UNIVERSIDADE FEDERAL DE SÃO CARLOS
CAMPUS SÃO CARLOS

Bruna Zamith Santos

CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO DE FUNÇÕES DE PROTEÍNAS VIA PREDIÇÃO DE INTERAÇÕES

BRUNA ZAMITH SANTOS

CLASSIFICAÇÃO HIERÁRQUICA MULTIRRÓTULO DE FUNÇÕES DE PROTEÍNAS VIA PREDIÇÃO DE INTERAÇÕES

Trabalho de Conclusão de Curso submetido à Universidade Federal de São Carlos, como requisito necessário para obtenção do grau de Bacharel em Engenharia de Computação

São Carlos, Junho de 2020

Dedico este trabalho primeiramente à minha avó Ruthe, com a qual aprendi o verdadeiro significado de resiliência. Ainda, ao meu avô Paulo, de quem herdei o sangue de engenheira e cientista; e ao meu avô Pedro, exemplo de dedicação e força de vontade. Por fim, dedico esse trabalho aos meus pais Regina e Hélio, minha fonte de suporte e inspiração para tudo que sou e faço até hoje.

Agradecimentos

Meus mais profundos agradecimentos ao Prof. Dr. Ricardo Cerri, por todo ensinamento ao longo dos meus 5 anos de graduação, sendo sempre extremamente solícito e me apoiando com todos os meus sonhos acadêmicos. Ainda, à Prof. Dra. Celine Vens, co-orientadora e idealizadora do projeto, com a qual muito aprendi e quem me deu todo o suporte para desenvolver o trabalho na *Katholieke Universiteit Leuven*. Aos meus amigos, obrigada pelo incentivo e grande ajuda com os empecilhos que surgiram ao longo da graduação, sempre me motivando e auxiliando. Agradeço à toda equipe de professores e funcionários do Departamento de Computação da Universidade Federal de São Carlos, assim como meus colegas do BioMaL (Bioinformatics and Machine Learning Group), em especial o Felipe Kenji Nakano. Por fim, meus agradecimentos ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelas diferentes bolsas-auxilio fornecidas, destacando-se os auxílios #2017/13218-5 e #2016/25078-0.

Resumo

Proteínas são macromoléculas responsáveis por praticamente todas as funções necessárias para a manutenção das células, tendo papel fundamental na regulação dos organismos. Avanços na área de Biologia Molecular permitiram uma listagem quase completa de todas as proteínas que compõem os organismos. Entretanto, existe um grande número de proteínas cujas funções ainda são desconhecidas, abrindo espaço para um novo foco de pesquisa em Biologia Molecular. Normalmente, a predição de funções de proteínas é feita usando-se ferramentas de Bioinformática baseadas em homologia, a qual consiste em comparar uma sequência com uma base de dados contendo várias sequências que pertencem a funções previamente conhecidas. Essa é uma estratégia limitada, uma vez que ignora as propriedades bioquímicas das sequências e as relações hierárquicas que podem existir entre as diferentes funções. Na literatura, o uso de Aprendizado de Máquina para predição de funções de proteínas tem se mostrado promissor, obtendo avanços significativos em relação ao uso de homologia e de outros métodos. Fazendo uso de Aprendizado de Máquina, é possível construir o problema de predição de funções de proteínas como um problema de Classificação Hierárquica Multirrótulo, devido ao fato de que funções de proteínas estão organizadas hierarquicamente e de que uma proteína pode exercer uma ou mais funções simultaneamente. Esse projeto propõe modelar o problema de predição de funções de proteínas como um problema de Classificação Hierárquica Multirrótulo, do inglês *Hierarchical Multi-label Classification* (HMC), através de dados interativos. Dados interativos são caracterizados por dois conjuntos de objetos, cada um descrito por seu próprio conjunto de atributos, o que permite a predição de interações entre duas instâncias. Em particular, adaptamos o método *Predictive Bi-Clustering Tree* (PBCT) para tarefas HMC. Nossos experimentos demonstraram que o PBCT-HMC é competitivo em relação ao concorrente estado-da-arte.

Palavras-chave: Aprendizado de Máquina. Aprendizado de Máquina Hierárquico. Aprendizado de Máquina Multirrótulo. Predição de Funções de Proteínas. Bioinformática. Aprendizado Supervisionado.

Abstract

Proteins are macro-molecules responsible for virtually every task necessary for the maintenance of cells, having a fundamental role in the behavior and regulation of organisms. Advances in the area of Molecular Biology have allowed an almost complete listing of the proteins that make up the organisms. However, there are a large number of proteins whose function is still unknown, opening space for a new research focus in Molecular Biology. Usually, protein function prediction is performed using homology-based Bioinformatic tools, comparing a sequence with a database with many sequences belonging to previously known functions. This is a limited strategy, since it ignores the sequences' biochemical properties, and also the hierarchical relationships that may exist between the different classes. In the literature, the use of Machine Learning for the protein function prediction has shown to be promising, obtaining significant advances regarding the use of homology and other methods. Making use of Machine Learning, it is possible to model the protein function prediction problem as a Hierarchical Multi-label Classification (HMC) problem, due to the fact that protein functions are hierarchically organized and that they can occur simultaneously. This project proposes modeling the protein function prediction task as a Hierarchical Multi-label Classification problem through interaction data. Interaction data are characterized by two sets of objects, each described by their own set of features, which makes it possible to predict the interactions between two instances. In particular, we adapt the *Predictive Bi-Clustering Tree* (PBCT) method to HMC tasks. Our experiments demonstrate that PBCT-HMC is competitive to the state-of-art competitor.

Keywords: Machine Learning. Hierarchical Machine Learning. Multi-label Machine Learning. Protein Functions Prediction. Bioinformatics. Supervised Learning.

Lista de abreviaturas e siglas

AM - Aprendizado de Máquina

AUPRC - do inglês, *Area Under Precision-Recall Curve*

ECC - do inglês, *Ensembles of Classifier Chains*

GAD - Grafo Acíclico Direcionado

HMC - do inglês, *Hierarchical Multi-label Classification*

IPP - Interação Proteína-Proteína

kNN - do inglês, *k-Nearest Neighbors*

MLP - do inglês, *Multi-Layer Perceptron*

PCT - do inglês, *Predictive Clustering Tree*

PBCT - do inglês, *Predictive Bi-Clustering Trees*

SVM - do inglês, *Support Vector Machine*

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Nos últimos anos, presenciamos um avanço na comunidade científica no que tange a capacidade de sequenciar genomas. Com mais de 150 genomas completos disponíveis [Roberts 2004], surgiu a necessidade de não só identificar as proteínas, mas de classificá-las de acordo com suas funções. Proteínas são macromoléculas responsáveis por praticamente todas as funções necessárias para a manutenção das células, que vão desde catalisar reações bioquímicas (enzimas) e formar estruturas (queratina e colágeno), à proteger o organismo (anticorpos).

