

ANÁLISE COMPARATIVA ENTRE OS ALGORITMOS C4.5 E C5.0 PARA IMPLEMENTAÇÃO NO CODIFICADOR DE VÍDEO HEVC

VINICIUS SIGNORI FURLAN; GUILHERME CORRÊA; LUCIANO AGOSTINI

Universidade Federal de Pelotas – Grupo de Arquiteturas e Circuitos Integrados
{vsfurlan, gcorrea, agostini}@inf.ufpel.edu.br

1. INTRODUÇÃO

A rápida evolução tecnológica dos dispositivos capazes de reproduzir mídia devido ao continuo aumento na capacidade computacional, combinada com a necessidade de reproduzir vídeos em alta resolução, resultou na necessidade de novas técnicas de compressão de vídeo, já que tais vídeos possuem uma grande quantidade de dados quando não comprimidos. Codificadores de vídeos adotam padrões, como o *High Efficiency Video Coding (HEVC)* (ITU-T, 2013), para implementar tais técnicas. O HEVC é atualmente o estado-da-arte entre os padrões e é capaz de reduzir em até 50% o *bit rate* – número de bits por segundos transmitidos por uma rede digital – de um determinado vídeo quando comparado com seus antecessores como o H.264/AVC (*Advanced Video Coding*) (ITU-T, 2003).

No entanto, o padrão HEVC implementa diversas novas ferramentas para atingir tais resultados e por esse motivo aumenta a complexidade de codificação em até 500% quando comparado ao seu predecessor, o H.264/AVC *High Profile*. Por esse motivo, é necessária a investigação e implementação de algoritmos capazes de reduzir tal complexidade, visando a maior redução possível no *bit rate* resultante (CORRÊA et al., 2012).

O trabalho de (CORRÊA et al., 2015) propõe a utilização de árvores de decisão, que permitem reduzir em até 86% a complexidade computacional do codificador HEVC. Porém, tais árvores de decisão precisam ser treinadas primeiramente, com um conjunto pré-definido de sequências de vídeo, para então serem implementadas de forma estática no codificador. Como os vídeos digitais são muito variáveis e as operações do processo de codificação são muito dependentes do conteúdo, é importante investigar a possibilidade de implementação de um algoritmo capaz de treinar árvores de decisão durante a codificação do vídeo, de forma que estas sejam mais apropriadas para o conteúdo em questão.

O C4.5 é um algoritmo utilizado para construção e interpretação de árvores de decisão e geração e interpretação de regras de produção, implementado em C (QUINLAN, 1993). Este algoritmo foi escolhido por ser considerado um dos 10 melhores algoritmos para mineração de dados, como visto em (WU et al., 2008) e para fins de comparação, tendo em vista que este algoritmo também foi utilizado em (CORREA et al., 2015).

Outro algoritmo que deve ser ressaltado é o algoritmo C5.0, uma variante do C4.5, que além de possuir todas as ferramentas presentes no C4.5 possui novas funcionalidades, como custo de erro de classificação de variáveis e suporte a novos tipos de variáveis (RULEQUEST, 2015). É importante ressaltar que o algoritmo tem sido defendido na literatura como uma alternativa mais rápida e precisa quando comparada ao C4.5 e por esse motivo foi escolhido para esta comparação.

Este resumo tem, portanto, o objetivo de analisar e comparar os algoritmos C4.5 e C5.0 em termos de tempo de execução e de precisão das árvores geradas, com a finalidade de escolher aquele que apresenta os melhores resultados para uma futura implementação no codificador de vídeo HEVC.

2. METODOLOGIA

Para a realização deste trabalho a seguinte metodologia foi adotada. Primeiramente, foi realizado o processo de coleta de informações de dez vídeos de diferentes resoluções com três configurações de tamanho das unidades de codificação que permitem sub-particionamento (16x16, 32x32, 64x64 pixels). Todas as informações coletadas dizem respeito a características do vídeo como variância entre pixels, profundidade das unidades de codificação vizinhas, entre outras. Além disso, para cada unidade de codificação, foi coletada a informação relativa ao particionamento da mesma (isto é, se houve sub-particionamento ou não). A coleta de dados foi realizada com a versão 12.0 do *HEVC Test Model*, utilizando parâmetro de quantização igual a 32 e a configuração *Random Access, Main*.

Após a etapa de coleta, a informação correspondente a cada vídeo foi reorganizada em dois grupos, onde 25% das informações foi utilizada para a geração das árvores de decisão (treinamento) e os outros 75% das informações foram utilizados para testar as árvores geradas (validação). Desta maneira, obtém-se o *dataset* necessário para as etapas de treino e de validação.

As árvores de decisão de cada um dos vídeos foram geradas e testadas em ambos os algoritmos (C4.5 e C5.0), computando o tempo de execução dos mesmos para cada uma das trinta árvores geradas – três árvores de decisão correspondentes aos três tamanhos de unidades de codificação para cada um dos dez vídeos.

Quanto à taxa de acerto, as árvores geradas são testadas com as informações de teste do *dataset* a fim de se comparar se as decisões das árvores foram iguais às tomadas pelo codificador sem a utilização das árvores. As árvores decidem se uma unidade de codificação deve ou não ser sub-particionada em outras unidades.

Finalmente, os algoritmos foram comparados em termos de precisão das árvores geradas e em termos de tempo de treinamento.

3. RESULTADOS E DISCUSSÃO

Utilizando o *dataset* gerado na primeira etapa da metodologia, os algoritmos geraram os resultados relacionados ao tempo de treinamento e taxa de acerto para cada árvore de decisão (Tabela 1 e 2, respectivamente). Cada linha da Tabela 1 apresenta o tempo de treinamento das três árvores de decisão geradas para o respectivo vídeo, considerando os algoritmos C4.5 e C5.0.

