

Relatório do Projeto 2 de Inteligência Artificial

Mihail Brinza

83533

Ricardo Brancas

83557

9 de Dezembro de 2017

1 Métodos de Classificação

Para classificar as palavras escolhemos de entre outras, com recurso a *cross validation*, as seguintes *features*:

1. Paridade do número de caracteres;
2. Paridade do número de caracteres “z”;
3. Paridade do número de vogais;
4. Paridade do número de consoantes;
5. Número de caracteres “a”.

Para escolher o classificador, usámos novamente *cross validation* com os classificadores *k-neighbors*, com $k = \{1, 3, 5, 7, 9\}$ e *decision tree*, obtendo os resultados indicados na tabela 1.

	Conjunto 1	Conjunto 2
<i>1-neighbor</i>	1.0	1.0
<i>3-neighbors</i>	1.0	1.0
<i>5-neighbors</i>	1.0	1.0
<i>7-neighbors</i>	1.0	1.0
<i>9-neighbors</i>	1.0	1.0
<i>Decision Tree</i>	1.0	1.0

Tabela 1: F_1 scores da validação cruzada para o problema 1.

Concluimos portanto que, para estas *features*, qualquer um dos classificadores testados escolhe sempre bem dentro do conjunto de treino. Mais tarde verificámos que a escolha também é sempre acertada no conjunto de testes.

Decidimos utilizar o *Decision Tree Classifier* porque escolhe corretamente com maior probabilidade mesmo quando utilizamos *features* piores.

2 Métodos de Regressão

Para escolher o método de regressão mais apropriado usamos *cross validation* pontuado pelo erro quadrático médio, obtendo os resultados dispostos na tabela 2.

¹Kernel Ridge Regression com radial basis function kernel.

²Kernel Ridge Regression com polynomial kernel.

	$g_1()$	$g_2()$
<i>Linear Regression</i>	0.94	816
KRR^1 ($\gamma = 0.05, \alpha = 0.1$)	0.67	1235
KRR^1 ($\gamma = 0.05, \alpha = 0.01$)	0.34	707
KRR^1 ($\gamma = 0.05, \alpha = 0.001$)	0.14	428
KRR^1 ($\gamma = 0.1, \alpha = 0.1$)	0.23	1265
KRR^1 ($\gamma = 0.1, \alpha = 0.01$)	0.10	811
KRR^1 ($\gamma = 0.1, \alpha = 0.001$)	0.10	548
KRR^1 ($\gamma = 0.2, \alpha = 0.1$)	0.40	1445
KRR^1 ($\gamma = 0.2, \alpha = 0.01$)	0.24	1104
KRR^1 ($\gamma = 0.2, \alpha = 0.001$)	0.66	799
KRR^2 (degree = 2)	2.27	3589
KRR^2 (degree = 3)	7.75	0.38
KRR^2 (degree = 4)	0.93	1.25
KRR^2 (degree = 5)	21.72	4.39
<i>Decision Tree</i>	0.73	1290

Tabela 2: *MSE scores* da validação cruzada para o problema 2. Os erros dentro do *threshold* definido estão marcados a azul.

Das duas parametrizações testadas que apresentamos resultados aceitáveis para ambas as funções decidimos utilizar a *Kernel Ridge Regression* com *kernels* do tipo *radial basis function* e parâmetros $\{\gamma = 0.1, \alpha = 0.001\}$. Apresentamos na figura 1 alguns dos resultados obtidos, em particular para a regressão escolhida.

3 Aprendizagem por Reforço

3.1 Trajetórias Aprendidas

3.1.1 Ambiente 1

$$5_0 \xrightarrow{0} 6_1 \xrightarrow{0} 6_1 \xrightarrow{0} 6_1 \xrightarrow{0} 6_{(1)}$$

3.1.2 Ambiente 2

$$5_0 \xrightarrow{0} 6_1 \xrightarrow{0} 1_0 \xrightarrow{1} 0_1 \xrightarrow{1} 0_{(1)}$$

3.2 Modelo do Mundo

3.2.1 Ambiente 1

Apresentamos na figura 2 um esquema do ambiente 1. Os movimentos são, maioritariamente,

