UNIVERSIDADE DE SÃO PAULO
FACULDADE DE FILOSOFIA, CIÊNCIAS E LETRAS DE RIBEIRÃO PRETO
DEPARTAMENTO DE COMPUTAÇÃO E MATEMÁTICA

ADRIANO HENRIQUE CANTÃO

Ranqueamento de atributos por meio de Random Forests e métricas de centralidade em redes complexas

Ribeirão Preto–SP

2022

ADRIANO HENRIQUE CANTÃO

Ranqueamento de atributos por meio de Random Forests e métricas de centralidade em redes complexas

Versão Corrigida

Versão original encontra-se na FFCLRP/USP.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) da Universidade de São Paulo (USP), como parte das exigências para a obtenção do título de Mestre em Ciências.

Área de Concentração: Computação Aplicada.

Orientador: Prof. Dr. José Augusto Baranauskas
Coorientador: Prof. Dr. Zhao Liang

Ribeirão Preto–SP

2022

ADRIANO HENRIQUE CANTÃO

Feature ranking from Random Forest through Complex Network's centrality measures

Corrected Version

The original version is found at FFCLRP/USP.

Dissertation presented to Faculdade de Filosofia, Ciências e Letras de Ribeirão Preto (FFCLRP) from the Universidade de São Paulo (USP), as part of the requirements to hold the Master of Science degree.

Field of Study: Applied Computing.

Supervisor: Prof. Dr. José Augusto Baranauskas

Co-supervisor: Prof. Dr. Zhao Liang

Ribeirão Preto–SP

2022

Adriano Henrique Cantão
Ranqueamento de atributos por meio de Random Forests e métricas de centralidade em redes complexas. Ribeirão Preto–SP, 2022.
72p. : il.; 30 cm.

Dissertação apresentada à Faculdade de Filosofia, Ciências e Letras
de Ribeirão Preto da USP, como parte das exigências para
a obtenção do título de Mestre em Ciências,
Área: Computação Aplicada.

Orientador: Prof. Dr. José Augusto Baranauskas
Coorientador: Prof. Dr. Zhao Liang

1. Ranqueamento de Atributos. 2. Random Forests. 3. Redes Complexas 4. Métricas de Centralidade.

Adriano Henrique Cantão

Ranqueamento de atributos por meio de Random Forests e métricas de centralidade em redes complexas

Modelo canônico de trabalho monográfico acadêmico em conformidade com as normas ABNT.

Trabalho aprovado. Ribeirão Preto–SP, 13 de setembro de 2022:

Orientador:
Prof. Dr. José Augusto Baranauskas

Professora
Dra. Solange Oliveira Rezende

Professor
Dr. Aluizio Fausto Ribeiro Araújo

Ribeirão Preto–SP
2022

Dedico este trabalho a toda minha família,
em especial aos meus pais Nilse e Valdir
que sempre me deram apoio e suporte
para aprender o que aprendi e chegar onde cheguei.

"I am just a child who has never grown up.
I still keep asking these 'how' and 'why' questions.
Occasionally, I find an answer."
(Stephen Hawking)

Resumo

O volume de dados disponíveis aumentou rapidamente nos últimos anos e, com isso, os *datasets* geralmente acabam tendo muitos atributos irrelevantes que podem dificultar a compreensão humana e até levar a modelos de aprendizado de máquina ruins. É possível lidar com esse problema ordenando os atributos de acordo com suas relevâncias e, se desejado, pode ser aplicado um valor de corte ou a estratégia dos top-k para reduzir o número de atributos, mantendo apenas os mais relevantes. Esta pesquisa aborda esse problema e propõe um novo método que emprega árvores de uma *Random Forest* para transformar um *dataset* em uma rede complexa na qual métricas de centralidade são aplicadas para ranquear os atributos. O processo representa cada árvore como um grafo, onde todos os atributos na Árvore de Decisão são vértices e as ligações entre os nós (pai $\rightarrow$ filho) da árvore são representados por uma aresta ponderada entre os dois respectivos vértices. A união de todos os grafos de árvores individuais leva à rede complexa. Experimentos foram realizados em 97 *datasets* de classificação e regressão rotulados, com variação nos níveis de ruído dos atributos e dos exemplos. Os resultados mostram que, para redes complexas geradas a partir de *Random Forests*, as métricas de peso de aresta unitário e *out-of-bag* apresentaram melhores resultados para *datasets* de classificação e regressão, respectivamente; as métricas de centralidade tiveram melhor desempenho em redes não orientadas, em geral. É possível concluir que a centralidade do autovetor e a importância dos atributos da *Random Forest* têm desempenho equivalente. Em outras palavras, não houve diferença estatisticamente significativa entre eles em todas, exceto em uma situação (com 40% de ruído nos exemplos para *datasets* de regressão), com nível de confiança de 95%.

Palavras-chave: Ranqueamento de Atributos. Random Forests. Redes Complexas. Métricas de Centralidade.

Abstract

In recent years, the volume of available data has rapidly increased, and datasets commonly end up with many irrelevant features which may disturb human understanding and even lead to poor machine learning models. It is possible to deal with that problem by sorting the features according to their relevancy, and if desirable, either a threshold or the best top-k strategy can be applied to reduce the number of features, keeping only the most relevant ones. This research addresses this problem and proposes a novel method that employs trees from a Random Forest to transform a dataset into a complex network to which centrality measures are applied to rank the features. The process represents each tree as a graph where all features in the Decision Tree are vertices, and the links within the nodes (father $\rightarrow$ child) of the tree are represented by a weighted edge between the two respective vertices. The union of all graphs from individual trees leads to the complex network. Experiments were performed in 97 labeled classification and regression datasets, with a variation in the feature and example noise levels. Results show that, for complex networks generated from Random Forests, the edge-weight metrics *unitary* and *out-of-bag* presented better results for classification and regression datasets, respectively; centrality measures had better performance in non-oriented networks, in general. It is possible to conclude that the eigenvector centrality and the Random Forest feature importance have equivalent performance. In other words, there was no statistically significant difference between them in all except one situation (at 40% noise in the examples for regression datasets), at 95% confidence level.

Keywords: Feature Ranking. Random Forests. Complex Networks. Centrality Measures.

Lista de figuras

Lista de tabelas

Sumário

1

Introdução

O aumento constante de tecnologias e aplicações tem gerado quantidades massivas de dados de forma rápida. Esses dados podem incluir números, vídeos, imagem, textos e, frequentemente, possuem alta dimensionalidade, ou seja, um número muito alto de atributos. Por exemplo, houve aumento na quantidade de dados epidemiológicos por conta da pandemia de Covid-19[1]. Essa alta dimensionalidade apresenta muitos desafios para a análise de dados, tomada de decisão, classificação e predição, pois comumente possuem grande quantidade de atributos redundantes, que não proporcionam novas informações, e irrelevantes, que não melhoram o resultado (HASHEMI; DOWLATSHAHI; NEZAMABADI-POUR, 2020).

A presença de atributos irrelevantes, redundantes e ruidosos nos *datasets* podem deixar os algoritmos mais lentos, degradar o desempenho em tarefas de aprendizado e aumentar a dificuldade da interpretabilidade desses dados. Métodos de seleção de atributos são capazes de eleger um subconjunto de atributos a fim de resolver esse problema, removendo os atributos que identificam como irrelevantes, redundantes e ruidosos (MIAO; NIU, 2016).

Assim, a utilização de algoritmos de aprendizado de máquina é uma abordagem eficiente para para extrair informação dos dados. Quando a dimensionalidade dos dados é muito alta, no entanto, o uso de um método de seleção de atributos é primordial tanto para acelerar a execução dos algoritmos quanto para restringir a quantidade de dados a serem utilizados em testes de *benchmark*. Como resultado, a seleção de atributos evoluiu de um exemplo instrutivo para um evidente pré-requisito para a construção de modelos (SAEYS; INZA; LARRAÑAGA, 2007). Um método de seleção de atributos pode auxiliar na redução da enorme quantidade de dados médicos, biológicos e textuais que estão disponíveis para análise atualmente. Uma das formas de reduzir o número de atributos de um *dataset* é por meio de um *ranker*.

De acordo com Guyon e Elisseeff (2003), o método de ranqueamento (*ranker*) é utilizado como mecanismo primário ou auxiliar em diversos algoritmos de seleção de atributos devido à sua

[1] Muitos dos dados públicos divulgados pelos municípios da região não eram estruturados, o que abriu espaço para projetos como o PICOVID: Portal de Informações sobre a COVID-19 (CANTÃO; FAZIO, 2020), que realiza a coleta, estruturação, análise e disponibilização de informações em formato de séries temporais. Esse processo trouxe conhecimento e também gerou e disponibilizou ainda mais dados.

simplicidade, escalabilidade e sucesso empírico. O *ranker* primeiramente mostra a importância de cada atributo para representar a classe no *dataset* e, em um segundo momento, pode ser feita uma seleção de um subconjunto de atributos que encontram-se mais ao topo do ranking. Existem diversos benefícios ao realizar seleção dos atributos, dentre os mais importantes estão: (i) melhoria no entendimento e mais clareza na visualização dos dados, (ii) redução de tempo e melhora na performance ao trabalhar com os dados, (iii) redução das medidas necessárias, facilitando novas coletas, (iv) redução de espaço de armazenamento e (v) melhora na qualidade de predição.

Com base na literatura da área, nesta pesquisa é proposto um *ranker* que consiste no uso de uma rede complexa gerada a partir de uma *Random Forest*, na qual os atributos são extraídos para análise de suas relevâncias. O ranking gerado reflete a relevância de cada atributo para a rede complexa utilizada.

Uma *Random Forest* é um classificador composto por uma coleção de árvores geradas a partir de amostras aleatórias independentes. A classe majoritária entre as árvores define a classe de um novo exemplo (BREIMAN, 2001; DUBATH et al., 2011; ZHAO; ZHANG, 2008).

Redes complexas são grafos não triviais que combinam conceitos de estatística, sistemas complexos e teoria dos grafos. As redes complexas têm a capacidade de descrever espaços físicos, funções e relações topológicas entre os elementos representados (CARNEIRO; ZHAO, 2017). Trivial, neste caso, quer dizer um grafo com padrões de conexões que não sejam meramente regulares nem completamente aleatórios.

O foco de aplicação do *ranker* proposto é em *datasets* rotulados, ou seja, em aprendizado supervisionado. Aprendizado supervisionado é um paradigma na criação de modelos de aprendizado de máquina no qual as informações obtidas de *datasets* rotulados, denominado conjunto de treinamento, são utilizadas posteriormente para rotular novos dados (CUPERTINO et al., 2018).

1.1 Motivação

Há uma variedade de motivos para realizar a seleção de atributos. Primeiro, a seleção de atributos geralmente promove um aumento na precisão, pois muitos algoritmos de aprendizado de máquina desempenham de forma insatisfatória quando recebem muitos atributos. Em segundo lugar, a seleção de atributos pode aumentar a compreensibilidade, que se refere à capacidade humana de compreender os dados e as regras produzidas por algoritmos de aprendizado de máquina simbólicos, como árvores de decisão. Finalmente, a coleta de alguns atributos pode ser cara em domínios específicos, a seleção de atributos pode reduzir esse custo de coleta ao diminuir a quantidade de atributos a ser utilizada (FOITONG; PINNGERN; ATTACHOO, 2012).

A seleção de atributos relevantes é uma das maiores dificuldades no aprendizado supervisionado. Embora a maioria das abordagens de aprendizado busque escolher um subgrupo

ou atribuir valores de importância aos atributos, análises teóricas e experimentais mostram que muitos algoritmos não se saem bem em domínios com grandes quantidades de atributos irrelevantes. Por exemplo, independentemente do objetivo desejado, o número de exemplos de treinamento necessários para que o algoritmo *Nearest Neighbors* atinja um certo nível de acurácia parece expandir exponencialmente em relação à quantidade de atributos irrelevantes. Para alguns tipos de tarefa, mesmo abordagens como as que geram árvores de decisão univariadas, que selecionam explicitamente alguns atributos em favor de outros, exibem esse tipo de comportamento. Algumas técnicas são robustas quando se tratam de atributos irrelevantes, assim como o algoritmo *Naïve Bayes*, em contrapartida, podem ser demasiadamente sensíveis ao trabalhar com domínios que contenham atributos correlacionados, mesmo que esses atributos sejam relevantes. Considerando que esse tipo de técnica seja baseado na independência dos atributos, outros métodos podem ser necessário para a seleção de atributos relevantes quando o número de atributos disponível for muito alto (HAN; KAMBER; PEI, 2011).

Número de amostras muito grande ou muito limitado, alta dimensionalidade, número alto de classes e classes desbalanceadas, por exemplo, são desafios comuns para o aprendizado de algoritmos nos domínios biológico e médico. Isso pode explicar porque, apesar de a pesquisa em seleção de atributos não ser nova na comunidade de aprendizado de máquina, os pesquisadores continuam propondo uma variedade de algoritmos (DEVARAJ; PAULRAJ, 2015; DITZLER et al., 2015; GOVINDAN; NAIR, 2014; MANDAL; MUKHOPADHYAY; MAULIK, 2015; PURKAYASTHA et al., 2015).

As abordagens para a seleção de atributos podem ser classificadas em três grupos: (i) empacotados (*wrappers*) (EL ABOUDI; BENHLIMA, 2016): selecionam um subconjunto de atributos que proporcionam maior precisão, com relação a um indutor específico; (ii) filtros (*filters*) (ROFFO et al., 2021): selecionam os atributos de forma independente do indutor que será usado para rotular os dados, durante uma etapa de pré-processamento; (iii) embarcados (*embedded*) (LIU; ZHOU; LIU, 2019): incorporam a seleção dos atributos durante o processo de aprendizado do algoritmo. Outra opção é tentar melhorar a eficiência do processo de seleção de atributos usando um método híbrido (filtro e *wrapper*) (UNCU; TURKSEN, 2007; ESTÉVEZ et al., 2009; MIN; FANGFANG, 2010; LAN et al., 2011).

Na abordagem do tipo filtro, que é de especial interesse no escopo do presente trabalho, os atributos são filtrados sem que haja dependência ao algoritmo de indução utilizado. Por um lado, a principal desvantagem dessa abordagem é que essa independência faz com que o desempenho do algoritmo seja totalmente ignorado. Por outro lado, uma vez filtrado, o *dataset* pode ser utilizado por diversos paradigmas ou indutores e terá maior eficiência computacional ao reduzir o tempo de processamento necessário.

Em *datasets* com quantidade relativamente grande de atributos, a alta dimensionalidade pode prejudicar o processo de aprendizado e levar a um aumento na taxa de erro. Um ranker é capaz de identificar e ordenar os atributos por relevância e, posteriormente, é possível diminuir a

quantidade de atributos, para que somente os mais relevantes sejam utilizados no processo de aprendizado. Há indicativos de que a utilização de um *ranker* é uma solução pertinente para esse tipo de problema.

Para representar um *dataset* em uma rede complexa deve-se usar alguma metodologia que permita identificar quais exemplos ou atributos devem estar conectados entre si. Assim sendo, foi utilizada, de forma inovadora, a estrutura de uma *Random Forest* como mecanismo para identificar essas conexões, uma vez que cada árvore da Random Forest é composta por atributos e conexões entre pares de atributos. Assim, ao transformar um *dataset* em uma *Random Forest* e então transformá-la em uma rede complexa, foi observado que as informações obtidas na caracterização topológica consideram não apenas a informação local dos atributos (presentes nos exemplos utilizados para a criação da rede), mas também passam a considerar informações globais, que são proporcionadas pela rede.

Embora a *Random Forest* possa fornecer medida de importância dos atributos (*feature importance*) (BREIMAN, 2001), poucos trabalhos têm estudado as propriedades teóricas e os mecanismos estatísticos dessas medidas (LOUPPE et al., 2013). Por outro lado, a estrutura topológica de uma rede complexa permite o uso de medidas de centralidade (BONACICH, 1987) que, como meio alternativo, podem contribuir para um melhor ordenamento da importância dos atributos. O ranking gerado reflete a relevância de cada atributo para aquela rede complexa. A maneira inovadora de realizar a transformação de conjuntos de atributos em redes complexas proposta neste estudo preenche uma lacuna que há na literatura entre *Random Forests* e redes complexas.

1.2 Objetivo

O objetivo principal desta pesquisa é propor um *ranker* de atributos utilizando redes complexas geradas a partir da estrutura de *Random Forests* que seja mais eficiente em identificar atributos relevantes do que o *ranker* da própria *Random Forest*.

Os objetivos secundários buscam responder, no Capítulo 5 às seguintes questões:

- Das taxas de ruído nos exemplos (5%, 10%, 20% e 40%) há alguma em que o ranking proposto teve resultado superior ao método RF(), com diferença estatística significativa?
- Das orientações de aresta nas redes utilizadas (entrada, saída e não-orientada) há alguma que proporcionou melhores resultados para as métricas de centralidade?
- Dentre as métricas de centralidade utilizadas (*strength*, *eigenvector* e Katz) há alguma que forneceu um ranking superior às demais?

- Dentre as métricas de peso de aresta (unitário, *out-of-bag*, Gini e média geométrica) há alguma que proporcionou melhores resultados para as métricas de centralidade?

1.3 Organização

O restante deste documento está organizado da seguinte forma:

No Capítulo 2 é apresentada a fundamentação teórica sobre aprendizado de máquina, árvores de decisão e *Random Forests*, redes complexas e métricas de centralidade, como força do vértice, centralidade de autovetor e índice Katz; bem como sobre seleção e ranking de atributos. Na Seção 2.6 são apresentados os trabalhos relacionados ao aqui proposto.

No Capítulo 3 é apresentado o projeto de pesquisa, a abordagem metodológica utilizada para gerar uma rede complexa a partir de árvores de uma *Random Forest*, assim como os detalhes sobre os Algoritmos 1, 2 e 3.

No Capítulo 4 é apresentada a configuração dos experimentos realizados, seguida de detalhes sobre a origem dos dados e sobre a utilização de datasets artificiais, inserção de ruído, métrica elaborada para avaliar a proposta aqui apresentada e também a forma de validação utilizada.

No Capítulo 5 são apresentados e discutidos os resultados experimentais separados em dois grupos (i) *datasets* de classificação e (ii) *datasets* de regressão.

No Capítulo 6 são apresentadas as principais conclusões e contribuições desta pesquisa.

2

Fundamentação Teórica

2.1 Considerações Iniciais

Neste capítulo será apresentada a fundamentação teórica sobre Aprendizado de Máquina, os algoritmos sobre árvores de decisão e *Random Forest*, assim como Redes Complexas e métricas de centralidade, seleção de atributos em Aprendizado de Máquina. Por último, são descritos alguns trabalhos relacionados ao temas desta pesquisa.

2.2 Aprendizado de Máquina

Mitchell (1997) define Aprendizado de Máquina supervisionado como um programa de computador que aprende a partir de uma experiência E com respeito a alguma classe de tarefas $\mathcal{T}$ e medida de performance P, se sua performance nas tarefas em $\mathcal{T}$, medida por P, melhora com a experiência E.

Em geral, os dados brutos coletados são transformados no formato padrão, previamente definido na Tabela 1 na página seguinte. Embora outros formatos possam ser definidos, o formato adotado como padrão representa os dados brutos de uma forma simples e uniforme, que é utilizado universalmente por várias técnicas de Mineração de Dados que utilizam algoritmos de Aprendizado de Máquina.

Assim, exemplos são tuplas $z_i = (x_{i1}, x_{i2}, \ldots, x_{im}, y_i) = (\vec{x}_i, y_i)$ também denotados por (x_i, y_i), onde fica subentendido o fato que x_i é um vetor. A última coluna, $y_i = f(x_i)$, é a função que tenta-se predizer a partir dos atributos. Observa-se que cada x_i é um elemento do conjunto $X_1 \times X_2 \times \ldots \times X_m$ e quando se trata de classificação y_i pertence a uma das k classes ($y_i \in \{C_1, C_2, \ldots, C_k\}$), no caso de regressão y_i assume valores reais.

Dado um *dataset* de treinamento, um indutor (algoritmo de aprendizado) gera como saída um classificador (também denominado hipótese ou descrição de conceito) de forma que,

Tabela 1 – *dataset* T no formato atributo-valor, com n exemplos e m atributos. A linha i refere-se ao i-ésimo exemplo ($i = 1, 2, \ldots, n$) e a entrada x_{ij} refere-se ao valor do j-ésimo ($j = 1, 2, \ldots, m$) atributo X_j do exemplo i.

Exemplo	X_1	X_2	$\cdots$	X_m	Y
z_1	x_{11}	x_{12}	$\cdots$	x_{1m}	y_1
z_2	x_{21}	x_{22}	$\cdots$	x_{2m}	y_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
z_n	x_{n1}	x_{n2}	$\ldots$	x_{nm}	y_n

dado um novo exemplo, ele possa predizer com a maior precisão possível a sua classe.

Formalmente, em classificação, um exemplo é um par $z_i = (x_i, f(x_i))$ onde x_i é a entrada e $f(x_i)$ é a saída. A tarefa de um indutor é, dado um *dataset*, induzir uma função h que aproxima f, normalmente desconhecida. Neste caso, h é chamada uma hipótese sobre a função objetivo f, ou seja, $h(x_i) \approx f(x_i)$.

