estruturas internas, de modo a **realizar melhor tarefas** do seu interesse, nas condições impostas pelo **ambiente**”

Aprendizagem Artificial

- Esta definição tem muito que ver com a definição de **adaptação** que ocorre nos seres vivos.

Existem no entanto algumas diferenças:

- **Aprender** é um ato individual, enquanto a **adaptação** se aplica a uma população de indivíduos
- A **escala temporal** também é importante, sendo curta no que se refere à aprendizagem e longa no que diz respeito à adaptação

Aprendizagem Artificial

Exemplo simples:

- Admitamos que pretendemos um agente especialista na atribuição de crédito (**tarefa**)
- O **ambiente** será composto, por exemplo, pelas centenas de processos de concessão de crédito cujo resultado é conhecido
- O **desempenho** do agente pode ser medido pela percentagem de decisões erradas que tomou
- O processo de aprendizagem procurará diminuir essa percentagem

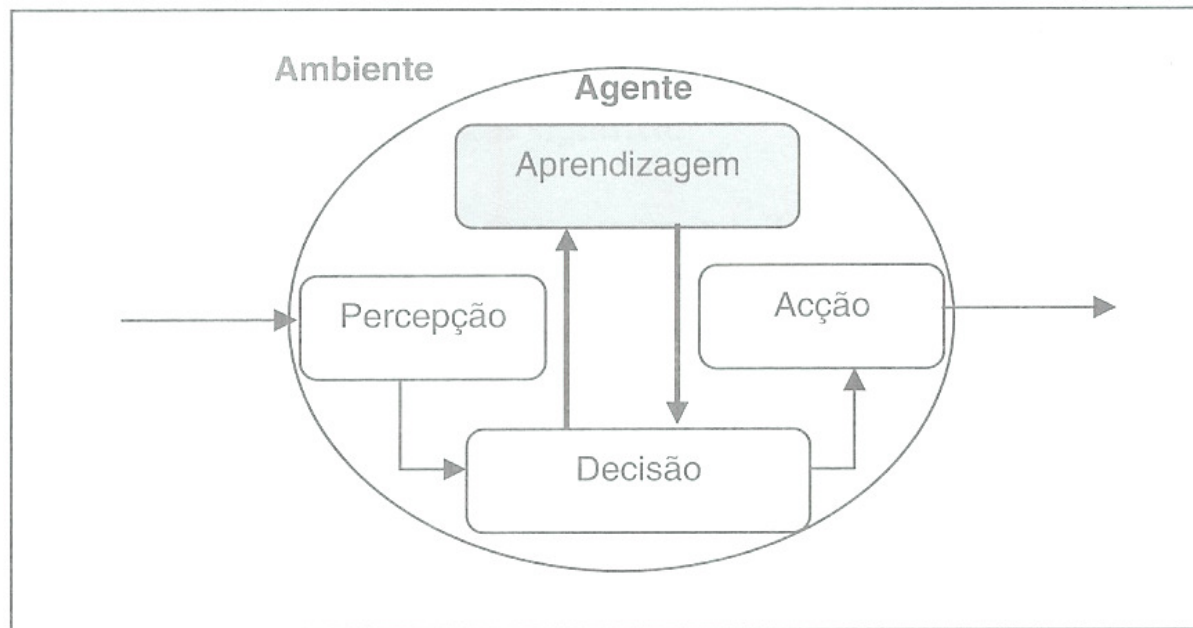
Aprendizagem Artificial

- A **capacidade de aprender** é uma condição essencial para que o **agente** seja **verdadeiramente autónomo**
- Por outro lado, um agente com capacidade de aprender pode substituir um engenheiro do conhecimento no processo difícil de adquirir conhecimento de peritos com a finalidade de **construir sistemas periciais**

Arquitetura de um Agente Aprendiz

- A arquitetura de um agente aprendiz baseia-se essencialmente em **quatro módulos**: o módulo de **perceção** e o módulo de **ação**, como nos agentes reativos; o módulo de **decisão**, como nos agentes baseados em conhecimento; e o módulo de **aprendizagem**.

Arquitetura de um Agente Aprendiz

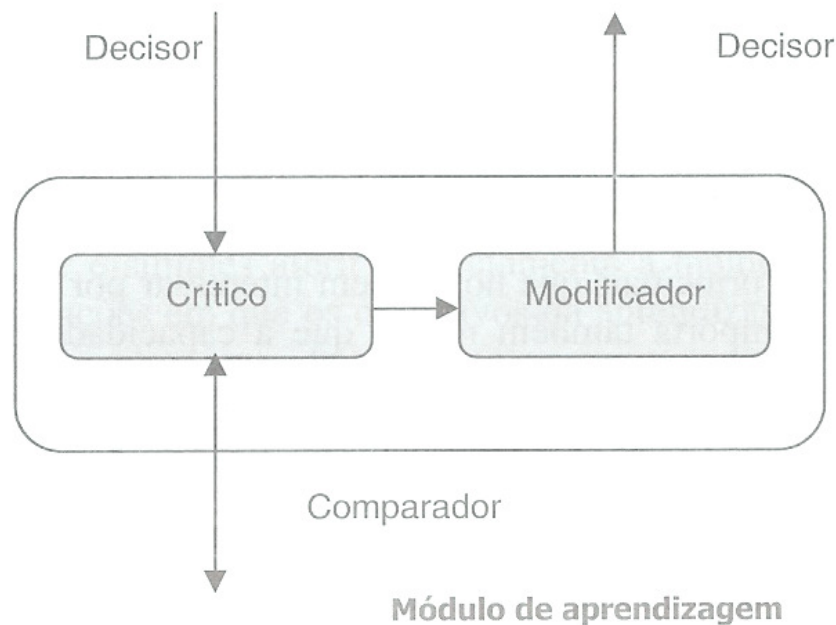


Arquitectura de um agente aprendiz

Arquitetura de um Agente Aprendiz

- O **módulo de percepção** corresponde à parte do agente responsável pela tradução dos dados dos sensores em percepções
- Essa informação é enviada para o **módulo de decisão** que a utiliza para determinar a melhor ação a executar
- No entanto, essa decisão é avaliada pelo **módulo de aprendizagem**
- Este módulo é composto, por sua vez, por dois submódulos essenciais: um **crítico** e um **modificador**.

