Relatório 3 - Definição de parâmetros de métodos de otimização

BCC407/PCC170 - Projeto e Análise de Experimentos Computacionais

Programa de Pós-Graduação em Ciência da Computação

Departamento de Computação

Universidade Federal de Ouro Preto

Nome:	Fernando Henrique Oliveira Duarte
Matrícula:	2022.10629
Grau:	Doutorando
Orientador:	VANDER LUIS DE SOUZA FREITAS

1 Contextualização do problema

A utilização de árvores de decisão em conjuntos de dados é um problema que possui uma ampla variedade de aplicações reais, que buscam o reconhecimento de padrões através de classificação ou regressão dos dados. Em conjuntos de dados desbalanceados, a classificação de classes é cara, pois, geralmente, uma pequena fração do total de atributos (do inglês, *features*) está correlacionada a uma determinada classe de interesse (do inglês, *target*).

A classificação de problemas multi-classe são mais desafiadores que problemas de classe binária, pois a frequência relativa e o custo de cada uma das classes podem variar amplamente de conjunto de dados para conjunto de dados (Hoens et al. (2012)).

O problema escolhido para esta etapa é um exemplo de um problema de Árvore de Decisão na classificação de um conjunto de dados de Máquinas de Vetores de Suporte (Prado Lima (2022)) em que ele utiliza o Algoritmo Genético do irace para melhorar o processo de classificação. O autor não fornece informações sobre o que exatamente é o problema que está tentando resolver. Assim apresentamos na próxima subseção alguns trabalhos com cenário similar ao do problema.

1.1 Seleção de Atributos na Geração de Árvores de Decisão

O autor Fayyad et al. (1992) aborda o problema de seleção de atributos na geração de árvores de decisão. Neste trabalho é apresentado dois caminhos (fig. 1) que, segundo o autor, proporcionam a geração de melhores árvores de decisão do que algoritmos que usam medidas de impurezas.

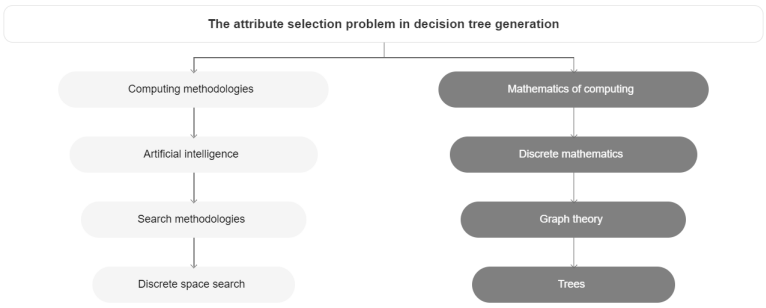

Figure 1: Duas abordagens para o problema de seleção de atributos na geração de árvores de decisão (Fayyad et al. (1992)).

No trabalho do autor Gavankar & Sawarkar (2017) foi proposto um algoritmo chamado de Árvore de Decisão Gulosa (do inglês, *Eager Decision Tree*) partindo de um único modelo de previsão, que considera todas as possibilidades de valores de atributos no momento do treinamento.

1.2 Algoritmos Genéticos e Árvores de Decisão

1.2.1 Problemas em Máquinas de Vetores de Suporte (SVMs)

Para resolver tarefas de diagnóstico de falhas, problema multi-classe, o autor Chen et al. (2011) combina SVM e árvore de decisão usando o conceito de dicotomia. Um algoritmo genético é introduzido na formação da árvore de decisão. A função do algoritmo genético é de identificar e separar as classes como nós da árvore de decisão. Segundo o resultado das simulações, demonstrou-se que o método proposto tem desempenho similar a métodos convencionais de classificação.

Prever tendências de mercado de ações é uma tarefa extremamente difícil devido a características não lineares do sistema. Neste cenário o autor Nair et al. (2010) apresenta em seu trabalho um sistema de previsão de tendências do mercado de ações baseado em mineração de dados. O sistema proposto é um híbrido de máquina de vetor de suporte de árvore de decisão otimizado por algoritmo genético, que tenta prever tendências de um dia à frente no mercado de ações.

O problema de regressão em máquinas de vetor é definido em como dividir um determinado conjunto de dados em dois clusters de forma que a margem entre os dois clusters atinja o máximo. No trabalho do autor Wang et al. (2005) é apresentado um algoritmo genético para definir a separação de clusters gerado por máquinas de vetores de suporte para a geração de árvores de decisão.

