Ministério da Educação
Universidade Tecnológica Federal do Paraná
Pró-Reitoria de Pesquisa e Pós-Graduação

Relatório Final de Atividades

Indução de Árvores de Decisão para a Inferência de Redes Gênicas

vinculado ao projeto

Integração de dados na biologia sistêmica: caracterização de fenômenos biológicos a partir de informações estruturais e funcionais

Maikon Aloan Marin

Voluntário

Tecnologia em Análise e Desenvolvimento de Sistemas

Data de ingresso no programa: 11/2012

Orientador: Prof. Dr. Fabrício Martins Lopes

Área do Conhecimento: 1.03.03.00-6 Metodologia e Técnicas da Computação

CAMPUS CORNÉLIO PROCÓPIO, 2013

MAIKON ALOAN MARIN
FABRÍCIO MARTINS LOPES

Indução de Árvores de Decisão para a Inferência de Redes Gênicas

Relatório Pesquisa do Programa de Iniciação Científica da Universidade Tecnológica Federal do Paraná.

CAMPUS CORNÉLIO PROCÓPIO, 2013

SUMÁRIO

INTRODUÇÃO

Nas últimas décadas houve um grande avanço e notoriedade na área da inteligência artificial (IA) devido à rápida evolução da tecnologia e principalmente da informática, porém o desenvolvimento de robôs inteligentes, pensando como humanos, é um produto apenas da ficção científica ou de um futuro ainda distante.

A ciência encara a IA de uma maneira bem menos fantasiosa e muito mais sutil, ela já está presente no cotidiano de todas as pessoas, seja em máquinas fotográficas que fazem o foco automático no rosto das pessoas, no desenvolvimento de videogames que utilizam esse tipo de estudo para criar jogos cada vez mais complexos, ou até mesmo nos corretores ortográficos dos processadores de texto de computador, pois é preciso um sistema inteligente para detectar um possível problema em uma frase e oferecer a suas opções de correção [1].

Os classificadores baseados em árvore de decisão, são um dos ramos da computação na área da inteligência artificial, conhecido como reconhecimento de padrões [13], campo esse dedicado ao desenvolvimento de algoritmos e técnicas que permitam ao computador aprender, ou seja, identificar padrões observando um conjunto de dados de interesse.

Dentro do contexto de reconhecimento de padrões, existe o raciocínio indutivo e o raciocínio dedutivo. Em geral, as técnicas desenvolvidas se preocupam com o raciocínio indutivo, que extrai regras e padrões de grandes conjuntos de dados [2].

Na área computacional, o processo de construção de uma árvore de decisão é conhecida como indução. A indução de árvores de decisão, tema central do presente trabalho, se trata de um importante tópico de pesquisa em reconhecimento de padrões [13] e um exemplo do aprendizado indutivo, sendo uma das formas mais simples e, ainda assim mais bem-sucedidas, de algoritmos de aprendizagem e classificação. Ela serve como uma boa introdução à área da aprendizagem indutiva, e é de fácil implementação [5].

Foi realizada então uma análise, mais especificamente, do algoritmo J48, algoritmo do software de mineração de dados WEKA [3], que é baseado no algoritmo de árvores de decisão C4.5. O estudo realizado concentrou-se no entendimento do conceito de árvores de decisão como um todo e da análise do funcionamento do algoritmo em questão dentro do ambiente WEKA com seus respectivos testes e resultados.

Para a exemplificação e avaliação dos conceitos estudados foram realizados testes com um banco de dados biológico, assim, foi utilizado o banco de dados de flores Íris disponibilizado no *Machine Learning Repository* pelo *Center for Machine Learning and Intelligent Systems* [4], que mantém 246 conjuntos de dados como um serviço para a comunidade de aprendizado de máquina.

Realizou-se então a análise do comportamento do software WEKA com o algoritmo J48 para a escolha dos atributos e consequente indução da árvore de decisão com os referidos dados.

MATERIAS E METÓDOS

Descrição da Base de Dados. O presente trabalho teve como entrada os dados do banco de dados Iris [4], um famoso conjunto de dados reais que foi criado por Fisher no ano de 1936 e doado por Michael Marshall em 01/07/1988. O trabalho de Fisher é um clássico no seu campo e é referenciado com frequência. Este banco de dados é considerado uma das melhores bases de dados conhecidas encontradas na literatura de reconhecimento de padrão e é um domínio extremamente simples, seus dados estão disponíveis para download e consulta em *<http://archive.ics.uci.edu/ml/datasets/Iris>*.

O conjunto de dados pertence à área biológica e refere-se à flor Íris. Ele contém três classes com 50 casos cada, são elas a classe Setosa, a Versicolour e a Virgínica, onde cada classe se refere a um tipo de planta Íris. A Figura 1 exibe as três classes das flores Íris.

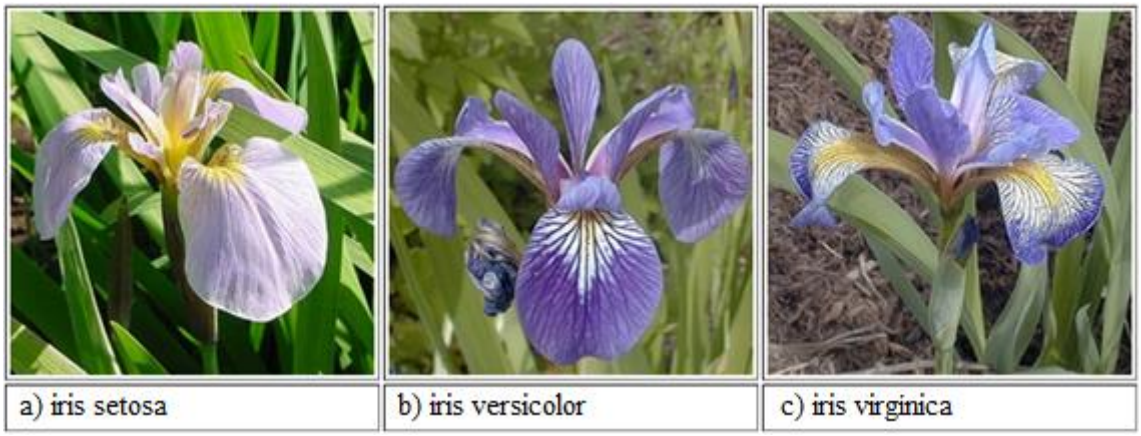

Figura 1. Imagem das três classes da flor Íris constante no conjunto de dados [16].

