

Introdução a Árvores de Decisão Adaptativas

F. Catae, R. L. A. Rocha

Abstract— A árvore de decisão é utilizada para classificar dados com base nos seus atributos. A estrutura é baseada em regras, que consegue mapear dados previamente observados em possíveis casos futuros. A adaptatividade permite que uma estrutura se auto-modifique como resposta às mudanças de ambiente, incorporando novas informações. No entanto, a adaptatividade não se restringe às modificações estruturais do núcleo. Através do não-determinismo, adiciona-se possibilidade de enumerar diferentes modelos e determinar o melhor deles usando uma métrica comum, como o conceito da complexidade de Kolmogorov. Em uma árvore de decisão adaptativa, há a possibilidade de criar novos atributos em função dos já existentes. Esses "atributos calculados" estendem sua capacidade para lidar com problemas simples, como paridade e multiplexação, mas que não podem ser bem modelados por uma árvore de decisão tradicional.

Palavras-chaves—Aprendizagem de Máquina, Árvore de Decisão, Tecnologia Adaptativa

I. INTRODUÇÃO

Nos últimos anos, observa-se maior disponibilidade de informações por meios digitais acompanhada pelo aumento da capacidade de armazenamento. Sabe-se que o excesso de informação afeta negativamente a produtividade e, por isso, torna-se necessária a utilização de ferramentas de apoio. Um sistema inteligente pode, por exemplo, filtrar as informações desnecessárias e mantê-las ordenadas de acordo com sua relevância.

A automação de processo pode ser realizada através de um dispositivo composto por conjuntos de regras e ações armazenados em uma árvore de decisão. Dessa forma, é possível classificar e ordenar um grande volume de dados, que seriam inviáveis de se realizar manualmente.

No campo de aprendizado de máquina a estrutura de árvore de decisão é utilizada para inferir propriedades e atributos sobre informações até então desconhecidas. Nesse caso, a árvore de decisão mapeia os dados previamente observados em possíveis casos futuros.

A tecnologia adaptativa [1] está diretamente relacionada à questão de aprendizagem de máquina. Um dispositivo adaptativo é baseado em um mecanismo base com a

capacidade de apresentar modificações na própria estrutura. Por exemplo, uma árvore de decisão adaptativa [2] pode assimilar novas informações através da inclusão de nós e ramos em sua estrutura.

II. CLASSIFICAÇÃO DE DADOS

Ao observar uma revoada de pássaros voando no céu, imaginamos que sejam todos da mesma espécie. Entretanto, sabemos que existem atributos que os diferenciam entre eles: formato dos olhos, tom das penas, tamanho das asas. Assim como existem outros, que são formas características do grupo: formato do bico e dos pés, proporção do tamanho da asa em relação ao corpo. Nesse cenário, a classificação corresponde ao problema inverso: a partir dos atributos de um indivíduo, seria possível descobrir qual a sua espécie?

O problema apresenta um universo de objetos, os quais podem ser descritos individualmente por meio de seus atributos. Um objeto possui uma série de atributos e pertence a uma única classe. A classificação pode ser feita por meio de regras, associando conjuntos de atributos às classes correspondentes. Por exemplo, o estudo de aves migratórias poderia ser modelado da seguinte forma:

Atributos:

Asas = { atrofiadas, desenvolvidas }

Bico = { curto, longo }

Corpo = { pequeno, grande }

Classes:

Espécie Migratória = { sim, não }

As regras são criadas com base nos estudos realizados sobre as diferentes espécies, buscando associar a morfologia ao comportamento de migração. Uma hipótese seria afirmar que somente as aves com asas desenvolvidas, bico longo e corpo grande são migratórias.

Regras:

{ desenvolvidas, curto, pequeno } → não

{ desenvolvidas, longo, pequeno } → não

{ desenvolvidas, curto, grande } → não

{ desenvolvidas, longo, grande } → sim

{ atrofiadas, curto, pequeno }

→ não

{ atrofiadas, longo, pequeno }

→ não

{ atrofiadas, curto, grande }

→ não

{ atrofiadas, longo, grande }

F. Catae é estudante de Mestrado junto ao Laboratório de Linguagens e Técnicas Adaptativas, Escola Politécnica da USP; e-mail: fcatae@usp.br.

R. L. A. Rocha é pesquisador do Laboratório de Linguagens e Técnicas Adaptativas, Escola Politécnica da USP, São Paulo/SP, Brasil; fone: 55 11-3091-5583; e-mail: luis.rocha@poli.usp.br.

→ não

A intenção é usar as regras para classificar indivíduos de acordo com seus atributos (asas, bico, corpo) em aves migratórias ou não.

O processo pode ser aplicado para os diferentes tipos de aves a partir de sua descrição. Por exemplo, observamos que o pelicano é uma ave migratória, enquanto que o avestruz e o pinguim não são.

indivíduo	= (asas, bico, corpo)
<i>avestruz</i>	= (<i>atrofiadas, curto, grande</i>)
<i>pinguim</i>	= (<i>atrofiadas, curto, pequeno</i>)
<i>pelicano</i>	= (<i>desenvolvidas, longo, pequeno</i>)

As regras foram dispostas semelhante a árvore de decisão, e representadas através de um autômato de estados finitos, como em ilustrado na Fig 1.