Arquitetura de um Agente Aprendiz



Arquitetura de um Agente Aprendiz

- Se a avaliação do comportamento do elemento de **decisão**, feita pelo **módulo crítico** por comparação com um *standard*, não é satisfatória, então o **elemento modificador** determina e promove as mudanças necessárias no elemento de **decisão** tendo em vista a melhoria do comportamento do agente

Arquitetura de um Agente Aprendiz

- O processo pode repetir-se até que o módulo de aprendizagem não promova mais modificações, altura em que o módulo de decisão interage com o módulo de ação para este, por sua vez, reagir sobre o ambiente

Arquitetura de um Agente Aprendiz

- O problema da aprendizagem pode ser sistematizado de um modo mais algorítmico, como no quadro seguinte:

Dados:

- INSTÂNCIAS (I)
- CONCEITO ALVO (C)
- EXEMPLOS DE TREINO DO CONCEITO ALVO ($E \subseteq I$)
- HIPÓTESES (H)
- TEORIA SOBRE O DOMÍNIO EXPLICATIVA DOS EXEMPLOS (T)

Determinar:

- Hipóteses $h_i \subseteq H$ para descrever o conceito alvo (C), consistentes com os exemplos de treino (E) e com a teoria sobre o domínio (T)
-

Arquitetura de um Agente Aprendiz

O significado é o seguinte:

- O **conjunto de instâncias (I)** refere-se ao conjunto de todos os objetos do mundo;
- O **conceito alvo (C)** é o que se pretende que o sistema aprenda;
- O **conjunto de exemplos (de objetos) de treino (E)** é formado pelos objetos usados na aprendizagem;
- O **conjunto de hipóteses (H) definidoras do conceito** refere-se às possibilidades de hipóteses definidoras para o conceito alvo;
- A **teoria sobre o domínio (T)** é o conhecimento pré-existente e explicativo dos exemplos.

Arquitetura de um Agente Aprendiz

- Com estes dados pretende-se **determinar hipóteses** consistentes com os **exemplos de treino** e com a **teoria sobre o domínio**, que **serão candidatos a soluções do problema de aprendizagem**

Taxionomia de Agentes Aprendizes

- Os agentes aprendizes distinguem-se fundamentalmente pelo modo como o **módulo de aprendizagem** é representado e implementado
- Os algoritmos propostos na literatura podem ser analisados de diferentes pontos de vista
- Consideremos as seguintes **5 dimensões**:

Taxionomia de Agentes Aprendizes

1. Do ponto de vista da **entrada** do algoritmo, esta pode ser definida através de características binárias identificadoras de
 - Presença ou ausência de determinada característica (**cor_clara**);
 - Pares atributo/valor (**cor(clara)**);
 - Expressões relacionais (**sobre(obj1,obj2)**);

Taxionomia de Agentes Aprendizes

2. Do ponto de vista da **saída** do algoritmo (o resultado da aprendizagem) podemos ter abstrações do tipo

- Hierarquias de conceitos (árvores de decisão);
- Expressões lógicas;
- Vetores de números;
- Regras do tipo se <condição> então <ação>;

Taxionomia de Agentes Aprendizes

3. Quanto ao **tipo de aprendizagem** os algoritmos podem ser

- Pobres em teoria, usando fundamentalmente **técnicas indutivas**;
- Ricos em teoria, socorrendo-se de **técnicas dedutivas**;
- De abordagens híbridas, **baseadas em casos**;

Taxionomia de Agentes Aprendizes

4. Outro aspeto importante diz respeito às condições proporcionadas pelo ambiente
- A aprendizagem pode ser **supervisionada**, quando os exemplos de treino vêm classificados;
 - A aprendizagem pode ser **incremental**, quando os exemplos aparecem sequencialmente ao longo do tempo;

Taxionomia de Agentes Aprendizes

5. Podemos olhar para os algoritmos do ponto de vista das **tarefas** para que estão vocacionados, podendo distinguir fundamentalmente
- Tarefas de **classificação**, quando se procura aprender a classificar instâncias concretas, num passo de aprendizagem e com precisão;
 - Tarefas de **resolução de problemas**, quando se procura resolver um problema em vários passos, de forma eficiente;

Taxionomia de Agentes Aprendizes

- A **aprendizagem** é uma disciplina baseada em algoritmos
- Para os classificar podemos usar aquelas 5 dimensões
- No entanto, também podemos adotar outros pontos de vista aglutinadores, de interesse por potenciarem um debate sobre questões essenciais não apenas para a **aprendizagem pela máquina**, mas também para a **Inteligência Artificial** no seu todo

Taxionomia de Agentes Aprendizes

- Esses diferentes pontos de vista derivam das diferentes **metáforas** que têm sido propostas para o **problema da cognição** e da **origem dos agentes inteligentes**
- Quais são então esses pontos de vista ?

Taxionomia de Agentes Aprendizes

1. O computador

- É um sistema físico de símbolos que possui os meios necessários e suficientes para a **ação inteligente**
- Esta é consequência da manipulação, baseada em **regras**, de **estruturas simbólicas**

Taxionomia de Agentes Aprendizes

2. O cérebro

- É uma **rede** de elementos simples
- A **inteligência** resulta da emergência de estados globais estáveis como resultado da interação baseada em regras locais para os elementos e regras para mudanças nas ligações

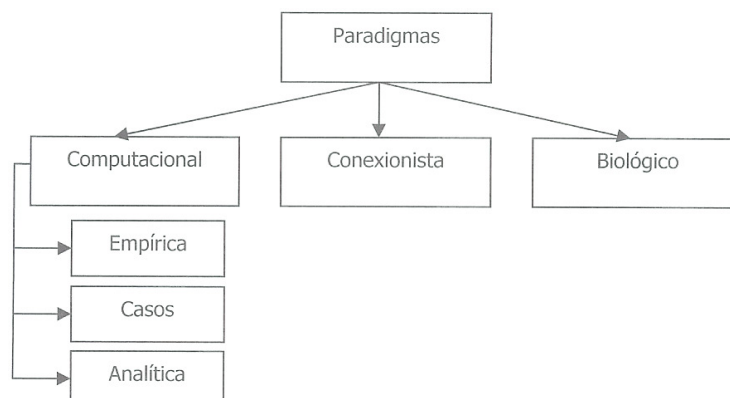
Taxionomia de Agentes Aprendizes

3. A evolução das espécies

- Os seres vivos são o resultado de biliões de anos de evolução
- O seu estabelecimento foi promovido pelo **mecanismo de seleção** (reproduzindo prioritariamente os mais aptos/adaptados) e **mecanismos que promovem a diversidade** (operadores genéticos de recombinação e mutação)
- Assim apareceram os **seres inteligentes**

Taxionomia de Agentes Aprendizes

- Historicamente, a abordagem simbólica clássica foi a que forneceu mais algoritmos alternativos
- Podemos tornar a nossa classificação mais fina se introduzirmos a natureza empírica, analítica ou híbrida do tipo de aprendizagem que cada um dos algoritmos privilegia



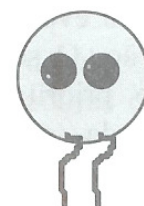
Exemplo: Classificação de objetos

- Vamos ver de seguida como cada uma destas três abordagens principais (**computacional**, **conexcionista** e **biológica**) pode ser usada para resolver um mesmo problema simples de aprendizagem, envolvendo **classificação de objetos**

Exemplo: Classificação de objetos

1. O Domínio

- Consideremos um domínio hipotético envolvendo células e possíveis doenças
- As células podem ser caracterizadas por diferentes atributos (caudas, cor e núcleos) que podem tomar valores numéricos ou nominais
- Uma célula pode ter uma ou duas caudas, ser de cor clara ou de cor escura e ter um ou dois núcleos
- O facto de apenas poderem tomar dois valores permite simplificar e variar a representação