Um dos métodos primários de classificação das funções de proteínas é a homologia, a qual baseia-se na similaridade de sequências. Quando duas sequências de proteínas apresentam similaridade estrutural significativamente relevante, esse pode ser um indicativo de que são homólogas, *i.e.* possuem um ancestral comum. No entanto, em alguns casos, proteínas, apesar de similares, podem ter funções diferentes; ou serem bastante distintas, mas terem a mesma função [Costa et al. 2007]. Métodos que consideram que sequências homólogas devem ter as mesmas funções estão sujeitos à propagação de incertezas, e falham em 20% a 40% das proteínas de genomas recentemente sequenciados [Letovsky e Kasif 2003].

Dentre os vários métodos para predição de funções de proteínas, o uso de Aprendizado de Máquina (AM) se destaca. Mais especificamente, aprendizado supervisionado, em que classificadores são treinados com base em objetos cujas classes (também chamadas de rótulos) já são conhecidas. Assim, cria-se um modelo capaz de mapear objetos desconhecidos em um espaço de rótulos determinado. Portanto, a extração de atributos adequados para as proteínas permite a criação de modelos preditivos de suas funções.

1.1 Problemática

Muitos dos trabalhos desenvolvidos na área de predição de funções de proteínas consideram que as categorias funcionais são isoladas e que existe apenas uma classificação final por proteína. Contudo, processos biológicos são altamente correlacionados e podem ocorrer simultaneamente.

Dentro do contexto de AM, existem diferentes abordagens para o problema de classificação e a escolha normalmente é determinada pelo domínio dos dados. Na classificação hierárquica, por exemplo, os rótulos são organizados hierarquicamente, isto é, possuem correlações e podem ser divididos em subclasses. Esse é o caso das funções de proteínas: Um exemplo é o rótulo “localização” como função num nível superior da hierarquia, e os

rótulos "localização subcelular" e "localização de tecido" no nível subsequente.

Outra abordagem é a classificação multirrótulo, em que um determinado objeto do domínio de estudo pode ser classificado em mais de um rótulo simultaneamente. Dado que uma proteína pode ser atribuída a mais de uma função, é benéfico considerar o problema de classificação de funções de proteínas como multirrótulo [Wang H. e Ding 2013]. Assim, a tarefa de predição de funções de proteínas é uma típica tarefa de Classificação Hierárquica Multirrótulo (do inglês, "*Hierarchical Multi-label Classification*" - HMC).

Como na tarefa de predição de funções de proteínas é valioso trabalhar com modelos de AM interpretáveis, a fim de extrair novas compreensões biológicas, este trabalho foca em modelos baseados em árvores de decisão. A fim de predizer as funções das proteínas, foi proposto na literatura um algoritmo de indução de árvores de decisão para problemas de classificação hierárquica multirrótulo [Vens et al. 2008]. Esse método constrói as chamadas "*Predictive Clustering Trees*" (PCTs).

Dentre as diferentes maneiras de se caracterizar problemas de classificação, destaca-se a representação por interação. Dados interativos são caracterizados por dois conjuntos de objetos, cada um descrito pelo seu próprio conjunto de atributos. São comumente modelados como redes e os valores de interesse são as possíveis interações entre dois objetos. Recentemente, um método de árvore de decisão foi proposto para predição de dados interativos [Pliakos, Geurts e Vens 2018], estendendo [Vens et al. 2008]. A árvores de decisão resultantes são chamadas de "*Predictive Bi-Clustering Trees*" (PBCTs), uma vez que são capazes de fazer divisões tanto no domínio dos objetos (divisões horizontais), quanto no domínio das classes (divisões verticais). No contexto deste projeto, os objetos representam as proteínas e as classes, as funções dessas proteínas. Ambos os algoritmos, PCTs e PBCTs, serão detalhados no Capítulo 4.

Nenhum dos trabalhos desenvolvidos até o momento abordou a predição de funções de proteínas, ou o problema do HMC em geral, como um problema de predição de interações. O desenvolvimento de métodos experimentais para análise de interação molecular possibilitou novas abordagens de inferência de funções de proteínas [Letovsky e Kasif 2003]. Proteínas podem interagir com diferentes componentes do organismo. Uma das interações mais estudadas é a Interação Proteína-Proteína (IPP). Ela desempenha papéis críticos em muitos processos biológicos celulares e é, inclusive, importante para elucidar funções de proteínas [Ding, Tang e Guo 2016, Sun et al. 2017].

Assim, abre-se espaço para o estudo da classificação hierárquica multirrótulo de funções de proteínas a partir de dados interativos. Com isso, pode-se também estender o uso de PBCTs para qualquer problema de HMC em geral onde a intenção primária não seja a determinação de interação entre dois objetos, mas sim a classificação. A vantagem da abordagem de *Predictive Bi-Clustering Tree* é que o modelo pode encontrar, através das divisões verticais, subconjuntos menores de funções a serem preditas juntas. Ou seja, cada

proteína pode pertencer a diferentes subconjuntos, dentro dos quais existe um domínio menor de funções de proteínas que apresentam uma relação mais significativa entre si. A classificação final, portanto, é dada pela concatenação desses diferentes subconjuntos. Na abordagem PCT, por sua vez, uma proteína pertence a um único subconjunto final, o qual não considera as relações entre as funções, apenas entre os objetos. Esse aspecto será discutido mais detalhadamente no Capítulo 5.

1.2 Objetivo

1.2.1 Objetivo Geral

O objetivo geral desta pesquisa foi modelar o problema de predição de funções de proteínas como um problema de Classificação Hierárquica Multirrótulo através de dados interativos. Para atingir esse objetivo, foi investigada a melhor maneira de se modelar a hierarquia de funções de proteínas como um conjunto de atributos. Como resultado, tendo ambas as proteínas e funções descritas por atributos, foi possível construir uma árvore de decisão que faz divisões ao longo do espaço de atributos das proteínas e do espaço de atributos das funções. Esse algoritmo desenvolvido foi denominado PBCT-HMC e pode ser aplicado para qualquer domínio de dados hierárquico e multirrótulo, não somente funções de proteínas.

1.2.2 Etapas de Desenvolvimento

Considerando o objetivo geral apresentado, é possível enumerar as etapas desenvolvidas para alcançá-lo:

- Modelagem do Conjunto de Dados: Esta etapa envolve a modelagem do conjunto de dados como dados interativos, principalmente no que tange o levantamento de atributos para o espaço de rótulos;

- Implementação do PBCT-HMC: Adaptação do algoritmo de *Predictive Clustering Tree* regular para selecionar o melhor teste dentre os atributos do espaço horizontal (proteínas) e vertical (funções), a fim de fazer divisões em ambas as direções;

- Refinamento do Método: Adoção do procedimento *lookahead* e adaptação do critério de parada;

- Análise e Comparação: Comparação da abordagem *Predictive Bi-Clustering Tree* com a abordagem *Predictive Clustering Tree* regular para aplicações HMC.