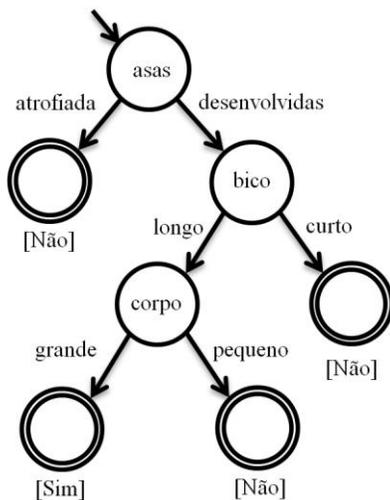


Figura 1. Representação das regras de classificação na estrutura de árvore de decisão

Formalmente, define-se um autômato de estados finitos determinístico por um quintupla $(Q, \Sigma, \delta, s_0, F)$ [3], no qual:

- Q – Conjunto finito de estados
- Σ – Conjunto finito de símbolos (alfabeto)
- δ – Transições entre estados, $\delta: Q \times \Sigma \rightarrow Q$
- s_0 – Estado inicial, $s_0 \in Q$
- F – Conjunto de estados finais, $F \subseteq Q$

Dado uma entrada inicial w composto por uma cadeia de símbolos de Σ , o autômato processa a informação a partir do estado inicial s_0 e realiza transições de estados conforme a função δ . Se a cadeia w apresenta tamanho n , então após n iterações o processamento finaliza no estado q . Dizemos que um autômato aceita w , quando o estado final q pertence ao conjunto F .

A representação de uma árvore de decisão em autômatos apresenta os nós e ramos mapeados para o conjunto de estados Q e transições δ . O alfabeto é definido pela união dos conjuntos de atributos, enquanto que as classes são partições do conjunto de estados finais.

No exemplo do estudo de aves, estamos interessados em determinar se o autômato finaliza em um estado que representa a classe de aves migratórias ou não. Assim, os conjuntos poderiam ser definidos da seguinte forma:

$$A = \{ \text{atrofiadas, desenvolvidas} \}$$

$$B = \{ \text{curto, longo} \}$$

$$C = \{ \text{pequeno, grande} \}$$

$$F_{[SIM]} = \{ q \in Q \mid \text{classe de aves migratórias} \}$$

$$F_{[NAO]} = \{ q \in Q \mid \text{classe de aves não migratórias} \}$$

$$\Sigma = A \cup B \cup C$$

$$F = (F_{[SIM]}) \cup (F_{[NAO]})$$

Permitindo a classificação de objetos que apresentem a descrição através da cadeia de entrada.

$$w = w_a w_b w_c, \text{ tal que } w_a \in A, w_b \in B, w_c \in C$$

O autômato classifica a ave em “sim” se o estado final pertencer ao conjunto $F_{[SIM]}$, caso contrário, o processamento termina $F_{[NAO]}$ e a resposta será “não”.

III. APRENDIZAGEM

Com o passar do tempo, novas regras de classificação podem surgir por conta de situações não consideradas inicialmente. Essas mudanças se refletem na mudança do autômato, forçando a criação de uma nova estrutura de classificação. A aprendizagem permite que um autômato incorpore as mudanças em sua própria estrutura de forma incremental, sem a necessidade de ser refeito. Consideramos, como exemplo, a classificação de uma andorinha seguindo as regras anteriormente definidas.

$$\text{andorinha} = (\text{desenvolvidas, curto, pequeno})$$

Apesar de conhecido o fato das migrações sazonais das andorinhas, o classificador por regras define que as características de asas desenvolvidas, bico curto e corpo pequeno correspondem a classe de aves não-migratórias. Esse comportamento pode ser corrigido durante o estágio de aprendizado, alterando-se os estados e transições do autômato.

Quando uma estrutura reage às mudanças impostas a ela e se auto-modifica, dizemos que houve uma adaptação. Essa adaptatividade incorpora informações do meio externo, permitindo o aprendizado incremental.

Na área de Tecnologia Adaptativa, estudam-se dispositivos que apresentam mudança de comportamento através de alterações em seu próprio núcleo. Um autômato adaptativo é uma extensão do autômato de estados finitos acrescido de regras de auto-modificação [1].

Um resultado relevante é que os autômatos de estados

finitos reconhecem somente linguagens regulares, enquanto que os autômatos adaptativos apresentam poder de Máquina de Turing [4].

O aprendizado do autômato ocorre por meio das operações elementares de consulta e inserção. A implementação da Adaptive-NDD Tree (Árvore Não-Determinística de Decisão Adaptativa) [2] se baseia em aprendizado por instância, na qual todos os dados de treinamento são incorporados na estrutura da árvore. A Fig. 2 apresenta os estados representando atributos e classificação após o treinamento.