Observando a Tabela 1, é possível notar uma grande discrepância de tempo entre os algoritmos, onde o C5.0 se mostrou mais rápido para a geração das árvores de decisão. Quanto à taxa de acertos observada na Tabela 2, novamente o algoritmo C5.0 se mostrou mais preciso, salvo duas exceções, nos vídeos *ChinaSpeed* para unidades de codificação 64x64 e *BQSquare* para unidades de codificação 32x32, sendo que a diferença entre as porcentagens foi de apenas 0.1% em ambos os casos.

Tabela 1 - Tempo de geração das três árvores de decisão em segundos para os algoritmos C4.5 e C5.0.

	16x16		32x32		64x64	
	C4.5 (s)	C5.0 (s)	C4.5 (s)	C5.0 (s)	C4.5 (s)	C5.0 (s)
BasketballDrill 832x480	8.3	0.91	0.7	0.27	0.08	0.07
BasketballPass 416x240	0.54	0.16	0.13	0.06	0.02	0.001
BQSquare 416x240	0.51	0.24	0.12	0.06	0.01	0.01
BQTerrace 1920x1080	390.1	4.95	19.38	1.14	0.92	0.28
Cactus 1920x1080	403.04	5.0	18.68	1.31	0.95	0.22
ChinaSpeed 1024x768	58.1	2.1	2.14	0.42	0.24	0.12
Kimono1 1920x1080	425.2	6.5	18.92	1.36	0.92	0.32
PeopleOnStreet 2560x1600	418.8	6.48	122.46	2.58	2.55	0.51
SlideEditing 1280x720	53.35	2.02	2.22	0.45	0.18	0.16
SteamLocomotive 2560x1600	526.03	6.5	124.5	2.83	3.18	0.56

Tabela 2 - Taxa de acerto em porcentagem das três árvores de decisão para os algoritmos C4.5 e C5.0.

	16x16		32x32		64x64	
	C4.5 (%)	C5.0 (%)	C4.5 (%)	C5.0 (%)	C4.5 (%)	C5.0 (%)
BasketballDrill 832x480	76.6	77.1	81.1	82.9	85.3	85.3
BasketballPass 416x240	77.8	79.8	83.0	83.0	81.5	81.9
BQSquare 416x240	83.6	83.6	84.7	84.6	77.2	77.3
BQTerrace 1920x1080	77.0	78.6	81.4	83.0	83.9	85.7
Cactus 1920x1080	80.1	80.5	81.8	82.0	88.0	88.4
ChinaSpeed 1024x768	78.8	80.0	82.6	83.0	81.4	81.3
Kimono1 1920x1080	71.9	72.0	74.1	75.2	78.7	79.3
PeopleOnStreet 2560x1600	81.3	81.5	83.8	84.3	89.4	90.2
SlideEditing 1280x720	72.6	72.8	70.7	72.0	78.3	80.7
SteamLocomotive 2560x1600	72.8	73.8	76.2	77.4	82.2	83.4

4. CONCLUSÕES

Neste trabalho, foi realizada uma análise comparativa entre os algoritmos C4.5 e onde se pode concluir que o algoritmo se mostra superior ao C4.5 tanto em tempo de geração das árvores de decisão como em taxas acerto. Desta

maneira, é possível afirmar que o algoritmo seria a melhor escolha de implementação em um codificador de vídeo HEVC.

Como trabalho futuro, pretende-se desenvolver um algoritmo capaz de incorporar as ferramentas de geração de árvores do no codificador de vídeo HEVC, permitindo assim a capacidade do codificador de gerar árvores durante a codificação de um vídeo mais precisas e em menor tempo quando comparado aos resultados obtidos em (CORRÊA et al., 2015), onde as árvores são geradas *offline* a partir de um conjunto pré-definido de vídeos.

5. REFERÊNCIAS BIBLIOGRÁFICAS

ITU-T. International Telecommunication Union. **Recommendation ITU-T H.265: High Efficiency Video Coding**. April, 2013.

ITU-T. Joint Video Team (JVT) of ITU-T VCEG and ISO/IEC MPEG. **Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification, ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC**, May 2003.

CORRÊA, G.; ASSUNCAO, P.; AGOSTINI, L.; DA SILVA CRUZ, L. A. Performance and Computational Complexity Assessment of High Efficiency Video Encoders. **IEEE Transactions on Circuits and Systems for Video Technology**, v.22, p 1899-1909, 2012.

CORRÊA, G.; ASSUNCAO, P. A.; AGOSTINI, L. V.; da Silva Cruz, L. A. Fast HEVC encoding decisions using data mining. **IEEE Transactions on Circuits and Systems for Video Technology**, v25, n.4 p.660-673, 2015.

QUINLAN, J. R. **C4.5: Programs for Machine Learning**. California: Morgan Kaufmann Publishers, 1993.

RULEQUEST. **See5/C5.0**. Rulequest Research Data Mining Tools, 2011. Acessado em 25 de julho de 2016. Disponível em <http://rulequest.com/see5-info.html>

WU, X.; KUMAR, V.; QUINLAN, J. R.; GHOSH, J.; YANG, Q.; MOTODA, H.; MCLACHLAN, G. J.; NG, A.; LIU, B.; YU, P. S.; ZHOU, Z.; STEINBACH, M.; HAND, D. J.; STEINBERG, D. Top 10 algorithms in data mining. **Knowledge and Information Systems**. Springer-Verlag. v.14, n.1, p. 1-37, 2008.