Na seção seguinte será descrito um indutor de interesse nessa pesquisa que consiste na criação de árvores de decisão também utilizadas na criação de *Random Forests*.

Definição de Dataset

Datasets são compostos por dados, exemplos e atributos independentes, podendo ou não haver um atributo dependente denominado classe e podem ser reais ou artificiais.

Dados são elementos que por si só não esclarecem incertezas, ou seja, são elementos sem significado quando em seu estado original (bruto), porém os dados são a matéria prima da informação. A informação é o resultado de uma situação de decisão baseada em um *dataset* (TEREZINHA, 2003).

Exemplos são vetores contendo dados sobre uma única observação, uma instância do contexto dos dados. Cada dado desse vetor é referente a um atributo, isto é, uma propriedade mensurável independente relativa ao evento sendo observado (CHANDRASHEKAR; SAHIN, 2014).

De acordo com Monard e Baranauskas (2003), os atributos podem ser vistos como descrições ou características analisadas de um exemplo; existem ao menos dois tipos de atributos, normalmente agrupados da seguinte maneira:

- Nominais: não existe ordem lógica estabelecida entre os valores. Por exemplo, cor: vermelho, verde, azul;
- Contínuos: existe uma ordem lógica estabelecida entre os valores. Por exemplo, peso: valor numérico pertencente ao conjunto dos números reais (peso $\in \mathbb{R}$).

Além dos atributos independentes, pode haver também um atributo especial, dependente dos demais, conhecido por classe. *Datasets* que possuem o atributo classe são denominados rotulados, enquanto os *datasets* que possuem somente os atributos independentes são denominados não-rotulados (CHANDRASHEKAR; SAHIN, 2014).

Um *dataset* pode ser real, se for gerado a partir de eventos do mundo real de um domínio específico, ou artificial, se for gerado de forma programática.

Ao trabalhar com *datasets* reais é necessário a cooperação de especialistas deste mesmo domínio para identificar quais são os exemplos que possuem ruído (se existir), o que pode aumentar o custo e a duração da etapa de pré-processamento. Esta limitação pode ser atenuada ao inserir ruído simulado de forma sistemática ou utilizando *datasets* artificiais (GARCIA et al., 2019).

Os *datasets*, sejam eles reais ou artificiais, são frequentemente transformados no formato atributo-valor definido na Tabela 1 na página precedente, popularmente conhecido por formato padrão (FERRO; MONARD; CAROLINA, 2011). Embora este formato não seja único, o formato denominado padrão é utilizado universalmente por representar os dados, coletados ou gerados, de forma descomplicada e universal.

2.3 Algoritmos de Árvores

2.3.1 Árvores de Decisão

A construção de uma árvore de decisão realiza-se da seguinte forma (BREIMAN et al., 1984; QUINLAN, 1986): utilizando o *dataset* de treinamento, um atributo é escolhido de forma a particionar os exemplos em subconjuntos, de acordo com valores deste atributo. Para cada subconjunto, outro atributo é escolhido para particionar novamente cada um deles. Este processo prossegue, enquanto um dos subconjuntos contenha uma mistura de exemplos pertencendo a classes diferentes. Uma vez obtido um subconjunto uniforme, quando todos os exemplos naquele subconjunto pertencem à mesma classe, um nó folha é criado e rotulado com o mesmo nome da respectiva classe.

Quando um novo exemplo deve ser classificado, começando pela raiz da árvore induzida, o classificador testa e desvia para cada nó com o respectivo atributo até que atinja uma folha. A classe deste nó folha será atribuída ao novo exemplo.

2.3.2 Random Trees e Random Forests

Considere um *dataset* de treinamento T com m atributos e n exemplos, seja T_l uma amostra *bootstrap* (EFRON; TIBSHIRANI, 1993) do *dataset* de treinamento a partir de T com reposição, contendo n exemplos. Uma *Random Tree* é construída da seguinte forma: em cada nó da árvore é escolhido um atributo a partir de a atributos aleatórios, onde $a \leq m$. Desta forma, ao criar diversas *Random Trees*, cada uma delas possui a mesma probabilidade de ser amostrada. A combinação dessa diversidade de *Random Trees* é conhecida como *Random Forest*, detalhada a seguir.

Uma *Random Forest* é definida formalmente como um classificador composto por uma coleção de árvores $\{h_l(x)\}, l = 1, 2, \ldots, L$, onde T_l são amostras aleatórias independentes e de distribuição idênticas. Cada árvore prediz a classe da entrada x e, em seguida, a classe mais popular entre as árvores é eleita para esta mesma entrada (BREIMAN, 2001; DUBATH et al., 2011; ZHAO; ZHANG, 2008).

Portanto, *Random Forests* aplicam o mesmo método que o *bagging* (BREIMAN, 1996) para produzir amostras aleatórias de *datasets* de treinamento (amostras *bootstrap*) para cada árvore. Breiman (2001) justifica o uso do método *bagging* em *Random Forests* por duas razões: o uso do *bagging* parece melhorar o desempenho quando atributos aleatórios são usados; *bagging* pode ser usado para fornecer estimativas contínuas do erro de generalização do conjunto combinado de árvores, assim como estimativas para força e correlação, usando o estimador *out-of-bag*.

O erro de classificação da floresta depende da força das árvores individuais da floresta, da correlação entre quaisquer duas árvores na floresta e da importância dos atributos (BREIMAN, 2001; BREIMAN, 2004; BREIMAN; CUTLER, 2004; MA; GUO; CUKIC, 2007), a saber:

- Força da árvore individual na floresta: pode ser interpretada como uma medida de desempenho para cada árvore. Uma árvore com uma taxa de erro baixa é um classificador forte. Assim, aumentando a força das árvores individuais, reduz-se a taxa de erro da floresta.

- Correlação entre as árvores da floresta: duas medidas de aleatoriedade (uso do *bagging* e seleção aleatória de atributos) fazem com que as árvores sejam diferentes e, portanto, diminui a correlação entre elas. A baixa correlação tende a diminuir a taxa do erro de classificação.

- Importância dos atributos: Após a construção da floresta, um dos atributos do *dataset* tem seus valores permutados nos exemplos *out-of-bag* e, após a permutação, os exemplos são apresentados à respectiva árvore e, por fim, é comparada a taxa de acerto na classificação com e sem a permutação deste atributo. Este processo de permutação é repetido para cada

atributo do *dataset*. Quão maior o aumento na taxa de erro gerado pela permutação de um atributo, maior é a importância deste atributo para a representação da classe nesse *dataset*.

A metodologia aqui proposta compreende a utilização de uma *Random Forest* para definir as ligações entre os atributos e o peso dessas ligações. A partir dessa definição é feita a representação do *dataset* em uma rede complexa. A seguir são apresentados os detalhes sobre Redes Complexas.

2.4 Redes Complexas

Redes complexas são grafos não triviais que combinam conceitos de estatística, sistemas complexos e teoria dos grafos. As redes complexas têm a capacidade de descrever espaços físicos, funções e relações topológicas entre os elementos representados (CARNEIRO; ZHAO, 2017). Trivial, neste caso, quer dizer um grafo com padrões de conexões que não sejam meramente regulares nem completamente aleatórios.

No domínio das Redes Complexas foi mostrado que estruturas complexas de rede são capazes de descrever uma variedade de sistemas como, por exemplo, a Internet que é uma rede complexa composta de roteadores, computadores, entre outros dispositivos, conectados por *links* físicos ou *wireless*, redes sociais, redes de distribuição de energia elétrica e rede de colaboração. Devido à circunstâncias como o aumento da quantidade de dados em diversas áreas, mapeados em topologia de rede, e do poder computacional, novos conceitos e métricas começaram a ser estudados, com destaque, até aquele momento, em três conceitos específicos: redes de pequeno-mundo (*small-world*), agrupamento (*clustering*) e distribuição de grau (*degree distribution*) (ALBERT; BARABÁSI, 2002).

Destacam-se, ainda, três principais modelos de Redes Complexas: (i) redes aleatórias, modelo mais simples, propostas por Erdos e Rény, cujas arestas são ligadas aos vértices de forma aleatória a partir de uma predeterminada probabilidade de conexão; (ii) redes pequeno-mundo, cujos vértices são altamente aglomerados e suas conexões ocorrem por meio de caminhos mínimos e as (iii) redes livres de escala, cuja distribuição de graus dos vértices segue uma lei de potência. Nas redes de livre escala existe uma tendência de um novo vértice conectar-se a outro que já tenha um grau elevado, o que gera uma rede com poucos vértices altamente conectados e muitos vértices pouco conectados (CARNEIRO; ZHAO, 2017; HUISMAN, 2016).

Diversas estruturas discretas, como listas e árvores, podem ser representadas por grafos. Devido a isso, há vários casos em Redes Complexas que uma estrutura de interesse é primeiramente representada por uma rede para em seguida analisar as características topológicas utilizando medidas informativas desta rede (COSTA et al., 2007).

Segundo (SILVA; ZHAO, 2012), algumas dessas medidas informativas são:

- centralidade, em que a importância de um vértice para a rede é calculada de acordo com a quantidade de caminhos que passam por este vértice;
- coeficiente de aglomeração, que demonstra o quão conectada está a vizinhança de um determinado vértice. Esta medida captura a estrutura quase-local por meio da contagem de conexões em triângulo formadas pelo vértice a ser analisado e dois de seus vizinhos;
- assortatividade, que quantifica a propensão de um novo vértice se conectar com determinados outros ao ser adicionado à rede; e
- modularidade, que identifica qual a força de divisão da rede em módulos (ou comunidades) após a remoção de determinadas arestas.

Redes Complexas têm se estabelecido como uma ferramenta poderosa; sua estrutura topológica tem se mostrado útil para a detecção de classes e agrupamentos, seja por algoritmos de aglomeração ou de classificação. Devido a este fato, houve um aumento na utilização de métodos de Aprendizado de Máquina baseados em redes (SILVA; ZHAO, 2016) que tornou-se uma área de pesquisa ativa com uma diversidade de aplicações bem sucedidas na abordagem de utilização de informações globais, como: aprendizado semi-supervisionado (VERRI; ZHAO, 2016), agrupamento de dados (FERREIRA; ZHAO, 2016), regressão (NI; YAN; KASSIM, 2012) e classificação (CARNEIRO; ZHAO, 2017; CUPERTINO et al., 2018; NETO; ZHAO, 2013).

2.4.1 Representação de Redes Complexas

Redes complexas são comumente representadas por meio da teoria de grafos que, inicialmente, nos anos 50, teve um foco em grafos regulares; ainda neste período redes de grande escala, as quais não apresentavam aparentes modelos estruturais, foram descritas como grafos aleatórios, inicialmente estudados pelos matemáticos Erdös e Rényi (1960). Eles definiram o modelo Erdos-Renyi com N nós e uma probabilidade p de conexão entre pares de nós. Tal modelo foi referência por muito tempo, mas, com o crescimento das pesquisas nesta área, acabou sendo questionado, pois intuitivamente muitos sistemas indicavam seguir alguns princípios de organização estrutural que representaria determinada topologia e não uma estrutura aleatória (ALBERT; BARABÁSI, 2002).

As representações de redes complexas ocorrem através de (i) grafos orientados (dígrafos) ponderados, e de suas derivações, (ii) dígrafos não-ponderados, (iii) grafos ponderados e (iv) grafos não-ponderados. Em um (dí)grafo ponderado, o peso de uma ligação é representado por w_{ij} sempre que houver uma aresta que ligue um par distinto de nós, indo do vértice i para o vértice j. Na literatura é comumente assumido que em um grafo não há laços (aresta entre um par de vértice (i, i)) e também não há múltiplas arestas, de uma mesma direção, entre um

mesmo par de vértices $\{(i_1, j_1) \ldots (i_m, j_m)\}$. Existem, porém, grafos com laços e também com múltiplas arestas, são denominados de multigrafos. (COSTA et al., 2007).

Para um grafo ser considerado orientado (direcionado) todas as suas arestas devem ser orientadas. Um grafo não-orientado pode ser representado por um grafo orientado que para cada par de vértice conectados há um par de arestas, uma apontando para cada direção (NEWMAN, 2003).

A representação de uma rede complexa adotada é a matriz de adjacência A. Para os grafos ponderados (redes), objeto de estudo nesta pesquisa, assume-se que a cada aresta (i, j) entre os vértices i e j $(1 \leq i, j \leq m)$ existe um peso associado w_{ij}. Assim, A_{ij} é igual ao peso associado à aresta, caso exista a aresta entre os vértices i e j; $A_{ij} = 0$, caso contrário, dado pela Equação (2.1).

$$A_{ij} = \begin{cases} 0, & \text{se não existe aresta } (i, j) \\ w_{ij}, & \text{se existe aresta } (i, j) \text{ com peso } w_{ij} \end{cases} \tag{2.1}$$

A partir da matriz de adjacência é possível calcular diversas métricas, desenvolvidas especialmente para redes complexas (Equações 2.2 – 2.10). De especial interesse nesta pesquisa são as métricas de centralidade, algumas das quais descritas a seguir. Métricas de centralidade são utilizadas para analisar a importância de um vértice para a rede.

2.4.2 Métricas de Centralidade

Uma métrica de centralidade do vértice i será representada de forma genérica por $C(i)$. Para redes orientadas, a métrica de centralidade que considera somente as arestas de entrada do vértice i, para calcular sua importância, será denotada por $C^{\text{in}}(i)$; a que considera somente as arestas de saída do vértice i será representada por $C^{\text{out}}(i)$.

Força do Vértice

Considerando uma rede complexa, orientada ou não-orientada, uma medida simples porém expressiva das propriedades desta rede, considerando os pesos, é obtida ao observar a grandeza chamada de força do vértice i (*vertex strength*), denotada por $\texttt{ST}(i)$ e dada pela Equação (2.2). Esta grandeza mede a força de um vértice por meio da soma dos pesos das arestas que lhe são incidentes, definida conforme as Equações (2.3) e (2.4) (BARRAT et al., 2004; COSTA et al., 2007). Essa métrica retorna valores no intervalo $[0, +\infty]$ e quanto maior, mais central é o vértice na rede.

$$C(i) \triangleq \mathtt{ST}(i) = \mathtt{ST}^{in}(i) + \mathtt{ST}^{out}(i) \tag{2.2}$$

$$C^{\text{in}}(i) \triangleq \mathtt{ST}^{\text{in}}(i) = \sum_j A_{ji} \tag{2.3}$$

$$C^{\text{out}}(i) \triangleq \mathtt{ST}^{\text{out}}(i) = \sum_j A_{ij} \tag{2.4}$$

Centralidade de Autovetor

Esta medida de centralidade de autovetor (*Eigenvector Centrality*), proposta por Bonacich (1987), utiliza uma função recursiva que calcula a centralidade dos vizinhos de um vértice para posteriormente determinar a centralidade deste mesmo vértice. Para cada elemento da rede, a centralidade de i corresponde a centralidade dos vizinhos atenuada pelo maior autovetor λ_1 da matriz de adjacência não negativa A, que é representada pelas Equações (2.5 – 2.7) (BILLIO; PELIZZON; SAVONA, 2016; BORBA; TREVIZAN, 2013). Essa métrica retorna valores no intervalo $[0, 1]$ e quanto mais próxima de 1, mais central é o vértice na rede.

$$C(i) \triangleq \mathtt{EC}(i) = \frac{1}{\lambda_1} \sum_j A_{ij} \times \mathtt{EC}(j) \tag{2.5}$$

$$C^{\text{in}}(i) \triangleq \mathtt{EC}^{\text{in}}(i) = \frac{1}{\lambda_1} \sum_j A_{ji} \times \mathtt{EC}^{\text{in}}(j) \tag{2.6}$$

$$C^{\text{out}}(i) \triangleq \mathtt{EC}^{\text{out}}(i) = \frac{1}{\lambda_1} \sum_j A_{ij} \times \mathtt{EC}^{\text{out}}(j) \tag{2.7}$$

Índice de Katz

O índice de Katz (*Katz Index*) é uma variação da centralidade de autovalor proposta por Bonacich. Na medida proposta por Katz (1953), há dois parâmetros adicionais, α e β, cujos valores são definidos pelo usuário. O parâmetro β é um termo arbitrário que pode ser utilizado para atribuir pesos extras aos vizinhos imediatos, recebe um valor escalar $\beta \geq 0$. As ligações com os vizinhos mais distantes são penalizadas por um fator de atenuação α. Este valor de atenuação deve ser estritamente menor do que o inverso do maior autovetor da matriz de adjacência para que possa convergir, ou seja, $\alpha < \lambda_1^{-1}$. Assim, as Equações (2.8 – 2.10) definem a centralidade de Katz de um vértice i (BILLIO; PELIZZON; SAVONA, 2016; BORBA; TREVIZAN, 2013). Essa métrica retorna valores no intervalo $[0, +\infty]$ e quanto maior, mais central é o vértice na rede.

$$C(i) \triangleq \texttt{Katz}(i) = \alpha \sum_j A_{ij} \times \texttt{Katz}(j) + \beta \tag{2.8}$$

$$C^{\text{in}}(i) \triangleq \texttt{Katz}^{\text{in}}(i) = \alpha \sum_j A_{ji} \times \texttt{Katz}^{\text{in}}(j) + \beta \tag{2.9}$$

$$C^{\text{out}}(i) \triangleq \texttt{Katz}^{\text{out}}(i) = \alpha \sum_j A_{ij} \times \texttt{Katz}^{\text{out}}(j) + \beta \tag{2.10}$$

Tanto a centralidade de autovetor como a centralidade de Katz são capazes de mensurar os efeitos dos vizinhos de um vértice; porém na segunda métrica há maior possibilidade de ajustes. Caso sejam utilizados os valores $\alpha = \lambda_1^{-1}$ e $\beta = 0$, esta equação é reduzida para a centralidade de autovetor (BILLIO; PELIZZON; SAVONA, 2016).

Ao aplicar uma das medidas de centralidade, acima descritas, em uma rede complexa composta por atributos, cada atributo recebe um valor (pontuação). O método proposto por essa pesquisa ordena os atributos de acordo com seus valores e gera um ranking. Esse método é denominado *ranker*. A seguir serão apresentados mais detalhes sobre um *ranker*.

2.5 Seleção de atributos

Os métodos para a seleção de atributos podem ser classificadas em três grupos: *wrappers* (EL ABOUDI; BENHLIMA, 2016), filtros (ROFFO et al., 2021) e *embedded* (LIU; ZHOU; LIU, 2019).

De acordo com Bolón-Canedo e Alonso-Betanzos (2019), Venkatesh e Anuradha (2019),

- *Wrappers*: escolhem e testam diversas combinações de atributos e, ao fim, selecionam o subconjunto de atributos que resulto em maior desempenho;
- Filtros: atuam na etapa de pré-processamento, o método utilizado aqui é independente do algoritmo a ser utilizado para trabalhar com esses dados. Como processo principal, os atributos são primeiramente pontuados de acordo com algum critério de relevância e são ranqueados com base nessa pontuação. O segundo passo é a seleção dos atributos melhores posicionados no ranking gerado; e
- *Embedded* buscam reduzir o tempo de processamento, que acontece nos *wrappers*, na seleção do melhor subconjunto e, para isso, inserem esse processo de seleção como parte do processo de treinamento dos algoritmos.

No entanto, é possível utilizar um método híbrido para tentar melhorar a eficiência do processo de seleção de atributos. O método híbrido realiza uma primeira seleção com um método filtro e, em seguida, utiliza um *wrapper* nos dados pré-selecionados para uma segunda

seleção (ESTÉVEZ et al., 2009; LAN et al., 2011; MIN; FANGFANG, 2010; UNCU; TURKSEN, 2007).

Os métodos de filtro utilizam mecanismos de ranking e seleção por causa de sua escalabilidade, simplicidade e sucesso empírico. A utilização do *ranker* é favorável, se comparado a outros métodos de seleção de atributos que testam subconjuntos de atributos, por causa de sua eficiência computacional e escalabilidade estatística (GUYON; ELISSEEFF, 2003).

Ranking de Atributos

Um *ranker* é um dos métodos utilizados para seleção de atributos. Este método primeiramente utiliza um determinado critério para pontuar os atributos e gera um ranking baseado nessa pontuação. Posteriormente, é aplicado um valor limite (*threshold*) t para selecionar os atributos com valor $\geqslant t$ ou, alternativamente, fixa-se um valor θ e seleciona-se apenas os θ primeiros atributos do ranking. *rankers* são comumente utilizados por conta de sua simplicidade e sucesso em aplicações práticas (CHANDRASHEKAR; SAHIN, 2014).

O ranking gerado por esse método fornece discernimento sobre os dados ao apresentar claramente a relevância de cada atributo e, quando apenas os atributos mais relevantes são selecionados, é capaz de melhorar o desempenho dos algoritmos de aprendizado. A qualidade dos atributos é um dos principais fatores responsáveis pela variância na taxa de erro na tarefa de classificação. O *ranker* simplifica a dimensão do problema ao diminuir o número de possíveis permutações para gerar o ranking (HALL; HOLMES, 2003).

A utilização de um *ranker* é favorável, mesmo em casos onde não é uma solução ótima, por causa de sua eficiência computacional, é preciso computar somente f atributos ao invés de diversos subconjuntos, e escalabilidade estatística, diminui consideravelmente a variância se tornando robusto contra *overfitting* (GUYON; ELISSEEFF, 2003).