Uma derivação da Adaptive-NDD Tree está ilustrada na Fig. 3, na qual os estados são agrupados conforme os atributos e classificações. Quando dados de treinamentos são inseridos, pode ocorrer uma operação denominada “split”, na qual são criados estados novos para armazenar as diferentes classes.

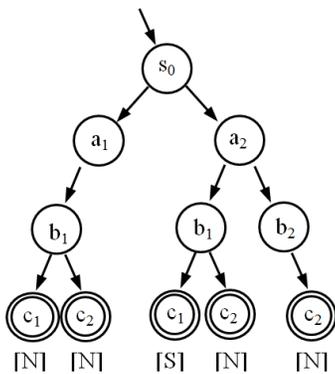


Figura 4. Em uma Adaptive-NDD Tree, todos os dados de treinamento são armazenados na própria estrutura.

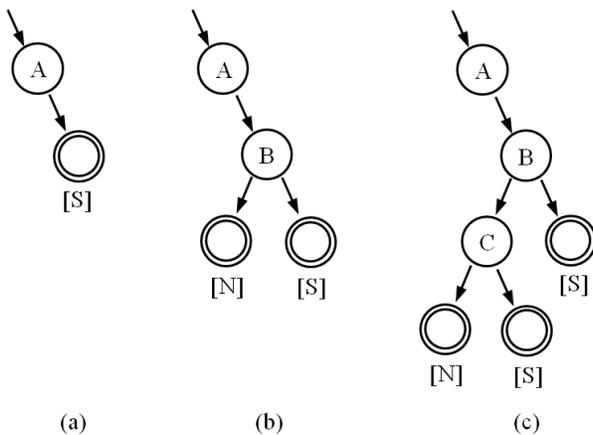


Figura 5. (a) Estado inicial contendo uma única classe [S]. (b), (c) Objetos com diferentes atributos são inseridos na árvore, forçando operações de split.

Os autômatos adaptativos são capazes de incorporar novas informações em sua própria estrutura, caracterizando a fase de aprendizado. Os conceitos mais fáceis de serem aprendidos podem ser atingidos por um pequeno número de auto-modificações.

IV. ÁRVORES DE DECISÃO

A árvore de decisão é uma estrutura bem conhecida [5], composta por nós e folhas relacionados aos atributos dos dados.

A aprendizagem ou processo de indução permite que uma árvore assimile novas informações e realize inferências sobre a classificação de dados. Para isso, utiliza-se um conjunto de dados de treinamento para ensinar ao dispositivo qual o resultado esperado de acordo com os atributos presentes. Durante o processo inicial de treinamento, a estrutura se automodifica. A operação adaptativa de adição corresponde ao chamado “split” de um nó da árvore.

O problema de aprendizado ótimo é conhecido como da classe NP-completo [6] e, por isso, os algoritmos são heurísticos e com base em melhor decisão local, porém, sem a garantia de retornar a melhor resposta global. Um método tradicional conhecido por ID3 e sua evolução C4.5 apresentados por Quinlan [7][8] utilizam o cálculo de alteração da entropia relacionada à operação de split.

Existem diferentes métodos para determinar a forma de crescimento da árvore, variando entre cálculo de entropia, chi-square ou diferentes medidas, e sempre buscando mínimos locais. Apesar de não ser possível determinar o mínimo global, observa-se que, pelo número de referência aos algoritmos ID3 e C4.5, os resultados são satisfatórios para grande parte dos casos de uso.

Em contrapartida, existem conceitos simples que não são bem expressos pela estrutura de árvore como, por exemplo, as operações de paridade, “ou exclusivo” (XOR), multiplexação. Nesses casos, a estrutura de árvore de decisão crescerá rapidamente sem, no entanto, melhorar sua capacidade de classificação ou predição.

Em algumas implementações de árvore de decisão, existe o mecanismo de poda (prunning), no qual são removidos os ramos e nós em excesso. Dessa forma, o tamanho total da estrutura diminui e fornece como resultado uma árvore de decisão mais concisa.

V. ERROS DE MODELAGEM

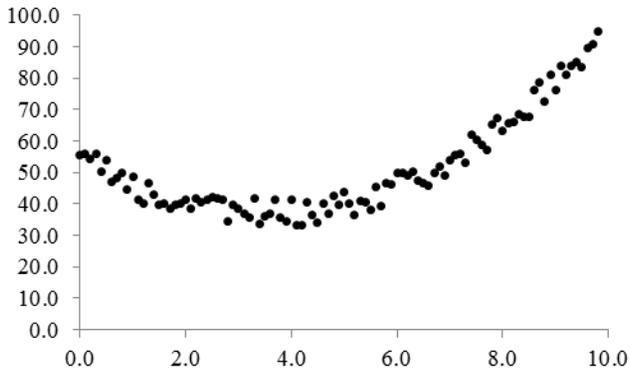
Um problema muito comum na modelagem de dados é a presença do fenômeno de overfitting [9]. Como exemplo, tomamos um conjunto de dados conforme ilustrado na Fig. 4(a), na qual foram disponibilizados os dados coletados durante um experimento. Com o objetivo de determinar uma curva que descreva o comportamento observado, selecionam-se arbitrariamente amostras de dados para serem utilizadas na definição do modelo. Será utilizado o método dos mínimos quadrados em polinômios de graus distintos.