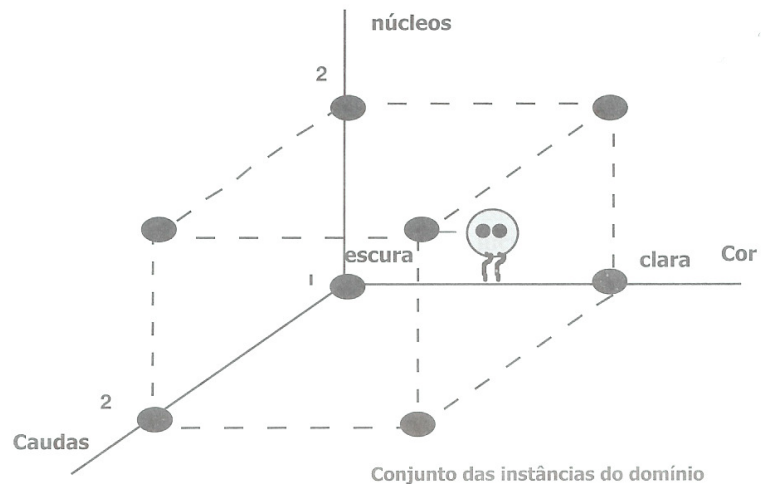


Exemplo de uma célula

Exemplo: Classificação de objetos

1. O Domínio

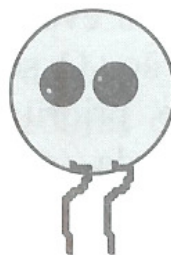
- O conjunto das instâncias é formado por 8 objetos, podendo ser representado pelo cubo da figura



Exemplo: Classificação de objetos

1. O Domínio

- A instância considerada é caracterizada por $i_1 = [\text{cor} = \text{clara}, \text{caudas} = 2, \text{núcleos} = 2]$



Exemplo de uma célula

Exemplo: Classificação de objetos

1. O Domínio

- Um conceito alvo poderia ser, por exemplo:



Exemplo: Classificação de objetos

1. O Domínio

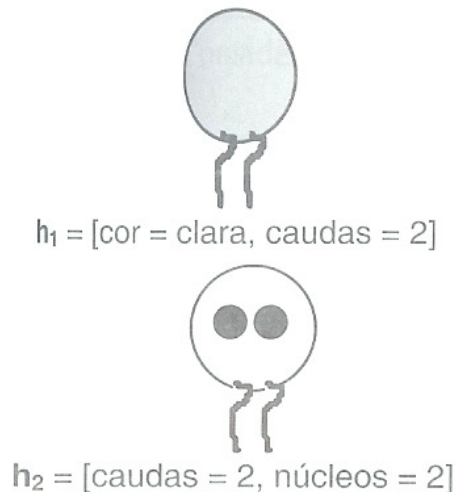
- Exemplos possíveis de treino



Exemplo: Classificação de objetos

1. O Domínio

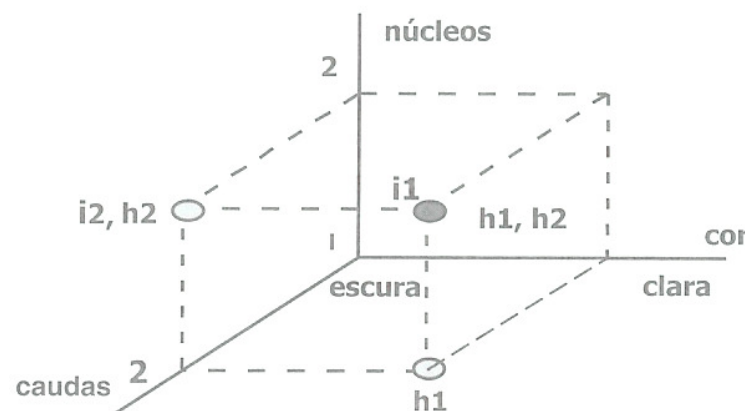
- São exemplos de **hipóteses** de conceito



Exemplo: Classificação de objetos

1. O Domínio

- As duas hipóteses e a instância referida cobrem várias instâncias assinaladas na figura



Instâncias e hipóteses do conceito a aprender

Exemplo: Classificação de objetos

1. O Domínio

- Como teoria teremos

Se simplóide **e** neoplasma **Então** letargia
Se uma-cauda **e** parede-fina **Então** simplóide
Se duas-caudas **Então** simplóide
Se um-núcleo **e** parede-grossa **Então** neoplasma
Se dois-núcleos **Então** neoplasma
Se duas-caudas **e** neoplasma **Então** doldroma

- Esta teoria refere-se a possíveis doenças das células
- A ideia é **dividir** um conjunto de células em **sãs e doentes**, definindo se possível um critério para as distinguir

Exemplo: Classificação de objetos

2. Abordagem computacional

- Vamos começar por considerar o **algoritmo ID3** desenvolvido por *Quinlan* em 1986 e que faz parte da família de algoritmos **TDIDT** (*Top Down Induction of Decision Trees*) em que o tipo de aprendizagem é empírico

Exemplo:

Classificação de objetos

- Trata-se de um algoritmo que, a partir de um conjunto de exemplos de treino positivos e negativos de uma classe, **constrói uma árvore de decisão** que define em intenção essa classe (um exemplo positivo corresponde a um exemplo do conceito)
- Este algoritmo promove uma **aprendizagem indutiva** em modo **supervisionado** e **não incremental**, sendo usado em tarefas de classificação

Exemplo:

Classificação de objetos

Algoritmo ID3

Dado:

- um conjunto S de exemplos de treino ($E+, E-$);
- uma família de conjuntos de atributos e respectivos valores, A ;
- um conceito alvo T

Determinar:

- uma árvore de decisão, AD , cujas folhas são formadas por elementos todos da mesma classe; o conceito alvo T é dado pela disjunção da caracterização da classe positiva. ($E+$).

1. Se todos os exemplos são da mesma classe

Então terminar, com a AD formada por um nó etiquetado pela classe dos elementos de S ;

2. Caso contrário:

2.1 Escolher um atributo $A = \{A_1, \dots, A_V\}$;

2.1.1. Dividir S em $\{S_1, \dots, S_V\}$ subconjuntos disjuntos de acordo com os diferentes valores de A ;

2.2. Chamar recursivamente o algoritmo para cada um dos subconjuntos S_i ;

2.3. Construir uma AD tendo por raiz o atributo A e os ramos etiquetados pelos valores A_j ligados às sub-árvores associadas a S_j .