O *ranker* proposto nesta pesquisa é caracterizado como um filtro, por suas características de gerar, como resultado, a ordenação dos atributos do mais relevante ao menos relevante no *dataset* ao qual eles pertencem.

2.6 Trabalhos Relacionados

No restante desta seção são apresentados os trabalhos relacionados encontrados na literatura. Destaca-se que, nos trabalhos encontrados, os mesmos fizeram uso de *Random Forest* para analisar dados que já estavam representados sob a forma de redes complexas. Ressalta-se que, até o momento em que a revisão bibliográfica foi realizada nesta proposta de pesquisa, nenhum outro trabalho que utilize uma *Random Forest* como etapa de pré-processamento na representação

de redes complexas foi encontrado na literatura.

Na literatura, estudos utilizam a representação de *datasets* em redes complexas em busca de resolver diversos problemas típicos de Aprendizado de Máquina. Por conta disso, metodologias foram propostas para a conversão de conjuntos tradicionais de exemplos em grafos, como *KNN-Graph* (EPPSTEIN; PATERSON; YAO, 1997) e grafos-K-associados (BERTINI; ZHAO; LOPES, 2013). Alguns métodos utilizam redes-híbridas cujos vértices podem ser exemplos ou atributos do conjunto de dados (VERRI; ZHAO, 2016). No entanto, o objeto de estudo desses modelos consiste nos vértices que representam exemplos do conjunto de dados, tipicamente encontrados em Aprendizado de Máquina.

No trabalho de Zanin et al. (2013), foram testados três métodos de seleção de atributos em redes de dados de espectrometria de massa (dois métodos baseados em informação mútua e um em particionamento [*binning*]). Como resultado, foi possível reduzir a quantidade de atributos (nós da rede) de forma segura, sem redução na qualidade dos dados, em duas ordens de magnitude — inicialmente a rede de dados envolvia 10.000 atributos e, após a seleção, passou a conter um total de 100 atributos.

Os trabalhos de Moradi e Rostami (2015), Zhu et al. (2016) utilizam redes complexas compostas por atributos. Esses trabalhos, porém, ao realizarem a representação do conjunto de dados em uma rede, aplicam no conjunto de dados medidas baseadas em distância para encontrar as ligações entre os atributos e seus pesos. A partir de uma rede já criada, em (MORADI; ROSTAMI, 2015) os autores aplicam a medida de centralidade de Laplace entre os vértices de um mesmo agrupamento, em seguida é realizada uma ordenação pelo valor da centralidade e, por fim também, é utilizado um *threshold* para remover os vértices com valores acima do limite definido. Já em (ZHU et al., 2016), os autores aplicam medida de similaridade nos vértices, que, inicialmente, calcula a distância dos atributos aplicando medida de kernel na matriz de ajacência da rede e, em seguida, empregam um *threshold* a fim de remover todos os vértices considerados não relevantes, com valor acima do limite estabelecido.

Na tese de Huisman (2016), é utilizada uma rede complexa contendo interações entre 850 usuários da rede social *Facebook*. Esses dados foram coletados por mais de dois anos e são utilizados para predizer se haveria a ocorrência de um encontro presencial (*offline*) entre os usuários com base apenas em dados *online*. O autor aplicou uma *Random Forest* para essa tarefa e conseguiu obter uma taxa de 78% de acurária.

No trabalho de Ikehara e Clauset (2017), os autores analisaram 986 redes reais e 575 redes sintéticas de diversos domínios em busca de identificar relações estruturais entre os diferentes domínios de redes complexas. Para essa análise, *Random Forests* e matriz de confusão foram utilizadas. Os resultados mostraram que ao analisar apenas as estruturas não foi possível evidenciar diferenças entre diversas redes que eram de domínios distintos. Isso levou os autores a concluírem que origem de uma rede não é um fator decisivo na formação de redes com estruturas similares.

No trabalho de Roffo et al. (2017), é apresentado um método de seleção de atributos latente probabilístico baseado em grafo, que analisa todos os possíveis subconjuntos de atributos para gerar o ranking. Aqui, a relevância dos atributos é modelada como um atributo latente. Esse método é realizado em três etapas. Primeiramente, um processo de quantização é realizado para mapear os valores dos atributos em um conjunto menor contável, denominado *tokens*. Em seguida, é criado um grafo completo não-orientado onde cada vértice representa um atributo do *dataset*. O peso das arestas é automaticamente gerado por um *framework* de aprendizado baseado na variação da análise semântica latente probabilística. Por último, é feito o ranqueamento que considera todos os possíveis caminhos do grafo e verifica a redundância de cada atributo. Esse método foi testado em dez datasets reais, com quantidade de atributos entre 970, no menor, e 20.000, no maior.

Em (HASHEMI; DOWLATSHAHI; NEZAMABADI-POUR, 2020) foi utilizada a métrica de centralidade *PageRank* para seleção de atributo em *datasets* multi-rótulo, no qual cada entrada possui mais de uma classe. *PageRank* é um método comumente utilizado em redes para calcular a importância de páginas web (*sites*) na Internet. Neste trabalho, primeiramente é criado um grafo completo, em que cada vértice é adjacente a todos os demais, não-orientado e ponderado, no qual cada vértice do grafo representa um atributo do *dataset*. O peso das arestas é obtido pela distância Euclidiana da matriz de correlação entre os atributos $\times$ classes do *dataset*. Em seguida, é aplicada a métrica *PageRank* para gerar uma pontuação para cada vértice e, por fim, os vértices são ordenados em ordem decrescente de pontuação, gerando um ranking dos atributos; o usuário define a quantidade de atributos de seu interesse para a seleção a partir do início do ranking. No trabalho, o método foi testado em sete *datasets* reais, sendo três do tipo texto, três do tipo imagem e um do tipo biológico, com quantidade de atributos entre 130, no menor, e 1.836, no maior.

Em um trabalho mais recente de Roffo et al. (2021), também apresenta um seletor de atributos baseado em grafo, que utiliza abordagem de de ranqueamento. Neste trabalho, porém, não é feita a etapa de pré-processamento dos dados. Ainda, para a atribuição de peso das arestas duas interpretações foram exploradas (i) utilizando as propriedades das séries de potência das matrizes, e (ii) utilizando o conceito de absorção da cadeia de Markov. Nos dois casos, foi computado o vetor que expressa a probabilidade de haver um atributo particular em um subconjunto de qualquer tamanho, somando todos os tamanhos possíveis até o infinito. Os datasets testados foram similares ao do trabalho anterior.

No trabalho de Wang et al. (2022), foi proposta uma combinação de métricas de centralidade em redes para identificar proteínas essenciais. Primeiramente, foram aplicadas 14 métricas em uma rede de proteínas e seus respectivos valores foram tratados como novos atributos. Em seguida, foi feita uma seleção entre esses novos atrubitos utilizando a importância dos atributos gerada pela *Random Forest*. Por fim, foi feita a média geométrica dos valores de centralidade entre os atributos previamente selecionados, gerando o valor final de cada atributo.

Em (FAN et al., 2022), foi utilizada técnica de redução de dimensionalidade utilizando Análise Discriminante Linear (LDA), para lidar com atributos altamente similares, antes de realizar a seleção dos atributos. Após ser feita a redução de dimensionalidade, os atributos resultantes foram avaliados de acordo com suas correlações com a classe, quão maior essa correlação, maior importância foi atribuída aos atributos.

Há também trabalhos que utilizam *datasets* não rotulados, quando não se tem a classe de nenhum dos dados (MORADI; ROSTAMI, 2015; TANG et al., 2018), e *datasets* semi-rotulados, quando somente parte dos dados possuem informação sobre qual a classe que pertencem (LAI et al., 2022; SHEIKHPOUR et al., 2020), para efetuar a seleção de atributos utilizando grafos.

2.7 Considerações Finais

Nesta seção foi detalhada a fundamentação teórica de sobre temas abordados neste projeto de pesquisa. Nesta pesquisa foram gerados e analisados grafos ponderados e dígrafos ponderados. Em seguida foram aplicadas medidas de centralidade nos grafos gerados. No capítulo a seguir são apresentados os detalhes sobre o projeto de pesquisa.

3

Projeto de Pesquisa

3.1 Considerações Iniciais

Neste capítulo é apresentado o projeto de pesquisa que propõe um método para o ranqueamento de atributos que consiste na representação de *datasets* em redes complexas utilizando *Random Forests* — *ensemble* composto por um conjunto de árvores de decisão — para definir as ligações entre os atributos e o peso dessas ligações.

Dessa forma, ao gerar uma *Random Forest* a partir de um *dataset* e então transformá-la em uma rede complexa, seja orientada, quando as arestas apontam o sentido da ligação entre o par de vértices, ou não-orientada, quando não há sentido direcional, foi observado que as informações obtidas na caracterização topológica consideram tanto a informação local dos atributos (presentes nos exemplos utilizados para a criação da rede) quanto informações globais, proporcionadas exclusivamente pela rede complexa.

3.2 Abordagem Proposta

A proposta geral desta pesquisa encontra-se representada sob a forma dos Algoritmos 1, 2 e 3 em alto nível, desenvolvidos e aprimorados durante a pesquisa. Essa proposta inicialmente utiliza uma *Random Forest*, método de aprendizado de máquina baseado em coleção de árvores de decisão, como estratégia para identificar as conexões entre pares de atributos e os respectivos pesos dessas conexões para compor uma rede complexa. Em seguida, métricas de centralidade são aplicas nessa rede complexa a fim de ranquear os atributos de acordo com seus valores de centralidade. Uma forma visual da proposta geral é mostrada esquematicamente na Figura 1 na próxima página. Os detalhes são explicados no texto que segue a definição dos algoritmos.

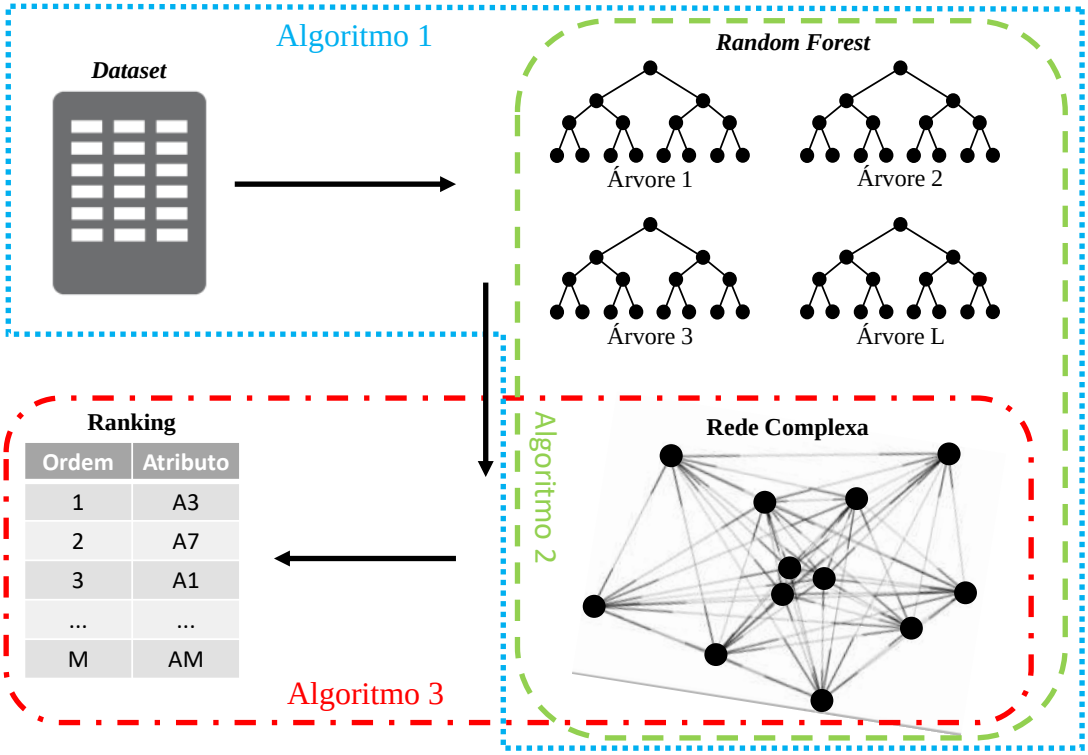


Figura 1 – Esquema geral do *ranker* proposto nesta pesquisa. Inicialmente, os dados encontram-se em um *dataset* e são utilizados para treinar uma *Random Forest* (Algoritmo 1). Os atributos utilizados e ligações geradas por cada árvore no classificador são então representados sob a forma de uma rede complexa (Algoritmo 2), na qual são aplicadas métricas de centralidade para obter o ranking dos atributos (Algoritmo 3).

3.2.1 Representação de Árvores como Redes Complexas

Utilizando a representação de redes complexas por meio da matriz de adjacência, conforme descrita na Seção 2.4.1 na página 25, os atributos (que aparecem tanto no *dataset* quanto na árvore de decisão) $\{X_1, \ldots, X_m\}$ são representados pelos vértices numerados $\{1, \ldots, m\}$, respectivamente, na rede complexa. A matriz de adjacência A é quadrada de ordem m, na qual o elemento A_{ij} representa a aresta entre os vértices i e j na rede complexa.

Havendo L grafos ponderados individuais, um para cada árvore da *Random Forest*, todos são representados por sua respectiva matriz de adjacência, acrescida de super-escrito entre parênteses, ou seja, denotada da forma $\{A_{ij}^{(1)}, A_{ij}^{(2)}, \ldots, A_{ij}^{(L)}\}$. Posteriormente, esses L grafos ponderados individuais são unidos por meio do Algoritmo 1 em um único grafo ponderado final, o qual nos referimos por rede complexa. Conforme descrito na Seção 2.4.2 na página 26, as métricas de centralidade são calculadas a partir da matriz de adjacência dessa rede complexa. Os detalhes tanto do Algoritmo 1 quanto da aplicação das métricas de centralidade, na abordagem proposta, são descritos nas próximas seções.

3.2.2 Algoritmo 1: *dataset* para rede complexa por meio de *Random Forest*

De acordo com o Algoritmo 1, inicialmente, é gerada uma *Random Forest* contendo L árvores considerando todos os m atributos do *dataset Instances* (Algoritmo 1, linhas 4–5). O número de $L = 64$ árvores é proveniente de resultado de pesquisa anterior (OSHIRO; PEREZ; BARANAUSKAS, 2012). Em seguida, cada árvore da floresta é representada por um grafo ponderado individual (Algoritmo 1, linhas 6–8). Essa representação é obtida pelo Algoritmo 2.

Algoritmo 1 *Dataset* para rede complexa por meio de *Random Forest*

Require: Instances: um *dataset* com n exemplos rotulados $\{(\mathbf{x}_i, y_i), i = 1, 2, \ldots, n\}$ contendo m atributos $\{X_1, X_2, \ldots, X_m\}$;
1: L: número de árvores na floresta, onde $L \geq 1$, *default* $L = 64$;
2: Directed: booleano indicando se a transformação de árvores é efetuada em grafos orientados (`true`) ou não-orientados (`false`), *default* `false`.
Ensure: N: rede complexa gerada a partir de Instances.
3: **function** convertDataToNet(Instances, L, Directed)
4: $T \leftarrow$ buildRandomForest(Instances, L)
5: Sejam $\{T_1, T_2, \ldots, T_L\}$ as árvores individuais contidas na floresta T
6: **for** $l \in \{1, 2, \ldots, L\}$ **do**
7: $A^{(l)} \leftarrow$ convertTreeToGraph(T_l, Directed)
8: **end for**
9: $\mathcal{N}_{ij} \leftarrow \sum_{l=1}^{L} A_{ij}^{(l)}$ para $1 \leq i, j \leq m$ ▷ $\mathcal{N}$ é a rede complexa resultante
10: **return** $\mathcal{N}$

3.2.3 Algoritmo 2: Árvore de Decisão para grafo

A função `convertTreeToGraph()`, Algoritmo 1, linha 7, é a responsável pela conversão de cada árvore da floresta em seu respectivo grafo, suas etapas são descritas pelo Algoritmo 2, detalhado a seguir.

Assuma que uma árvore de decisão possua E arestas de árvore numeradas de $e = 1, 2, \ldots, E$. Um ponto importante a ser ressaltado é que em uma árvore de decisão a ligação entre dois nós de decisão (aresta de árvore) que liga os atributos (X_i, X_j) pode ocorrer mais de uma vez em um mesmo nível da árvore ou mesmo em outros (sub)níveis. Assim, é interessante denotar a e-ésima aresta de árvore (X_i, X_j) com peso $w_{ij}^{(e)}$ da forma $(e, X_i, X_j, w_{ij}^{(e)})$ que pode ser simplificada para $(X_i, X_j, w_{ij}^{(e)})$ onde fica implícito o fato que a referência é à e-ésima aresta de árvore. O conjunto de arestas de árvore será denotado por $\mathcal{E} = \{(X_i, X_j, w_{ij}^{(1)}), (X_i, X_j, w_{ij}^{(2)}), \ldots, (X_p, X_q, w_{pq}^{(E)})\}$, onde é claro que $|\mathcal{E}| = E$.

Na transformação da árvore em grafo realizada pelo Algoritmo 2, inicialmente, a matriz de adjacência que representará a árvore convertida em grafo contém apenas zeros, ou seja, $A_{ij} \leftarrow 0$ para $1 \leq i, j \leq m$ (linha 5). A seguir, para cada aresta e na árvore ligando os atributos

$(X_i, X_j, w_{ij}^{(e)})$ será adicionado o peso $w_{ij}^{(e)}$ ao valor atual da aresta (i, j) no grafo representado pela matriz A, ou seja, $A_{ij} \leftarrow A_{ij} + w_{ij}^{(e)}$ (linha 8). Esse processo é repetido para todas as E arestas de árvore (Algoritmo 2, linhas 6-12). Para a atribuição dos pesos das arestas foram utilizadas as seguintes métricas de peso:

- un: atribuição de peso unitário (1) sempre que uma aresta existir nas árvores (métrica global em relação a floresta);
- oob: pontuação *out-of-bag* de uma árvore para todas as arestas ali encontradas (métrica global à cada árvore, mas local à floresta);
- gini: índice Gini (BREIMAN et al., 1984) do atributo pai para cada ligação da árvore entre os nós pai $\rightarrow$ filho (métrica local à aresta); e
- mg: média geométrica entre as métricas (oob) e (gini). Essa operação serve para indicar a tendência central entre as duas métricas utilizadas no cálculo, se o valor de uma delas for muito inferior que a outra para uma mesma aresta esta medida terá um valor resultante menor do que a média aritmética. A ideia aqui é que a medida seja valorizada (tenha uma pontuação mais alta) quando a aresta tiver um peso maior em ambas as métricas (oob) e (gini).

Na Figura 2 são mostradas as representações de duas árvores de uma determinada floresta em seus respectivos grafos ponderados individuais e, posteriormente, a união desses grafos gera duas representações de uma mesma rede (orientada e não-orientada). Essa união é realizada somando os pesos das arestas dos grafos individuais (Algorithm 1, line 9). As orientações de aresta são representadas por 'out', 'in' e 'g'. Pormenores, 'out' indica redes orientadas em que um vértice i considera as arestas que estão no sentido $i \rightarrow j$, 'in' é o oposto de 'out', indica redes orientadas em que um vértice i considera as arestas que estão no sentido $i \leftarrow j$ e 'g' indica redes não-orientadas em que um vértice i considera todas as arestas que lhe são incidentes, ao trabalharem com as métricas de centralidade. Os nós folha das árvores, rotulados com 'F', não são representados nos grafos. Os nós rotulados com números indicam atributos.

3.2.4 Algoritmo 3: ranqueamento dos atributos via métrica de centralidade

O ranqueamento dos atributos, objetivo desta pesquisa, é aplicado na rede complexa gerada pelo Algoritmos 1 e 2 utilizando as métricas de centralidade descritas na Seção 2, tais como *strength*, *eigenvector* e Katz que calculam a centralidade dos vértices (atributos) na rede complexa por meio da função $C(\cdot)$ no Algoritmo 3 (linhas 6–8). Os valores resultantes da função $C(\cdot)$ são ordenados (Algoritmo 3, linha 9) com o objetivo de identificar os atributos mais centrais que

Algoritmo 2 Árvore de Decisão para grafo

Require: Tree: uma árvore de decisão induzida a partir de um *dataset* rotulado contendo m atributos $\{X_1, X_2, \ldots, X_m\}$;
1: Weight: indica a métrica de peso a ser utilizada;
2: Directed: booleano indicando se a transformação de árvore é efetuada em grafo orientado (`true`) ou não-orientado (`false`), *default* `false`
Ensure: A: matriz de adjacência representando a árvore Tree transformada em rede complexa
3: **function** convertTreeToGraph(Tree, Directed)
4: Seja $\mathcal{E} = \{(X_i, X_j, w_{ij}^{(1)}), \ldots, (X_p, X_q, w_{pq}^{(E)})\}$ a lista de arestas de árvore em Tree, onde $|\mathcal{E}| = E$
5: $A_{ij} \leftarrow 0$ para $1 \leq i, j \leq m$
6: **for** $e \in \{1, 2, \ldots, E\}$ **do**
7: Seja $(X_i, X_j, w_{ij}^{(e)})$ a e-ésima aresta de árvore em $\mathcal{E}$
8: $A_{ij} \leftarrow A_{ij} + w_{ij}^{(e)}$
9: **if** not Directed **then**
10: $A_{ji} \leftarrow A_{ji} + w_{ji}^{(e)}$
11: **end if**
12: **end for**
13: **return** A

tendem a ter maior importância que os demais (BORBA; TREVIZAN, 2013). No Algoritmo 3, $C(\cdot)$ representa uma das métricas de centralidade, descritas na Seção 2.4.2, $C, C^{\text{in}}, C^{\text{out}}$, de acordo com a orientação das arestas da rede em questão.