1.2.3 Resumo dos Resultados Obtidos

Comparamos o método proposto, PBCT-HMC, com o estado-da-arte em PCTs, o Clus-HMC [Vens et al. 2008]. Ambos os algoritmos são detalhados no Capítulo 4.

Em relação à medida de avaliação considerada neste estudo, a *Pooled Area Under Precision-Recall Curve*, o PBCT-HMC se destacou em 6 dos 16 conjuntos de dados, e obteve uma média de resultados próxima do Clus-HMC: 0.197 *versus* 0.202, respectivamente. Não há diferença estatisticamente significante entre os resultados dos dois algoritmos.

No que tange ao tamanho dos modelos, o PBCT-HMC gerou modelos maiores que o Clus-HMC. Esse resultado já era esperado, uma vez que o Clus-HMC faz somente divisões horizontais e, para o PBCT-HMC, cada divisão vertical levou a seis novos nós comparado com apenas dois nós para as divisões horizontais.

O PBCT-HMC também apresenta um maior tempo de indução, o que pode ser explicado pela aplicação do procedimento *lookeahead* (a ser explicado no Capítulo 5). Nos conjuntos de dados considerados, a média de diferença entre os tempos de execução do PBCT-HMC e do Clus-HMC não se mostrou significativa. No entanto, ela tende a crescer em conjuntos de dados com hierarquias muito extensas.

Em geral, o PBCT-HMC se mostrou um excelente modelo para classificação hierárquica multirrótulo de funções de proteínas e é extensível para qualquer domínio de dados hierárquico e multirrótulo. Assim, os resultados indicaram que tratar a tarefa de predição de funções de proteínas via predição de interações é relevante, principalmente na busca por modelos mais interpretáveis.

1.3 Organização

O restante deste trabalho está organizado da seguinte forma:

- Capítulo 2: Apresenta os principais conceitos de Classificação de Dados;
- Capítulo 3: Apresenta uma revisão bibliográfica de Classificação Hierárquica Multirrótulo e de Predição de Interações;
- Capítulo 4: Apresenta os principais aspectos de *Predictive Bi-Clustering Trees*;
- Capítulo 5: Apresenta em detalhes o algoritmo proposto, PBCT-HMC;
- Capítulo 6: Apresenta os resultados obtidos;
- Capítulo 7: Apresenta as conclusões e trabalhos futuros.

2 Classificação de Dados

Existem diferentes formas de se abordar um problema de aprendizado de máquina, o que normalmente é determinado pelo domínio dos dados. A Figura 1 expõe as principais abordagens.

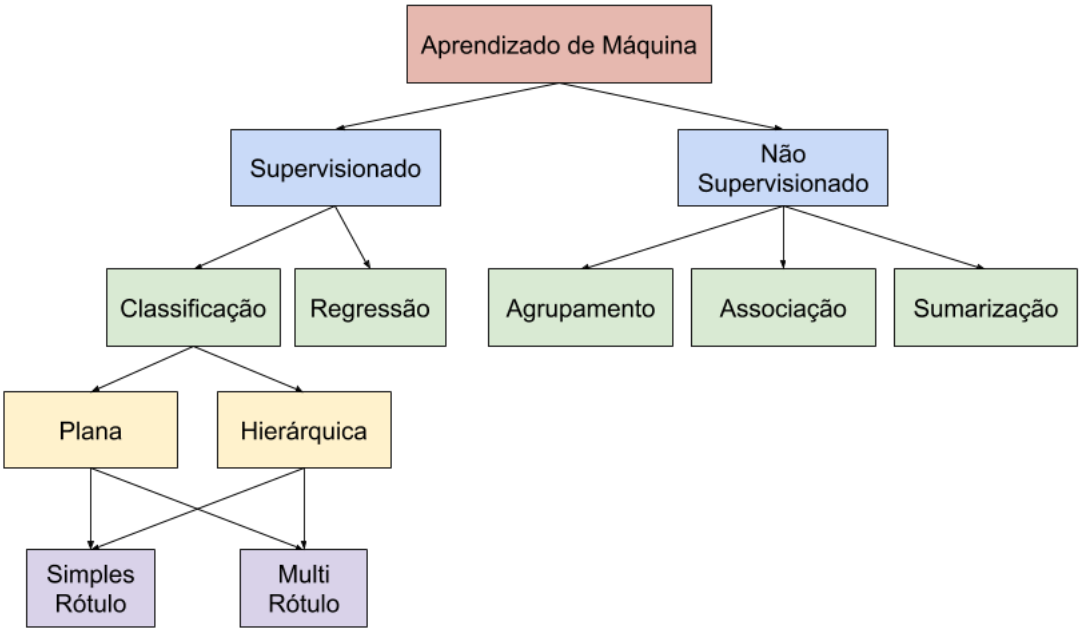


Figura 1 – Abordagens de aprendizado de máquina.

Nesse estudo, foca-se em classificação, uma tarefa de aprendizado de máquina supervisionado em que busca-se encontrar uma função f capaz de mapear um conjunto de atributos x em um conjunto de classes $y \subset Y$, onde Y é o domínio total de classes do problema. Esse modelo é construído através de um conjunto de dados de treinamento, onde, para todos os n objetos, as classes são conhecidas. O modelo pode ser representado de diversas formas, como uma árvore, uma rede neural ou até simplesmente uma tabela de probabilidades. O modelo representa as relações entre o espaço de atributos e o espaço de classes, e é dito capaz de classificar corretamente um objeto (x, y) se $f(x) = y$ [Tan, Steinbach e Kumar 2005].

De acordo com o interesse para este projeto, nas seções a seguir serão detalhados os principais aspectos da Classificação Multirrótulo e da Classificação Hierárquica. Por fim, a Classificação Hierárquica Multirrótulo será brevemente discutida.

2.1 Classificação Multirrótulo

Como exposto anteriormente, um problema de classificação convencional visa associar cada objeto x a um conjunto de classes $y \subset Y$. Se $|Y| = 2$, temos um domínio binário. Caso $|Y| > 2$, trata-se de um problema multi-classe. Problemas multi-classe cujo $|y| \geq 2$ são chamados de multirrótulo, *i.e*, um objeto pode ser classificado em mais de uma classe simultaneamente. Diferentemente de uma classificação multi-classe, na classificação multirrótulo as escolhas de rótulos não são consideradas mutualmente exclusivas. Assim, múltiplos rótulos podem estar associados a um único objeto [Elkafrawy, Mausad e Esmail 2015].