A fim de oferecer um melhor serviço aos clientes móveis, o autor Liu & Fan (2014) em seu trabalho, apresenta um algoritmo de árvore de decisão para classificação de processos de utilização do serviço por seus usuários. Ele também introduz um algoritmo genético para otimizar os resultados do algoritmo de árvore de decisão.

1.2.2 Problemas em Máquinas de Vetores de Suporte (SVMs)

Invasões a sistemas são comumente relatadas, a detecção dessas intrusões faz parte do sistema de rede de diversas empresas. O autor Azad & Jha (2015) propõe um sistema de detecção de intrusão baseado em algoritmo genético em árvore de decisão. O algoritmo genético é usado para melhorar a cobertura das regras que lidam com o problema de intrusão.

2 Descrição dos parâmetros

Os parâmetros foram definidos como os do autor. Onde é considerado uma variação de população de 50 a 300. Mutação e cruzamento variando de 1% a 100%.

Parâmetro	Identificador	Tipo	Variação
population	”–pop=”	i	(50, 300)
crossover	”–cros=”	r	(0.01, 1.0)
mutation	”–mut=”	r	(0.01, 1.0)

Table 1: Parameters.txt

2.1 Características do Problema

O problema possui um total de três mil instâncias, em que cada instância possui cento e trinta e dois atributos (*features*). Cada instancia está relacionada a uma das dez classes (*targets*). As instâncias são definidas pelo autor do repositório Prado Lima (2022).

2.2 Descrição das máquinas e Passmark

Dois experimentos foram realizados em um hardware, Intel Core i7-10700 CPU, 2.90GHz (PassMark CPU de 16789). E outro experimento no hardware Apple M1 Pro 8 Core 3.2 GHz (PassMark CPU de 17245).

3 Relato do experimento

Os dados foram gerados a partir das ferramentas Oñate Marín et al. (2022) e de Souza et al. (2021). Os experimentos 1 (E1) e 2 (E2) foram realizados na máquina Abstergo (Intel I7) e o experimento 3 (E3) no Mac.

4 Dados do experimento

Indicar o *link* temporário contendo códigos-fonte, instâncias e arquivos de configuração, para validação dos resultados.

	ID	n_instances	mean	sd	median	min	max
E1	83	8	-86.26	0.96	-86	-87.9	-85.2
	52	8	-86.22	1.01	-86.1	-87.9	-85
	143	8	-86.2	1.13	-86.45	-87.4	-83.9
E2	119	8	-86.25	1.16	-86.45	-87.3	-83.6
	144	8	-86.13	1.14	-85.9	-88.2	-84.8
	108	8	-86.01	0.95	-85.95	-87.2	-84.4
E3	64	9	-80.52	1.20	-80.29	-81.95	-78.29
	89	9	-80.31	1.19	-80.125	-82.16	-78.54
	72	9	-80.01	2.67	-80.75	-82.41	-74

Table 2: Desempenho das melhores configurações para as instâncias de treinamento.

Na tabela acima (tab.2) temos que os experimentos E1 e E2 tem resultados aproximados. Esse fato se deve a eles utilizarem 33% do dataset para teste e 67% para treino e validação (50% para cada). Já o E3 teve 80% de dados do dataset para teste e apenas 20% em treinamento e validação (50% para cada).