Consta também no conjunto de dados quatro atributos diferentes de valores numéricos indicando medidas de partes da flor.

Informação dos Atributos:

- Comprimento da Sépala em cm;
- Largura da Sépala em cm;
- Comprimento da Pétala em cm;
- Largura da Pétala em cm;
- Classe: Íris Setosa, Íris Versicolour e Íris Virgínica.

Estes dados diferem dos dados apresentados no artigo de Fisher na amostra 35 e 38, classe Íris Setosa. A amostra 35 deve ser: 4.9,3.1,1.5,0.2, "Iris-setosa" onde ocorre um erro no quarto atributo e a amostra 38 deve ser: 4.9,3.6,1.4,0.1, "Iris-setosa" onde os erros estão no segundo e terceiro atributos.

Árvore de Decisão. Uma árvore de decisão é uma forma gráfica de visualizar os resultados de decisões atuais e futuras, ela tem como entrada um conjunto de atributos que compõem um objeto ou situação, e como saída sua “decisão”, ou seja, uma previsão do valor de saída de acordo com a entrada. Os atributos de entrada e o valor de saída podem ser discretos ou contínuos, a aprendizagem de uma função de valores discretos é chamada aprendizagem de classificação, a aprendizagem de uma função contínua é chamada regressão [5].

As decisões são tomadas pela árvore através da realização de uma sequência de testes. Cada nó interno na árvore corresponde a um teste do valor de um dos atributos, e as ramificações a partir do nó são identificadas com os possíveis valores do teste. Cada nó de folha na árvore específica o valor a ser retornado se aquela folha for alcançada.

Na Figura 2 é apresentado um exemplo de árvore de decisão que classifica os dias, conforme eles são satisfatórios ou não, para se jogar tênis.

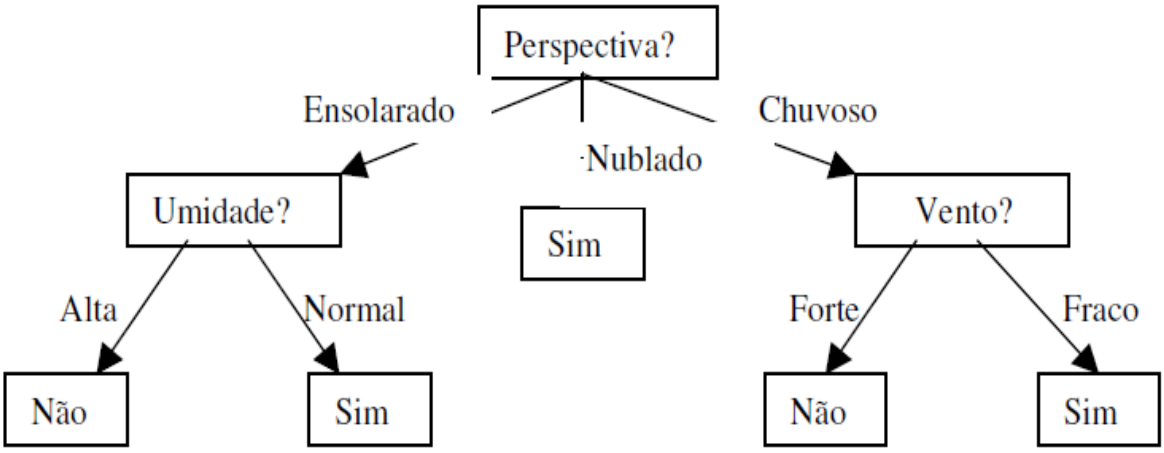

Figura 2. Exemplo de Árvore de Decisão [14].

No contexto computacional, as árvores de decisão constituem uma técnica muito eficiente e amplamente utilizada em problemas de reconhecimento de padrões [13], como é o caso utilizado neste trabalho. Uma das razões para que esta técnica seja muito utilizada é o fato de que o conhecimento adquirido pode ser representado por meio de regras. Essas regras podem ser expressas em linguagem natural, facilitando o entendimento e interpretação dos resultados. Outra vantagem das árvores de decisão é que podem ser aplicadas em grandes bases de dados, dado que sua indução é computacionalmente rápida.

Conjunto de Teste e de Treinamento. Uma parte importante da construção de uma árvore de decisão é a separação dos dados nos conjuntos de treinamento e teste. Normalmente, quando você separa um conjunto de dados em um conjunto de treinamento e um conjunto de teste, a maior parte dos dados é usada para treinamento e uma parte menor dos dados é usada para o teste. O conjunto de treinamento são os dados retirados de seu conjunto de exemplos para a utilização na construção da árvore de decisão e o conjunto de teste, os dados restantes utilizados para testar o desempenho e precisão da árvore construída.

Um processo de treinamento pode ser divido em treinamento supervisionado e não supervisionado [13]. O treinamento supervisionado ocorre quando o número de classes das suas instâncias for definido anteriormente, já o treinamento não supervisionado ocorre quando esse número for definido automaticamente a partir dos dados disponíveis. Neste trabalho foi utilizada a técnica de aprendizado supervisionado.

Seleção de Atributos. A construção de uma árvore de decisão, também conhecida como indução, é realizada escolhendo os atributos que irão separar seus dados em cada nó até a classificação total do conjunto. A chave para o sucesso do algoritmo de aprendizado por árvores de decisão irá depender do critério utilizado para escolher o atributo que particiona o conjunto de exemplos em cada iteração.