Um domínio comum de problemas multirrótulo é o de classificação de imagens. Uma fotografia de Copacabana, por exemplo, pode ser classificada, ao mesmo tempo, nas categorias "Praia", "Turismo", "Surf" e "Rio de Janeiro" (RJ).

De acordo com [Elkafrawy, Mausad e Esmail 2015], os métodos de Classificação Multirrótulo podem ser agrupados em três categorias principais: 1) Transformação do Problema, 2) Adaptação do Algoritmo e 3) Métodos Ensemble.

2.1.1 Transformação do Problema

Existem diversos métodos baseados em transformação do problema descritos na literatura. Os dois mais comuns, que serão abordados a seguir, são o *Label Power Set* e o *Binary Relevance*. Para fins didáticos, considere o conjunto de dados multirrótulo descrito na Tabela 1. O espaço de atributos foi omitido, pois o processo de transformação de problema modifica apenas o espaço de rótulos.

Tabela 1 – Exemplo de conjunto de dados multirrótulo.

Objeto	Turismo	Surf	Praia	RJ
1	X			X
2			X	X
3	X			
4		X	X	

2.1.1.1 *Label Power Set*

Esse método transforma o problema multirrótulo em um problema de classificação simples-rótulo, construindo um novo conjunto de rótulos. Esse novo conjunto é composto por todas as possíveis combinações de rótulos do conjunto original. Assim, gera-se o classificador $C : X \rightarrow P(L)$, onde $P(L)$ é o *power set* de L [Cherman, Monard e Metz 2011]. A Tabela 2 apresenta a transformação *Label Power Set* aplicada sobre os dados anteriormente apresentados na Tabela 1.

Tabela 2 – Exemplo de transformação *Label Power Set*.

Objeto	Turismo	(Turismo ∧ RJ)	(Praia ∧ RJ)	(Surf ∧ Praia)
1		X		
2			X	
3	X			
4				X

2.1.1.2 *Binary Relevance*

O método *Binary Relevance* decompõe um problema multirrótulo em vários problemas simples-rótulo binários, um para cada um dos n rótulos do conjunto $L = y_1, y_2, ..., y_n$ [Cherman, Monard e Metz 2011]. Cada um dos n conjuntos de dados binários D_j contém todos os objetos do conjunto de dados original, mas apenas dois rótulos: y_j e $\neg y_j$. Para completar a tarefa de classificação, portanto, serão necessários n classificadores. O resultado é a combinação de cada um dos resultados dos classificadores individuais. A Tabela 3 apresenta a transformação *Binary Relevance* aplicada sobre os dados anteriormente apresentados na Tabela 1.

Tabela 3 – Exemplo de transformação *Binary Relevance*.

Objeto	Turismo	¬ Turismo
1	X	
2		X
3	X	
4		X

Objeto	Surf	¬ Surf
1		X
2		X
3		X
4	X	

Objeto	Praia	¬ Praia
1		X
2	X	
3		X
4	X	

Objeto	RJ	¬ RJ
1	X	
2	X	
3		X
4		X

2.1.2 Adaptação do Algoritmo

Métodos de adaptação de algoritmo se baseiam em transformar um algoritmo de classificação convencionalmente desenvolvido para classificação simples-rótulo para ser capaz de classificar problemas multirrótulo. Exemplos são:

1. No trabalho de [Clare e King 2001], a fórmula da entropia do algoritmo de árvore de decisão C4.5 foi adaptada para englobar dados multirrótulo.

2. O algoritmo ML-kNN [Min-Ling Zhang e Zhi-Hua Zhou 2005] é uma adaptação do algoritmo *k-Nearest Neighbors* (kNN) regular. Ele combina um método de transformação do problema com um método de ranqueamento de rótulos, sendo capaz de classificar dados multirrótulo.

3. O trabalho de [Elisseeff e Weston 2001] apresenta um algoritmo de ranqueamento para ser aplicado em classificação multirrótulo. Baseia-se no princípio das *Support Vector Machines* (SVMs), onde um modelo linear tenta minimizar uma determinada função de custo.

4. MMAC [Thabtah, Cowling e Yonghong Peng 2004] é um algoritmo indutor de regras, que cria conjuntos de regras de classificação a partir de mineração de regras de associação. Alguns subconjuntos similares dentre os diversos conjuntos de regras criados podem ser combinados em uma única regra multirrótulo.

5. Os algoritmos IBLR-ML e IBLR-ML+ são outra adaptação de kNN [Cheng e H'Ullermeier 2009]. Eles calculam o vizinho mais próximo de um novo objeto e usam os rótulos desse vizinho como atributos para calcular as probabilidades *a priori* e obter a equação logística de regressão Bayesiana.

6. O trabalho de [Read, Pfahringer e Holmes 2008] apresenta a abordagem de "*Pruned Sets*". Essa abordagem considera grafos em que os nós representam as rótulos dos dados e os vértices representam as coocorrências desses rótulos. Então, o algoritmo foca em determinar as relações mais importantes dentre os rótulos e, para tal, vai podando, no grafo, as relações menos frequentes.

7. "*Classifier Chains*" [Read et al. 2009] se baseiam na estratégia *Binary Relevance* para criar um cadeia de classificadores binários. O espaço de atributos é estendido para conectar esses classificadores, passando informações sobre os rótulos entre eles. Assim, as correlações entre os rótulos são exploradas.

2.1.3 Métodos Ensemble

As abordagens convencionais de aprendizado multirrótulo muitas vezes ignoram as correlações entre os rótulos, como é o caso do *Binary Relevance*. Outras buscam extrair essas correlações construindo-se vários classificadores simples-rótulo e combinando os resultados individuais, como o *Label Power Set*. Ainda, é possível obter essas correlações através de um único classificador global, como em alguns métodos baseados na Adaptação de Algoritmo. No entanto, a capacidade de generalização dessas abordagens pode ser fraca [Shi et al. 2011].

Uma outra abordagem é a classificação multirrótulo por método *ensemble* (em português, "combinação"). Métodos ensemble constroem vários modelos individuais heterogêneos a fim de obter um único modelo final, baseado na combinação desses modelos individuais, com maior capacidade de generalização e com menor risco de *overfitting* (em português, "sobreajuste") [Moyano et al. 2018]. O *overfitting* em AM ocorre justamente quando o modelo foi capaz de se adaptar extremamente bem aos dados de treinamento,

porém não é capaz de fazer uma boa generalização para dados inéditos de teste, o que é um comportamento indesejado.

Em [Dietterich 2000], são apresentados alguns motivos pelos quais um classificador ensemble pode ser melhor que um único classificador, como por exemplo: 1. Ao escolher um único classificador, é possível que essa escolha seja ruim; 2. Vários algoritmos nem sempre são capazes de encontrar a solução ótima, então rodar o mesmo algoritmo múltiplas vezes (com diferentes parâmetros) e combinar as respostas pode levar a uma melhor aproximação da solução ótima.

