P1 – Métodos de Classificação

Análise e Escolha das *Features*

Para o problema dado foram inicialmente escolhidas como *features* (para além do **(0)** comprimento da palavra); **(1)** o número de vogais; **(2)** se o comprimento é par; **(3)** se a palavra contém a letra ‘a’; e **(4)** se a palavra contém caracteres acentuados.

Após algumas análises às *features* enumeradas, concluiu-se que **(3)** a palavra conter a letra ‘a’e **(2)** o tamanho da palavra ser par são suficientes para uma boa classificação usando uma árvore de decisão.



Figura - Árvore gerada para as features acima, para o ficheiro de treino e teste wordsclass2.npy

Análise e Escolha de Parâmetros

Como obtivemos bons resultados com todos os classificadores experimentados (com valores de *cross validation* (*5-fold*) iguais a 1.0), e tendo em conta que não só os dados em análise são binários como estamos a trabalhar com um pequeno número de *features* com um número limitado de resultados, escolhemos utilizar como método classificador **uma árvore de decisão** (com número mínimo de divisão = 2). Para corroborar a nossas escolhas, efetuámos uma matriz classificativa de forma a sabermos o número de erros do tipo I (falsos positivos) e de tipo II (falsos negativos), tendo obtido 0 classificações erradas para os grupos de teste e treino dados.

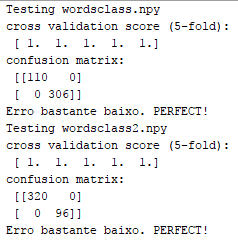


Figura - resultados de P1, com os valores da validação cruzada e matrizes de classificação para cada ficheiro de teste.

P2 – Métodos de Regressão

Métodos de aprendizagem

Utilizámos como métodos de regressão **kernel ridge regression** e **máquina de vetores de suporte (SVR)**.Ambos estes métodos aprendem funções não-lineares através do *kernel trick,* provando-se adequados aos conjuntos de dados em análise. Para ambos os métodos, foi utilizado como *kernel* **função de base radial (RBF)**.

Afinação de parâmetros

Para determinar a melhor combinação de parâmetros a aplicar aos métodos de regressão, foi efetuado um conjunto de testes com *cross validation (5-fold)* para vários valores de parâmetros, utilizando a ferramenta *GridSearchCV.* O melhor estimador foi escolhido por comparação do **erro quadrático médio** para a *cross validation* efetuada.

Para *kernel ridge* foram testados os parâmetros *alpha* e *gamma.* Para SVR testou-se *C* e *gamma.*

Comparação de resultados

P3 – Aprendizagem por Reforço

Ambiente 1

Representação Gráfica

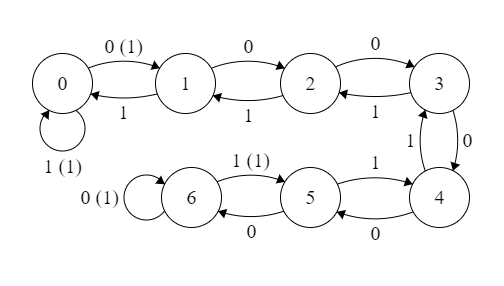


Figura - Representação gráfica do ambiente 1. As setas representam as ações possíveis, os valores entre parêntesis representam a recompensa associada à mesma ação.

Função de Recompensa

Em que

***Total Discounted Reward* =**

Tendo em conta o movimento do agente, a recompensa final é TDR = 0 + 1\*0.9 + 1\*0.9^2 +1\*0.9^3 = 2.439

Movimento do Agente

Para o trajeto recebido e tendo em conta o estado inicial, o caminho feito pelo agente é o seguinte:

Através da variação do estado inicial, pela análise dos valores de Q e da política (determinística) utilizada, a politica ótima do agente será:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Estado atual | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Ação | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Estado seguinte | 0 | 0 | 1 | 2 | 5 | 6 | 6 |

Ambiente 2

Representação Gráfica

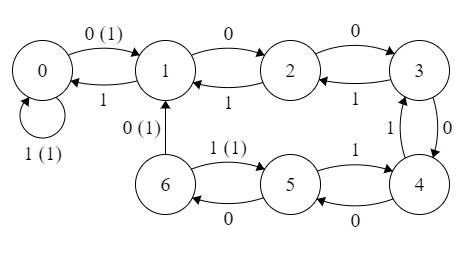


Figura - Representação gráfica do ambiente 2.

Função de Recompensa

Em que **TDR =**

Tendo em conta o movimento do agente, a recompensa final é TDR = 0 + 0 + 1\*(0.9^2) + 1\*(0.9^3) = 1.629

Movimento do Agente

Para o trajeto recebido e tendo em conta o estado inicial, o caminho feito pelo agente é o seguinte:

Através da variação do estado inicial e pela análise dos valores de Q e da política (determinística) utilizada, o movimento do agente será:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Estado atual | 0 | 1 | 2 | 3 | 4 | 5 | 6 |
| Ação | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| Estado seguinte | 0 | 0 | 1 | 2 | 5 | 6 | 1 |