A seleção desses atributos é efetuada de acordo com critérios estatísticos que buscam selecionar os atributos mais relevantes para a classificação. Os critérios de seleção para a melhor divisão são baseados em diferentes medidas, tais como impureza, distância e dependência. A maior parte dos algoritmos de indução busca dividir os dados de um nó-pai de forma a minimizar o grau de impureza dos nós-filhos [6].

Algumas possibilidades para escolher esse atributo são:

- Aleatória: seleciona qualquer atributo aleatoriamente;

- Menos valores: seleciona o atributo com a menor quantidade de valores possíveis;
- Mais valores: seleciona o atributo com a maior quantidade de valores possíveis;
- Ganho de informação máximo;
- Razão de ganho;
- Índice Gini.

Entropia. Uma das medidas de seleção de atributos baseada em impureza é o Ganho de Informação. Para definir o ganho de informação e consecutivamente o ganho de informação máximo, começa-se definindo uma medida comumente usada em teoria de informação, chamada entropia ou informação esperada, que caracteriza a impureza de uma coleção arbitrária de exemplos.

O cálculo da entropia total de um conjunto, definido na equação (1), é referente à sua classificação final, ou seja, ao seu atributo que delimita a classe das amostras.

$$info(S) = entropia(S) = -\sum_{j=1}^{k}\left(\frac{C_j}{S}\right) * log_2\left(\frac{C_j}{S}\right) \quad (1)$$

Onde:
$C_{j:}$ quantidade de amostras da classe.
S: quantidade total das amostras.

A equação (2) refere-se ao cálculo da entropia para cada atributo de decisão utilizado para classificar suas amostras.

$$info(S, A) = \sum_{i=1}^{m}\left(\frac{S_i}{S}\right) * info(S_i) \quad (2)$$

Onde:
S_i: quantidade de amostras para a partição.
S: quantidade total dos amostras.
m: quantidade de partições.
$info(S_i)$: entropia total para a partição.

Ganho de Informação, Ganho Máximo e Razão de Ganho. O ganho máximo seleciona o atributo que possui o maior ganho de informação esperado, isto é, seleciona o atributo que resultará no menor tamanho esperado das subárvores, assumindo que a raiz é o nó atual. Ele possui tendência em favor de testes com muitos valores.

A escolha do atributo para particionar o conjunto de exemplos é dada pelo cálculo do ganho de informação de cada atributo. Esse cálculo consiste na subtração da entropia de todo o conjunto pela entropia de cada atributo, como definido pela equação (3).

$$gain(S, A) = info(S) - info(S, A) \quad (3)$$

O atributo entre todos os utilizados na classificação das amostras que possuir o maior valor de ganho de informação é o atributo com ganho máximo, e assim o mais relevante para particionar os exemplos classificando-os.

A partir da primeira seleção de um atributo para particionar os exemplos é feita as escolhas para a partição e classificação nas sub árvores até a classificação total de todos os exemplos do conjunto.

Como o ganho máximo tem uma tendência em favor de testes com muitos valores, para testes com poucos valores pode ser utilizada como critério de avaliação a Razão de Ganho, que nada mais é do que o ganho de informação relativo. A razão de ganho é definida pela equação (4) e constitui-se da divisão do ganho de informação do atributo por sua entropia.

$$ratio(S,A) = \frac{gain(S,A)}{info(S,A)} \quad (4)$$

A razão de ganho expressa a proporção de informação gerada pela partição que é útil, ou seja, que aparenta ser útil para a classificação.

GINI. Proposto em 1912 pelo estatístico italiano Corrado Gini, o índice GINI é outra medida bastante conhecida e utilizada [10]. Ele é um índice de dispersão estatística que mede a heterogeneidade dos dados e é utilizado tanto para a seleção de atributos como também em análises econômicas e sociais para verificar a distribuição de renda em um certo país.

O índice GINI para um conjunto de dados *S*, que contém *n* registros, cada um com uma classe C_i é dado pela equação (5).

$$gini(S) = i - \sum_{i=1}^{k} p(C_i|n)^2 \quad (5)$$

Onde:
p_i: probabilidade relativa da classe C_i em *S*.
n: número de registros no conjunto *S*.
k: número de classes.

Se o conjunto *S* for particionado em dois ou mais subconjuntos S_i, O índice GINI dos dados particionados será definido pela equação (6).

$$gini(S,A) = \sum_{i=1}^{k} \frac{n_i}{n}\ gini(S_i) \quad (6)$$

Onde:
n_i: número de registros no subconjunto S_i.
n: número de registros no conjunto *S*.

Quando este índice é igual a zero, o conjunto de dados é puro, ou seja, todos os registros pertencem a uma mesma classe. Por outro lado, quando ele se aproxima do valor um, o conjunto apresenta os registros distribuídos igualmente entre todas as classes. Quando se utiliza o critério Gini na indução de árvores de decisão binárias, tende-se a isolar num ramo os registros que representam a classe mais frequente, assim, utilizando o atributo com menor valor do índice para a classificação, já, ao utilizar-se da entropia, balanceia-se o número de registros em cada ramo.

Um algoritmo de indução de árvore de decisão bastante conhecido que utiliza o índice GINI para a seleção de atributos é o algoritmo CART (*Classification and Regression Trees*), ele realiza a indução pela abordagem top-down e constrói uma árvore de decisão binária simples e legível. O atributo a ser particionado é escolhido como aquele que gera grupos com a menor heterogeneidade.

Atributos Numéricos. Diferente dos atributos discretos, quando se está lidando com atributos contínuos tem-se um conjunto infinito de valores possíveis. Para que isso não gere um número infinito de ramificações, os algoritmos de aprendizagem de árvore de decisão em

geral encontram um ponto de divisão, também chamado de limiar, para esses atributos, que dividirá as amostras em dois conjuntos.

Alguns dos testes utilizados para a divisão de atributos contínuos são: os testes simples, os testes múltiplos e a combinação linear de características. O mais utilizado é o teste simples, também conhecido como pesquisa exaustiva, sendo implementado, por exemplo, pelo algoritmo C4.5 [10].