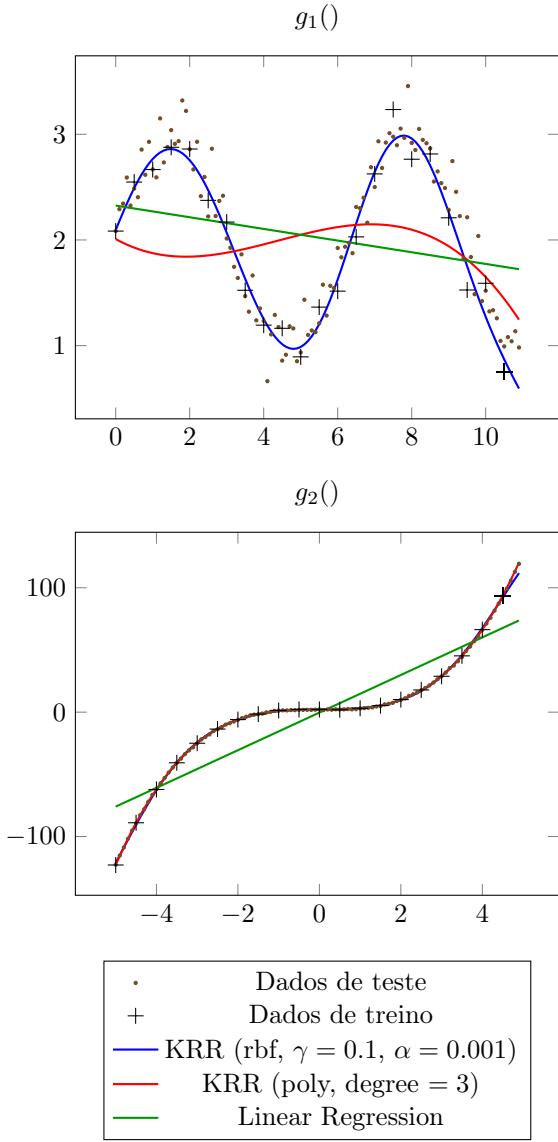


Figura 1: Resultados obtidos para as regressões.

sequenciais com exceção da ação 0 no estado 5 que é não-determinística. Tentar andar para os estados anteriores/seguíntes a partir dos estados 0/6, respetivamente, não tem qualquer efeito.

A função de recompensa é a seguinte:

$$r(s) = \begin{cases} 1, & s \in \{0, 6\} \\ 0, & s \notin \{0, 6\} \end{cases}$$

Assim, a melhor ação a tomar é sempre dirigirmo-nos o mais rapidamente possível para um dos estados de recompensa, tendo eventualmente cuidado com o não-determinismo do estado 5. Começando no estado 5, a melhor ação é sem duvida movermo-nos para o estado 6 e depois mantermo-nos lá.

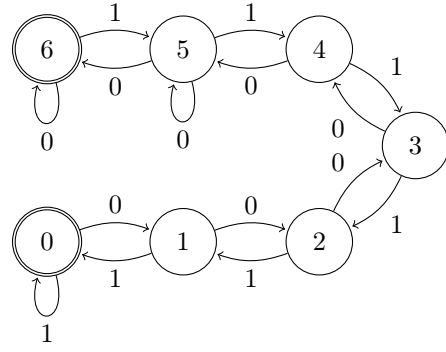


Figura 2: Ambiente 1. Os nós com duplo contorno são os nós de recompensa.

3.2.2 Ambiente 2

O ambiente 2 é muito semelhante ao primeiro com exceção do estado 6. Agora quando tentamos tomar a ação 0 nesse estado voltamos para o estado 1, tal como representado na figura 3.

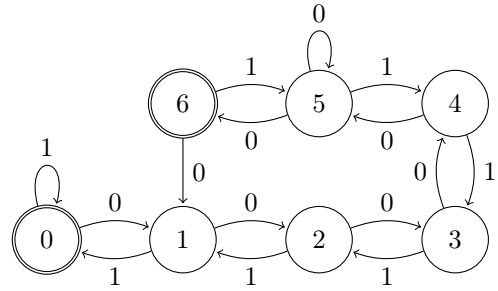


Figura 3: Ambiente 2. Os nós com duplo contorno são os nós de recompensa.

A função de recompensa mantém-se também inalterada:

$$r(s) = \begin{cases} 1, & s \in \{0, 6\} \\ 0, & s \notin \{0, 6\} \end{cases}$$

Como tal [quando começamos no estado 5] já não é possível usar a estratégia anterior de ficar parado no estado 6. Assim, o melhor a fazer nesta situação é tentarmos dirigir para o estado 0, de modo a maximizar a recompensa a longo prazo, que é exatamente o que obtemos com o algoritmo Q-learning.