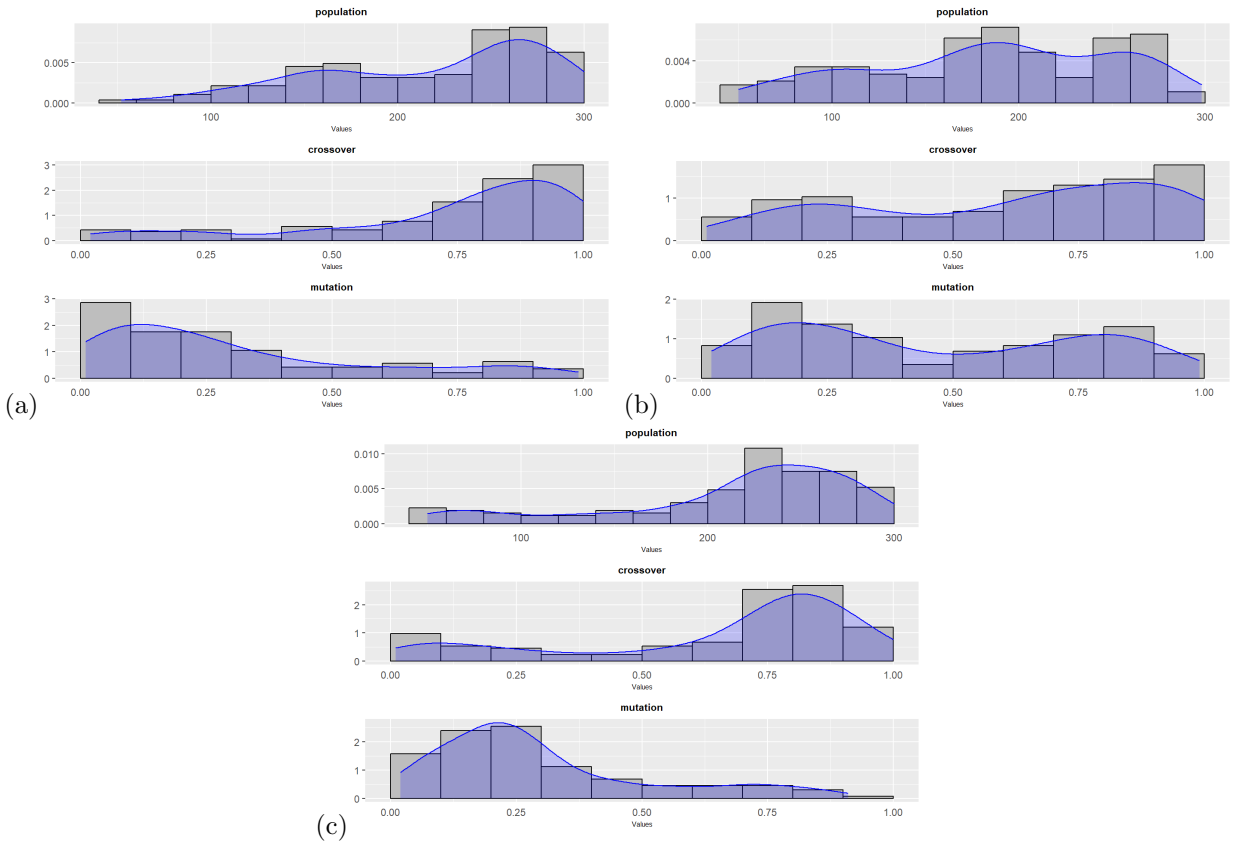

Figure 2: Curvas de variação dos parâmetros (a) E1 (b) E2 (c) E3.

Nas figuras acima (fig.2) temos as curvas de máximos e mínimos encontradas nos experimentos E1, E2 e E3, respectivamente. Neste problema procuramos os maiores valores. Podemos observar que os picos variam de posição em cada gráfico, isso se deve a que cada experimento não utilizamos os mesmo dados de entrada para treinamento, escolhidos de modo estratificado. Os melhores valores encontrados dos parâmetros estão na tabela (tab.3) de cada experimento.

Finalmente podemos observar na (fig.3) a evolução de como o algoritmo genético trabalha para conseguir tunar os parâmetros. Mesmo se tratando de um dataset de experimentação, vemos que ele possui outliers e que

	.ID.	population	crossover	mutation
E1	83	278	0.93	0.12
	52	264	0.98	0.07
	143	292	0.8	0.13
E2	119	202	0.68	0.31
	144	262	0.94	0.11
	108	196	0.86	0.27
E3	64	256	0.84	0.2
	89	234	0.83	0.25
	72	264	0.84	0.31

Table 3: Melhores valores de parâmetros conhecidos (BKV) encontrados para cada experimento.