No teste simples o método de escolha do limiar é iniciado com a ordenação de todos os valores do atributo de forma crescente. Após esse passo é calculado o ganho de informação para cada valor diferente do atributo, sendo cada um desses valores, possíveis pontos de divisão do atributo. É escolhido então aquele que fornecer o maior ganho de informação.

Para que a árvore construída apresente melhores resultados para exemplos que não participaram do conjunto de treinamento é utilizado como limiar o ponto médio, definido pela equação 7, entre o valor escolhido com o maior ganho de informação e o seu respectivo sucessor.

$$A = \frac{v_i + v_{i+1}}{2} \tag{7}$$

Sendo assim o particionamento de atributos contínuos implica em uma maior complexidade de cálculo e é a parte mais dispendiosa das aplicações de aprendizagem em árvores de decisão do mundo real [5].

Poda. Quando árvores de decisão são induzidas, muitas das arestas ou sub-árvores podem refletir ruídos ou dados ausentes. Ruídos referem-se a situações em que dois ou mais registros, composto por atributos que possuem os mesmos valores e que chegam a classes distintas, já dados ausentes, correspondem a registros que não possuem todos os valores dos atributos preenchidos.

Em ambas as situações, os registros redundantes ou mal formados devem ser eliminados, ou modificados, de tal forma que tenham a mesma classe ou todos seus valores preenchidos, respectivamente. Uma maneira para detectar e excluir essas ramificações e sub-árvores é utilizando métodos de poda (*pruning*) da árvore, com o objetivo de melhorar a taxa de acerto do modelo para novos exemplos [5].

A poda funciona impedindo a divisão recursiva de atributos que são claramente irrelevantes, até mesmo quando os dados nesse nó da árvore não são uniformemente classificados. O ganho de informação, por exemplo, pode ser utilizado como critério de poda. Caso todas as divisões possíveis utilizando um atributo X gerem ganhos menores que um valor pré-estabelecido, então esse nó vira folha, representando a classe mais frequente no conjunto de exemplos.

A árvore podada se torna mais simples, facilitando a sua interpretabilidade por parte do usuário. Junto ao método de seleção, o método de poda também varia de acordo com os diferentes algoritmos de indução de árvores de decisão.

Algoritmo J48. O algoritmo J48 permite a criação de modelos de árvore de decisão. Ele utiliza uma tecnologia *greedy* para induzir as árvores para a classificação posterior. O modelo de árvore de decisão é construído pela análise dos dados de treino e o modelo utilizado para classificar dados ainda não classificados. O J48 gera árvores de decisão, em que cada nó da árvore avalia a existência ou significância de cada atributo individualmente. As árvores de decisão são construídas do topo para a base, através da escolha do atributo mais importante, ou seja, que faz maior diferença para a classificação de um exemplo, para cada situação de acordo com regras matemáticas e estatísticas utilizadas pelo algoritmo. Uma vez que o atributo é escolhido, os dados de treino são divididos em sub-grupos correspondendo aos

diferentes valores dos atributos, o processo é repetido para cada sub-grupo até que uma grande parte dos atributos em cada sub-grupo pertençam a uma única classe. Desse modo pretende-se chegar à classificação correta com um número pequeno de teste, gerando a menor árvore de decisão possível [12].

A indução por árvore de decisão é um algoritmo que habitualmente aprende um conjunto de regras com elevada importância. Este algoritmo é escolhido para comparar a percentagem de acerto com outros algoritmos.

O J48 se baseia no algoritmo de árvores de decisão C4.5, que forma a árvore mais adequada sobre o conjunto de dados, podando as regras que melhoram a sua acurácia [9]. Os algoritmos de árvores de decisão são conhecidos pelo seu poder de expressividade, encadeando um conjunto de testes, os quais atuam diretamente no ganho de informação a respeito dos dados. Há a possibilidade de transformarmos árvores de decisão em regras de classificação.

Validação de Resultados. Um importante fator no desenvolvimento e construção de sistemas de classificação de dados, como as árvores de decisão, é validação de seus resultados. Ela qualificará o poder discriminativo do sistema identificando o método usado como bom ou não para à determinada análise de dados.

Validação Cruzada. A validação cruzada é uma técnica para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados, ela pode ser utilizada em conjunto com qualquer método de construção de árvore, inclusive poda. Sempre que existe um grande conjunto de hipóteses possíveis, devemos ser cuidadosos para não utilizar a liberdade resultante para encontrar uma "regularidade" sem significado nos dados, acarretando assim o que é chamado de superadaptação.

A ideia básica por trás da validação cruzada dos dados é estimar até que ponto cada hipótese irá prever dados não vistos. Isto é feito separando-se alguma fração dos dados conhecidos e usando-se esses dados para testar o desempenho da previsão de uma hipótese induzida a partir dos dados restantes. A validação cruzada de k vias significa que você deve executar k experimentos reservando de cada vez uma fração 1/k diferente dos dados para testes, e calcular a média dos resultados. Os valores mais utilizados para k são 5 e 10. Após a validação cruzada deve-se medir o desempenho da previsão com um novo conjunto de testes [5].

Análise ROC. Uma forma eficiente de demonstrar a relação entre a sensibilidade e a especificidade são as Curvas de Característica de Operação do Receptor (Curvas ROC - *Receiver Operating Characteristic*) [11], elas são úteis quando se deve levar em consideração diferentes custos/benefícios para os diferentes erros/acertos de classificação ou em domínios nos quais existe uma grande desproporção entre as classes.

A Curva ROC é um gráfico da taxa de verdadeiros positivos pela taxa de falsos positivos, exemplificada na figura 3. A taxa de verdadeiros positivos é a porcentagem de amostras corretamente classificadas como positivas dentre todas as positivas reais e a taxa de falsos positivos é a porcentagem de amostras erroneamente classificadas como positivas dentre todas as negativas reais.