Algoritmo 3 Ranqueamento dos atributos via métrica de centralidade

Require: $\mathcal{N}$: uma rede complexa contendo m vértices representada pela sua matriz de adjacência
1: $C(\cdot)$: métrica de centralidade, *default* $C(\cdot) = \texttt{EC}(\cdot)$. A métrica considera o grau dos vértices conforme valor dos parâmetros Directed e InDegree
2: Directed: booleano indicando se a rede complexa é orientada (`true`) ou não-orientada (`false`), *default* `false`
3: InDegree: booleano indicando se a métrica de centralidade deve ser calculada usando o grau de entrada (`true`) ou o grau de saída (`false`), *default* `true`. Somente aplicável se Directed = `true`.
Ensure: Ranked: lista de atributos da forma $< i, C(i) >$ ordenados pela métrica $C(\cdot)$
4: **function** featureRanker($\mathcal{N}$, $C(\cdot)$, Directed, InDegree)
5: centralityList $\leftarrow \emptyset$ ▷ conterá pares (atributo, valor da métrica de centralidade)
6: **for** $j \in \{1, 2, \ldots, m\}$ **do**
7: centralityList $\leftarrow$ centralityList $\cup \{ < j, C(j) > \}$
8: **end for**
9: Ranked $\leftarrow$ Ordene centralityList pela métrica $C(\cdot)$
10: **return** Ranked

3.3 Considerações Finais

Neste capítulo foram apresentado os detalhes do projeto de pesquisa no qual, em resumo, a partir de um *dataset*, uma *Random Forest* é induzida como estratégia para gerar uma rede complexa. Cada árvore da *Random Forest* é representada por um grafo. Quanto às arestas,

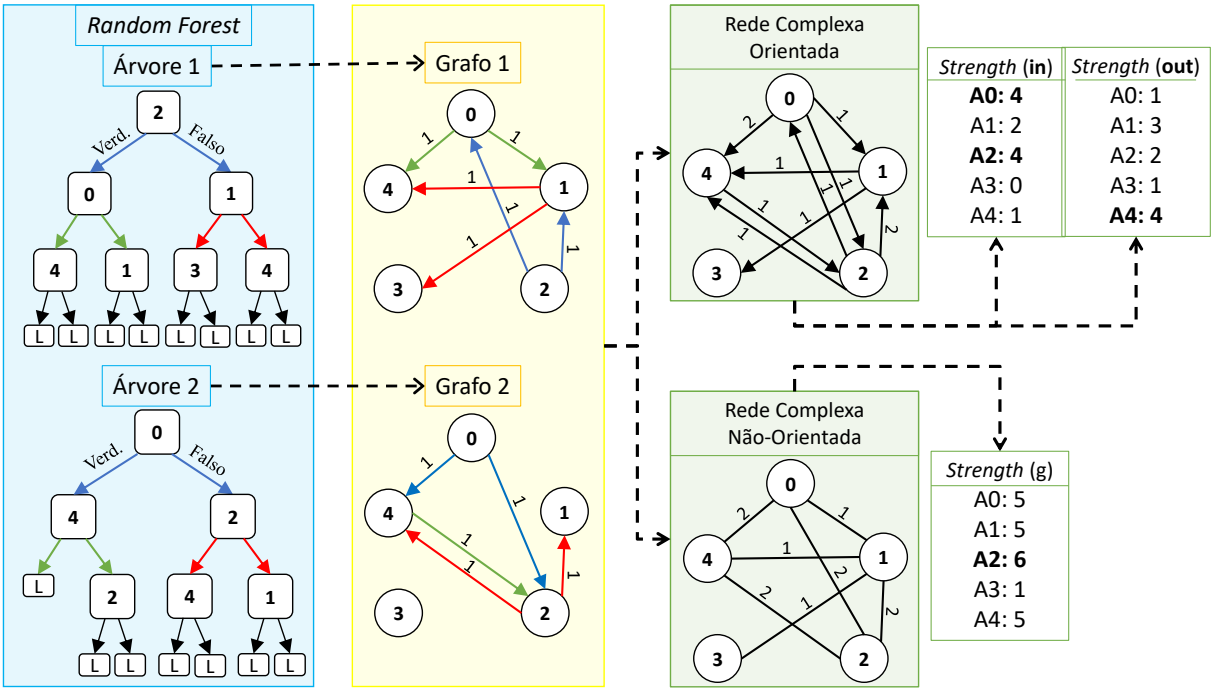


Figura 2 – Representação de duas árvores de uma floresta (à esquerda) em seus respectivos grafos ponderados individuais representando as ligações entre os atributos nas árvores (ao centro) e as redes complexas, orientada e não-orientada, geradas a partir da união entre os grafos (à direita). Nas árvores os nós folha, rotulados com 'F', não são representados no grafo. Os nós rotulados com números indicam atributos.

pesos são associados para representar a força de conexão entre os atributos adjacentes a cada aresta. A união dos grafos gerados resultam na rede complexa, cujos vértices são atributos. Por fim, métricas de centralidade são aplicadas à rede complexa a fim de identificar os atributos relevantes para o respectivo *dataset*. A abordagem proposta foi descrita, em alto-nível, por meio dos Algoritmos 1, 2 e 3. Para avaliar o desempenho desta abordagem proposta, foram realizados diversos experimentos cujos detalhes encontram-se no capítulo seguinte.

4

Configuração Experimental

4.1 Considerações Iniciais

Neste capítulo é descrita a configuração dos experimentos realizados para avaliar o método proposto neste trabalho. A seguinte notação é definida e empregada na apresentação e discussão de resultados e na conclusão, nos Capítulos 5 e 6:

A(ruído, orientação, centralidade, peso) na qual $A()$ representa os resultados da execução dos Algoritmos 1, 2 e 3 usando os seguintes parâmetros

- ruído $\in$ {5%,10%,20%,40%}: é a porcentagem de exemplos que se tornaram ruidosos naqueles *datasets*;
- orientação $\in$ {g, in, out}: 'g' indica redes não orientadas, 'in' indica redes orientadas em que um vértice i considera as arestas que estão no sentido $i \leftarrow j$ e 'out' é o oposto de 'in' – indica redes orientadas em que um vértice i considera as arestas que estão no sentido $i \rightarrow j$;
- centralidade $\in$ {eigen, katz, str}: indica qual a métrica de centralidade aplicada à rede, 'eigen', 'katz' e 'str' são respectivamente as abreviações de centralidade de autovetor (*eigenvector centrality*), índice de Katz (*Katz Index*) e força do vértice (*vertex strength*);
- peso $\in$ {un, oob, gini, mg}: indica qual a métrica de peso aplicada às arestas, ´un', 'oob',´gini',´mg' são respectivamente as abreviações de unitário, *out-of-bag*, índice Gini e média geométrica.

Para facilitar a análise dos resultados, foram feitos dois recortes utilizando os parâmetros taxa de ruído nos exemplos e métrica de peso de aresta. Dessa forma, para evitar repetição de texto dentro de uma mesma tabela ou figura em cada recorte, os valores de ruído e de peso foram substituídos por um ponto — A(·, orientação,centralidade,·). Esses pontos podem ser lidos como

os valores da taxa de ruído e do peso de aresta descrito na legenda de suas respectivas tabelas ou figuras.

O método importância dos atributos, utilizado como linha de base para comparação dos resultados, foi aplicado pela mesma *Random Forest* utilizada para gerar as redes cujas métricas de centralidades foram aplicadas.

4.2 Origem dos Dados

Há diversas fontes de dados públicos disponíveis para acesso que dispõem de dados obtidos a partir de diversos domínios do conhecimento humano. Existem também diversos métodos para a geração de dados artificiais, de maneira controlada, com quantidade diversificada de exemplos, classes e atributos.

A escolha dos *datasets*, bem como ter conhecimento sobre as propriedades dos dados disponíveis (saber exatamente quais atributos são os mais relevantes), são pontos fundamentais para a obtenção de análises significativas, com cenários e complexidades diversificadas. Levando em consideração esses fatores, nesta pesquisa foram utilizados como fonte de dados *datasets* artificiais, com ampla variedade de dimensões, domínios e dificuldades originados a partir das seguintes ferramentas:

- Scikit-learn (PEDREGOSA et al., 2011): biblioteca da linguagem *Python* que possibilita a geração de dados artificiais de acordo com a necessidade do usuário, por meio de diversos parâmetros como quantidade de exemplos, quantidade de classes, quantidade total de atributos e quantidade de atributos relevantes <https://scikit-learn.org/stable/modules/classes.html#samples-generator>;
- MLBench (LEISCH; DIMITRIADOU, 2010): pacote da linguagem R que contém *datasets* reais e artificiais. De nosso interesse, nesse pacote, são as funções geradoras de dados artificiais disponíveis <https://CRAN.R-project.org/package=mlbench>.
- KODAMA (CACCIATORE et al., 2016): pacote da linguagem R para descoberta de conhecimento e mineração de dados. De nosso interesse, nesse pacote, são as funções geradoras de dados artificiais disponíveis <https://CRAN.R-project.org/package=KODAMA>.

Foram gerados datasets rotulados, os quais temos os rótulos de todos os exemplos, tanto para problemas de classificação, cujos rótulos são discretos e separam os dados em grupos (classes), quanto para problemas de regressão, cujos rótulos são valores reais (contínuos). Com isso, a análise de resultados foi dividida em dois recortes, um somente para *datasets* de classificação e outro somente para os *datasets* de regressão.

As informações sobre a geração dos *datasets* encontram-se na seção seguinte, bem como os detalhes sobre a inserção de ruído nos *datasets* gerados estão situados na Seção 4.4.

4.3 Utilização de Datasets Artificiais

Datasets gerados de forma artificial permitem um melhor conhecimento sobre o domínio dos dados e também é possível garantir que não haja ruído previamente inserido nos dados. A inserção de ruído em *datasets* que já possuem algum tipo de ruído dificulta a análise de eficiência de um *ranker* (KHOSHGOFTAAR; HULSE, 2009).

Segundo Albuquerque, Lowe e Magnor (2011), um *dataset* pode ser gerado em três etapas:

- É definida a estrutura central do *dataset* ao coletar algumas informações previamente necessárias para sua criação, tais como quantidade de exemplos, quantidade de atributos, quantidade de classes, uma distribuição de probabilidade padrão e o tipo de dado dos atributos.
- Com base nos parâmetros anteriormente coletados, é atribuído um valor aleatório a cada um dos atributos de um exemplo, este valor é relativo a distribuição de probabilidade escolhida. Esta atribuição se repete para todos os exemplos do *dataset* a serem gerados. Com os dados já criados, é opcional a inserção de alguma estrutura (padrão de agrupamento) em um ou mais atributos, para simular *datasets* reais que também possam conter tais estruturas.
- Por último, pode ser inserido ruído aleatório de forma controlada no *dataset* gerado, a fim de simular quaisquer irregularidades que possam haver em dados reais.

Uma forma visual deste processo encontra-se na Figura 3.

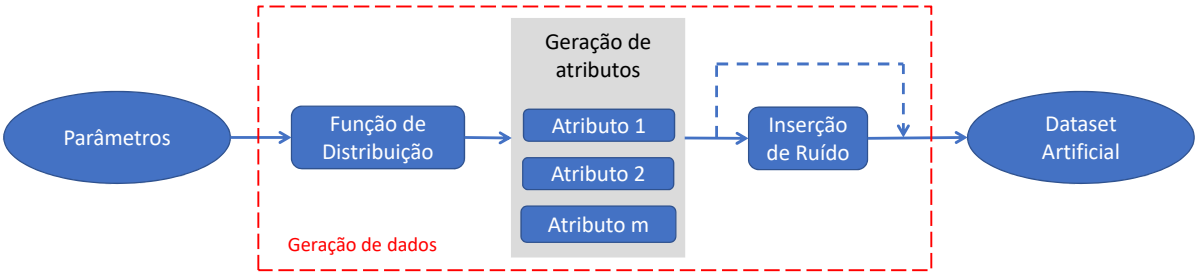


Figura 3 – Esquema geral da geração de um *dataset* artificial. Inicialmente são inseridos parâmetros como quantidade de exemplos, atributos, função de distribuição. Em seguida é gerada a função de probabilidade para a escolha dos valores de cada um dos m atributos e, por fim, pode ser inserido ruído nos dados. Após estas etapas o *dataset* artificial está pronto para uso.

Primeiramente, para analisar a robustez do *ranker* proposto, foram gerados *datasets* com uma ampla variedade de domínios e dificuldades. Em seguida, foi analisada a eficiência de cada métrica em ranquear, de forma correta, os atributos relevantes (sem ruído).

Foram utilizados 28 geradores de dados numéricos, mostrados na Tabela 2, a partir dos quais foram gerados 97 *datasets* distintos e sem ruído. Esses *datasets* foram gerados para representar problemas de diferentes complexidades. Do total de *datasets*, 85 são de classificação e 12 são de regressão, todos de aprendizado supervisionado. Essa quantidade de *datasets* foi gerada para representar problemas de diferentes dimensões e complexidades, tais como *datasets* com grande número total de atributos, poucos atributos relevantes em meio a muitos ruidosos, poucos exemplos, atributos altamente similares e classes sobrepostas. A quantidade de exemplos foi fixada em 300 e foi variada a quantidade de atributos relevantes ($\rho \in \{2, 3, 5, 7, 21\}$) e de classes ($k \in \{2, 3, 4, 5, 7, 8\}$).

Alguns exemplos visuais de *datasets* artificiais com dois e três atributos utilizados nesta pesquisa são mostrados nas Figuras 4 na página seguinte e 5 na página 44, respectivamente. Por último, após a geração, foi inserido ruído nos *datasets*.

Tabela 2 – Descrição dos datasets artificiais gerados.

Pacote	**Nome da Função**	**Tarefa**	**# Atributos Rel.**	**# Classes**	**Total**
mlbench	spirals	cla.	2	2	1
mlbench	cassini	cla.	2	3	1
mlbench	shapes	cla.	2	4	1
mlbench	smiley	cla.	2	4	1
mlbench	2dnormals	cla.	2	2,3,5,7	4
mlbench	cuboids	cla.	3	4	1
mlbench	waveform	cla.	21	3	1
mlbench	circle	cla.	2,3,5,7	2	4
mlbench	ringnorm	cla.	2,3,5,7	2	4
mlbench	threenorm	cla.	2,3,5,7	2	4
mlbench	twonorm	cla.	2,3,5,7	2	4
mlbench	xor	cla.	2,3	2,4	2
mlbench	hypercube	cla.	2,3	4,8	2
KODAMA	spirals	cla.	2	2,3,5,7	4
KODAMA	dinisurface	cla.	3	3	1
KODAMA	helicoid	cla.	3	3	1
KODAMA	swissroll	cla.	3	3	1
sklearn	make_circles	cla.	2	2	1
sklearn	make_moons	cla.	2	2	1
sklearn	make_classification	cla.	2,3,5,7	2,3,5,7	14
sklearn	make_gaussian_quantiles	cla.	2,3,5,7	2,3,5,7	16
sklearn	make_blobs	cla.	2,3,5,7	2,3,5,7	16
sklearn	make_sparse_uncorrelated	reg.	4	$\mathbb{R}$	1
sklearn	make_regression	reg.	2,3,5,7	$\mathbb{R}$	4
mlbench	friedman2	reg.	4	$\mathbb{R}$	1
mlbench	friedman3	reg.	4	$\mathbb{R}$	1
mlbench	friedman1	reg.	5	$\mathbb{R}$	1
mlbench	peak	reg.	2,3,5,7	$\mathbb{R}$	4
Total	28				97

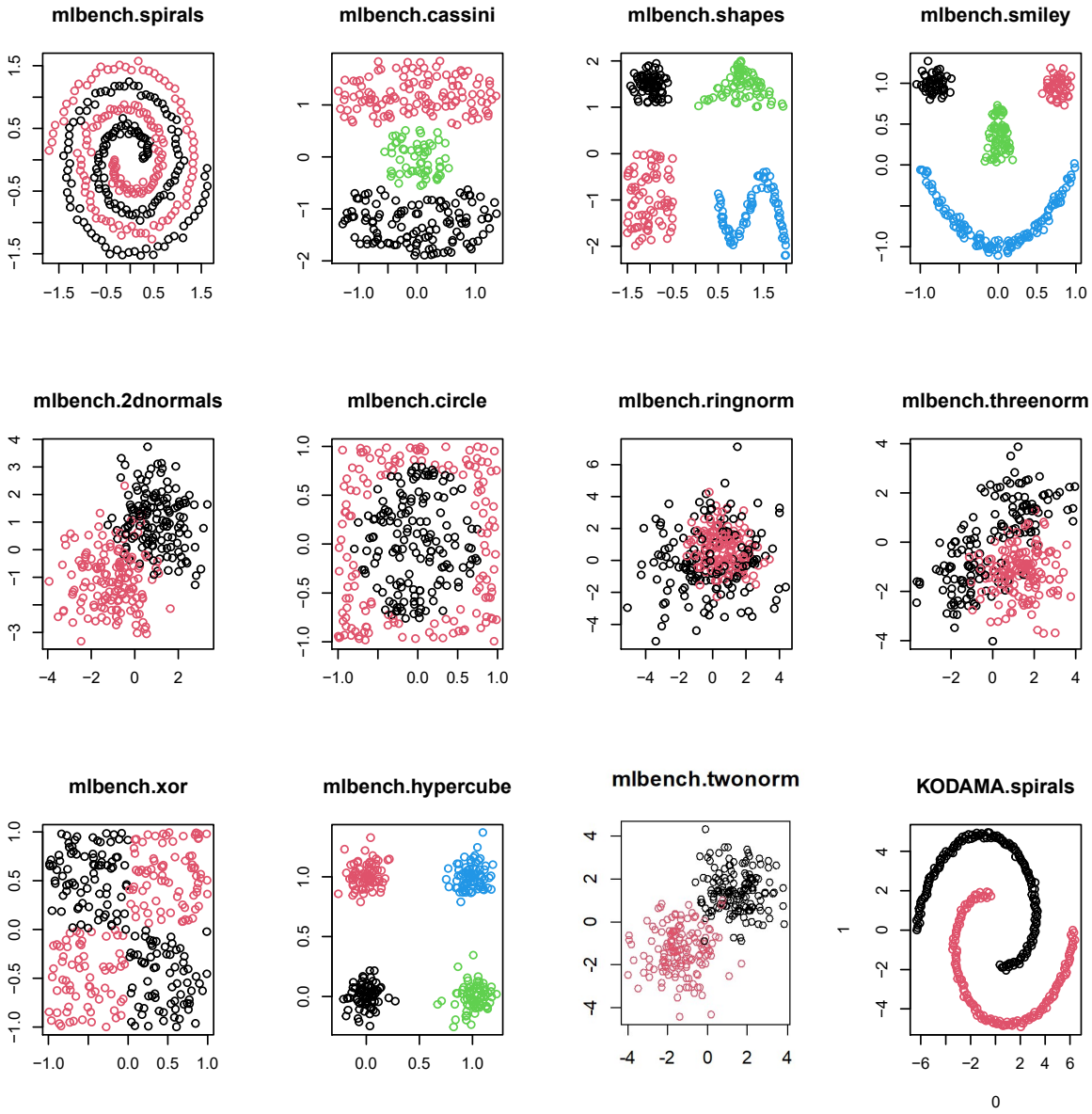


Figura 4 – Visualização de 16 funções geradoras de dados artificias para classificação, todos contendo 2 atributos relevantes e duas, três ou quatro classes.

4.4 Inserção de Ruído

Ao trabalhar com dados gerados de forma artificial é possível ter conhecimento sobre quais atributos são relevantes ($\mathcal{R}$). A inserção de ruído nos *datasets* de forma controlada torna possível a avaliação da capacidade de ranqueamento e da eficiência do *ranker* proposto nesta pesquisa (KHOSHGOFTAAR; HULSE, 2009).