A Fig. 4(b) apresenta a aproximação dos pontos de amostragem por um polinômio de segundo grau e seu resultado é visualmente coerente. Dessa forma, podemos dizer que um polinômio de grau 2 representa o fenômeno observado dado dentro de uma tolerância de erro razoável.

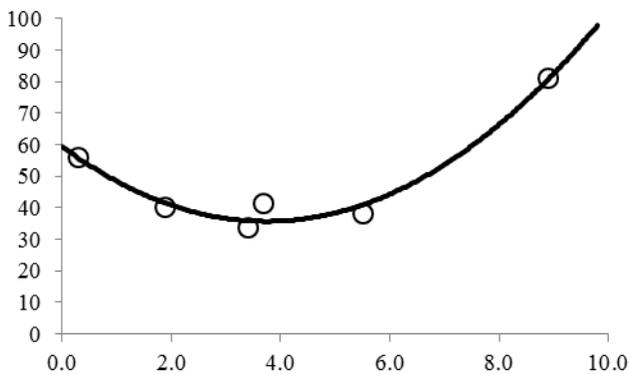
Ao utilizar um polinômio de grau maior, espera-se obter um erro de aproximação igual ou menor. Repetimos o processo de aproximação usando o método dos mínimos quadrados e obtemos um polinômio de grau 6, conforme

observado na Fig. 4(c). Apesar do gráfico do polinômio passar exatamente sobre cada ponto de amostragem e zerar o erro quadrado, a curva diverge sobre o resultado esperado da Fig. 4(a).

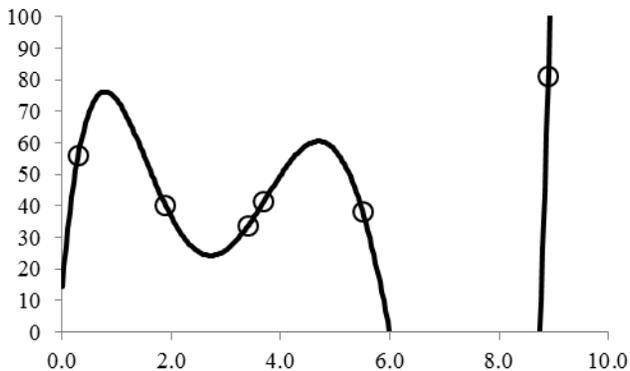
Nesse segundo caso, o modelo não representa bem a realidade e dizemos que o modelo sofreu um overfitting. Apesar do polinômio de grau 6 eliminar o erro de aproximação dos pontos de amostragem, existe um polinômio mais simples que consegue uma boa representação para os dados coletados experimentalmente.



(a)



(b)



(c)

Figura 4. (a) medição experimental de um determinado comportamento, (b) aproximação adequada usando um polinômio de grau 2 com presença de erro, (c) aproximação

usando um polinômio de grau 6, que resultou no fenômeno chamado overfitting.

Em uma árvore de decisão, esse fenômeno está associado ao desejo de representar usando um número de nós e ramos maior do que o necessário. Isso significa que o espaço foi subdividido em diversas partes e não há um número suficiente de dados de amostragem para determinar as classes em cada região. O mesmo ocorreria em uma modelagem adaptativa que apresentasse somente operações de adição, mas sem remoção de parte da estrutura.

A. Princípio da Navalha de Occam

“Se há diversas possibilidades, opte pela mais simples. Ela provavelmente é a mais correta”. O princípio da navalha de Occam é intuitivamente um ótimo fator de decisão para a escolha de modelos. Por exemplo, se existem polinômios de grau 2 e 6 que representam o mesmo comportamento, então deve-se optar pelo de menor grau.

De fato, existem inúmeras formas de interpretar o significado da palavra “simples”. Em uma árvore de decisão, a simplicidade pode estar relacionada com o número de nós, profundidade da árvore ou número de folhas. A decisão se torna subjetiva. A fim de complementar o Princípio da Navalha de Occam e definir o termo “simplicidade”, introduzimos os conceitos de Complexidade Algorítmica ou também conhecida como Complexidade de Kolmogorov.

B. Complexidade de Kolmogorov ou Algorítmica

A complexidade de Kolmogorov é denotada por $K_L(x)$ e corresponde ao tamanho da menor descrição da cadeia de entrada x em uma linguagem L . Uma definição equivalente é feita através da formalização de complexidade utilizando uma Máquina de Turing Universal [10]. Nesse caso, define-se $K_M(x)$ como o tamanho da menor representação $\langle M, w \rangle$ de uma Máquina de Turing M , que a partir de uma entrada w , gera a cadeia de saída x .