Ela foi desenvolvida por engenheiros elétricos e engenheiros de radar durante a Segunda Guerra Mundial para detecção de objetos inimigos nas batalhas, sendo implementada na psicologia para detecção perceptual de estímulos [15]. A análise ROC vem sendo também amplamente utilizada nas áreas da medicina, radiologia, biometria e outras áreas por muitas décadas e, mais recentemente, foram cada vez mais inseridas em áreas como aprendizado de máquina e mineração de dados.

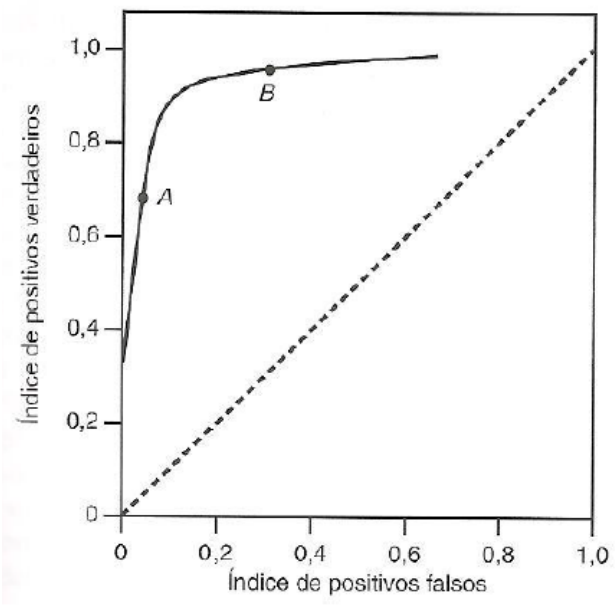

Figura 3. Curva de Características de Operação do Receptor (Curva ROC) [11].

Um modelo de classificador perfeito corresponderia a uma linha horizontal no topo do gráfico no ponto (0,1), onde todos os exemplos positivos e negativos seriam corretamente classificados, já uma linha horizontal no ponto (1,0) representaria o pior caos, onde um modelo sempre faz predições erradas. A linha diagonal no centro do gráfico indica um modelo que selecione as saídas como positivas ou negativas aleatoriamente, ela parte do ponto (0,0), que representa a estratégia de nunca classificar um exemplo como positivo e vai ao ponto (1,1) com a estratégia inversa de sempre classificar um novo exemplo como positivo [8].

Ferramenta WEKA. *Waikato Environment for Knowledge Analysis* (WEKA) [3] é um pacote de diversas implementações de algoritmos de aprendizagem de máquina para tarefas de mineração e classificação de dados. Ele foi desenvolvido utilizando a tecnologia JAVA na universidade de Waikato na Nova Zelândia e é um software de código aberto que se encontra disponível em <*http://www.cs.waikato.ac.nz/ml/weka/*>.

O formato de arquivo de dados utilizado pelo software WEKA é o formato ARFF, um formato próprio do software que corresponde a um arquivo de texto constituído de um cabeçalho e o conjunto de todas as instâncias (dados a serem analisados).

O cabeçalho fornece informações a respeito dos campos que compõem o conjunto de instâncias. Ele é definido pela notação *@relation* juntamente com o nome do conjunto de dados, posteriormente vem à sequência de atributos definidos para cada atributo pela notação *@attribute*, o nome do atributo e seu tipo ou os valores que ele pode representar, quando utilizado valores estes devem estar entre "{ }" separados por vírgulas. Para finalizar o cabeçalho possui a notação *@data* que define o início das instâncias. Por padrão, o ultimo atributo apresentado na relação será o atributo classe, porém isso pode ser modificado na interface do programa.

O WEKA possui uma interface gráfica, demonstrada na figura 4, com quatro aplicações: *Explore*r, *Experimenter*, *KnowledgeFlow e Simple CLI*.

Figura 4. Tela inicial do pacote WEKA [3].

- *Explorer*: módulo gráfico utilizado para explorar dados e executar os algoritmos a partir do carregamento de um arquivo de dados;
- *Experimenter*: utilizado para realizar experimentos, testes estatísticos e manipular a base de dados;
- *KnowledgeFlow*: similar ao *Explorer*, porém em uma interface *drag-and-drop*;
- *Simple CLI*: interface para a execução dos algoritmos em linha de comando.

Dentro do ambiente *Explorer* do WEKA, ambiente qual foi utilizado neste trabalho, encontram-se as opções *Preprocess* onde se pode abrir, editar e salvar a base de dados, *Classify* que contém o conjunto de algoritmos que implementam os esquemas de aprendizagem que funcionam como classificadores, *Cluster* onde contém os algoritmos para geração de grupos, *Associate* que possui o conjunto de algoritmos para gerar regras de associação, *Select Attributes* onde se pode determinar a relevância dos atributos e *Visualise* que explora os dados.

Dentro do software WEKA foi utilizado, como objeto de estudo do trabalho, o algoritmo de indução de árvores de decisão *weka.classifiers.trees.J48 -C 0.25 -M 2*.

RESULTADOS E DISCUSSÕES

Foram realizados os testes a partir do conjunto de dados já descrito com 150 instâncias referentes à classificação de flores "Íris".

Para a classificação das instâncias utilizou-se o algoritmo de árvore de decisão *weka.classifiers.trees.J48 -C 0.25 -M 2*.

Primeiramente foi transformado o arquivo da base de dados de um arquivo de texto para um arquivo do tipo ARFF devido à necessidade de processamento dos dados pelo software WEKA. O conjunto de dados foi definido dentro do arquivo com o nome iris e os atributos de suas instancias foram definidos como *slength* para o comprimento da sépala, *swidth* para a largura da sépala, *plength* para o comprimento da pétala, *pwidth* para a largura da pétala e *class* para as classes da flor. Todos os atributos do tipo real, exceto o atributo *class* que é classificação do tipo da flor.

Dentro das opções disponíveis de acordo com esse algoritmo foram utilizadas algumas opções padrões como a opção de poda ativa. Entre as opções de teste foi selecionada a opção *"percentage Split"* com a porcentagem padrão de 66%, dividindo-se assim os conjuntos de

instâncias de acordo com a porcentagem e utilizando 99 ocorrências dos dados para treino e as 51 restantes para teste.