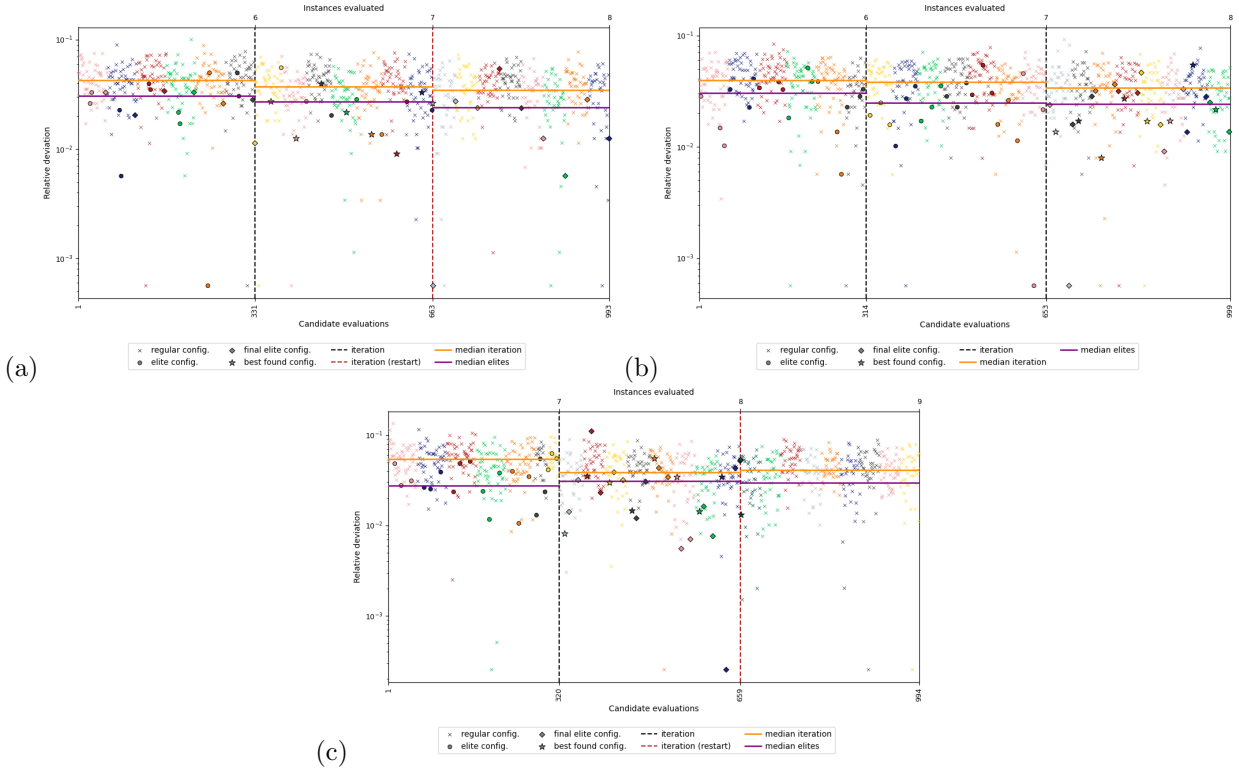

Figure 3: Evolução dos experimentos (a) E1 (b) E2 (c) E3 buscando os melhores parâmetros.

estes são descartados. Isso mostra que o algoritmo genético consegue eliminar os parâmetros ruins e se manter dentre de uma bolha que representa as classes.

4.1 Declarações Finais

Não foi possível gerar um arquivo com os melhores valores encontrados (BKV.txt), pois não encontrei nenhuma referência para isso no irace (López-Ibáñez et al. (2016)) ou no Acviz (de Souza et al. (2021)). O pacote AcViz permite criar um ranqueamento e média desses dados.

Não consegui recuperar os dados dos terminais para os experimentos E1 e E2, apenas para o E3, por isso não o inseri, mas está no Github. Existem mais alguns dados e métricas bem interessantes que deixei no Github, geradas pelos pacotes AcViz e iraceplot. Não foram inseridos por eu não saber interpretá-los.

Acredito que o pacote irace é excelente na otimização de parâmetros, mas foi um pouco difícil apresentar os dados, pois seu guia é apenas um "How To Do" e não existe documentação clara caso ocorra erros ou até mesmo como os dados são armazenados em arquivos *.Rdata*.

5 Dados do experimento

Os dados para verificação e replicabilidade do experimento estão disponíveis no link Github.

References

Azad, C., & Jha, V. K. 2015, International Journal of Computer Network and Information Security, 7, 56

Chen, H., Wang, Q., & Shen, Y. 2011, Journal of Systems Engineering and Electronics, 22, 322

de Souza, M., Ritt, M., López-Ibáñez, M., & Cáceres, L. P. 2021, Operations Research Perspectives, 8, 100

Fayyad, U. M., Irani, K. B., et al. 1992, in AAAI, Vol. 92, 104–110

Gavankar, S. S., & Sawarkar, S. D. 2017, in 2017 2nd International Conference for Convergence in Technology (I2CT), IEEE, 837–840

Hoens, T. R., Qian, Q., Chawla, N. V., & Zhou, Z.-H. 2012, in Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 122–134

Liu, D.-s., & Fan, S.-j. 2014, The Scientific World Journal, 2014

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., & Stützle, T. 2016, Operations Research Perspectives, 3, 43

Nair, B. B., Mohandas, V., & Sakthivel, N. 2010, International journal on computer science and engineering, 2, 2981

Oñate Marín, P., Pérez Cáceres, L., & López-Ibáñez, M. 2022, iraceplot: Plots for Visualizing the Data Produced by the 'irace' Package

Prado Lima, J. 2022, Examples for execute irace: Iterated Racing for Automatic Algorithm Configuration

Wang, X.-z., He, Q., Chen, D.-G., & Yeung, D. 2005, Neurocomputing, 68, 225