Ruído pode ocorrerem em um *dataset* de duas formas: na classe ou nos atributos. Classe com ruído é quando um exemplo está rotulado com a classe incorreta. Atributo com ruído é quando um ou mais atributos de um exemplo contém valores incorretos. Há várias

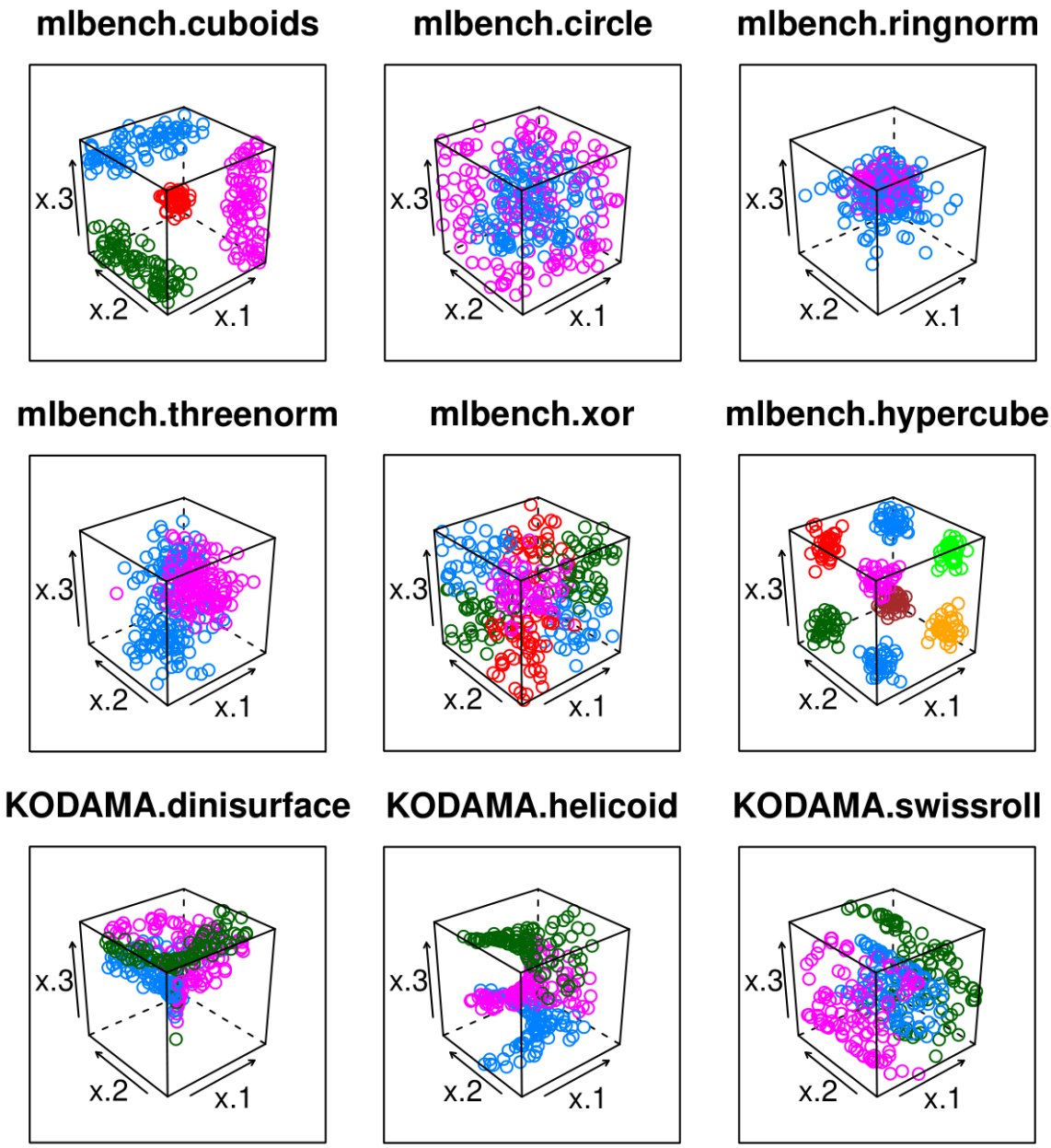


Figura 5 – Visualização de 9 funções geradoras de dados artificias para classificação, todas contendo 3 atributos relevantes e duas, três, quatro e oito classes.

técnicas disponíveis para a identificação de exemplos com ruído na classe, por outro lado, há poucos métodos para a detecção de exemplos com ruído nos atributos, isso ocorre por conta da complexidade de trabalhar com dados que possuem ruído nos atributos ser maior do que ruído nas classes (HULSE; KHOSHGOFTAAR; HUANG, 2007).

Ao inserir ruído em um *dataset*, qualquer seja a estratégia utilizada (ruído nos atributos ou na classe), é preciso utilizar uma taxa de ruído, também denominada taxa de corrompimento. Em diversos trabalhos sobre inserção de ruído as taxas utilizadas vão de 5% a 40% (GARCIA et al., 2019).

A inserção de ruído em atributos geralmente ocorre em três etapas que se repetem em ciclo até atingir a quantidade desejada de ruído no *dataset*:

1. É escolhido um atributo sem ruído do conjunto de atributos do *dataset*;
2. É selecionado uma amostra de exemplos ao acaso do conjunto de exemplos. A quantidade de exemplos da amostra deve ser previamente definida;
3. Os exemplos da amostra têm os valores do atributo escolhido modificados de acordo com alguma regra pré estabelecida.

O procedimento de alteração dos valores, ou seja, procedimento de inserção de ruído utilizado nesta pesquisa se baseia no modelo proposto por Khoshgoftaar e Hulse (2009) que seleciona uma amostra de atributos de *datasets* reais para inserir o ruído. A forma de escolher os atributos, porém, foi adaptada para o nosso contexto em que são utilizados *datasets* artificiais. O procedimento completo é representado sob a forma do Algoritmo 4.

Algoritmo 4 Algoritmo de inserção de ruído nos exemplos

Require:
T é um *dataset* com n exemplos rotulados $\{(\mathbf{X}_i, y_i), i = 1, 2, \ldots, n\}$ e m atributos $\{X_1, X_2, \ldots, X_m\}$;
$\hat{T}$ é o *dataset* com ruído, contendo os atributos originais (relevantes) e os atributos ruidosos (não relevantes)
X_m é o atributo a ser corrompido
X_i é o valor do atributo X_m para o exemplo i, que foi selecionado para ser corrompido
max(X_m) é o maior valor do atributo X_m
min(X_m) é o menor valor valor do atributo X_m
$\hat{X}_{mi}$ é o novo valor do atributo para o exemplo i, após o corrompimento
k = 20% * max(X_m)
tar é a taxa de atributos ruidosos, por exemplo, $\{2^0, 2^1, \ldots, 2^{10}\}$
ter é a taxa de exemplos ruidosos, por exemplo, $\{5, 10, 20, 40\}$
Ensure:
for $tar \in \{2^0, 2^1, \ldots, 2^{10}\}$ **do**
    $\hat{T}emp \leftarrow tar \times T\{X\}$
    for $ter \in \{5, 10, 20, 40\}$ **do**
        for $X \in \hat{T}emp\{X\}$ **do**
            se $X_{mi} <$ mediana(X_m) então $\hat{X}_m i \leftarrow$ max(X_m) + k
            se $X_{mi} >$ mediana(X_m) então $\hat{X}_m i \leftarrow$ min(X_m) - k
            se $X_{mi} =$ mediana(X_m) então $\hat{X}_m i \leftarrow$ aleatorio(min(X_m) - k, max(X_m) + k)
        end for
    end for
end for

A inserção de ruído proposta por Khoshgoftaar e Hulse (2009) é feita ao escolher uma parte dos atributos do *dataset*, depois escolher uma amostra dos exemplos destes atributos e, por fim, alterar os valores dos exemplos desta amostra. A quantidade de atributos totais não é alterada, com isso, os atributos que tiveram valores alterados passem a ser considerados ruidosos enquanto todos os demais são considerados relevantes. Na presente pesquisa, primeiramente são feitas cópias de todos os atributos do dataset original e, em seguida, é inserido ruído somente nos exemplos das cópias, mantendo-se assim todos os atributos originais intactos. Dessa forma é

possível testar também a capacidade do *ranker* de identificar os atributos relevantes em meio a atributos ruidosos mesmo quando 60%, 80%, 90% e 95% dos dados entre os atributos originais e os ruidosos são similares (quantidade de exemplos não alterados).

Para a quantidade de ruído a ser inserida, utilizaremos a mesma faixa de valores utilizada na pesquisa de Garcia et al. (2019), trabalhando com 4 taxas de ruído 5%, 10%, 20% e 40%. Quando é inserido ruído em 5% dos exemplos de um atributo cópia, este passa a ter uma similaridade de 95% com o atributo original, da mesma forma 10% de ruído equivale a 90% de similaridade, 20% de ruído a 80% de similaridade e 40% de ruído a 60% de similaridade.

Para cada atributo cópia, os exemplos a serem corrompidos são escolhidos ao acaso. Por exemplo, para um *dataset* que contém 2 atributos ($m = 2$), 300 exemplos ($n = 300$) e taxa de 5% de ruído, são escolhidos 15 exemplos para cada atributo cópia de forma aleatória. Os valores dos exemplos escolhidos são trocados para o extremo oposto da distribuição univariada, se o valor for abaixo da mediana de seu atributo, é alterado para ficar acima do valor máximo; se o valor for acima da mediana, é alterado para ficar abaixo do valor mínimo; se for igual a mediana, qualquer extremo é escolhido ao acaso. Essa troca modifica a distribuição dos atributos corrompidos (KHOSHGOFTAAR; HULSE, 2009).

Taxas de Ruído

Cada um dos 97 *datasets* foi gerado 10 vezes, permitindo assim que fosse testado *datasets* com pequenas variações entre si, causadas pela função geradora dos dados. Após a criação dos *datasets* foi inserido ruído sob a forma de atributos adicionais. Todos os atributos originais de cada *dataset* gerado são considerados relevantes e todos os atributos adicionais são cópias dos relevantes com ruído adicionado a uma parte de seus exemplos, como é mostrado na Figura 6.

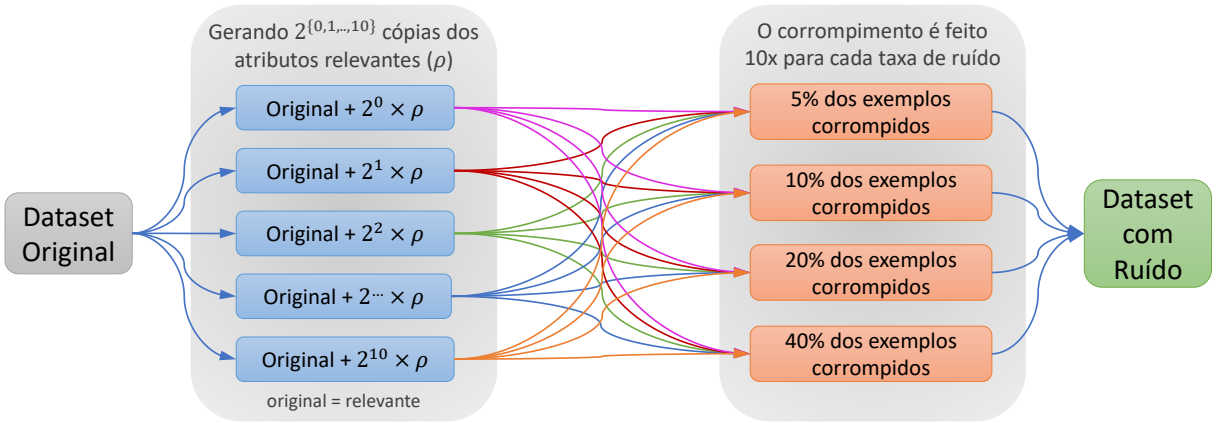


Figura 6 – Esquema utilizado para gerar *datasets* com diferentes quantidades de atributo-ruído. A partir deste esquema, para cada *dataset* sem ruído são gerados 440 *datasets* com ruído.

Foram utilizadas 4 taxas de ruído nos exemplos 5%, 10%, 20% e 40% e 11 taxas de atributos com ruído. A quantidade de atributos com ruído aumenta de forma exponencial, assim sendo, para cada *dataset* sem ruído com ρ atributos relevantes, $2^i \times \rho\,(i = 0, 1, \cdots, 10)$ novos atributos com ruído foram gerados e adicionados ao *dataset* sem ruído. O número total de atributos no *dataset* com ruído é definido por

$$m = \{\rho + 2^i \times \rho\} \tag{4.1}$$

Por exemplo, para um *dataset* sem ruído com dois atributos relevantes ($\rho = 2$), para cada uma das taxas de ruído nos exemplos, foram criados *datasets* com ruído contendo um total de $m = 4, 6, 10, 18, 34, 66, 130, 258, 514, 1026, 2050$ atributos.

Para cada *dataset*, este processo de inserção de ruído foi repetido 10 vezes devido à natureza estocástica do processo. Em outras palavras, o processo foi repetido para garantir que os resultados não fossem afetados caso houvesse algum viés, gerado ao acaso, durante a escolha aleatória dos exemplos a serem corrompidos.

Dessa forma, cada *dataset* original resultou em 4400 *datasets* com ruído. Evidenciando, foram gerados 97 datasets originais distintos, 10 vezes cada, utilizadas 11 taxas de atributos com ruído e 4 taxas de ruído nos exemplos desses atributos – também realizada 10 vezes. Ao fim, foi gerado um total de 426.800 *datasets* com ruído ($97 \times 10 \times 11 \times 4 \times 10$). A quantidade total de atributos nos *datasets* varia de $m = 4$, nos menores, a $m = 21.525$, nos maiores. Para os resultados, foi feita a média das execuções.

4.5 Métrica de Avaliação

Cada um dos *datasets* foi representado por uma rede atributo-atributo, na qual as arestas que ligam pares de atributos são obrigatoriamente ponderadas. Quatro métricas de pesos foram utilizadas para a atribuição de peso nas arestas, são elas (i) peso unitário, (ii) pontuação *out-of-bag*, (iii) índice Gini e (iv) média geométrica entre os pesos (ii) e (iii). A partir da rede criada, foram aplicadas três métricas de centralidade.

Cada uma das métricas de centralidade foi aplicada com a finalidade de ranquear os atributos de acordo com sua importância para a rede. As métricas utilizadas foram (i) *strength*, (ii) *eigenvector* e (iii) Katz. Além das métricas de centralidade foi testada também a medida *feature importance*, fornecida pela *Random Forest*, representando um ranking dos atributos de acordo com sua importância para o classificador gerado pela floresta. Portando, o número total de execuções foi $426.800 \times 4 \times 3 \times 3 = 15.364.800$.

Após a execução do *ranker* proposto foi avaliada a eficiência das métricas de centralidade em identificar os atributos relevantes em meio aos atributos ruidosos. Para essa avaliação foi elaborada uma métrica de desempenho denominada *ranking score* $\mathcal{R}$ ($0 \leq \mathcal{R} \leq 1$) (Eq. 4.5).

Essa métrica calcula o *score* com base na posição em que os atributos relevantes são encontrados no ranking gerado pelas métricas de centralidade e pela RF, para cada respectivo *dataset*. O $\mathcal{R}$ é composto por três termos, $\mathcal{S}$, $\mathcal{B}$ e $\mathcal{W}$, detalhados a seguir:

- O termo $\mathcal{S}$, Equação 4.2, representa a soma das posições dos ρ atributos relevantes no ranking (r_i é a posição do atributo relevante i no ranking gerado pelo Algoritmo 3) . Para um dataset com dois atributos relevantes ($\rho = 2$), se estes atributos estiverem nas posições 3 e 7 o valor de $\mathcal{S} = 3 + 7 = 10$.

$$\mathcal{S} = \sum_{i=1}^{\rho} r_i \tag{4.2}$$

- O melhor cenário possível, denotado $\mathcal{B}$ e dado pela Equação 4.3, é quando todos os atributos relevantes encontram-se nas primeiras ρ posições do ranking. Ou seja, para um dataset contendo dois atributos relevantes ($\rho = 2$), independente da quantidade total de atributos, o melhor cenário é obtido quando os dois atributos relevantes estão nas duas primeiras posições, dessa forma $\mathcal{B} = 1 + 2 = 3$.

$$\mathcal{B} = \sum_{i=1}^{\rho} i = \frac{\rho + \rho^2}{2} \tag{4.3}$$

- Pelo contrário, o pior cenário possível, denotado $\mathcal{W}$ e dado pela Equação 4.4, é quando todos os atributos relevantes econtram-se nas últimas ρ posições do ranking. Melhor dizendo, para um dataset com dois atributos relevantes ($\rho = 2$) e um total de 10 atributos ($m = 10$), o pior caso é quando os dois atributos relevantes estão nas duas últimas posições, dessa forma $\mathcal{W} = 9 + 10 = 19$.

$$\mathcal{W} = \sum_{i=(m-\rho+1)}^{m} i = \frac{((m-\rho+1)+m)[m-((m-\rho+1)+1)]}{2} = \frac{(2m-\rho+1)\rho}{2} \tag{4.4}$$

- Por fim, a métrica $\mathcal{R}$, representada pela Equação 4.5, analisa a posição dos atributos relevantes no ranking para gerar o *score*. Quanto mais ao topo do ranking os atributos relevantes forem encontrados, maior será o *score* até atingir o $\mathcal{B}$. Quanto mais ao fim do ranking os atributos relevantes forem encontrados, menor será o *score* até atingir o $\mathcal{W}$. Por exemplo, para um *dataset* que tenha $\rho = 2$ e um total de 10 atributos, o *score* é 1 se os dois relevantes estiverem nas primeiras duas posições, 0 se os dois relevantes estiverem nas duas últimas posições ou $0 < score < 1$ caso estejam nas demais posições. Ou seja, o score gerado por $\mathcal{R}$ varia entre 0, no pior caso, e 1, no melhor caso.

$$\mathcal{R} = 1 - \left(\frac{\mathcal{S} - \mathcal{B}}{\mathcal{W} - \mathcal{B}}\right) \tag{4.5}$$

4.6 Validação

Nesta pesquisa, ao realizar a geração de cada *dataset* original 10 vezes e repetir a inserção de ruído também 10 vezes, foi possível realizar 100 execuções para cada configuração, garantindo a mesma quantidade de ruído nos exemplos em cada uma dessas execuções. Para analisar os resultados, foi feita a média das execuções.

Para as as múltiplas comparações entre os métodos analisados nesta pesquisa, foi utilizado o teste de Friedman (1940). Esse é um teste estatístico não paramétrico que assume como hipótese nula que a diferença encontrada nos dados é ocorrida ao acaso, considerando um nível de confiança de 95%. A hipótese nula assume que todos os métodos possuem desempenho equivalente. No caso de rejeição da hipótese nula, foi aplicado o teste post-hoc de Bonferroni-Dunn (DUNN, 1961), também com nível de confiança de 95%, para detectar quaisquer diferenças significativas entre os métodos utilizados.

4.7 Considerações Finais

Neste capítulo foram apresentados detalhes sobre a origem dos dados, geração de datasets artificiais, inserção de ruído, forma de validação, bem como as métricas de peso de aresta e a métrica de desempenho utilizada para calcular a eficiência do *ranker*. No capítulo seguinte são reportados os resultados dos experimentos realizados para avaliação do *ranker* proposto no Capítulo 3.

5

Resultados e Discussão

5.1 Considerações Iniciais

Neste capítulo são reportados e discutidos os resultados de experimentos realizados com base na configuração experimental descrita no capítulo anterior sobre a proposta apresentada no Capítulo 3 na página 33. Nas seções seguintes são apresentados os resultados que permitem responder às questões presentes na Seção 1.2 na página 18, apresentados separadamente para problemas de classificação e de regressão.

5.2 Datasets de Classificação

Na Figura 9 na página 55 é mostrado o comportamento das métricas de centralidade para cada uma das 11 taxas de ruído nos atributos (eixo x). O *score* médio e o desvio padrão compreendem todos os *datasets* de classificação. Nessa figura os resultados são separados em quatro subgrupos, cada um representa a execução das métricas de centralidade em redes geradas com uma das seguintes métricas de peso de aresta (a) unitário, (b) *out-of-bag*, (c) Gini e (d) média geométrica. Dentro de cada subgrupo as linhas representam as taxas de ruído nos exemplos de 5%, 10%, 20% e 40% e as colunas representam as orientações de aresta consideradas pelas métricas de centralidade para calcular a importância do vértice.

De forma a generalizar um pouco mais as informações que são mostradas na Figura 9 na página 55 apresentamos o diagrama de diferença crítica, Figura 7 na página 53. Este diagrama analisa os *scores* das métricas de centralidade criando um ranking entre elas para cada uma das 11 taxas de ruído nos atributos e, como resultado, exibe o ranking médio e quais métricas não possuem diferença estatística significativa entre si. Por último, a Tabela 3 na página 52 agrupa e estrutura os rankings médios mostrados na Figura 7 na página 53, facilitando o entendimento do desempenho de cada uma das métricas de centralidade em cada uma das taxas de ruído nos exemplos, bem como em cada uma das métricas de peso de aresta.

Analisando os resultados mostrados na Tabela 3 na página seguinte, nota-se que os três melhores resultados foram obtidos em redes complexas não-orientadas (g), utilizando *eigen*, Katz e *str*, respectivamente. O método A(·,g,eigen,·) não apresentou diferença estatística significativa para os melhores posicionados (RF(·) e A(·,out,str,·)) para todas as situações.

Na Figura 7 na página 53, os diagramas de diferença crítica mostram o ranking médio dos *scores* dos métodos RF(·) e A() em todos os tamanhos de *datasets*. Cada linha representa uma taxa de ruído nos exemplos e cada coluna uma métrica de peso de aresta. Em quatro, dos dezesseis cenários (25%) o método RF(·) não ficou na primeira posição do ranking — cenários destacados com borda. A sub-figura destacada com borda na com azul é mostrada de forma isolada (e ampliada) na Figura 8 na página 54, cujo diagrama apresenta o melhor resultado obtido para *datasets* de classificação, que foi obtido com 40% de ruído nos exemplos e utilizando a métrica de peso de aresta unitário (detalhado no final desta seção).