Exemplos de métodos ensemble multirrótulo na literatura são:

1. O algoritmo Random k-labelsets [Tsoumakas e Vlahavas 2007] estende o algoritmo *Label Powerset*, construindo vários subconjuntos aleatórios de rótulos e treinando um classificador simples-rótulo para cada elemento no *power set* desse conjunto. O resultado então é a combinação desses classificadores individuais.

2. Ensembles de cadeias de classificadores (do inglês, "*Ensembles of Classifier Chains*" - ECC) [Read et al. 2009] buscam aumentar a acurácia e reduzir o *overfitting* dos *Classifier Chains* (apresentados anteriormente na Seção 2.1.2). Para tanto, os *classifier chains* são combinados e um esquema genérico de votação é incluído, onde os rótulos recebem votos de acordo com as predições ao longo dos classificadores individuais.

3. As *Predictive Clustering Trees* estudadas nesse projeto também podem ser combinadas em um método ensemble. Uma proposta é o uso do algoritmo de Random-Forests [Kocev et al. 2007] para melhorar a performance geral das PCTs.

4. O KFHE-HOMER, apresentado por [Pakrashi e Namee 2019], propõe treinar múltiplos classificadores multirrótulo baseados na abordagem HOMER [Tsoumakas, Katakis e Vlahavas 2008] e combinar suas saídas usando propriedades de sensor de fusão do filtro de Kalman.

5. Outro método é o ensemble de *Pruned Sets* [Read, Pfahringer e Holmes 2008] (apresentados anteriormente na Seção 2.1.2). Ao combinar os *Pruned Sets*, novos conjuntos de rótulos podem ser encontrados e adaptados para classificar dados irregulares ou mais complexos.

2.2 Classificação Hierárquica

Na literatura de AM, problemas de classificação convencionais são geralmente resolvidos com métodos de classificação plana (não hierárquicos). Este tipo de classificação desconsidera as relações hierárquicas entre as classes. Portanto, supõe independência entre

as classes, o que nem sempre é verdade. Ignorar as possíveis correlações entre as classes pode levar a um classificador com pouca capacidade de generalização.

Em problemas de classificação mais complexos, as classes estão normalmente dispostas em uma hierarquia, isto é, as classes podem ser divididas em subclasses e agrupadas em superclasses. Nesses casos, os classificadores consideram as relações mais relevantes entre os dados de treinamento na classificação. O classificador f deve respeitar as restrições da taxonomia hierárquica. Ou seja, quando uma classe é predita, todas as suas superclasses devem ser preditas. A Figura 2 ilustra a diferença entre os problemas de classificação plana e hierárquica.

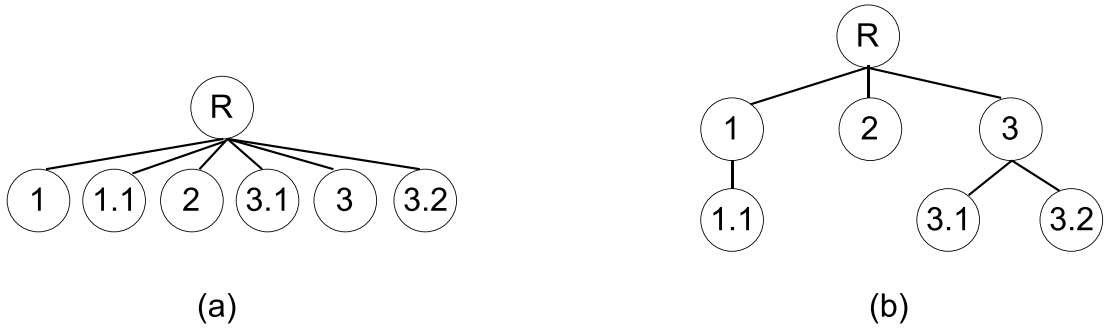


Figura 2 – Diferença entre problemas de classificação. (a) Plana; (b) Hierárquica.

A distinção entre classificadores hierárquicos se dá de acordo com o tipo de hierarquia em que podem ser aplicados. Essas hierarquias são nomeadas de acordo com a maneira que organizam suas classes [Silla e Freitas 2010]: Árvore ou Grafo Acíclico Direcionado (GAD). A diferença entre elas está no fato de que na árvore, um nó filho possui um único pai, enquanto que no GAD um nó pode possuir mais de um pai, sendo classificado simultaneamente em duas ou mais classes distintas. A Figura 3 exemplifica as duas hierarquias.

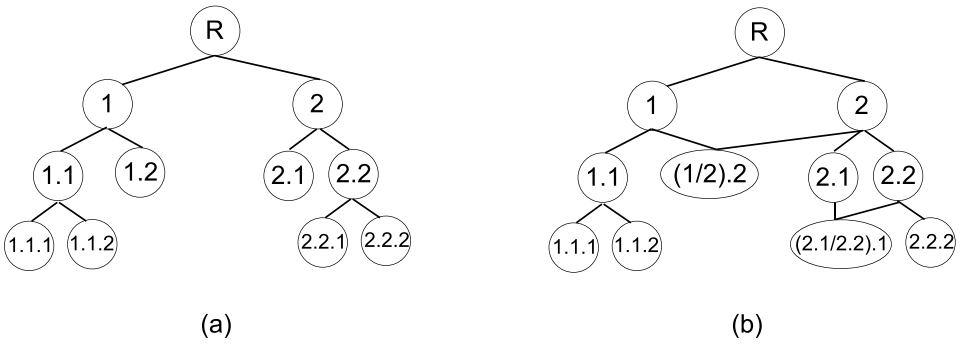


Figura 3 – Exemplos de hierarquias. (a) Árvore; (b) Grafo Acíclico Direcionado.

Problemas hierárquicos são muito estudados no domínio da biologia, devido ao fato de que os objetos biológicos frequentemente compartilham propriedades em comum

entre si (sendo agrupados em uma família), que não compartilham com outros objetos mais distantes (ou com outras famílias) [Zimek et al. 2010].

Duas principais abordagens são utilizadas para classificação hierárquica: local e global. Elas são descritas a seguir.

2.2.1 Abordagem Local

Em [Koller e Sahami 1997], foi apresentado o primeiro classificador baseado na abordagem local com um procedimento *top-down* (descrito mais a frente). Na abordagem local, a hierarquia é levada em consideração a partir de informações locais, o que pode ser feito por meio de três estratégias diferentes [Silla e Freitas 2010]:

- Classificador local por nó;
- Classificador local por nó pai;
- Classificador local por nível;

Nas três estratégias, classificadores convencionais são empregados diretamente para a classificação hierárquica. O classificador local por nó se baseia na associação de um classificador binário por nó (exceto a raiz). Já o classificador local por nó pai associa um classificador a cada nó interno da hierarquia, com o objetivo de distinguir as subclasses desses nós. São usados ou classificadores multirrótulo ou combinações de classificadores binários. Por último, o classificador local por nível associa um classificador multirrótulo por nível da hierarquia. Na literatura, o classificador local por nó é um dos mais utilizados, sendo que diferentes estratégias podem ser usadas para determinar quais objetos serão considerados positivos e quais serão considerados negativos no momento do treinamento de cada nó [Silla e Freitas 2010].