Usando a linguagem associada aos computadores podemos dizer, por exemplo, que $K_M(00000000)$ corresponde ao tamanho em bytes do menor programa capaz de imprimir a cadeia de zeros “00000000” na tela. Observa-se que, para cadeias previsíveis, a complexidade de programa se mantém baixa independente da quantidade de símbolos a serem impressos.

$$K_M(00) \sim K_M(000000000) \sim K_M(000\dots000)$$

O mesmo não ocorre para sequências que apresentam um maior grau de aleatoriedade, como 011000101011. Esse tipo de cadeia requer uma descrição detalhada e o tamanho K_M do programa cresce de acordo com o tamanho da cadeia. De forma independente, Kolmogorov, Solomonoff e Chaitin [11] observaram que o conceito de complexidade está diretamente relacionado com aleatoriedade e compressão de dados.

Um resultado importante da teoria de complexidade algorítmica é ao Teorema da Invariância, demonstrado por Solomonoff [12].

Teorema da Invariância: $|K_{M_1}(x) - K_{M_2}(x)| < \text{constante}$

Isso significa que as complexidades em diferentes máquinas são próximas, apesar da diferença por uma constante. Além disso, esse teorema reforça a idéia de que o conceito de complexidade é universal e não depende da linguagem, máquina utilizada ou estrutura de dados.

Em uma árvore de decisão, a complexidade de Kolmogorov está associada ao tamanho do menor programa necessário para reconstruir a estrutura da árvore. Portanto, uma árvore de decisão “simples” é aquela que possui um menor número de nós, ramos ou folhas.

C. As subdivisões do espaço

Dizemos que um modelo apresenta overfitting quando a complexidade do modelo é superior a Complexidade de Kolmogorov, $K(x)$, associada ao comportamento observado. Na Fig. 5, a aproximação da borda do círculo por retas horizontais e verticais caracteriza uma situação de overfitting. Isso ocorre porque o espaço será dividido em várias regiões menores, que requerem uma quantidade maior de dados de amostragem.

Similar ao comportamento de subdivisão sucessiva da região espacial, a Fig. 6 ilustra o crescimento desbalanceado da estrutura de árvore que pode ser um problema de overfitting. A árvore de decisão possui muitos nós e ramos e, conseqüentemente, necessitam amostras para classificar cada nó ou folha presente.

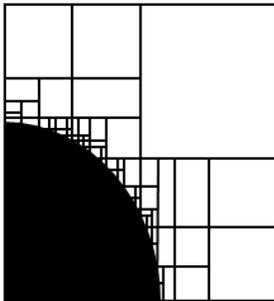


Figura 5. Aproximação de um círculo por retas horizontais e verticais. Próximo à borda do círculo, há uma série de regiões cada vez menores que são incapazes de aproximar corretamente.

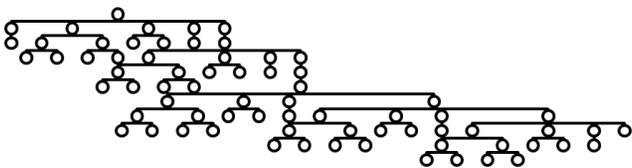


Figura 6. Árvore de decisão com grande quantidade de nós, representando o comportamento de overfitting.

Esse problema afeta uma modelagem de autômato adaptativo que apresenta somente operações de adição de dados e não há remoção. A fim de se evitar o overfitting de dados, é importante retirar estruturas redundantes ou com baixa utilização. Esse mecanismo é conhecido, nas árvores de

decisão, como poda (pruning) [13].

VI. ÁRVORES DE DECISÃO ADAPTATIVAS

Na área de Tecnologia Adaptativa, os dispositivos apresentam mudança de comportamento para se adequar melhor ao ambiente. Uma árvore de decisão adaptativa é uma extensão da árvore de decisão, que atua como núcleo, acrescido de uma camada adaptativa.

A. Alteração Estrutural

A camada adaptativa permite que a estrutura hierárquica de nós e ramos seja modificada ao longo da sua utilização a partir das ações elementares de consulta, inserção e remoção. O crescimento da árvore corresponde ao conhecimento que se acumula durante o processo inicial de treinamento.

Usando o formalismo da Árvore Não-Determinística de Decisão Adaptativa, Hemerson [2] propôs uma notação eficiente para descrever as ações adaptativas relacionadas ao modelo da árvore.

? [(no, atributo)]
 + [(no, atributo, (resultado1, filho1), (resultado2, filho2))]
 - [(no, atributo)]

Uma árvore adaptativa pode ser implementada através de métodos incremental ou completo. O método completo avalia todos os elementos antes de iniciar a criação da árvore de decisão. Após a fase de treinamento, a estrutura continua reagindo aos dados fornecidos e pode se auto-modificar. Esse processo de adquirir novas informações a partir do meio externo é chamado de aprendizagem incremental.