Exemplo:

Classificação de objetos

- O algoritmo parte de um conjunto S de exemplos positivos e negativos, descrito numa representação baseada em pares atributo/valor e de um conceito alvo T , para a construção da árvore de decisão (AD), cujas folhas são formadas por elementos todos da mesma classe
- O conceito alvo é dado pela disjunção da caracterização da classe positiva
- Esta construção é feita de modo repetido

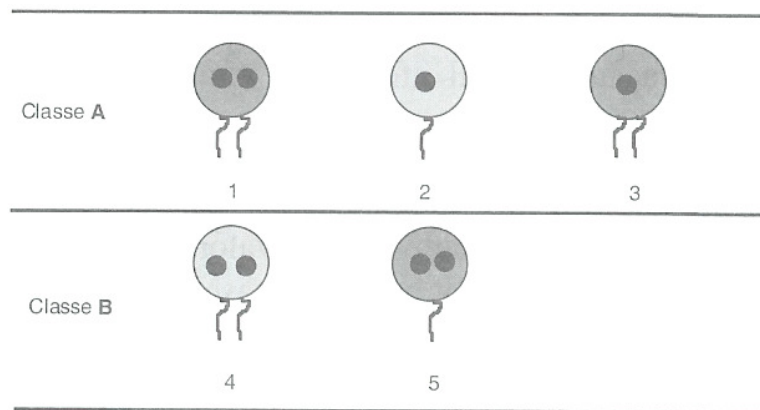
Exemplo:

Classificação de objetos

- Se o conjunto dos elementos em análise pertence todo à mesma classe, então o processo pára com a formação de um nó etiquetado pelo nome da classe
- Se não, escolhe-se um atributo, seja A_j , que permite dividir os exemplos de S em subconjuntos S_i , indexados pelos valores possíveis do atributo A_j
- A cada S_i ficam associados os elementos de S que têm o mesmo valor para A_j
- O processo é retomado recursivamente com cada um dos S_i
- Finalmente, constrói-se a AD tendo por raiz um nó etiquetado pelo nome do atributo A_j ligada às subárvores associadas a S_i , sendo os valores do atributo A_j associados aos respetivos ramos
- O atributo escolhido em cada etapa é aquele que for mais discriminante, isto é, aquele que permite criar mais nós com todos os elementos da mesma classe ou então, no caso de isto ser impossível, com nós contendo muitos elementos de uma classe e poucos da outra
- Deste modo, garante-se a construção de uma árvore de decisão ótima, isto é, envolvendo em média o menor número de testes para classificar corretamente novos exemplos

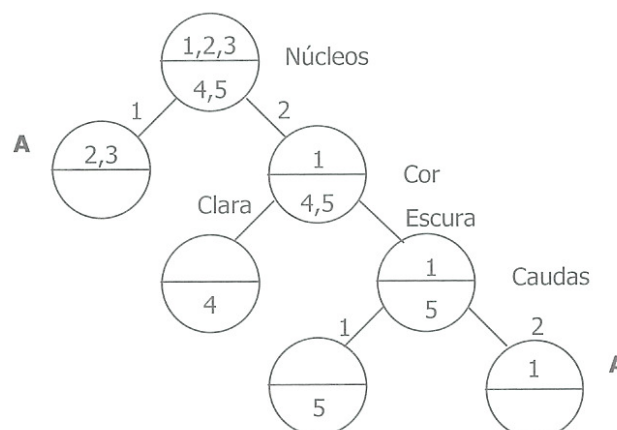
Exemplo: Classificação de objetos

- Exemplo de aplicação do **ID3**, usando o domínio das células descrito anteriormente
- Consideremos os 5 exemplos positivos e negativos seguintes:



Exemplo: Classificação de objetos

- Estes exemplos serão usados pelo algoritmo **ID3** como exemplos de treino, cujo resultado será a construção da árvore de decisão seguinte



Exemplo: Classificação de objetos

- Em cada nó a parte superior contém os números dos exemplos da **classe A**, enquanto a parte inferior tem os números dos exemplos da **classe B**
- É fácil de ver que o atributo **Núcleos** é o mais discriminante
- Na etapa seguinte, os atributos restantes são igualmente discriminantes, pelo que é escolhido aleatoriamente um deles

Exemplo: Classificação de objetos

- Da árvore retira-se a definição da **classe A** (também se pode retirar para a classe B):

Classe A = [núcleo=1] ou [cor=escura, caudas=2, núcleos=2]

- Notar que se trata de uma definição disjuntiva
- Esta definição pode agora ser usada para classificar exemplos de teste

Exemplo: Classificação de objetos

- O ID3 é um algoritmo simples capaz de aprender a partir de um conjunto vasto de exemplos
- Tem no entanto alguns problemas associados:
- Ruído

Se a classificação for mal efetuada, o algoritmo colapsa fornecendo resultados inconsistentes;

- Valores em falta/desconhecidos

Em situações reais pode acontecer que alguns exemplos de teste tenham atributos sem valores associados

Coloca-se o problema de saber que valor atribuir nesses casos. Uma solução é escolher o valor mais frequente para a classe;

Exemplo: Classificação de objetos

- Polarização (*bias*)
O algoritmo prefere as hipóteses simples às complexas, princípio genericamente usado em aprendizagem empírica;
- Sobreajustamento (*overfitting*)
Problema que ocorre quando o sistema, dispondo de muitas hipóteses onde podem ocorrer erros, descobre regularidades sem sentido, classificando corretamente os exemplos de treino, mas incorretamente os de teste
Uma solução possível poderá ser limitar o crescimento da árvore de decisão;
- Valores contínuos
necessitam ser discretizados, podendo também ser definidos intervalos;

Exemplo:

Classificação de objetos

3. Abordagem conexionista

- Consideremos agora os modelos conexionistas
- Os modelos conexionistas inspiram-se no **cérebro**, que é tratado como uma rede intrincada de elementos simples, os **neurónios**, **funcionando em paralelo** e, por isso mesmo **tolerante a falhas e à incerteza**

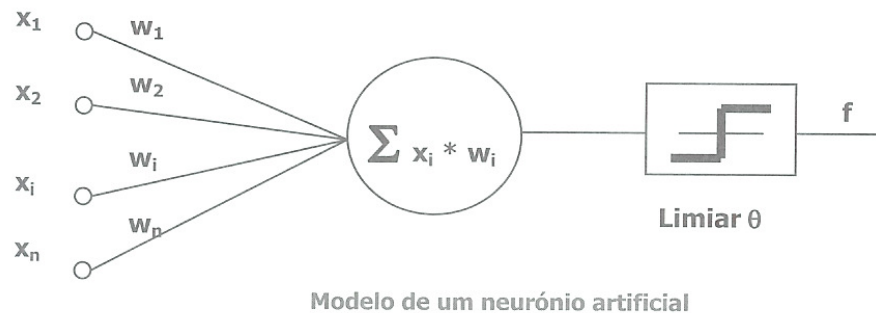
Exemplo:

Classificação de objetos

- O neurónio biológico funciona, de forma simplificada, do seguinte modo:
- Através das **dendrites** recebe de outros neurónios sinais de ativação.
- Se a soma desses sinais alcançar um determinado **limiar**, o neurónio dispara enviando um sinal ao longo do **axónio** que, por sua vez, se propaga aos outros neurónios a que se encontra ligado através dos espaços **sinápticos**.