Apesar das três estratégias locais diferirem na fase de treinamento, elas utilizam um procedimento *top-down* na fase de teste. Nesse procedimento, para cada novo objeto no conjunto de teste, o classificador prediz sua classe mais genérica (classe localizada no nível mais próximo da raiz). Outro classificador, associado ao nível ou classe predita, é responsável pela predição das subclasses no próximo nível (classes filhas). Todo o processo é feito recursivamente, seguindo o princípio de divisão e conquista. Exemplificando: Se no primeiro nível o objeto foi classificado como pertencente à classe 2, no próximo nível ela só poderá pertencer a uma ou mais classes filhas de 2 (2.1, 2.2, 2.3,...2.n). A desvantagem do procedimento *top-down* é que erros cometidos em um nível são propagados para os próximos níveis. A Figura 4 exemplifica as três estratégias locais.

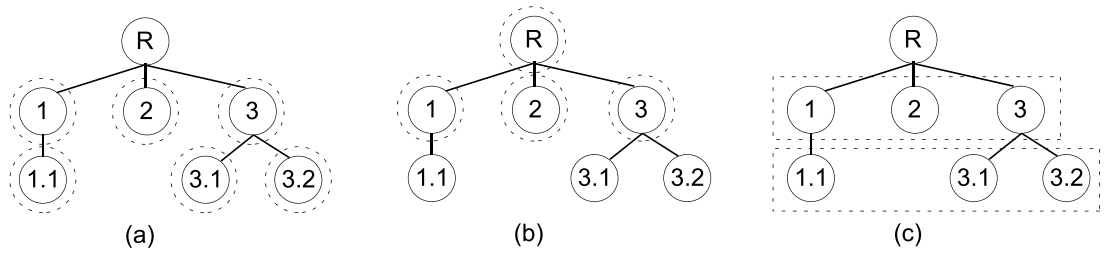


Figura 4 – Abordagem local: (a) Um classificador local por nó; (b) Um classificador local por nó pai; (c) Um classificador local por nível. Adaptado de [Faceli et al. 2011].

2.2.2 Abordagem Global

Na abordagem global, apenas um modelo de classificação é induzido a partir do conjunto de treinamento, levando em consideração a hierarquia de classes como um todo [Silla e Freitas 2010]. A vantagem desta abordagem é que, normalmente, a complexidade do modelo de classificação é menor que a complexidade de vários classificadores juntos. Porém, informações locais não são utilizadas durante o treinamento. Como informações utilizadas para a classificação de objetos em níveis mais próximos da raiz são diferentes de informações utilizadas para classificar objetos em níveis mais profundos, métodos locais podem apresentar vantagens sobre métodos globais.

Existem diversos classificadores globais desenvolvidos e eles possuem em comum o fato de considerarem toda a hierarquia de classes de uma única vez no momento do treinamento, além do fato de não possuírem modularidade para treinamento local de classificadores [Madjarov G. 2014].

2.3 Classificação Hierárquica Multirrótulo

A Classificação Hierárquica Multirrótulo é a abordagem que trata o domínio do problema como sendo hierárquico e multirrótulo simultaneamente. Isto é, considera as relações hierárquicas entre as classes, ao mesmo tempo em que é capaz de associar um objeto a mais de uma classe. Assim, um objeto pode ser classificado simultaneamente em dois ou mais caminhos de uma hierarquia de classes. Um típico domínio hierárquico multirrótulo é o domínio estudado neste projeto, das proteínas e suas funções. Isto porque:

1. As funções de proteínas estão organizadas em uma hierarquia. Um exemplo é o rótulo “localização” como função num nível superior da hierarquia, e os rótulos “localização subcelular” e “localização de tecido” no nível subsequente.

2. Uma mesma proteína pode exercer múltiplas funções. Um exemplo disso são as Cristalinas (em inglês, “Crystallins”), que constituem grande parte das lentes dos

olhos de aves e répteis, ao mesmo tempo em que contribuem como enzimas para a desidrogenase do lactato [Jeffery 2018].

No próximo capítulo, portanto, o foco será a revisão bibliográfica de estudos que aplicaram a Classificação Hierárquica Multirrótulo nos mais diversos domínios. Além da revisão de outros estudos que abordaram o problema de predição de interações. Nenhum dos trabalhos desenvolvidos até o momento abordou a predição de funções de proteínas, ou o problema de Classificação Hierárquica Multirrótulo em geral, como um problema de predição de interações.

3 Revisão Bibliográfica

3.1 Classificação Hierárquica Multirrótulo

Três métodos de PCTs foram investigados por [Vens et al. 2008]: Clus-HMC, um método global para induzir uma árvore de decisão única considerando todas as classes hierárquicas; Clus-SC local, que treina uma árvore de decisão binária para cada classe, ignorando os relacionamentos entre as classes; e Clus-HSC, que induz uma árvore de decisão binária para cada classe, explorando as relações hierárquicas entre elas. [Schietgat et al. 2010] também usa uma técnica de agrupamento para combinar as árvores de decisão induzidas pelo Clus-HMC.

Um conjunto de classificadores locais foi proposto por [Valentini 2009]. Nesse método, cada classificador treinado estima a probabilidade local de um determinado objeto pertencer a uma determinada classe. Uma fase de combinação estima a probabilidade consensual global. Os mesmos autores [Valentini e Re 2009, Valentini 2011] modificaram esse método para modular a relação entre a predição de uma classe e a predição de suas subclasses.

[Cesa-Bianchi, Re e Valentini 2011] investigaram a sinergia entre diferentes estratégias locais relacionadas à tarefa de predição de funções gênicas. Eles integraram ferramentas de fusão de dados baseados em kernel e algoritmos de conjunto com métodos sensíveis ao custo [Cesa-Bianchi e Valentini 2010, Valentini 2011]. Os autores definiram a sinergia como a melhoria na precisão da predição, considerando qualquer medida de avaliação, devido ao uso de estratégias de aprendizagem concorrente. A sinergia é detectada quando a ação combinada de duas estratégias alcança melhores taxas de classificação corretas do que a média da classificação correta das duas estratégias utilizadas separadamente [Cesa-Bianchi, Re e Valentini 2011].