Nem sempre uma abordagem incremental é adequada, uma vez que as atualizações na estrutura da árvore de decisão seguem a ordem dos dados apresentados. A dependência da ordem dos dados é chamada de influência ou tendência (bias, em inglês), na qual resultam diferentes estruturas de acordo com a ordem de apresentação dos dados.

B. Poda da Árvore

Seguindo o princípio da navalha de Occam, uma estrutura mais simples apresenta uma melhor generalização para as predições de resultados. Sob esse ponto de vista, a remoção de estrutura deve ser vista como parte da aprendizagem da estrutura.

A poda da árvore de decisão remove os ramos e os nós que não acrescentam informação suficiente à estrutura e, portanto, pode ser descartada. A definição sobre relevância é feita pela camada adaptativa usando algum cálculo ou heurística.

De fato, ao simplificar a estrutura da árvore de decisão, o dispositivo requer um menor número de amostras para classificação.

C. Não-Determinismo

O comportamento não-determinístico permite que o processamento seja feito em paralelo. A vantagem do paralelismo não é em termos de maior poder computacional, mas a possibilidade de explorar outras alternativas. Isso significa que mais de uma hipótese pode ser verificada antes de chegar ao resultado final.

Em uma árvore não-determinística de decisão, a camada adaptativa pode enumerar diferentes modelos para serem comparados com o comportamento a ser previsto.

D. Uso de atributos contínuos

A modelagem de árvores de decisão depende do uso de atributos discretos para classificar objetos. Entretanto, uma grande parte dos problemas apresenta atributos contínuos a serem considerados [14]. Uma forma trivial de mapear atributos contínuos em discretos seria utilizar uma correspondência entre intervalos de valores contínuos com atributos discretos. Por exemplo, um animal de 28 metros pode ser caracterizado como de grande porte, enquanto que animais menores que 5 metros seriam considerados de pequeno porte.

Atributos que possuam uma grande quantidade de resultados, como os números de 1 a 1000, podem ser considerados como atributos contínuos. Nesse caso, poderia ser realizada uma divisão pré-definida em regiões de dados, como por exemplo usar as faixas de 1-100, 101-201, e assim por diante, agrupando os valores em um total de 10 conjuntos.

Outra possibilidade é transformar os valores numéricos em expressões booleanas, como por exemplo: predicado[t <= 500] retorna verdadeiro para os primeiros 500 registros e falso para os últimos 500. Assim reduz-se um problema de 1000 possibilidades a apenas dois casos: verdadeiro ou falso.

Um atributo contínuo pode, também, ser associado a mais de um atributo discreto. Por exemplo, usando uma série de atributos para representar intervalos cada vez menores. É possível representar um número na base 2 e utilizar os dígitos mais significativos como atributos discretos. A Fig. 7 ilustra a definição dos atributos discretos a₁,..., a₈ de acordo com os dígitos, que representam o intervalo dos números inteiros de 0 a 255.

a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈
1	0	0	0	0	0	1	1

Figura 7. Número 131 apresenta os atributos discretos a₁,..., a₈, que apresentam somente os valores 0 ou 1. Dessa forma, representa-se um atributo contínuo através de 8 atributos.

A utilização de um modelo não-determinístico permite a escolha entre as diferentes formas de discretização de dados, determinando aquela que melhor se adequa ao problema proposto.

E. Inclusão de novos atributos

Durante a construção da árvore de decisão, assume-se que a relação entre os atributos é independente, ou seja, um não depende um do outro. Em muitos casos, essa suposição é incorreta e pode levar incapacidade do modelo ser uma boa representação do objeto real.

Uma das causas de ocorrência de overfitting está relacionada com limitações do modelo em relação aos dados obtidos experimentalmente. Um exemplo é a tentativa de modelar uma figura através da leitura de atributos contínuos x e y como

apresentados na Fig. 5, observa-se uma figura circular (primeiro quadrante) sendo aproximado por uma série de quadrados menores. Embora os atributos x e y sejam tratados independentes, sabemos que seguem a equação $x^2 + y^2 < c$.

A inclusão de novos atributos calculados com base nos existentes cria a interdependência entre eles. Por exemplo, a substituição das coordenadas cartesianas (x,y) por coordenadas polares aumentaria a eficiência da árvore de decisão para modelar o círculo. Portanto, a camada adaptativa pode realizar transformações entre diferentes sistemas de coordenadas, assim como operações de translação, rotação e escala.

Conhece-se a limitação de árvores de decisão não possuírem uma boa representação para operações de paridade ou multiplexação, uma vez que o tamanho da árvore resultante é grande. A utilização de um mecanismo gerador de novos atributos combinados pode favorecer sua representação.

VII. EXEMPLO

Abordamos o problema de uma figura geométrica em um espaço bidimensional usando as coordenadas cartesianas x e y. A capacidade de representação do objeto por meio de uma árvore de decisão depende da modelagem adotada, na qual formas geométricas mais complexas apresentam maior propensão ao erro. Por exemplo, usando retas é mais difícil mapear a borda de uma figura curva do que os lados de um quadrado.