Exemplo: Classificação de objetos

- Estes princípios foram traduzidos por *McCulloch & Pitts* num modelo de neurónio artificial



Exemplo: Classificação de objetos

- Podemos identificar as **entradas**, uma **unidade somadora**, uma **função de ativação** associada a um **limiar** e a **saída**, daí o nome de **unidades lineares de limiar**
- Os modelos conexionistas são formados por redes de neurónios artificiais, daí serem vulgarmente designados por **redes neuronais**

Exemplo: Classificação de objetos

- As redes neuronais funcionam normalmente em ambiente **supervisionado** e de modo **incremental**
- As suas entradas são vetores de números (inteiros ou reais) e a **aprendizagem é de tipo empírica**, traduzindo-se pela mudança nos pesos associados às ligações entre neurónios

Exemplo: Classificação de objetos

- Uma TLU é uma rede neuronal com um único neurónio, com uma função de ativação do tipo **degrau** (*hard limiter*) e usando uma regra denominada **regra delta** como regra de aprendizagem
- A partir de um conjunto de exemplos de treino e de um parâmetro que especifica a dimensão dos ajustes dos pesos (**o ritmo de aprendizagem η**), vai aprender o conjunto de pesos que permite classificar corretamente os exemplos de treino (e de teste)

Exemplo: Classificação de objetos

- Os **pesos** e o **limiar** começam por ser inicializados aleatoriamente
- Entra-se depois num ciclo em que, para cada exemplo de treino, se verifica se a rede responde corretamente
- Em caso negativo, procede-se ao **ajuste dos pesos na proporção do erro verificado e do ritmo de aprendizagem**, seguindo o princípio *Hebbiano* de que dois neurónios que estão ativos em simultâneo devem ter a sua ligação reforçada

Exemplo: Classificação de objetos

Algoritmo TLU

Dado:

- ritmo de aprendizagem: η
- conjunto de exemplos de treino

Determinar:

- conjunto de pesos que permite classificar todos os elementos do conjunto de treino de forma consistente.

1. Inicializar Pesos \mathbf{W} e Limiar θ de forma aleatória (com valores baixos)

2. Seleccionar Exemplo de treino $(x_i(t), d(t))$

3. Calcular saída real

$$O(t) = f_{hl}(\sum w_i(t) * x_i(t) - \theta)$$

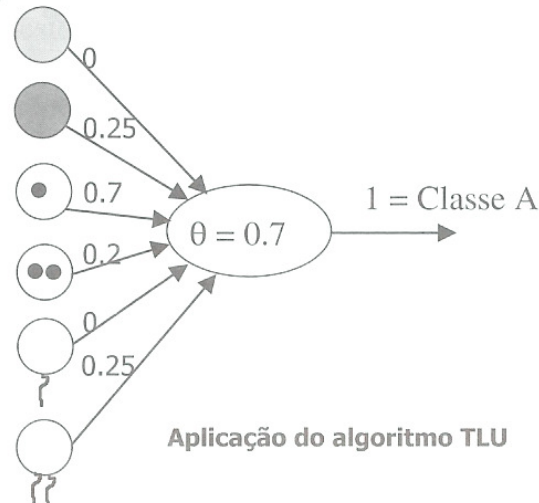
4. Adaptar pesos

$$w_i(t+1) = w_i(t) + \eta * [d(t) - O(t)] * x_i(t)$$

5. Volta a 2. e repete até ser consistente com o conjunto de treino

Exemplo: Classificação de objetos

- Para o exemplo anterior, uma possível solução resultante da execução do algoritmo seria a dada pelos pesos e limiar indicados na figura



Exemplo: Classificação de objetos

- Não é solução única
- Exemplo de outra solução para o mesmo limiar:

$$W [\text{cor} = \text{escura}] = 0.3$$

$$W [\text{caudas} = 2] = 0.2$$

$$W [\text{núcleos} = 2] = 0.3$$

mantendo-se os restantes pesos.

Exemplo: Classificação de objetos

- As entradas correspondem às seis possibilidades de valores para os três atributos
- Se for presente à TLU uma entrada correspondente ao vetor

$$i = \langle 001000 \rangle$$

a saída será 1, correspondendo à **classe A**

Exemplo: Classificação de objetos

- Esta TLU reconhece corretamente todos os exemplos da **classe A** e não reconhece nenhum da **classe B**
- Dizemos nesse caso que o sistema é **completo** (reconhece todos os positivos) e **consistente** (não reconhece os negativos) relativamente ao conjunto de treino

Exemplo: Classificação de objetos

- Um modelo tão simples como este coloca vários problemas
- Um diz respeito à **garantia de convergência**, isto é, se ao fim de um tempo finito o algoritmo pára com uma solução
- A convergência nestes casos, por um lado, é lenta e, por outro, embora exista um teorema de convergência devido a *Rosenblatt*, ele depende da verificação de determinadas condições
- **A questão que se coloca é a de saber qual o poder computacional das TLU**

Exemplo: Classificação de objetos

4. Abordagem biológica

- O próximo algoritmo baseia-se nos princípios que governam a evolução das espécies naturais
- Trata-se de um algoritmo genético
- Os **Algoritmos Genéticos (AG)** são algoritmos de **procura cega**, **paralela** e **estocástica** guiados pelos princípios da seleção natural e da genética

Exemplo: Classificação de objetos

- As características principais de um AG são o facto de privilegiarem uma **representação binária** para a entrada e saída, não usarem teoria sobre o domínio para aprender (tendo características empíricas ou indutivas), funcionarem de modo **não supervisionado** e **não incremental** e estarem adaptados para **tarefas de classificação**, ou ainda, adaptados para **tarefas de resolução de problemas** envolvendo otimização

Exemplo: Classificação de objetos

Algoritmo Genético Simples

Dados:

- função de adaptabilidade ou mérito, f_a
- probabilidade de recombinação, p_r
- probabilidade de mutação, p_m
- critério de paragem, cp

Determinar:

- indivíduo que maximiza f_a

1. Definir aleatoriamente e avaliar a população inicial, p_0

2. Se existir um indivíduo em p_i que satisfaz cp **então** devolve esse indivíduo e pára!

3. Caso contrário

3.1. Selecciona indivíduos de p_i de acordo com f_a

3.2. Recombina os indivíduos de acordo com p_r

3.3. Muta os indivíduos de acordo com p_m

3.4. Define e **avalia** nova população p_{i+1}

3.5. Volta a 2.

Exemplo: Classificação de objetos

- O AG atua sobre uma **população** (de dimensão fixa ou variável) de indivíduos
- Normalmente a **população inicial** é gerada aleatoriamente
- Cada indivíduo é caracterizado pelo seu conjunto de **cromossomas** (normalmente um)
- A dimensão de um cromossoma pode também ser fixa ou variável
- Os cromossomas, por sua vez, são formados por **genes** que podem tomar diferentes valores (designados por **alelos**)
- O conjunto de material genético disponível constitui o **genótipo**
- Da interação do **genótipo** com o ambiente emerge um organismo designado por **fenótipo**