Observando as informações mostradas na Figura 9 na página 55, em todas as quatro linhas de diagramas, ou seja, para 5%, 10%, 20% e 40% de ruído nos exemplos e considerando todas as orientações de aresta, o *score* médio de A() começa relativamente alto e vai diminuindo para valores maiores de i, ou seja, conforme aumenta o número total de atributos dos *datasets*.

Quanto aos resultados do método RF(·), é possível notar que seu desempenho médio tende a acompanhar o padrão de comportamento do método A(). Em alguns casos o *score* médio de RF(·) foi ligeiramente superior e, em outros, igualou-se ao A().

Em relação à orientação, redes não-orientadas, em geral, apresentaram os melhores resultados. Sobre a métrica de peso de aresta, para os três melhores resultados, não houve diferença estatisticamente significativa nas taxas de ruído de 5% e 10%. Já para 20%, as métricas de peso unitário e Gini se saíram melhor. Para 40% as melhores foram peso unitário, média geométrica e *out-of-bag*. Considerando todas as taxas de ruído, a melhor métrica de peso nas arestas para classificação foi unitário.

Tabela 3 – Agrupamento dos rankings do Diagrama de Diferença Crítica para as métricas de peso unitário (UN), média geométrica (MG) e pontuação *out-of-bag* (OOB), para cada taxa de ruído nos exemplos (5%, 10%, 20% e 40%), em *datasets* de classificação. Os valores destacados, dentro de uma mesma coluna, representam os métodos melhores ranqueados que não possuem diferença significativa entre si, que correspondem à primeira linha (mais à esquerda) dos diagramas de diferença crítica, mostrados na Figura 7.

Método	5% dos exemplos com ruído				10% dos exemplos com ruído				20% dos exemplos com ruído				40% dos exemplos com ruído			
	UN	MG	GI	OOB	UN	MG	GI	OOB	UN	MG	GI	OOB	UN	MG	GI	OOB
A(·,g,eigen,·)	2.91	2.91	2.91	2.91	3.36	3.27	3.27	3.27	3.18	3.27	3.18	3.27	3.45	4.27	4.27	4.18
A(·,in,eigen,·)	6.73	6.73	6.73	6.73	6.27	6.00	5.91	6.00	4.91	4.82	4.45	5.09	3.73	3.82	3.55	3.91
A(·,out,eigen,·)	6.36	6.36	6.36	6.36	6.55	7.00	7.45	6.55	7.45	8.36	8.55	8.18	8.82	8.82	8.91	8.73
A(·,g,katz,·)	3.55	3.55	3.55	3.55	3.55	3.73	3.73	4.09	4.36	4.36	4.45	4.36	5.45	5.82	5.91	5.73
A(·,in,katz,·)	8.09	8.09	8.09	8.09	7.36	7.18	7.09	7.45	6.18	6.36	6.27	6.45	5.45	5.36	5.27	5.64
A(·,out,katz,·)	7.18	7.18	7.18	7.18	7.45	8.36	8.36	8.18	9.55	9.55	9.64	9.45	9.73	9.82	9.91	9.82
A(·,g,str,·)	4.27	4.27	4.27	4.27	4.55	4.45	4.45	4.36	4.73	4.82	5.00	4.64	4.09	4.18	4.55	3.82
A(·,in,str,·)	7.82	7.82	7.82	7.82	8.00	7.55	7.27	7.64	8.09	7.73	7.55	7.55	8.00	8.00	7.82	8.00
A(·,out,str,·)	7.09	7.09	7.09	7.09	6.91	6.45	6.45	6.45	4.73	4.73	4.18	5.00	**2.64**	**2.45**	**2.27**	**2.55**
RF(·)	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.82**	**1.00**	**1.73**	**1.00**	3.64	**2.45**	2.55	2.64

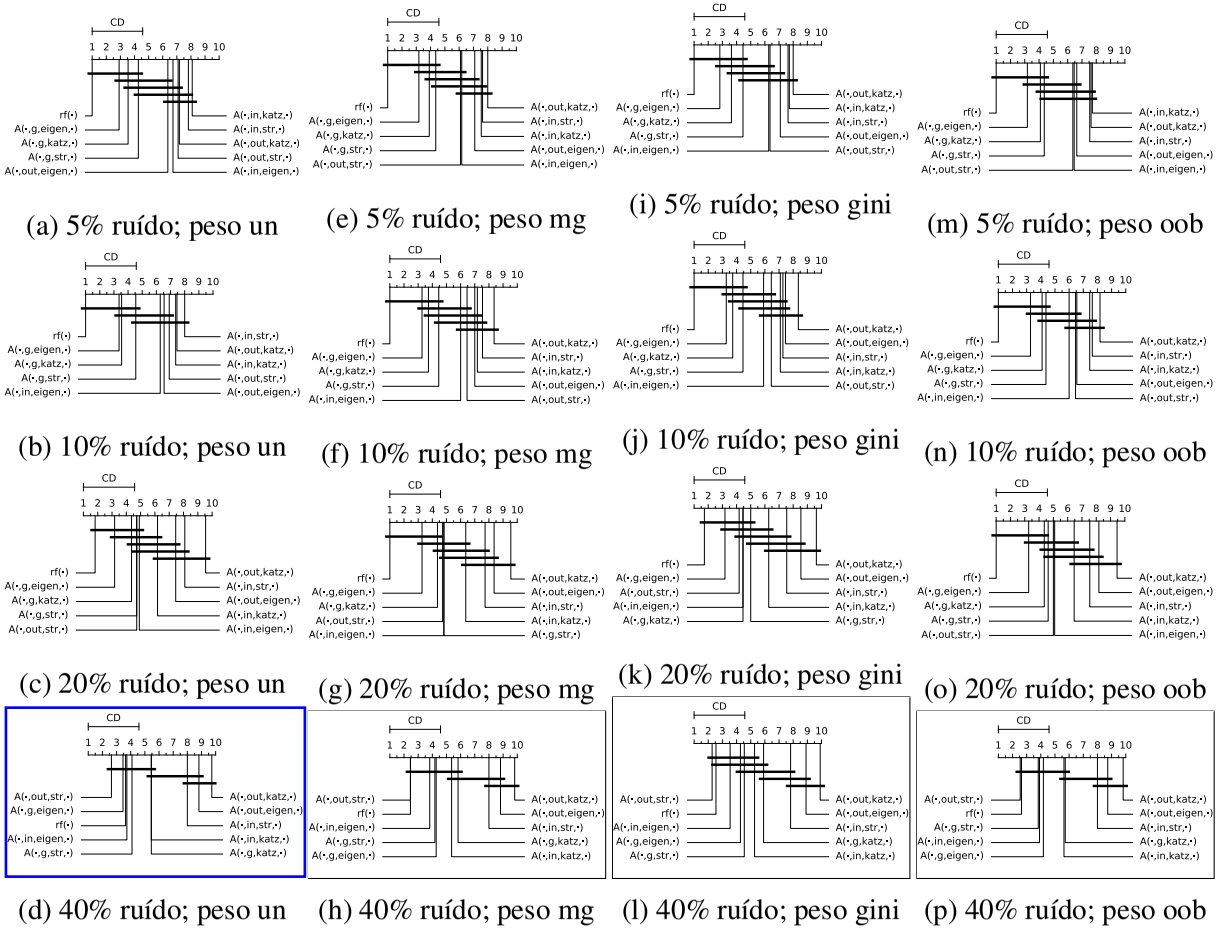


Figura 7 – Diagrama de diferença crítica utilizando as métricas de peso de aresta unitário (a-d), média geométrica (e-h), Gini (i-l) e *out-of-bag* (m-p) em *datasets* de classificação com teste post-hoc Bonferroni-Dunn. Em destaque estão os cenários em que RF(·) não ficou na primeira posição do ranking médio

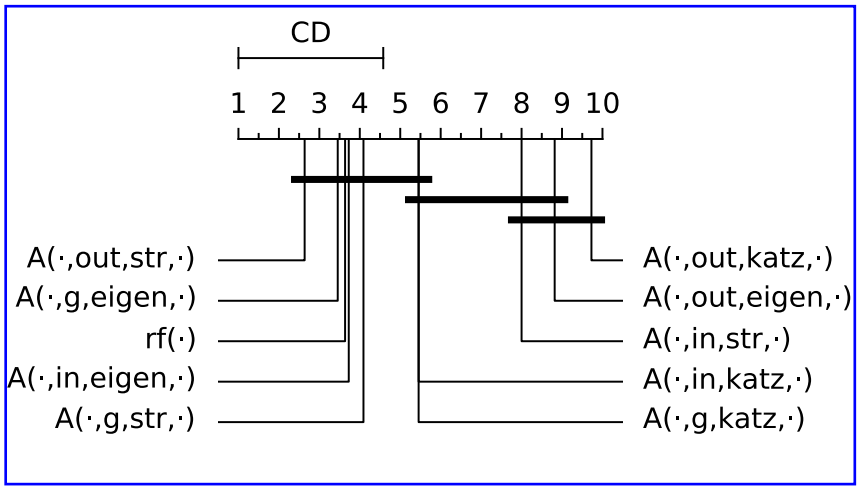
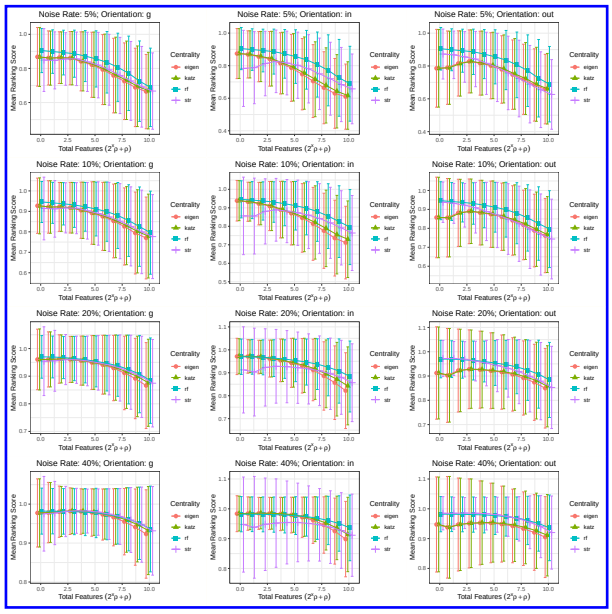
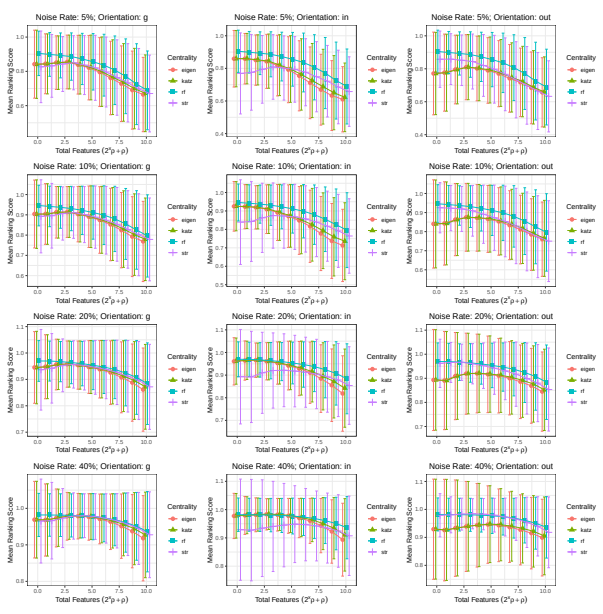


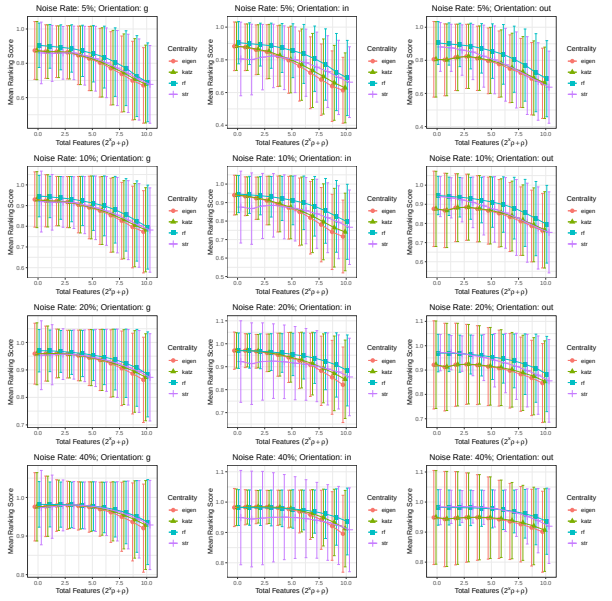
Figura 8 – Diagrama de diferença crítica do melhor resultado obtido para *datasets* de classificação, 40% ruído nos exemplos e métrica de peso de aresta unitário (un)



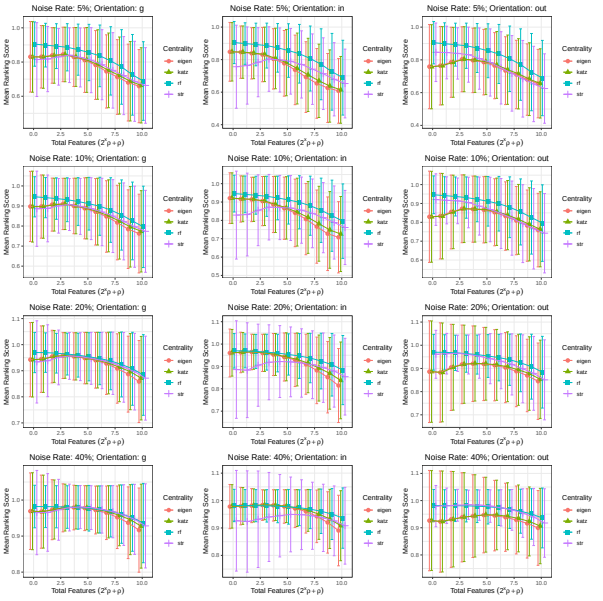
(a) Unitário



(b) Média geométrica



(c) Gini



(d) *out-of-bag*

Figura 9 – *Score* médio de cada métrica de centralidade executadas em redes com métrica de peso de aresta unitário, média geométrica, Gini e *out-of-bag* em *datasets* de classificação. Cada coluna representa uma orientação de aresta da rede e cada linha representa uma taxa de ruído nos exemplos. Observa-se que, para melhor comparação, os limites do eixo vertical (y) são diferentes entre os diagramas.

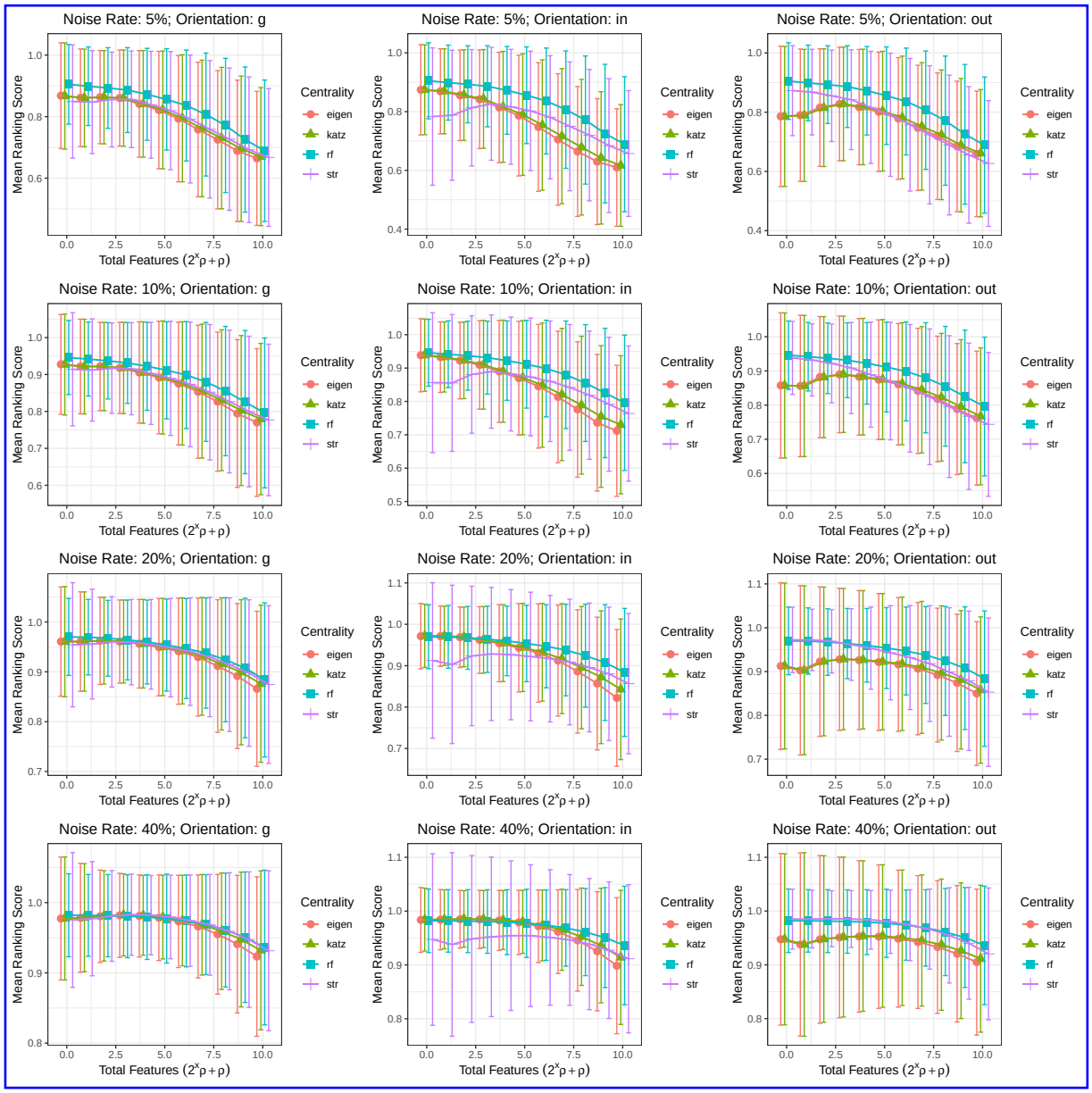


Figura 10 – *Score* médio da cada métrica de centralidade executadas em redes com métrica de peso de aresta unitário em *datasets* de classificação. Cada coluna representa uma orientação de aresta da rede e cada linha representa uma taxa de ruído nos exemplos. Observa-se que, para melhor comparação, os limites do eixo vertical (y) são diferentes entre os diagramas.

5.3 Datasets de Regressão

Analisando os resultados mostrados na Tabela 4 na página 58, observa-se que o método A() apresentou desempenho sem diferença estatística significativa (células com fundo verde) para as taxas de ruído de 5%, 10% e 20% em várias das configurações. Além disso, nota-se que o melhor resultado foi obtido em redes complexas não-orientadas (g), utilizando a métrica de centralidade *eigen*. Em específico, o método A(·,g,eigen,·) nunca apresentou diferença estatística

significativa em relação ao método RF(·) em *datasets* com 5%, 10% e 20% de ruído nos exemplos e, adicionalmente, foi melhor significativamente do que RF(·) em *datasets* com 40% de ruído nos exemplos.

Na Figura 11 na página 59, os diagramas de diferença crítica mostram o ranking médio dos *scores* dos métodos RF(·) e A() em todos os tamanhos de *datasets*. Cada linha representa uma taxa de ruído nos exemplos e cada coluna uma métrica de peso de aresta. Em doze, dos dezesseis cenários (75%) o método RF(·) não ficou na primeira posição do ranking — cenários destacados com borda. A sub-figura destacada com borda na com azul é mostrada de forma isolada (e ampliada) na Figura 12 na página 60, cujo diagrama apresenta o melhor resultado obtido para *datasets* de regressão, que foi obtido com 40% de ruído nos exemplos e utilizando a métrica de peso de aresta oob (detalhado no final desta seção).

Observando as informações mostradas na Figura 13 na página 61, nas duas primeiras linhas dos diagramas, ou seja, para 5% e 10% de ruído nos exemplos, considerando todas as métricas e orientações, nota-se que o *score* é relativamente alto mais à esquerda do gráfico, para $i = \{0, 1\}$, diminui para $i = \{2, \ldots, 6\}$ e volta a aumentar para $i \geq 7$. Para as duas últimas linhas, ou seja, para 20% e 40% de ruído nos exemplos, isso ocorre de forma mais sutil, nas quais ocorre uma queda menos acentuada do que nas duas primeiras linhas.

Considerando todas as orientações e para as métricas de peso de aresta un, oob, e mg, para 5% e 10% de ruído nos exemplos, RF(·) teve desempenho similar ao A() para $i = \{0, 1\}$, se apresentou melhor entre $i = \{2, \ldots, 7\}$ e voltou a igualar-se ao A() a partir de $i \geq 8$. Para 20% o desempenho médio do RF(·) praticamente igualou-se ao A(). Já para 40% de ruído, o desempenho do RF(·) ficou abaixo do A() para $i \leq 7$, tendo desempenho similar para $i > 7$. Ressalta-se ainda que para 40% de ruído, para o peso de aresta un, o desempenho médio de A() foi superior ao RF(·) nas três orientações e em todos os tamanhos de dataset ($i = 0, \ldots, 10$). Em relação ao peso de aresta gini, para 5% e 10% de ruído o método RF(·) teve desempenho similar para $i = \{0, 1, 2\}$ e $i = \{7, 8, 9, 10\}$ e ficou acima do A() para $i = \{3, \ldots, 6\}$. Já para 20% e 40%, RF(·) teve desempenho similar ao A().