[Borges e Nievola 2012] propuseram uma rede neural competitiva formada por uma camada de entrada e uma camada de saída. As distâncias entre os nós de hierarquia e cada objeto de treinamento são calculadas. Os neurônios com as menores distâncias são considerados vencedores, influenciando seus ancestrais. Os pesos da rede neural são ajustados de acordo com as classes associadas aos neurônios vencedores.

O trabalho de [Stojanova et al. 2013] propõe um método para considerar a auto-correlação, *i.e*, as relações estatísticas entre a mesma variável em objetos diferentes, mas relacionados. A ideia é usar, durante o treinamento, uma combinação de características e auto-correlações entre objetos. Uma rede é usada para modelar as auto-correlações, que são então usadas pelo método enquanto se aprende.

[Bi e Kwok 2014] usam a estratégia "*Mandatory Leaf Node Prediction*". Eles usam informações de hierarquia e buscam encontrar os vários rótulos com a maior probabilidade posterior sobre todos os rótulos. Os autores estenderam a propriedade de aproximação aninhada [Baraniuk et al. 2010] para lidar com problemas estruturados como GADs, o que foi resolvido usando um algoritmo guloso.

Problemas de predição de funções de proteínas com rótulos hierárquicos incompletos foram investigados por [Yu, Zhu e Domeniconi 2015]. Semelhanças hierárquicas e planas (não hierárquicas) entre funções foram consideradas e a similaridade combinada entre os rótulos foi definida. Os rótulos conhecidos e essa similaridade são usados para estimar funções ausentes na hierarquia. Informações sobre interações proteína-proteína também são usadas. Situações em que rótulos estão faltando foram simulados aleatoriamente mascarando as funções foliares de uma proteína.

Em [Triguero e Vens 2016], os autores propuseram um estudo envolvendo alternativas para realizar a rotulação final em problemas hierárquicos. Os autores avaliaram o uso de limiares únicos e múltiplos para transformar as pontuações de predição com valor real em rótulos binários reais. Para escolher os limiares, duas abordagens foram propostas: Otimizar uma determinada medida de avaliação; ou simular as propriedades do conjunto de treinamento no conjunto de testes. Como medidas de avaliação, os autores utilizaram a função de perda hierárquica e a *f-measure*, concluindo que selecionar limiares para cada classe é uma boa alternativa, resultando em rótulos melhorados e tempo de execução mais rápido.

[Sun et al. 2016] formulou a tarefa de classificação como um problema de seleção de caminhos, onde cada caminho começa na raiz e termina em uma folha ou nó interno. Eles usaram técnicas de mínimos quadrados parciais para resolver o problema de predição de rótulos como um problema de predição de caminho ótimo. Cada predição é então um subgrafo conectado, que pode ser formado por um pequeno número de caminhos. O método proposto então encontra os caminhos ótimos, e a predição final é dada pela união desses caminhos.

Em [Cerri et al. 2016] foi proposto um método chamado "*Hierarchical Multi-label Classification with Local Multi-Layer Perceptrons*" (HMC-LMLP). O método associa uma rede neural *multi-layer perceptron* (MLP) a cada nível hierárquico, sendo cada MLP responsável pelas predições em seu nível associado. As predições em um nível são usadas para complementar os vetores de atributos dos objetos usados para treinar a rede neural associada ao próximo nível. Assim, o treinamento e os testes são realizados de maneira *top-down*.

3.2 Predição de Interações

Uma das interações proteicas mais comuns estudadas atualmente são as chamadas Interações Proteína-Proteína (IPP). Entender essas interações é uma tarefa importante, pois regulam os processos celulares fundamentais e ser capaz de prever os IPPs significa possivelmente poder identificar também as funções das proteínas envolvidas. [O. Nussinov R. 2008] apresenta um algoritmo para predição de IPPs que emprega uma abordagem *bottom-up* e considera a similaridade estrutural e a conservação evolucionária das proteínas. Ele executa um método completo para verificar se duas regiões de proteínas são semelhantes a cadeias parceiras complementares de uma interface de modelo ou não. Em [Hamp e Rost 2015], os métodos evolutivos e os métodos baseados em *profile-kernel support vector machines* são empregados para predizer os IPPs. Muitos outros estudos tentam predizer IPPs usando novas estratégias e algoritmos, como [You, Chan e Hu 2015], [Kamada M. 2014] e [Fields 1989].

As IPPs não são a única tarefa de predição de interação em estudo na literatura de predição de interações. Muitas outras interações podem ser usadas como dados para algoritmos preditivos. Em [Li 2013], os sistemas de recomendação são modelados como uma tarefa de predição de conexões em grafos bipartidos. Um novo kernel gráfico é projetado para prever conexões entre usuários e itens, fazendo uso de caminhos aleatórios. [Liben-Nowell e Kleinberg 2003] analisam diferentes abordagens para predizer as arestas (interações) que serão adicionadas a uma rede social em um intervalo de tempo futuro, caracterizando outro problema de predição de conexões.

[Zhang W. e Li 2017] apresentam uma integração de diferentes modelos com regras de conjunto adequadas para predizer as interações medicamentosas. O estudo teve como objetivo predizer interações medicamentosas não observadas, utilizando métodos de classificação no âmbito da aprendizagem semi-supervisionada. Três modelos representativos foram utilizados para construir os modelos preditivos: o método de recomendação de vizinho [Bobadilla et al. 2013], o método de passeio aleatório e o método de perturbação de matriz [Lü et al. 2015].

Recentemente, um método foi introduzido no contexto de aprendizagem em pares [Pliakos, Geurts e Vens 2018]. Ele é capaz de predizer interações entre dois conjuntos de dados ao criar *bi-clusters* da matriz de interação. Isto é, criar pequenos subconjuntos (*clusters*) ao fazer divisões verticais e horizontais na matriz. Esses subconjuntos devem representar objetos com interações mais fortes entre si quando comparadas às interações com objetos que caem em diferentes *clusters*. A pesquisa desenvolvida neste Trabalho de Conclusão de Curso se baseia nesse algoritmo, mas trazendo-o para o contexto de HMC.

4 Predictive Bi-Clustering Trees

A fim de conceituar as *Predictive Bi-Clustering Trees*, primeiro são introduzidos os conceitos de Árvore de Decisão e de *Predictive Clustering Trees*. O algoritmo implementado nesse trabalho, PBCT-HMC, é apresentado futuramente na Seção 5.1.

4.1 Árvores de Decisão

Árvores de Decisão são uma categoria de algoritmos de AM Supervisionado e destacam-se em várias aplicações por sua simplicidade e interpretabilidade. Nesses algoritmos, o modelo é representado como uma árvore em que cada nó representa um teste sobre um atributo e os nós-folha representam classes. Os objetos são classificados um a um, descendo-se pela árvore da raiz até um nó-folha. Nesse processo, em cada nível da árvore, um teste é aplicado em algum atributo, para decidir qual o próximo nó a seguir. O nó-folha representa a classificação final encontrada para aquele objeto, após ter percorrido a árvore. Um exemplo ilustrativo de uma árvore de decisão é mostrado na Figura 5, onde os dados constituem aspectos do tempo, com atributos como umidade, temperatura e força do vento. A classificação esperada, nesse mesmo exemplo, é se vai chover ("Sim" ou "Não").