Exemplo: Classificação de objetos

- Para se utilizar um AG é necessário que o problema seja representado de acordo com os conceitos referidos: **população**, **indivíduo**, **cromossoma**, **gene**, **alelos**.
- É também necessário especificar qual a função para **avaliar a qualidade** dos indivíduos, em que consistem as operações de **recombinação** e **mutação** e respetivas probabilidades e definir o **critério de paragem** (número de gerações ou indivíduo com determinada qualidade)

Exemplo: Classificação de objetos

- O problema a resolver pelo AG consiste em encontrar um indivíduo de qualidade máxima de acordo com a função de avaliação

Exemplo: Classificação de objetos

- Consideremos o mesmo exemplo das duas abordagens anteriores: definir uma classe a partir de exemplos positivos e negativos da classe

Exemplo: Classificação de objetos

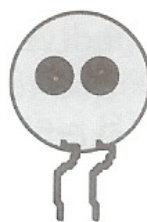
- Temos objetos (**células**) caracterizados por três atributos, cada um deles podendo assumir dois valores
- O facto de um objeto pertencer a uma classe pode ser representado por uma **regra** do tipo

Se <condição> então <classe>

Exemplo: Classificação de objetos

- Por exemplo, para o elemento da figura, podemos dizer que:

Se [cor=escura, cauda=2, núcleos=2] então Classe=A



Exemplo da classe A

Exemplo: Classificação de objetos

- Esta regra pode ser traduzida por uma cadeia binária do seguinte modo:

r1 = 10 10 10 1
(cor) (cauda) (núcleos) (classe)

10 – cor escura

01 – cor clara

11 – a cor é irrelevante

0 – Classe B

1 – Classe A

Exemplo: Classificação de objetos

- Este esquema pode ser facilmente estendido a situações em que os atributos têm mais do que dois valores possíveis, não sendo forçoso também que o conjunto de valores de cada atributo tenha a mesma cardinalidade

Exemplo: Classificação de objetos

- Sabemos como representar um objeto.
Mas o que serão os indivíduos ?
- Uma possibilidade é serem hipóteses de conceito formadas por **sequências de regras** como a indicada
- Tratando-se de sequências binárias, os genes e os alelos ficam definidos

Exemplo: Classificação de objetos

- **Como será avaliado cada indivíduo ?**
- Adotaremos o critério de considerar como mérito de um indivíduo a **percentagem de classificações corretas** que se obtêm quando é aplicada ao nosso conjunto de treino formado pelos 5 exemplos (3 positivos e 2 negativos)

Exemplo: Classificação de objetos

- Por exemplo, o indivíduo

i1 = 11 01 10 1 10 10 11 1 01 11 01 0

é formado por 3 regras, as duas primeiras para classificar um indivíduo como sendo da classe A, e a última para classificar na classe B

Exemplo: Classificação de objetos

- Tradução das regras:

r11 = Se [caudas=1, núcleos=2] então [classe=A]
r12 = Se [cor=escura, caudas=2] então [classe=A]
r13 = Se [cor=clara, núcleos=1] então [classe=B]

- Em termos de mérito teremos:

$$\text{mérito}(i1) = (0 + 2 + 0) / 5 = 2/5$$

ou seja, 40%

Exemplo:

Classificação de objetos

- No que diz respeito ao operador de **mutação** basta usar o operador simples que complementa um bit
- Já o problema do operador de **cruzamento** não pode ser resolvido de forma clássica sob pena de gerarmos indivíduos sem sentido
- Assim, iremos usar um operador de cruzamento de **dois pontos** que respeita a representação escolhida

Exemplo:

Classificação de objetos

- Exemplo: Sejam as hipóteses
$$h1 = 10\ 01\ 1\ 11\ 10\ 0$$
$$h2 = 01\ 10\ 0\ 10\ 11\ 1$$
- Suponhamos que são definidos aleatoriamente dois pontos de corte na primeira hipótese: pontos 1 e 7
$$h1 = 1|0\ 01\ 1\ 11\ |10\ 0$$

Exemplo: Classificação de objetos

- Calculemos a distância a que cada um destes pontos se encontra do início da primeira regra à sua esquerda
- Neste caso teremos $d1=1$ e $d2=2$
- Vamos agora forçar estes valores no segundo indivíduo
- Os pontos de corte possíveis serão os pares $(1,2)$, $(1,7)$ e $(6,7)$

Exemplo: Classificação de objetos

- Admitamos que é escolhido o primeiro par
- A segunda hipótese será então marcada do seguinte modo

$$h2 = 0|1|10010111$$

Exemplo: Classificação de objetos

- Agora basta trocar o material genético entre esses dois pontos, gerando dois novos indivíduos:

fh1 = 11 10 0
fh2 = 00 01 1 11 10 0 10 11 1

- Como se pode verificar neste exemplo, o comprimento dos cromossomos que representam os indivíduos não é fixo

Exemplo: Classificação de objetos

- Os restantes elementos do algoritmo são *standard* para este tipo de problemas:

Probabilidade de cruzamento	60%
Probabilidade de mutação	0.1%
Tamanho da população	100
Critério de paragem	10000 gerações

Exemplo: Classificação de objetos

- Usando agora o algoritmo iríamos obter uma população contendo um cromossoma:

$h = 11\ 11\ 01\ 1\ 10\ 10\ 10\ 1\ 11\ 11\ 10\ 0$

- Esta hipótese corresponde às três regras:

$r1h = \text{Se } [\text{núcleos}=1] \text{ então } [\text{classe}=A]$

$r2h = \text{Se } [\text{cor}=escura, \text{caudas}=2, \text{núcleos}=2] \text{ então } [\text{classe}=A]$

$r3h = \text{Se } [\text{núcleos}=2] \text{ então } [\text{classe}=B]$

- É fácil de constatar que o mérito desta hipótese relativamente ao conjunto de treino é de 100%

Exemplo: Classificação de objetos

- Os AG são algoritmos que promovem uma **procura paralela** no espaço das hipóteses de solução, guiada pelo mecanismo de seleção e pelos operadores genéticos
- Contrasta assim com os métodos simbólicos clássicos (um estado inicial e heurísticas dependentes do domínio para guiar a procura)

Exemplo: Classificação de objetos

São de referir alguns aspetos importantes:

- **Existem restrições computacionais**

Qual a combinação ótima tamanho da população / número de gerações ?

- **Adequação da representação binária**

Porque não usar vetores de números reais, regras, árvores, autómatos finitos ?

Exemplo: Classificação de objetos

- **Probabilidade de recombinação/mutação**

Valores muito baixos atrasam a convergência, valores altos introduzem procura aleatória
Que valores escolher ?

- **Foram referidos os operadores mais conhecidos e na forma mais simples**

Existem outros operadores (inversão) e variantes dos anteriores (recombinação por ordem, posição)

Que operadores usar ?