Em relação à orientação, redes não-orientadas, em geral, apresentaram os melhores resultados. Em relação à métrica de peso de aresta, para os três melhores resultados, não houve diferença estatisticamente significativa nas taxas de ruído de 5%, 10% e 20%. Já para 40%, a métrica de peso *out-of-bag* se saiu melhor. Considerando todas as taxas de ruído, a melhor métrica de peso nas arestas para regressão foi *out-of-bag*, destacado com borda na Figura 13 na página 61 e mostrada de forma isolada (e ampliada) na Figura 14 na página 62.

Tabela 4 – Agrupamento dos rankings do Diagrama de Diferença Crítica para as métricas de peso unitário (UN), média geométrica (MG) e pontuação *out-of-bag* (OOB), para cada taxa de ruído nos exemplos (5%, 10%, 20% e 40%), em *datasets* de regressão. Os valores destacados, dento de uma mesma coluna, representam os métodos melhores ranqueados que não possuem diferença significativa entre si, que correspondem à primeira linha (mais à esquerda) dos diagramas de diferença crítica, mostrados na Figura 11

Método	5% dos exemplos com ruído				10% dos exemplos com ruído				20% dos exemplos com ruído				40% dos exemplos com ruído			
	UN	MG	GI	OOB	UN	MG	GI	OOB	UN	MG	GI	OOB	UN	MG	GI	OOB
A(·,g,eigen,·)	5.00	5.27	4.82	4.73	5.55	5.09	4.73	5.27	4.82	4.82	4.18	4.73	3.68	4.09	4.09	3.91
A(·,in,eigen,·)	9.00	8.73	8.45	8.64	8.45	8.36	8.36	8.36	6.36	5.82	5.09	6.55	**2.09**	3.36	2.82	**2.64**
A(·,out,eigen,·)	4.64	3.82	4.00	4.27	3.91	4.09	4.27	4.09	6.27	6.82	6.91	6.27	6.55	6.91	7.18	6.95
A(·,g,katz,·)	3.91	3.82	3.55	3.82	4.18	3.18	3.09	4.18	**2.91**	**2.55**	**2.73**	3.09	6.00	6.82	6.91	5.45
A(·,in,katz,·)	8.45	7.91	7.73	8.36	7.82	7.55	7.55	7.73	4.64	4.64	4.27	5.27	5.18	6.91	6.64	5.73
A(·,out,katz,·)	3.18	**2.27**	**2.73**	**2.91**	2.91	**2.82**	**2.91**	3.00	4.55	3.82	4.64	4.45	8.64	9.55	9.82	8.59
A(·,g,str,·)	5.18	5.64	5.09	5.27	6.09	6.00	5.55	5.73	6.55	6.91	6.36	6.55	3.95	2.36	2.18	3.91
A(·,in,str,·)	5.55	5.09	5.82	5.45	5.18	5.36	5.27	5.27	8.18	7.82	7.27	8.00	6.27	5.82	5.73	6.64
A(·,out,str,·)	8.18	9.09	8.82	8.45	8.82	9.09	9.09	9.00	7.73	7.73	7.73	7.45	2.64	**1.73**	**1.09**	3.64
RF(·)	**1.91**	3.36	4.00	3.09	**2.09**	3.45	4.18	**2.36**	3.00	4.09	5.82	**2.64**	10.00	7.45	8.55	7.55

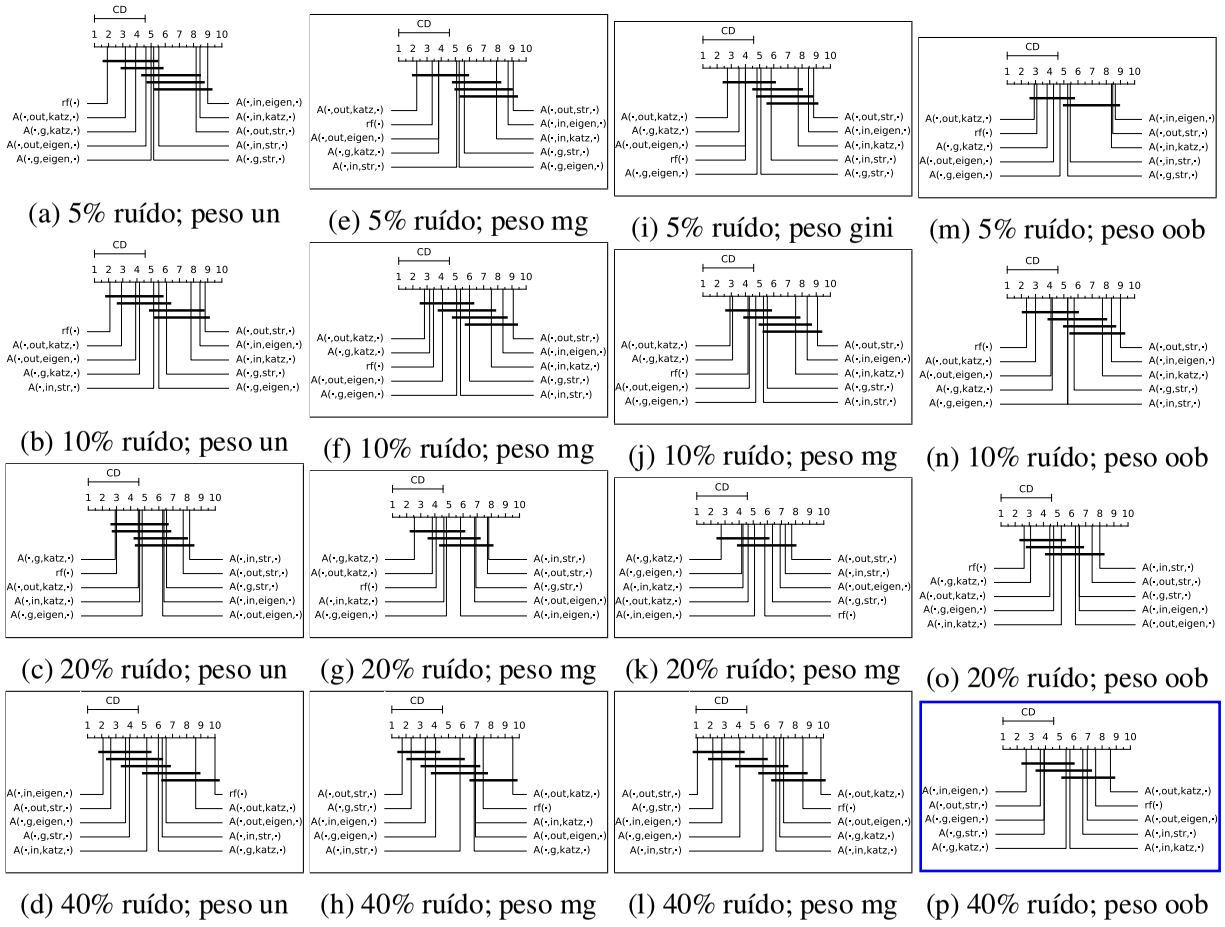


Figura 11 – Diagrama de diferença crítica utilizando as métricas de peso de aresta unitário (a-d), média geométrica (e-h), Gini (i-l) e *out-of-bag* (m-p) em *datasets* de regressão com teste post-hoc Bonferroni-Dunn. Em destaque estão os cenários em que RF(·) não ficou na primeira posição do ranking médio.

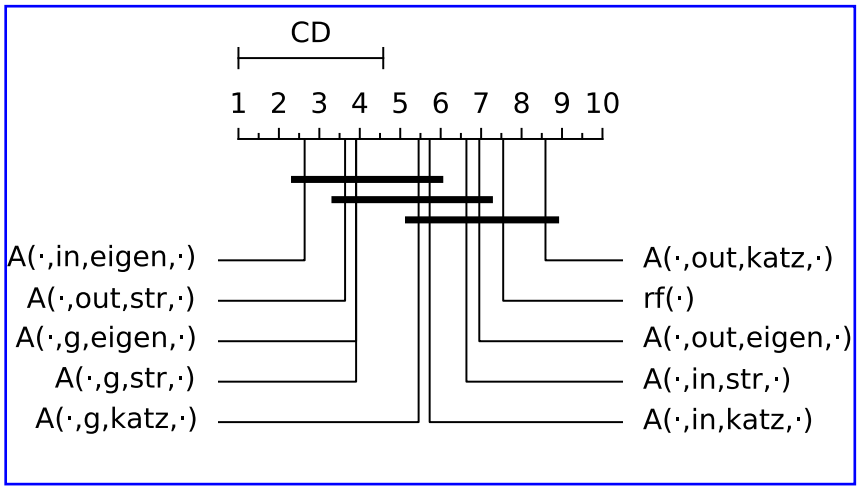
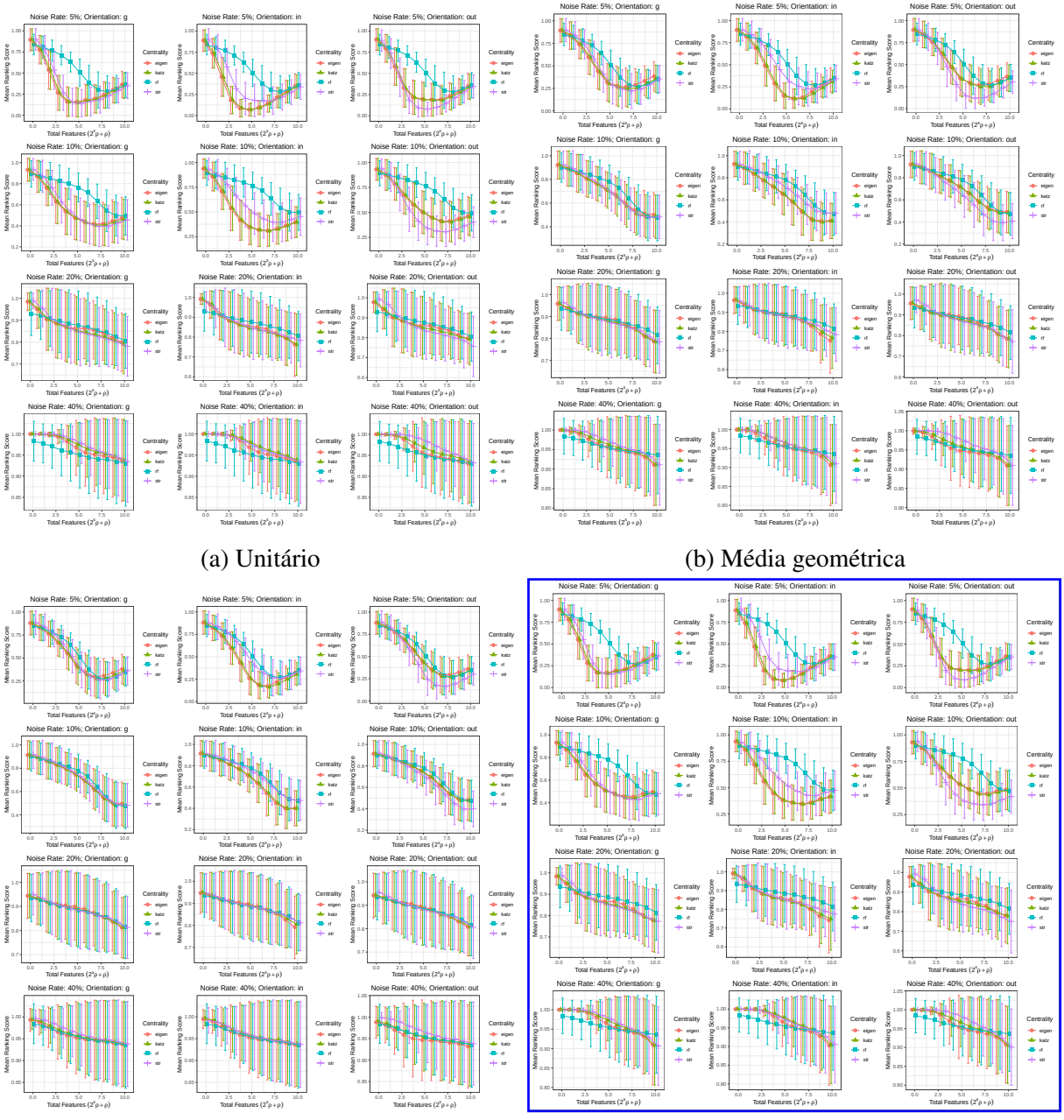


Figura 12 – Diagrama de diferença crítica do melhor resultado obtido para *datasets* de regressão, métrica de peso de aresta *out-of-bag* (oob) e 40% ruído nos exemplos



(a) Unitário

(b) Média geométrica

(c) Gini

(d) *out-of-bag*

Figura 13 – *Score* médio de cada métrica de centralidade executadas em redes com métrica de peso de aresta *unitário*, *média geométrica*, *Gini* e *out-of-bag* em *datasets* de regressão. Cada coluna representa uma orientação de aresta da rede e cada linha representa uma taxa de ruído nos exemplos. Observa-se que, para melhor comparação, os limites do eixo vertical (y) são diferentes entre os diagramas.

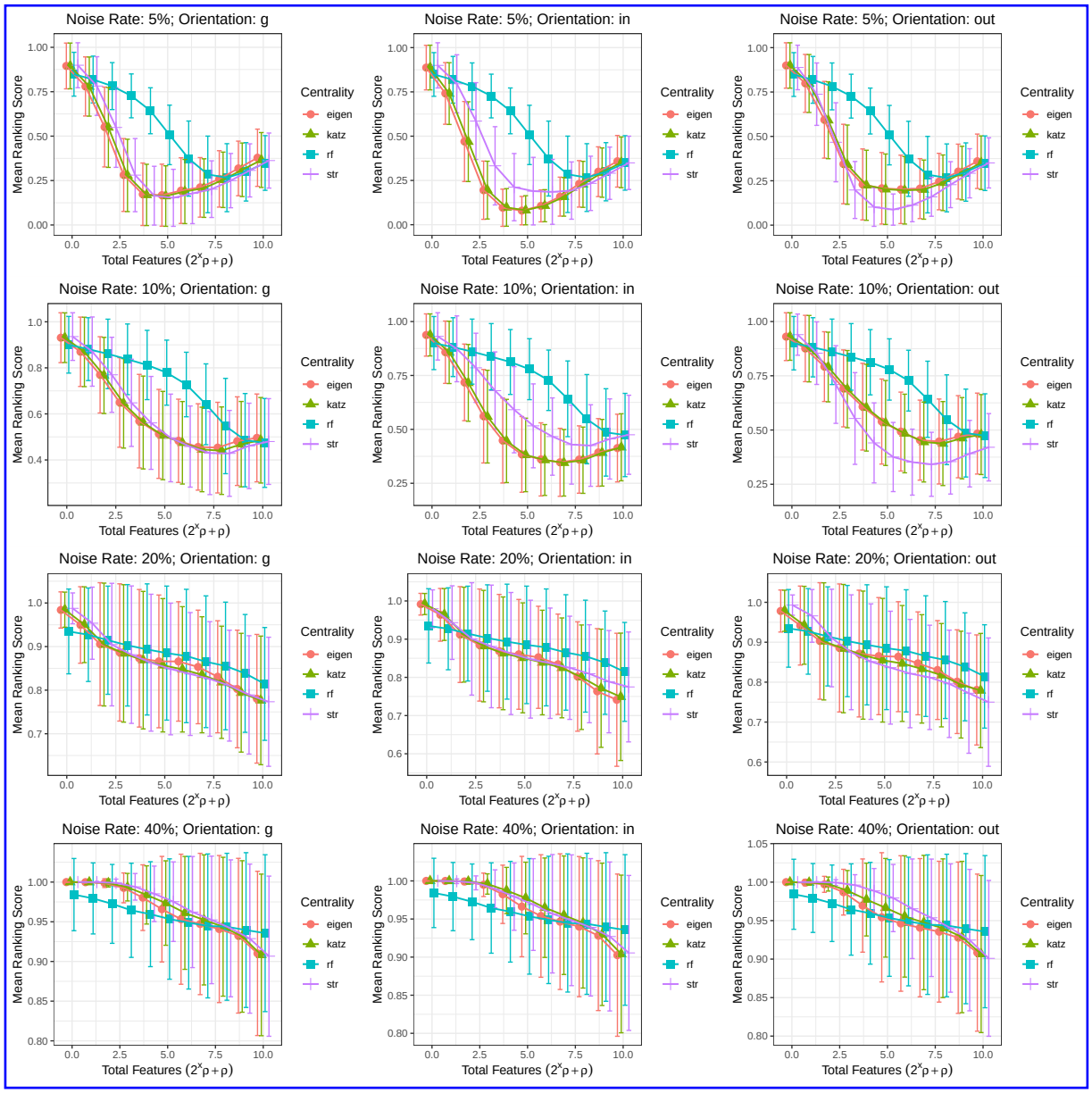


Figura 14 – *Score* médio da cada métrica de centralidade executadas em redes com métrica de peso de aresta *out-of-bag* (oob) em *datasets* de regressão. Cada coluna representa uma orientação de aresta da rede e cada linha representa uma taxa de ruído nos exemplos. Observa-se que, para melhor comparação, os limites do eixo vertical (y) são diferentes entre os diagramas.

5.4 Considerações Finais

Neste capítulo foram apresentados os resultados dos experimentos para avaliação do *ranker* proposto. Um resumo dos principais resultados é fornecido a seguir.

Para *datasets* de classificação, as métricas de centralidade apresentaram melhor desempenho em redes não-orientadas. Quanto à métrica de peso de aresta, para datasets com 5% e 10% de ruído nos exemplos não houve diferença. Para 20%, as métricas de peso unitário e Gini

obtiveram melhores resultados. Já para 40% as melhores métricas foram peso unitário, média geométrica e *out-of-bag*. Considerando todas as taxas de ruído nos exemplos, o melhor peso foi o unitário. O método A(·,g,eigen,·), teve desempenho equivalente aos melhores posicionados, para todas as taxas de ruído (RF(5%), RF(10%), RF(20%) e A(40%,out,str,·)), isto é, não apresentou diferença estatística significativa para todas as situações.

Para *datasets* de regressão, as métricas de centralidade apresentaram melhor desempenho, em geral, em redes não-orientadas. Sobre as métricas de peso de aresta, não houve diferença estatística significativa para as taxas de ruído de 5%, 10% e 20%. A métrica de peso *out-of-bag* obteve melhor desempenho para 40% de ruído. Dessa forma, considerando todas as taxas de ruído, a melhor métrica de peso foi *out-of-bag*.

A partir dos resultados, tanto em classificação como em regressão, é possível concluir que os métodos A(·,g,eigen,·) e RF(·) possuem desempenho equivalentes, ou seja, não houve diferença estatística significativa entre ambos em todas, exceto uma única situação (em 40% de ruído nos exemplos para datasets de regressão).

Notoriamente, o método A(40%,g,eigen,·) foi melhor significativamente do que o RF(40%) em problemas de regressão e A(20%,g,eigen,·), A(10%,g,eigen,·) e A(5%,g,eigen,·) tiveram desempenho equivalentes ao RF(20%), RF(10%) e RF(5%), respectivamente, pois não houve diferença estatística significativa entre eles.

Tais resultados indicam que a utilização de redes complexas não-orientadas são as mais indicadas quando geradas a partir de uma *Random Forest* para o ranqueamento de atributos. Também há uma indicação que o método A(·,g,eigen,·) é mais apropriado, visto que a *Random Forest* teve seu desempenho deteriorado com o aumento na taxa de ruído, sendo pior significativamente nesta situação (40% de ruído).

Adicionalmente, ambas as estratégias são bastante diferentes quanto ao uso de dados adicionais. A *Random Forest* calcula a importância dos atributos em cada árvore usando dados não vistos durante o treinamento (amostra *out-of-bag*). Nossa abordagem procura somente pelas arestas das árvores da *Random Forest*, não sendo necessário dados *out-of-bag*. Ainda assim, os resultados obtidos mostram uma possível correlação entre essas duas estratégias. Se essa correlação existir, de fato, nossos resultados mostram que a importância dos atributos das *Random Forests* é equivalente a ranquear redes complexas por métricas de centralidade; além disso, nossos resultados são suficientemente robustos para mostrar que é possível identificar a importância de cada atributos sem qualquer dado adicional ao treinamento (amostra *out-of-bag*).

6

Conclusão

6.1 Considerações Iniciais

Neste trabalho foi proposto um método de ranqueamento de atributos utilizando métricas de centralidade aplicadas à redes complexas geradas por meio de *Random Forests*.

No Capítulo 1 foram apresentados a introdução e o objetivo da pesquisa, seguidos da metodologia utilizada. No Capítulo 2 foram apresentados detalhes teóricos sobre Aprendizado de máquina, ranking e seleção de atributos, algoritmos de árvores de decisão, além de redes complexas e métricas de centralidade.

No Capítulo 3 encontra-se a proposta de desenvolvimento desta pesquisa, bem como sua representação sob a forma dos Algoritmos 1, 2 e 3, descritos em alto nível.