A árvore gerada possui um número de 5 folhas e um tamanho de 9 nós. É exibida na Figura 5 a árvore de decisão gerada pelo J48 com a opção de Poda.

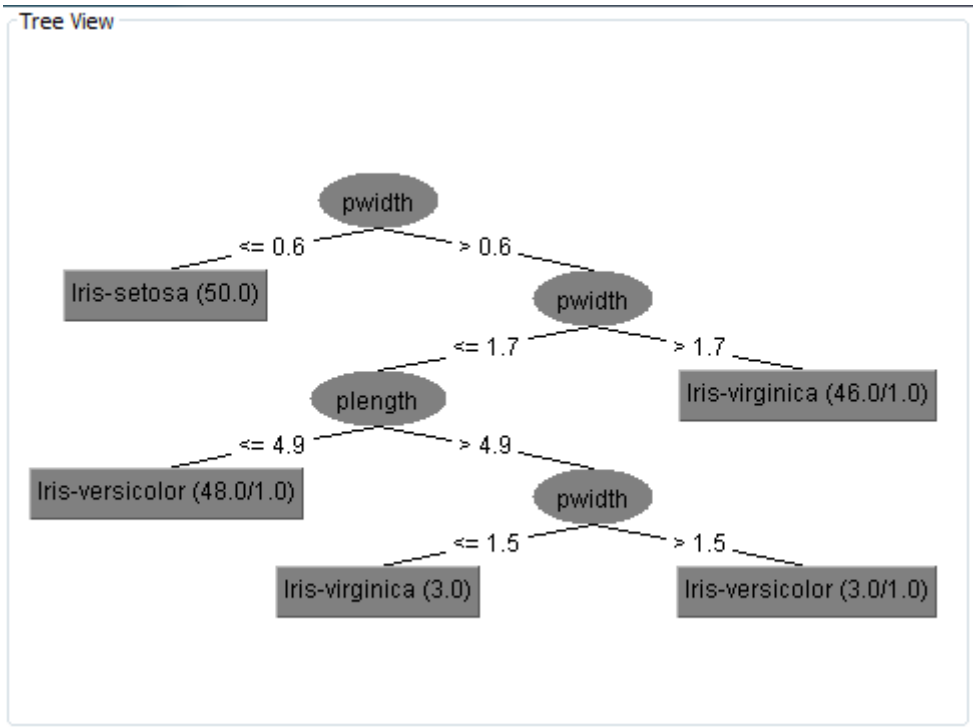

Figura 5. Árvore de Decisão Induzida.

O algoritmo J48, utilizado no trabalho, se baseia no algoritmo de árvores de decisão C4.5, que por sua vez, utiliza como para a seleção de atributos contínuos o método de testes simples.

Para a construção da árvore de decisão gerada o algoritmo começou calculando o ganho de informação para cada valor diferente de *pwidth*, de *plength*, de *swidth* e de *slength* do conjunto de dados de treinamento, e com isso delimitou o limiar de cada um deles. Após essa etapa ele calculou o ganho de informação total para cada um desses atributos a partir do limiar encontrado e selecionou o atributo que apresentou o ganho de informação máximo. O atributo escolhido foi o *pwidth* com o limiar de 0.6, que se tornou o nó raiz da árvore de decisão gerada.

Assim, o algoritmo foi repetindo o mesmo processo recursivamente escolhendo os atributos até que todas as instâncias do conjunto de treinamento fossem classificadas de uma maneira satisfatória, construindo a árvore de decisão apresentada na figura 5.

Descrição textual, por forma de algoritmo, da árvore de decisão construída:

```
pwidth <= 0.6: Iris-setosa (50.0)
pwidth > 0.6
|   pwidth <= 1.7
|   |   plength <= 4.9: Iris-versicolor (48.0/1.0)
|   |   plength > 4.9
|   |   |   pwidth <= 1.5: Iris-virginica (3.0)
|   |   |   pwidth > 1.5: Iris-versicolor (3.0/1.0)
|   pwidth > 1.7: Iris-virginica (46.0/1.0)
```

Após a construção da árvore de decisão foram utilizados as instâncias do conjunto de teste para analisar o desempenho e precisão da árvore construída.

O tempo necessário para a construção do modelo foi 0,05 segundos e o tempo necessário para testá-lo foi de 0,01 segundos.

O algoritmo apresentou um índice de 96.0784% de classificações corretas e 3.9216% de erro, classificando assim corretamente 49 das 51 instâncias utilizadas no teste.

Alguns valores estatísticos resultantes do teste:

- Erro médio absoluto 0,0396
- Erro quadrático 0,1579
- Erro absoluto relativo 8,8979%
- Raiz relativa erro quadrado 33,4091%
- Cobertura dos casos (0,95 nível) 96,0784%
- A média relativa tamanho da região (0,95 nível) 33,3333%
- Número total de Instâncias 51

Matriz de Confusão. Foi gerada também a matriz de confusão, exibida pela Tabela 1, para verificar o desempenho do algoritmo. A matriz de confusão de uma hipótese h oferece uma medida efetiva do modelo de classificação, ao mostrar o número de classificações corretas versus as classificações preditas para cada classe, sobre um conjunto de exemplos T.

Verificou-se na matriz a classificação correta de todas as flores da classe Íris setosa e Íris versicolor utilizadas e a classificação errônea de 2 flores Íris virgínica que foram classificadas como Íris versicolor.

Tabela 1. Matriz de confusão gerada.

Classificado como	A	B	C
A = Iris-setosa	15	0	0
B = Iris-versicolor	0	19	0
C = Iris-virginica	0	2	15

Precisão Detalhada por Classe. Para a análise da precisão da árvore de decisão gerada foram verificadas as curvas ROC de cada classe, atentando-se para os valores da Taxa de Verdadeiros Positivos (*TP Rate*), Taxa de Falsos Positivos (*FP Rate*), Precisão (*Precision*), Sensitividade (*Recall*) e Área da Curva ROC (*ROC Area*), apresentados na Tabela 2.