Exemplo: Classificação de objetos

- Existe com frequência o problema do AG ser atraído para máximos locais

Este problema está ligado à topologia do espaço de procura

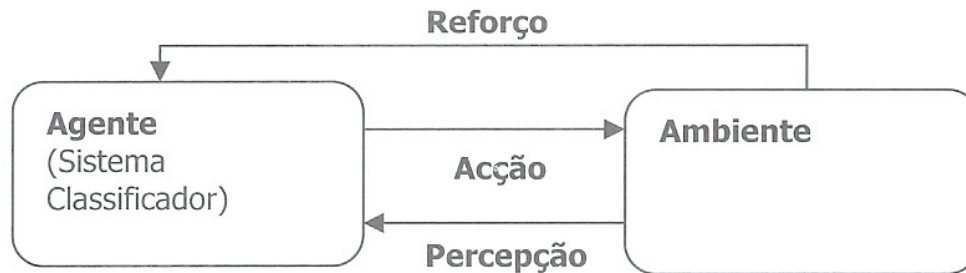
Como contrariar essa tendência ?

Sistemas Classificadores: Generalidades

- Um sistema classificador (SC) é um sistema semelhante a um sistema de produção mas capaz de aprender regras, denominadas classificadores.
- Essas regras permitem ao sistema interagir de forma adaptada com o ambiente.
- O nome de classificadores dado às regras tem a ver com o facto dessas regras permitirem classificar as mensagens que o ambiente envia ao sistema.

Sistemas Classificadores: Generalidades

- À semelhança de um sistema de controlo clássico, um **SC** usa a reação do ambiente para adaptar os classificadores ao ambiente.



Interacção de um SC com o ambiente

- A reação traduz-se por um sinal dito de **reforço**.

Sistemas Classificadores: Generalidades

- Quando os sistemas são não adaptativos, o sinal de reforço não existe ou é simples e está embutido nos sinais enviados pelo ambiente e capturados pelos mecanismos de percepção do agente.
- Um sistema classificador pode ainda ser visto como a **junção de sistemas periciais baseados em regras com os algoritmos genéticos**.
- O AG é utilizado para **aprender as regras que melhor se adaptam à tarefa e ao ambiente**, permitindo deste modo ultrapassar a dificuldade de adquirir conhecimento nos sistemas baseados em conhecimento clássicos.

Sistemas Classificadores: Generalidades

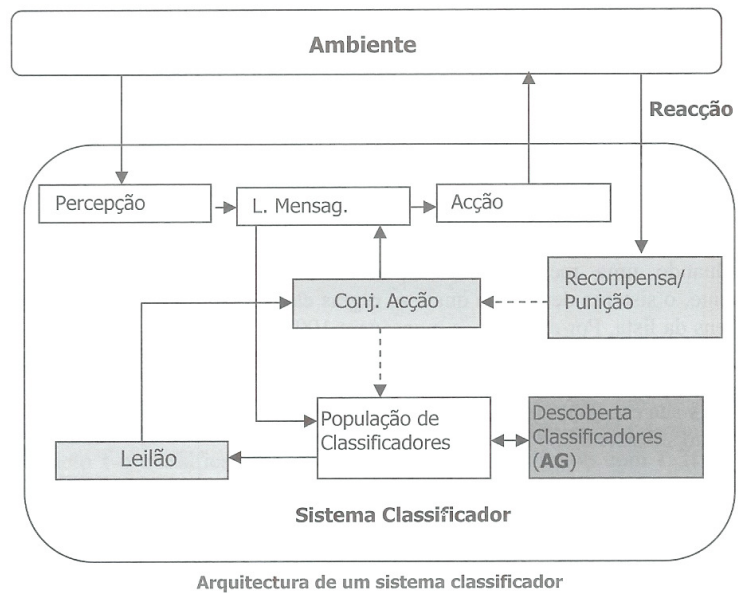
- Existem diferentes variantes de SC, mas que mantêm um conjunto das características comuns.
- As duas abordagens mais conhecidas:
 - *Abordagem de Michigan*
Cada elemento da população é (a representação de) *uma única regra*
 - *Abordagem de Pittsburgh*
Cada indivíduo da população é (uma representação de) *um sistema completo de regras*

Arquitectura de um Sistema Classificador

- Na *abordagem de Michigan* um sistema classificador é composto basicamente por três subsistemas:
 - Um subsistema de regras (ou classificadores) e mensagens;
 - Um subsistema de atribuição de crédito;
 - Um subsistema de descoberta de regras ou classificadores;

Arquitetura de um Sistema Classificador

- O subsistema de regras e mensagens é formado pelos módulos de percepção, ação, lista de mensagens e população de classificadores;
- O subsistema de atribuição de crédito é constituído pelos módulos conjunto de ação, leilão e recompensa/punição;
- O AG corresponde ao subsistema de descoberta de classificadores



Regras e Mensagens

- Cada regra ou classificador tem a forma:
Se <condição> então <ação> com força <n>
- muitas vezes também representadas por
<condição> : <ação> (<n>)

Regras e Mensagens

- A <condição> corresponde a uma palavra sobre o alfabeto {0, 1, #} e a <ação> a uma palavra sobre o alfabeto {0, 1}
- O símbolo # corresponde a uma situação em que é indiferente o valor ser 1 ou ser 0
- A força <n> é uma medida do desempenho passado da regra

Regras e Mensagens

- Cada classificador pode ser mais específico ou mais geral, sendo a especificidade dada pelo número de símbolos # presentes na condição.
- Uma condição apenas com #s tem especificidade mínima (igual a 0), enquanto uma condição sem #s tem uma especificidade máxima.
- A lista de mensagens descreve o estado corrente do ambiente.

Atribuição de Crédito

- Este subsistema está ligado ao modo como a força de uma regra é afetada pelo processo de aprendizagem.
- A modificação da força é feita através de três processos: **leilão**, **reforço** ou **punição** e **taxação**.

Atribuição de Crédito

- Cada **classificador** cuja condição é compatível com a mensagem captada pelo mecanismo de perceção **entra em competição** através de um processo de licitação num leilão.
- O classificador vencedor pode determinar o **envio de uma mensagem** (**ação**) **para o ambiente** que reage dando indicação se a ação foi “boa” ou “má”.
- Essa **reação**, **positiva** (**recompensa**) ou **negativa** (**punição**), vai determinar alterações na força da regra ativada.