- Com base em trabalhos na área de Aprendizado de Máquina (MITCHELL, 1997) envolvendo Árvores de Decisão (QUINLAN, 1993), bem como Redes Complexas (ALBERT; BARABÁSI, 2002) e Métricas de Centralidade foi efetuada a proposta dos Algoritmos 1, 2 e 3 que realizam a transformação de *datasets*, tipicamente encontrados em Aprendizado de Máquina, em forma de uma rede complexa;
- Utilizou-se, neste trabalho, o *dataset* sob a perspectiva de uma matriz bidimensional em que as linhas são compostas por exemplos (casos) e as colunas por atributos que descrevem esses exemplos; na representação desse *dataset* em uma rede complexa foram utilizadas as colunas dessa matriz, ou seja, os atributos;
- O Algoritmo 3 é capaz de realizar a ordenação de atributos por sua relevância utilizando métricas de centralidade em redes complexas. Ressalta-se que os Algoritmos 1, 2 e 3 compõem a metodologia de pesquisa.

No Capítulo 4 encontram-se informações sobre a origem, a geração e a inserção de ruídos nos datasets artificiais, a forma de validação, a avaliação experimental dos resultados.

No Capítulo 5 encontram-se os resultados dos experimentos realizados obtidos tanto para *datasets* de classificação quanto para *datasets* de regressão.

- A proposta foi analisada de forma empírica em diversos *datasets* artificiais, com dimensões e complexidades variadas. Para avaliação, o método RF(·), que fornece o ranking dos atributos gerado pela *Random Forest*, foi comparado com as métricas de centralidade força do vértice (*vertex strength*), autovetor (*eigenvector centrality*) e índice de Katz (*Katz Index*). Cada uma das três métricas de centralidade foi testada em três situações, (i) redes não-orientadas, em que todas as arestas adjacentes aos vértices são analisadas, (ii) redes orientadas de entrada, em que somente as arestas convergentes são analisadas e (iii) redes orietadas de saída, em que somente as arestas divergentes são analisadas;
- *Datasets* com 40% de ruído nos exemplos possuem menor similaridade entre os atributos, uma vez que os atributos com ruído são cópia dos atributos relevantes com com ruído adicionado a seus exemplos. Para esse caso, o método A(40%,out,str,·) superou, mas não significativamente, a RF(40%) em *datasets* de classificação. Já para *datasets* de regressão, os métodos A(40%,out,str,·), A(40%,in,str,·), A(40%,g,str,·) e A(40%,g,eigen,·) superaram a RF(40%) com diferença estatística significativa, para 95% de confiança;
- Para a maior taxa de ruído, em *datasets* de regressão, o método A(40%,g,eigen,·) apresentou desempenho significativamente superior ao RF(40%), para todas as métricas de peso de aresta. Posto isso, A(·,g,eigen,·) foi o único método que apresentou desempenho equivalente ao RF(·), sem diferença estatística significativa, para todas as demais situações.

Ademais, a *Random Forest* faz uso de dados adicionais, não vistos durante o treinamento (dados *out-of-bag*) para calcular a importância dos atributos. Nossa abordagem, por outro lado, ao analisar as árvores da *Random Forest* utiliza somente as arestas e seus atributos adjacentes. Por fim, nossos resultados são suficientemente robustos para mostrar que é possível identificar a importância dos atributos sem utilizar dados *out-of-bag*.

6.2 Resumo das Principais Contribuições

As principais contribuições deste trabalho são:

- Publicação de um artigo (CANTÃO et al., 2022) intitulado *Feature Ranking from Random Forest Through Complex Network's Centrality Measures* no congresso internacional *26th European Conference on Advances in Databases and Information Systems* (ADBIS 2022) e publicado em *Lecture Notes in Computer Science* (LNCC); DOI: 10.1007/978-3-031-15740-0_24.

- Proposta e aplicação de melhorias no algoritmo de inserção de ruído de Khoshgoftaar e Hulse (2009), para que fosse inserido ruído em cópias dos atributos relevantes, assim mantendo o mesmo número de atributos relevantes do dataset original e permitindo obter a quantidade necessária de atributos ruidosos (Seção 4.4).
- Elaboração da abordagem inovadora, que realiza a representação de um *datasets*, tipicamente encontrados em Aprendizado de Máquina, sob a forma de uma rede complexa cujos vértices são atributos (Algoritmos 1 e 2) e que utiliza o conceito de métricas de centralidade como ferramenta para ranquear esses atributos (Algoritmo 3) (Seção 3.2).
- Com os resultados realizados, é possível recomendar a utilização da métrica de centralidade *eigen* em redes complexas não-orientadas (Seção 5.4).
  - Para redes geradas a partir de *Random Forests*, é possível indicar a métrica de peso de aresta unitário para *datasets* de classificação e peso *out-of-bag* para *datasets* de regressão.

Adicionalmente, o autor realizou outras contribuições ao longo do desenvolvimento desta pesquisa, como a colaboração com a pesquisa sobre indução de árvores de decisão por meio de meta-aprendizado, publicada em congresso internacional (FERREIRA; CANTÃO; BARANAUSKAS, 2022) e colaboração com a pesquisa sobre encadeamento de classificadores de múltiplos níveis (*multi-level stacking*), publicada em congresso nacional (BOLDRIN et al., 2022). Além disso, colaborou com o projeto Covid-19 Brasil que realizou a coleta de dados de Covid-19 dos 645 municípios do estado de São Paulo para gerar inferências estatísticas sobre a situação regional e nacional (BERNARDI et al., 2021). Esse projeto teve um resumo publicado na conferência *University Social Responsibility Network Summit* 2021 e um artigo aceito pela Revista de Saúde Digital e Tecnologias Educacionais (RESDITE) (CARVALHO I et al., 2022). Ademais, cofundou o projeto voluntário PICOVID: Portal de Informações sobre a Covid-19 para auxiliar tanto os gestores quanto a população a melhor compreender os números relacionados à Covid-19 intramunicipal (CANTÃO; FAZIO, 2020).

6.3 Trabalhos Futuros

Os resultados obtidos nesta pesquisa de mestrado utilizando *datasets* artificiais são promissores, uma vez que os métodos A(·,g,eigen,·) se apresentou tão bom quanto a *Random Forest* para *datasets* de classificação, ou seja, não houve diferença estatística significativa, e se apresentou estatisticamente superior em uma situação, para *datasets* de regressão.

Sendo assim, trabalhos futuros poderão ser desenvolvidos utilizando *datasets* reais, aplicando a métrica de centralidade *eigenvector* uma rede complexa não orientada gerada a apartir de uma *Random Forest*.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior Brasil (CAPES) - Código de Financiamento 001.

Referências

ALBERT, R.; BARABÁSI, A.-L. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, American Physical Society, v. 74, p. 47–97, Jan 2002.

ALBUQUERQUE, G.; LOWE, T.; MAGNOR, M. Synthetic generation of high-dimensional datasets. *IEEE Transactions on Visualization and Computer Graphics*, v. 17, n. 12, p. 2317–2324, Dec 2011.

BARRAT, A. et al. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 101, n. 11, p. 3747–3752, 2004.

BERNARDI, F. et al. *From social sensibility to interdisciplinary collaboration: the Brazilian COVID-19 epidemiology reference portal [abstract]*. Virtualy hosted from South Africa: University Social Responsibility Network, 2021. Abstract nr 2.

BERTINI, J. R.; ZHAO, L.; LOPES, A. A. An incremental learning algorithm based on the k-associated graph for non-stationary data classification. *Information Sciences*, v. 246, n. Supplement C, p. 52 – 68, 2013. ISSN 0020-0255.

BILLIO, M.; PELIZZON, L.; SAVONA, R. *Systemic Risk Tomography*. 2016. 300 p. Elsevier.

BOLDRIN, F. C. et al. Multi-level stacking. In: BRAZILIAN INSTITUTE OF DATA SCIENCE. *Encontro Nacional de Inteligência Artificial e Computacional*. Campinas, Brasil, 2022. p. 12.

BOLÓN-CANEDO, V.; ALONSO-BETANZOS, A. Ensembles for feature selection: A review and future trends. *Information Fusion*, v. 52, p. 1–12, 2019. ISSN 1566-2535.

BONACICH, P. Power and centrality: A family of measures. *American Journal of Sociology*, v. 92, n. 5, p. 1170–1182, March 1987.

BORBA, E. M.; TREVIZAN, V. *Medidas de Centralidade em Grafos e Aplicações em redes de dados*. Dissertação (Mestrado) — IME-UFRGS, 2013.

BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, n. 2, p. 123–140, 1996.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, L. *Wald Lecture II, Looking Inside the Black Box*. 2004.

BREIMAN, L.; CUTLER, A. *Random Forests: Classification/Clustering*. 2004.

BREIMAN, L. et al. *Classification and Regression Trees*. 1984. Wadsworth & Books.

CACCIATORE, S. et al. KODAMA: an R package for knowledge discovery and data mining. *Bioinformatics*, v. 33, n. 4, p. 621–623, 11 2016. ISSN 1367-4803.

Referências

CANTÃO, A. H.; FAZIO, R. B. de. *PICOVID: Portal de Informações sobre a Covid-19*. 2020. Acessado em: 13/05/2022. Disponível em: <https://picovid.com.br/>.

CANTÃO, A. H. et al. Feature ranking from random forest through complex network's centrality measures. In: *Advances in Databases and Information Systems*. Cham: Springer International Publishing, 2022. p. 330–343. ISBN 978-3-031-15740-0.

CARNEIRO, M. G.; ZHAO, L. Organizational data classification based on the importance concept of complex networks. *IEEE Transactions on Neural Networks and Learning Systems*, p. 1–13, 2017. ISSN 2162-237X.

CARVALHO I et al. Informação em saúde na pandemia: O processo colaborativo de coleta e auditoria de dados no portal covid-19 brasil. In: . [online]: Revista de Saúde Digital e Tecnologias Educacionais, 2022. v. 7, n. 1, p. 92–108. ISSN 2525-9563.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. *Computers & Electrical Engineering*, v. 40, n. 1, p. 16 – 28, 2014. ISSN 0045-7906. 40th-year commemorative issue.

COSTA, L. da F. et al. Characterization of complex networks: A survey of measurements. *Advances in Physics*, Taylor & Francis, v. 56, n. 1, p. 167–242, 2007.

CUPERTINO, T. H. et al. A scheme for high level data classification using random walk and network measures. *Expert Systems with Applications*, v. 92, p. 289–303, 2018. ISSN 0957-4174.

DEVARAJ, S.; PAULRAJ, S. An efficient feature subset selection algorithm for classification of multidimensional dataset. *The Scientific World Journal*, n. Article ID 821798, p. 9 p., 2015.

DITZLER, G. et al. Fizzy: feature subset selection for metagenomics. *BMC Bioinformatics*, v. 16, n. 1, p. 358, 2015. ISSN 1471-2105.

DUBATH, P. et al. Random forest automated supervised classification of hipparcos periodic variable stars. *Monthly Notices of the Royal Astronomical Society*, Blackwell Publishing Ltd, v. 414, n. 3, p. 2602–2617, 2011. ISSN 1365-2966.

DUNN, O. J. Multiple comparisons among means. *Journal of the American Statistical Association*, American Statistical Association, Taylor & Francis, Ltd., v. 56, n. 293, p. 52–64, 1961. ISSN 01621459.

EFRON, B.; TIBSHIRANI, R. *An Introduction to the Bootstrap*. [S.l.]: Chapman & Hall, 1993.

EL ABOUDI, N.; BENHLIMA, L. Review on wrapper feature selection approaches. In: *International Conference on Engineering & MIS)*. [S.l.: s.n.], 2016. p. 1–5.

EPPSTEIN, D.; PATERSON, M. S.; YAO, F. F. On nearest-neighbor graphs. v. 17, p. 263–282, 04 1997.

ERDÖS, P.; RÉNYI, A. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, v. 5, p. 17–61, 1960.

ESTÉVEZ, P. et al. Normalized mutual information feature selection. *Neural Networks, IEEE Transactions on*, IEEE, v. 20, n. 2, p. 189–201, 2009.

FAN, M. et al. Adaptive data structure regularized multiclass discriminative feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, v. 33, n. 10, p. 5859–5872, 2022.

Referências

FERREIRA, C. A.; CANTÃO, A. H.; BARANAUSKAS, J. A. Decision tree induction through meta-learning. In: *International Federation for Information Processing*. Cham: Springer International Publishing, 2022. p. 101–111. ISBN 978-3-031-08337-2.

FERREIRA, L. N.; ZHAO, L. Time series clustering via community detection in networks. *Information Sciences*, Elsevier Ltd., v. 326, p. 227–242, 2016. ISSN 00200255.

FERRO, M.; MONARD, M.-C.; CAROLINA, M. Aquisição de conhecimento de conjuntos de exemplos no formato atributo valor utilizando aprendizado de máquina relacional. 2011.

FOITONG, S.; PINNGERN, O.; ATTACHOO, B. Feature subset selection wrapper based on mutual information and rough sets. *Expert Systems with Applications*, v. 39, p. 574–584, 2012.

FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, JSTOR, v. 11, n. 1, p. 86–92, 1940. ISSN 0003-4851.

GARCIA, L. P. et al. New label noise injection methods for the evaluation of noise filters. *Knowledge-Based Systems*, v. 163, p. 693 – 704, 2019. ISSN 0950-7051.

GOVINDAN, G.; NAIR, A. S. Sequence features and subset selection technique for the prediction of protein trafficking phenomenon in eukaryotic non membrane proteins. *International Journal of Biomedical Data Mining*, OMICS International, v. 3, n. 2, p. 1–9, 2014. ISSN 2090-4924.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 1157–1182, mar. 2003. ISSN 1532-4435.

HALL, M. A.; HOLMES, G. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, v. 15, n. 6, p. 1437–1447, Nov 2003. ISSN 1041-4347.

HAN, J.; KAMBER, M.; PEI, J. *Data mining: concepts and techniques*. [S.l.]: Morgan Kaufmann, 2011.

HASHEMI, A.; DOWLATSHAHI, M. B.; NEZAMABADI-POUR, H. Mgfs: A multi-label graph-based feature selection algorithm via pagerank centrality. *Expert Systems with Applications*, v. 142, p. 113024, 2020. ISSN 0957-4174.

HUISMAN, J. S. *Theoretical and Computational Analysis of Dynamics on Complex Networks*. Dissertação (Mestrado) — Technical University of Denmark, the Netherlands, 2016.

HULSE, J. D. V.; KHOSHGOFTAAR, T. M.; HUANG, H. The pairwise attribute noise detection algorithm. *Knowledge and Information Systems*, v. 11, n. 2, p. 171–190, Feb 2007. ISSN 0219-3116.

IKEHARA, K.; CLAUSET, A. Characterizing the structural diversity of complex networks across domains. *CoRR*, abs/1710.11304, 2017.

KATZ, L. A new status index derived from sociometric analysis. *Psychometric Society*, v. 18, n. Issue 1, p. 39–43, March 1953.

KHOSHGOFTAAR, T. M.; HULSE, J. V. Empirical case studies in attribute noise detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, v. 39, n. 4, p. 379–388, July 2009. ISSN 1094-6977.

Referências

LAI, J. et al. Adaptive graph learning for semi-supervised feature selection with redundancy minimization. *Information Sciences*, v. 609, p. 465–488, 2022. ISSN 0020-0255.

LAN, Y. et al. A hybrid feature selection method using both filter and wrapper in mammography cad. In: IEEE. *Image Analysis and Signal Processing (IASP)*. [S.l.], 2011. p. 378–382.

LEISCH, F.; DIMITRIADOU, E. *mlbench: Machine Learning Benchmark Problems*. [S.l.], 2010. R package version 2.1-1.

LIU, H.; ZHOU, M.; LIU, Q. An embedded feature selection method for imbalanced data classification. *IEEE/CAA Journal of Automatica Sinica*, v. 6, n. 3, p. 703–715, 2019.

LOUPPE, G. et al. Understanding variable importances in forests of randomized trees. In: BURGES, C. J. C. et al. (Ed.). *Advances in Neural Information Processing Systems 26*. [S.l.]: Curran Associates, Inc., 2013. p. 431–439.

MA, Y.; GUO, L.; CUKIC, B. Statistical framework for the prediction of fault-proneness. In: *Advances in machine learning applications in software engineering*. [S.l.]: Idea Group, 2007.

MANDAL, M.; MUKHOPADHYAY, A.; MAULIK, U. Prediction of protein subcellular localization by incorporating multiobjective pso-based feature subset selection into the general form of Chou's PseAAC. *Medical & Biological Engineering & Computing*, Springer Berlin Heidelberg, v. 53, n. 4, p. 331–344, 2015. ISSN 0140-0118.

MIAO, J.; NIU, L. A survey on feature selection. *Procedia Computer Science*, v. 91, p. 919–926, 2016. Information Technology and Quantitative Management.

MIN, H.; FANGFANG, W. Filter-wrapper hybrid method on feature selection. In: IEEE. *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*. [S.l.], 2010. v. 3, p. 98–101.

MITCHELL, T. M. *Machine Learning*. [S.l.]: WCB McGraw-Hill, 1997.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. In: *Sistemas Inteligentes Fundamentos e Aplicações*. 1. ed. Barueri-SP: Manole Ltda, 2003. cap. 4, p. 89–114. ISBN 85-204-168.

MORADI, P.; ROSTAMI, M. A graph theoretic approach for unsupervised feature selection. *Engineering Applications of Artificial Intelligence*, v. 44, p. 33 – 45, 2015. ISSN 0952-1976.

NETO, F. A.; ZHAO, L. High level data classification based on network entropy. In: *International Joint Conference on Neural Networks*. Dallas, TX, USA: IEEE, 2013. p. 1–5.

NEWMAN, M. The structure and function of complex networks. *Computer Physics Communications*, v. 147, p. 40–45, 03 2003.

NI, B.; YAN, S.; KASSIM, A. Learning a propagable graph for semisupervised learning: Classification and regression. *Knowledge and Data Engineering, IEEE Transactions on*, v. 24, n. 1, p. 114–126, 2012.

OSHIRO, T. M.; PEREZ, P. S.; BARANAUSKAS, J. A. How many trees in a random forest? In: *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Berlin, Germany: Lecture Notes in Computer Science, 2012. v. 7376, p. 154–168. ISBN 978-3-642-31536-7.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PURKAYASTHA, P. et al. Effect of feature selection on kinase classification models. In: *Computational Intelligence in Medical Informatics*. [S.l.]: Springer Singapore, 2015, (SpringerBriefs in Applied Sciences and Technology). p. 81–86. ISBN 978-981-287-259-3.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, p. 81–106, 1986.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. [S.l.]: Morgan Kaufmann, 1993.

ROFFO, G. et al. Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In: *IEEE International Conference on Computer Vision*. Venice, Italy: IEEE, 2017. p. 1407–1415. ISSN 2380-7504.

ROFFO, G. et al. Infinite feature selection: A graph-based feature filtering approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 43, n. 12, p. 4396–4410, 2021. ISSN 1939-3539.

SAEYS, Y.; INZA, I.; LARRAÑAGA, P. A review of feature selection techniques in bioinformatics. *Bioinformatics*, v. 23, n. 19, p. 2507–2517, 08 2007. ISSN 1367-4803.

SHEIKHPOUR, R. et al. A robust graph-based semi-supervised sparse feature selection method. *Information Sciences*, v. 531, p. 13–30, 2020. ISSN 0020-0255.

SILVA, T. C.; ZHAO, L. Network-based high level data classification. *IEEE Transactions on Neural Networks and Learning Systems*, v. 23, n. 6, p. 954–970, June 2012. ISSN 2162-237X.

SILVA, T. C.; ZHAO, L. *Machine Learning in Complex Networks*. [S.l.]: Springer, 2016. ISBN 978-3-319-17289-7.

TANG, C. et al. Robust graph regularized unsupervised feature selection. *Expert Systems with Applications*, v. 96, p. 64–76, 2018. ISSN 0957-4174.

TEREZINHA, A. M. Elementos intervenientes na tomada de decisão. v. 32, 04 2003.

UNCU, O.; TURKSEN, I. A novel feature selection approach: Combining feature wrappers and filters. *Information Sciences*, v. 177, n. 2, p. 449–466, 2007. ISSN 0020-0255.

VENKATESH, B.; ANURADHA, J. A review of feature selection and its methods. *Cybernetics and Information Technologies*, v. 19, p. 3, 03 2019.

VERRI, F. A. N.; ZHAO, L. Random walk in feature-sample networks for semi-supervised classification. In: *Brazilian Conference on Intelligent Systems*. [S.l.: s.n.], 2016. p. 235–240.

WANG, H. et al. Centrality combination method based on feature selection for protein interaction networks. *IEEE Access*, v. 10, p. 112028–112042, 2022.

ZANIN, M. et al. Feature selection in the reconstruction of complex network representations of spectral data. *PLOS ONE*, Public Library of Science, v. 8, n. 8, p. 1–7, 08 2013.

ZHAO, Y.; ZHANG, Y. Comparison of decision tree methods for finding active objects. *Advances in Space Research*, v. 41, p. 1955–1959, 2008.

ZHU, Y. et al. Graph feature selection for dementia diagnosis. *Neurocomputing*, v. 195, p. 19 – 22, 2016. ISSN 0925-2312. Learning for Medical Imaging.