Tabela 2. Precisão detalhada por classe.

	TP Rate	FP Rate	Precision	Recall	ROC Area
Iris-setosa	1,000	0,000	1,000	1,000	1,000
Iris-versicolor	1,000	0,063	0,905	0,905	0,969
Iris-virginica	0,882	0,000	1,000	1,000	0,967
Média Ponderada	0,961	0,023	0,965	0,961	0,977

Os valores de *TP Rate* indicam as proporções de casos verdadeiros entre todos os casos com teste positivo, logo, quanto mais próximos de 1 melhor será a classificação. Já os valores para *FP Rate*, indicam as proporções de casos falsos entre todos os casos com teste falso, portanto, demonstram uma melhor classificação quando mais próximos de 0, evidenciando que essas medidas são complementares.

Para as classes de Íris setosa e Íris versicolor foi obtido um valor de *TP Rate* de 1 visto que todas as suas instâncias foram classificadas corretamente, já para a Íris virgínica esse valor foi menor devido ao fato de 2 de suas instâncias terem sidos classificadas como Íris versicolor. O valor de *FP Rate* maior que 0 para Íris versicolor indica que uma outra classe foi classificada erroneamente como versicolor, como sabemos a outra classe classificada como versicolor foi a da Íris vigínica.

Assim como os valores de verdadeiro positivo os valores de *ROC Area* também demonstram uma melhor classificação das instâncias quando mais próximos de 1, esse valor se refere a área abaixo da curva ROC da classe em questão e fornece uma medida para comparar a performances dos classificadores.

A medida de *Precision* se refere a porcentagem de amostras positivas classificadas corretamente sobre o total de amostras positivas e a medida de *Recall* a porcentagem de amostras positivas classificadas corretamente sobre o total de amostras classificadas como positivas [7].

As figuras 6, 7 e 8 ilustram respectivamente as Curvas ROC das flores Íris setosa, Íris versicolor e Íris virginica.

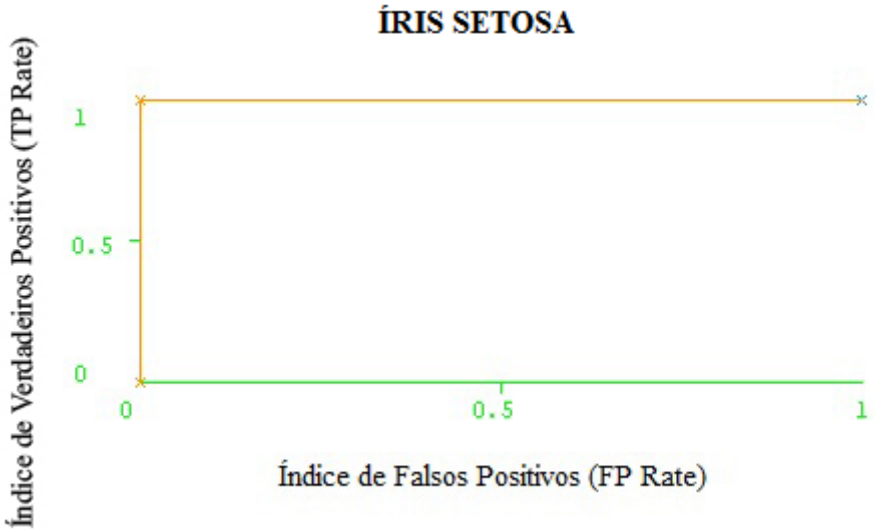

Figura 6. Curva ROC para Classe de Flores Íris Setosa.

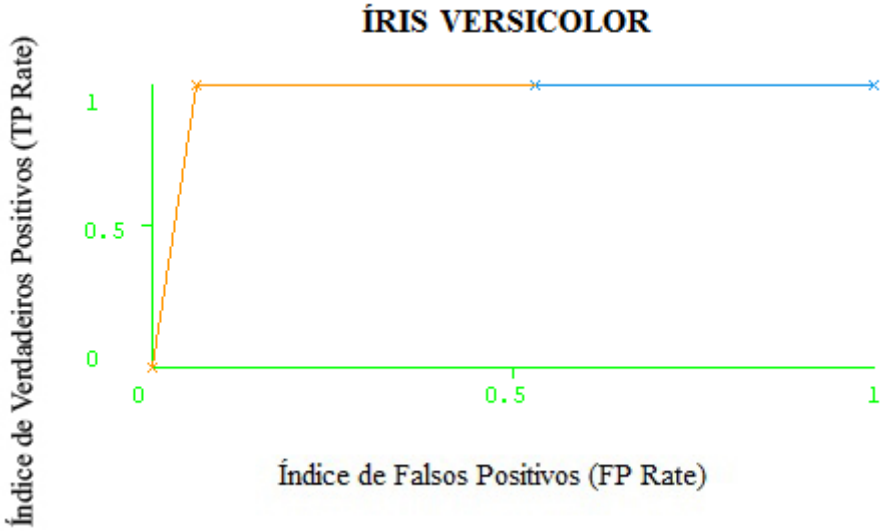

Figura 7. Curva ROC para Classe de Flores Íris Versicolor.