Atribuição de Crédito

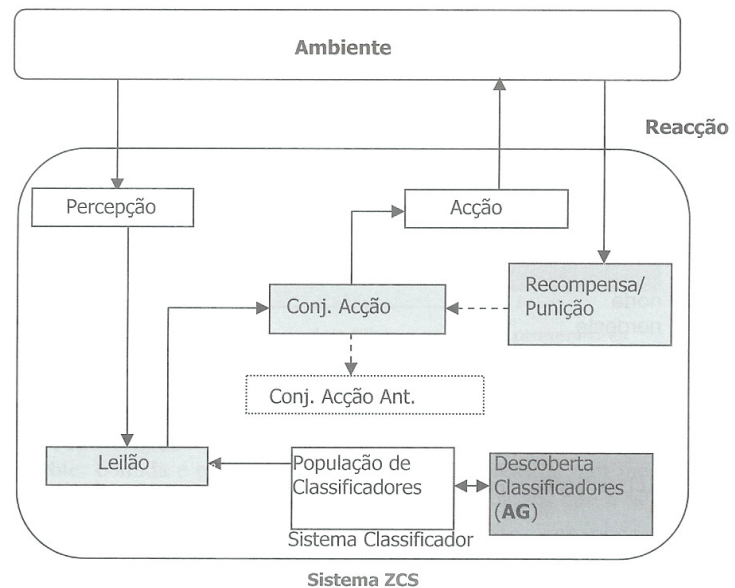
- O valor da **licitação do classificador vencedor** é pago aos classificadores que permitiram a sua seleção.
- Por outro lado, em cada ciclo de aprendizagem **cada classificador paga ainda uma taxa**, quer tenha licitado ou não.
- Deste modo, classificadores que não são ativados durante um longo tempo, e portanto não são úteis, vêem a sua força diminuída, pelo que a sua possibilidade de licitar e ganhar também se vai tornando menor.

Algoritmo Genético

- O algoritmo genético tradicionalmente empregue num **SC** é um **AG simples** que recorre a mecanismo de **seleção por roleta**, operador de **cruzamento de n pontos de corte** e **mutação por troca aleatória** do valor de um gene.
- A **função de mérito** corresponde à **força do classificador**.
- A atualização da população é feita substituindo uma dada proporção de indivíduos da população antiga pelos novos indivíduos gerados.
- **O mecanismo empregue favorece a substituição dos classificadores de menor força (ou mérito).**

Um Exemplo: O Sistema ZCS

- Vamos usar o sistema ZCS (Zeroth Classifier System) como exemplo – SC mais simples

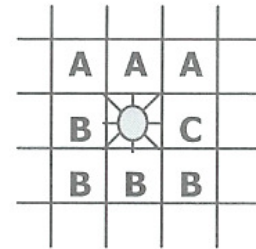


Um Exemplo: O Sistema ZCS

- O sistema ZCS tem três diferenças fundamentais:
 - Não existe lista de mensagens (todas as mensagens correspondem a ações sobre o ambiente);
 - O comprimento das condições e ações não é necessariamente o mesmo;
 - Para além do conjunto ação, é mantido o conjunto ação do ciclo anterior.

Um Exemplo: O Sistema ZCS

- A aplicação:
- Imaginemos um robô, **Animat**, que navega num espaço a duas dimensões à procura de comida e evitando obstáculos.
- À semelhança dos **agentes reativos**, **Animat** tem uma noção de vizinhança.
- Exemplo de uma situação concreta:



Robô Animat no espaço
a duas dimensões

Um Exemplo: O Sistema ZCS

- Os mecanismos de perceção do sistema irão retirar do ambiente a mensagem:
AACBBBBA
- **A** representa uma **árvore**, **C** comida e **B** um **espaço vazio** (em branco).
- De modo a que a mensagem seja binária, usa-se a codificação:
 - A – 01
 - B – 00
 - C – 11

Um Exemplo: O Sistema ZCS

- Com esta codificação a mensagem anterior será:

$m = 0101110000000001$

- Uma mensagem é uma cadeia binária de comprimento fixo igual a 16.

Um Exemplo: O Sistema ZCS

- As ações possíveis são: deslocar-se para norte, nordeste, este, ...
- Existem 8 possibilidades que podem ser codificadas por 3 bits:
 - 000 – norte
 - 001 – nordeste
 - 010 – este
 - ...
 - 111 – noroeste

Um Exemplo: O Sistema ZCS

- Os classificadores têm **condições** de comprimento 16 e **ações** de comprimento 3.
- **Exemplo:**
##011#00000#00# : 010 (50.2)
- Notar que a mensagem m é compatível com este classificador

m = 0101110000000001

Um Exemplo: O Sistema ZCS

- O sistema **Animat** apresenta alguns aspetos que vale a pena referir:
 - O conjunto inicial de classificadores está vazio;
 - Quando nenhum classificador é compatível com a mensagem enviada pelo ambiente é então criado um classificador;
 - O processo baseia-se na generalização da mensagem: alguns 1's e 0's são transformados em #s e formam a condição do classificador;
 - A ação é escolhida aleatoriamente e a força é também inicializada com um valor baixo.

Um Exemplo: O Sistema ZCS

- Exemplo:
Se a mensagem
 1000110000000101
- não for compatível com nenhum classificador, então pode ser gerado o classificador:
 $1\#001\#000000\#0\# : 011 (20)$

Um Exemplo: O Sistema ZCS

- A **geração de novas regras** é da responsabilidade do algoritmo genético.
- Em cada ciclo são **escolhidos** aleatoriamente, e proporcionalmente à sua força, **dois classificadores** que funcionarão como **progenitores**.
- Os filhos gerados irão substituir dois elementos escolhidos da mesma forma.

Um Exemplo: O Sistema ZCS

- A sua **força** será igual à média da força dos progenitores.

#11###00 0###0##0 : 000 (78) ###0#000 #0110000 : 001 (55)	Progenitores
#11###00 #0110000 : 001 (66.5) ###0#000 0###0##0 : 000 (66.5)	Filhos

Geração de dois filhos a partir dos progenitores

Um Exemplo: O Sistema ZCS

- Os resultados obtidos pelo **ZCS**, num ambiente como o da figura, mostram que, com uma **visão puramente local** e sem nenhum conhecimento específico sobre o domínio, ele é capaz de aprender a obter comida e evitar árvores, passando a necessitar, em média, de cerca de **3.06 passos** para obter comida (**27 passos** para estratégia aleatória e **1.7 passos** para estratégia ótima).

B	B	B	B	B
B	A	A	F	B
B	A	A	A	B
B	A	A	A	B
B	B	B	B	B

Resultado ZCS

Um Exemplo: O Sistema ZCS

- Aquele resultado foi obtido na seguinte situação:
 - AG é chamado com uma probabilidade de 0.25;
 - Probabilidade de cruzamento: 0.5;
 - Probabilidade de mutação: 0.002;
 - População: 400 classificadores;
 - O reforço recebido pelo ambiente é afetado por uma constante de 0.2;
 - Taxa de licitação, T_L : 0.1;
 - Factor de desconto da regra vencedora, C_L : 0.71;

Um Exemplo: O Sistema ZCS

- Os dez melhores classificadores obtidos, de entre os 400 gerados, após 5000 ciclos são:

```
#11###000###0##0 : 000 ( 86.66069)
#11###000###0##0 : 000 (78.39397)
###0000##0111000 : 101 (78.04076)
#11###000###0##0 : 000 (73.33034)
#0#0000##0111000 : 101 (65.70401)
#0#0000##0111000 : 101 (65.70401)
#0#0#000#01100#0 : 001 (64.49016)
#0#0000010###000 : 101 (60.01391)
###0#000#0110000 : 001 (55.10122)
#11###000###0##0 : 000 (53.33184)
```