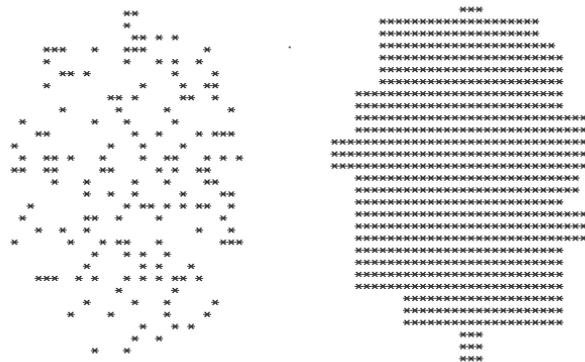


Figura 8. Mapeamento de figuras

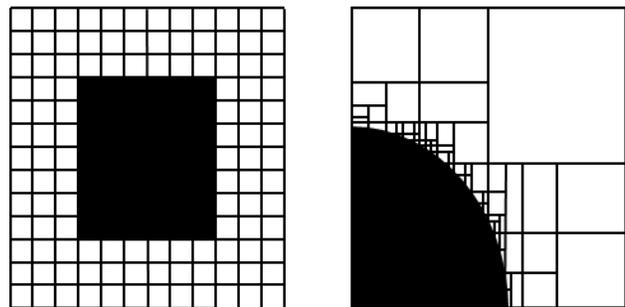


Figura 9. Diferença no mapeamento de regiões entre um quadrado e um arco

Em posições distintas em um espaço de 50x50, foram desenhados círculos de diferentes tamanhos. A tabela 1

apresenta as posições do centro (x,y) do círculo seu respectivo raio. O problema proposto é construir uma árvore de decisão que modela a figura proposta. Em outras palavras, é possível determinar se um ponto qualquer está dentro ou fora da região do da figura com base nas suas coordenadas.

Foi escolhida como estrutura núcleo a implementação de árvores de decisão usando autômatos adaptativos, baseado no Adaptive-NDD, com seleção de atributos seguindo uma ordem pré-estabelecida. A partir desse modelo, modificamos a estrutura para apresentar um comportamento determinístico e que ocorresse a poda da árvore em casos triviais, como um conjunto de sub-ramos que pertencem à mesma classe.

A. Árvore de Decisão

Os atributos contínuos foram discretizados usando a representação do número na base binária e associando cada dígito a um novo atributo discreto, caracterizando um algoritmo não-supervisionado. Por exemplo, a representação da posição no eixo X utiliza uma série de atributos booleanos $a_{x1}, a_{x2}, \dots, a_{xn}$ que os dígitos da coordenada x na base binária.

Na Tabela 1, as colunas x , y e *raio* representam as características do círculo, enquanto o tamanho resultante da árvore de decisão está apresentado na coluna *tree*. Observamos que os tamanhos das árvores de decisão variaram entre 15 a 200.

x	y	size	tree
6	6	3	21
12	12	3	27
17	17	3	15
22	22	3	15
25	25	3	33
28	28	3	26
8	8	5	35
14	14	5	44
19	19	5	55
24	24	5	64
27	27	5	50
30	30	5	37
10	10	7	57
16	16	7	50
21	21	7	76
26	26	7	46
29	29	7	56
32	32	7	39
14	14	11	110
20	20	11	140
25	25	11	110
30	30	11	121
33	33	11	122
36	36	11	107
18	18	15	137
24	24	15	116
29	29	15	200
34	34	15	184
37	37	15	119
40	40	15	61

Como observado pela variação dos valores na coluna *tree* da Tabela 1, foram geradas árvores de decisão com diferentes complexidades. Dependendo da posição da figura geométrica, foi necessário um determinado número de nós para representá-la. O método de discretização, portanto, influencia bastante no

tamanho da árvore de decisão.

B. Árvore de Decisão Adaptativa

Em uma árvore não-determinística de decisão, torna-se possível enumerar diferentes modelos a serem considerados. A camada adaptativa atua na geração de novos atributos, que são calculados em função dos já existentes. Utilizaremos, nesse exemplo, uma ação adaptativa que transforma as coordenadas cartesianas em coordenadas polares, adicionado de uma operação de translação.

A representação feita na Fig. 10 ilustra algumas das possibilidades de transformações entre coordenadas, acrescido de um vetor T de translação entre os sistemas. A idéia é utilizar o método adaptativo para fazer a regulagem correta de parâmetros até determinar um valor adequado de T , cujo valor é inicialmente desconhecido.

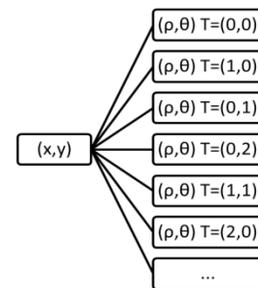


Figura 10. Enumeração entre sistemas de coordenadas cartesianas em coordenadas polares, com a adição de um vetor de translação T .