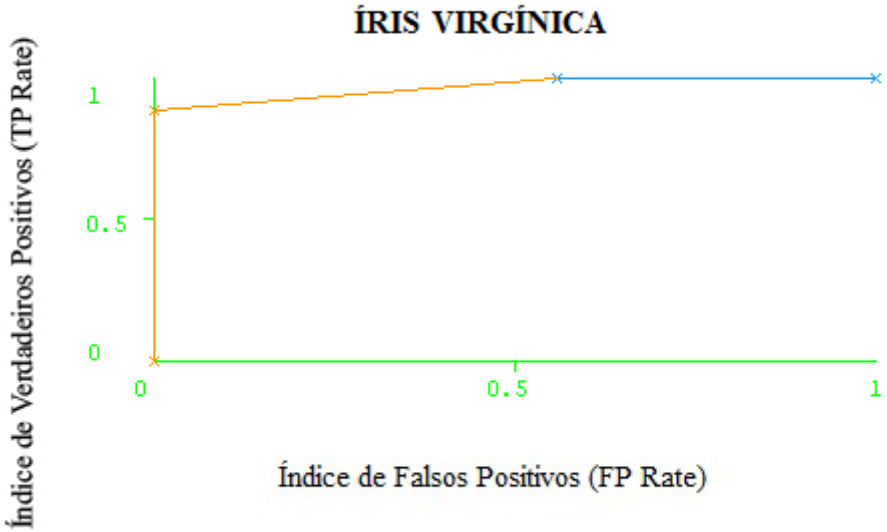

Figura 8. Curva ROC para Classe de Flores Íris Virgínica.

O gráfico da Curva ROC para a classe Íris setosa (Figura 6) corresponde a um modelo ideal de classificação, definido pela linha horizontal traçada a partir do ponto (0,1). As outras duas curvas porém apresentam pequenas variações, a da Íris versicolor (Figura 7) no valor de *FP Rate* um pouco maior que zero e a da Íris virgínica (Figura 8) com o valor de *TP Rate* um pouco menor que 1, evidenciando assim os pequenos erros de classificação já comentados para essas duas classes.

CONCLUSÕES

Este trabalho teve como o objetivo a revisão bibliográfica sobre os algoritmos de indução de árvores de decisão e suas principais características, bem como o entendimento de todo o conceito por de trás das árvores de decisões. Também foi alvo de estudo a análise das

características do algoritmo de indução de árvores de decisões J48 dentro do software WEKA e a realização de testes para a exemplificação, análise e entendimento dos conceitos e do algoritmo de indução de árvores de decisão utilizado.

Os resultados obtidos através dos testes mostraram uma excelente classificação dos dados das flores Íris utilizados. Foi obtido um índice de classificação correta das amostras do conjunto de dados de testes de mais de 95%, o que demonstra uma alta precisão da árvore de decisão construída pelo algoritmo. Pôde-se observar também a rapidez de construção e teste do modelo sendo respectivamente de 0,05 segundos e 0,01 segundos. O algoritmo foi extremamente satisfatório dentro do que ele se propôs, se mostrando muito bom para a classificação do tipo de dados os quais ele estava lidando e ajudou no entendimento de todo o conteúdo estudado.

Posteriormente é pretendido utilizar os conhecimentos adquiridos neste trabalho visando uma aplicação em dados de expressões gênicas (GRNs). Propondo assim, o desenvolvimento de um novo método de inferência de GRNs utilizando árvores de decisão a partir de informações funcionais dos genes, aprimorando, nesse contexto, o algoritmo utilizado e efetuando comparações com outras metodologias de inferência de redes gênicas e análise dos resultados.

REFERÊNCIAS

[1] SATO, Paula. **O que é inteligência artificial? Onde ela é aplicada?** Nova Escola. Disponível em: <http://revistaescola.abril.com.br/ciencias/fundamentos/inteligencia-artificial-onde-ela-aplicada-476528.shtml>.

[2] PUC-Rio. **Aprendizado de Máquina.** Disponível em: <http://www.maxwell.lambda.ele.puc-rio.br/10970/10970_4.PDF>.

[3] WAIKATO, U. O. Weka Data Mining Software in Java. **Weka - The University of Waikato**. Disponível em: <http://www.cs.waikato.ac.nz/ml/weka/>.

[4] UCI MACHINE LEARNING REPOSITORY, Center for Machine Learning and Intelligent Systems. **Iris Data Set**. Disponível em: <http://archive.ics.uci.edu/ml/datasets/Iris>.

[5] NORVIG, Peter. RUSSEL, Stuart Jonathan. **Inteligência artificial:** tradução da segunda edição. Elsevier, Rio de Janeiro, 2004.

[6] BARANAUKAS, José Augusto. **Indução de Árvores de Decisão.** Departamento de Física e Matemática – FFCLRP-USP. Disponível em: <http://professor.ufabc.edu.br/~ronaldo.prati/MachineLearning/AM-I-Arvores-Decisao.pdf>.

[7] **MEDIDAS DE DESEMPENHO:** Classificação SUPERVISIONADA. Disponível em: <http://adessowiki.fee.unicamp.br/media/Attachments/iaOPF/MainPage/Classificadores.ppt>

[8] COLONHEZI, Thiago Pereira. **Relatório Final de Atividades:** Caracterização de Bioimagens. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2012.

[9] HALMENSCHLAGER, Carine. **Um algoritmo para indução de árvores e regras de decisão.** Universidade Federal do Rio Grande do Sul – Instituto de Informática, Porto Alegre, 2002.

[10] ATTUX, Romis R. F. ZUBEN, Fernando J. Von. **Tópico 7:** Árvores de Decisão. DCA/FEEC/Unicamp.

[11] MARGOTTO, Paulo R. **CURVA ROC:** Como fazer e interpretar no SPSS. Escola Superior de Ciências da Saúde. Distrito Federal.

[12] COSTA, Paulo Dias. MARQUES, João Miguel. MARTINS, Antonio Cardoso. **Estudo Comparativo de Três Algoritmos de Machine Learning na Classificação de Dados Eletrocardiográficos.** Faculdade de Medicina da Universidade do Porto. Porto, 2009.

[13] R. O. Duda, P. E. Hart, and D. G. Stork. **Pattern Classification.** John Wiley and Sons, 2001.
[14] Mitchell, Tom M. **Machine Learning.** McGraw-Hill, New York, 1997.
[15] MARTINEZ, E. Z. **A curva ROC para testes diagnósticos.** Rio de Janeiro, p. 7-31, 2011.
[16] **Data set:** Iris.data. Disponível em: <http://www.statlab.uni-heidelberg.de/data/iris/>