Ao escolher a árvore de decisão com a menor complexidade associada, encontraremos os valores ideais de T . Esse ajuste pode ser feito, inclusive, em tempo real e favorecer a utilização de métodos incrementais.

Após o ajuste do modelo, verifica-se que a árvore de decisão escolhida apresentou de 10 a 12 nós. Essa baixa complexidade é independente da posição do centro do círculo, mas está relacionado com o total de dígitos necessários para representar o raio do círculo na base binária. Esse resultado é muito superior ao caso da árvore de decisão sem a camada adaptativa, no qual o tamanho variava entre 15 a 200 nós.

VIII. CONSIDERAÇÕES FINAIS

As árvores de decisão adaptativas são dispositivos capazes de incorporar novas informações em sua própria estrutura através das auto-modificações. A camada adaptativa, que corresponde a um segundo nível e independente da estrutura núcleo, é responsável por enumerar diferentes modelos e decidir qual deles possui a melhor representação.

Como contribuição fundamental do artigo, propõe-se que o modelo de árvore de decisão adaptativa seja capaz de associar novos atributos com base nos já existentes. Os atributos calculados permitem obter melhores resultados mantendo-se a mesma estrutura do núcleo. O resultado imediato é a possibilidade de modelar adequadamente operações simples, como paridade, ou exclusivo (“XOR”) e multiplexação, que

não apresentavam bons resultados com uma árvore de decisão tradicional.

A complexidade de Kolmogorov, apesar de incomputável, formaliza um conceito importante na escolha do melhor modelo: a simplicidade. Durante o artigo, o tamanho da árvore foi a métrica utilizada para determinar a complexidade do modelo, mas pode ser inadequado para outras situações.

O tamanho final da árvore de decisão adaptativa pode ser visto como uma forma de caracterizar o problema [14], ou seja, transforma-se um problema de classificação em reconhecimento de objeto.

REFERÊNCIAS

- [1] J. J. Neto, “Adaptive rule-driven devices – general formulation and a case study”. In CIAA’2001 Sixth International Conference on Implementation and Application of Automata. Springer-Verlag.
- [2] H. Pistori, “Adaptive Non-Deterministic Decision Trees: General Formulation and Case Study”.
- [3] H. Lewis and C. Papadimitriou, *Elements of the Theory of Computation*, Prentice-Hall, 1997.
- [4] R. Rocha e J. Neto, Autômato adaptativo, limites e complexidade em comparação com máquina de Turing. In: Proceedings of the second Congress of Logic Applied to Technology - LAPTEC 2000. São Paulo 2001.
- [5] S. B. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, *Informatica* 31(2007) 249-268, 2007
- [6] L. Hyafil, R.L. Rivest, Constructing optimal binary decision trees is NP-Complete. *Information Processing Letters*, 5(1), 15–17, 1976
- [7] J. R. Quinlan, Induction of Decision Trees. *Machine Learning* (Mar. 1986), 81-106
- [8] J. R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993
- [9] T. G. Dietterich, Overfitting and under-computing in machine learning. *Computing Surveys*, 27 (3), 326-327, 1995
- [10] Li, Ming and Vitányi, Paul, *An Introduction to Kolmogorov Complexity and Its Applications*, Springer, 1997
- [11] R. Solomonoff, Complexity-Based Induction Systems: Comparisons and Convergence Theorems, *IEEE Trans. on Information Theory*, Vol. IT-24, No. 4, pp. 422-432, July 1978
- [12] R. Solomonoff, A Formal Theory of Inductive Inference Part I, *Information and Control*, Part I: Vol 7, No. 1, pp. 1-22, March 1964
- [13] J. Mingers, An Empirical Comparison of Pruning Methods for Decision Tree Induction, *Machine Learning*, Vol. 4, 1989, pp.227-243
- [14] P.M. Murphy, An empirical analysis of the benefit of decision tree size biases as a function of concept distribution, TR. 95-29, Department of Information and Computer Science, University of California, Irvine, 1995

Fabricao S. Catae é graduado em Engenharia Elétrica (2002) na ênfase de Automação e Controle e aluno regular do programa de mestrado da Escola Politécnica, Universidade de São Paulo. Sua área de atuação está relacionada com Mecanismos de Inferência e Linguagem Natural, trabalhando em linhas de pesquisa do Laboratório de Linguagens e Técnicas Adaptativas.

Ricardo Luis A. Rocha é natural do Rio de Janeiro-RJ e nasceu em 29/05/1960. Graduou-se em Engenharia Elétrica modalidade Eletrônica na PUC-RJ, em 1982. É Mestre e Doutor em Engenharia de Computação pela EPUSP (1995 e 2000, respectivamente). Suas áreas de atuação incluem Tecnologias Adaptativas, Fundamentos de Computação e Modelos Computacionais. Dr. Rocha é membro da ACM (Association for Computing Machinery) e da SBC (Sociedade Brasileira de Computação).