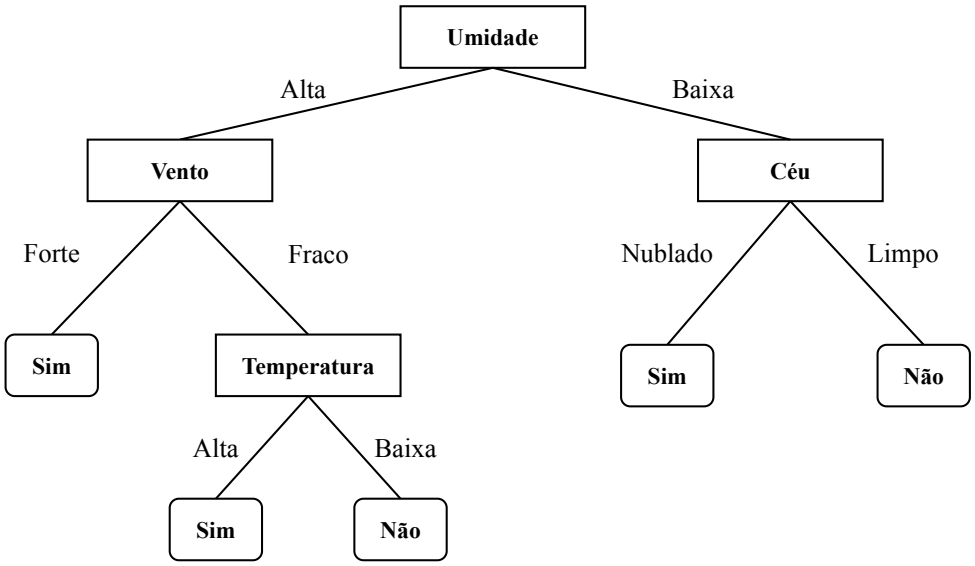


Figura 5 – Ilustração de uma Árvore de Decisão. Adaptado de [Mitchell 1997].

Exemplificando, considere um objeto com os seguintes valores para seus atributos: Umidade = Alta, Vento = Fraco, Temperatura = Alta, Céu = Nublado. Percorrendo a árvore de decisão da Figura 5 de acordo com os valores dos atributos do objeto, obtemos a

classificação "Não", indicando que para não irá chover: $f(Alta, Fraco, Alta, Nublado) =$ Não.

Existem diferentes algoritmos baseados em Árvores de Decisão propostos na literatura, cada um com diferentes métodos para construção do modelo e diferentes heurísticas. Os mais conhecidos na literatura são os algoritmos ID3 [Quinlan 1986] e seu sucessor, C4.5 [Quinlan 1993]. Na maioria dos exemplos, a árvore de decisão é construída de acordo com um procedimento *top-down*, começando com a raiz e descendo nível a nível. A primeira pergunta a ser feita, portanto, é "Qual o atributo deveria ser testado na raiz da árvore?" [Mitchell 1997]. Para responder essa pergunta, heurísticas são aplicadas a cada atributo e aquele que obtiver o melhor resultado é escolhido. Os dados então são particionados, de um lado do nó ficam os objetos que retornam positivo para aquela heurística, e do outro ficam os objetos que retornam negativo. Isso considerando-se divisões binárias. Vale ressaltar que para divisões não binárias, o procedimento ocorre da mesma maneira, com a diferença de que cada nó pode ter mais de 2 filhos. O procedimento então é repetido recursivamente até atingirmos um critério de parada previamente definido, gerando então um nó-folha. A classificação final dada por aquele nó-folha leva em consideração a classe majoritária dos objetos que atingiram aquele nó.

4.2 *Predictive Clustering Trees*

Uma *Predictive Clustering Tree* é uma generalização de uma árvore de decisão, em que cada nó corresponde a um *cluster*. Um *cluster* é, neste contexto, uma coleção de objetos similares entre si. Dependendo das informações contidas nos nós-folha, as PCTs podem ser usadas para diferentes tarefas de aprendizado, incluindo agrupamento, classificação (simples ou multirrótulo, plana ou hierárquica) e regressão (simples ou *multi-target*). Elas estão implementadas no *software* Clus[1] e são construídas em um procedimento *top-down*, isto é, o nó raiz corresponde ao conjunto de treinamento completo, que é particionado recursivamente em cada divisão [Vens et al. 2008].

Cada objeto é atribuído a um vetor binário de rótulos, em que o i-ésimo componente do vetor é 1 se aquele objeto pertence à classe c_i e 0 caso contrário. Assim, para um *cluster*, é calculado um vetor com a média aritmética dos vetores do conjunto, chamado vetor-protótipo. Então, o j-ésimo componente do vetor-protótipo corresponde à proporção de objetos do *cluster* pertencentes à classe c_j.

A heurística usada para selecionar testes a serem incluídos na árvore é a redução da variância intra-cluster. Na classificação multirrótulo, como os rótulos são representados

[1] <https://dtai.cs.kuleuven.be/clus/>

por um vetor binário, a variância de um conjunto de objetos S é definida como:

$$Var(S) = \frac{\sum_i d(\mathbf{v}_i, \overline{\mathbf{v}})^2}{|S|} \tag{4.1}$$

onde d é a distância entre cada vetor de rótulos $\mathbf{v}_i$ e o vetor-protótipo do conjunto, $\overline{\mathbf{v}}$.

Em tarefas de HMC, é conveniente considerar que a similaridade entre nós localizados em níveis mais altos da hierarquia é mais importante do que a similaridade em nós localizados em níveis mais baixos. Levando-se em consideração esse conceito, a implementação HMC de PCTs no Clus (Clus-HMC) faz uso da distância Euclidiana d ponderada na Equação 4.1 [Vens et al. 2008]. O peso da classe w decresce exponencialmente com a profundidade da classe na hierarquia. O peso w de uma classe c é definido como:

$$w(c) = w_0^{profundidade(c)} \tag{4.2}$$

A classificação de um objeto de teste é feita de forma semelhante a uma árvore de decisão comum: Percorrendo-se a árvore da raiz até um nó-folha, aplicando-se os testes em cada nível. O nó-folha, como exposto anteriormente, fornece um vetor-protótipo. A classificação final baseia-se em um limiar t, onde o objeto de teste é dito pertencente a uma determinada classe c_i se $\overline{\mathbf{v}}[i] \geq t$.

4.3 *Predictive Bi-Clustering Trees*

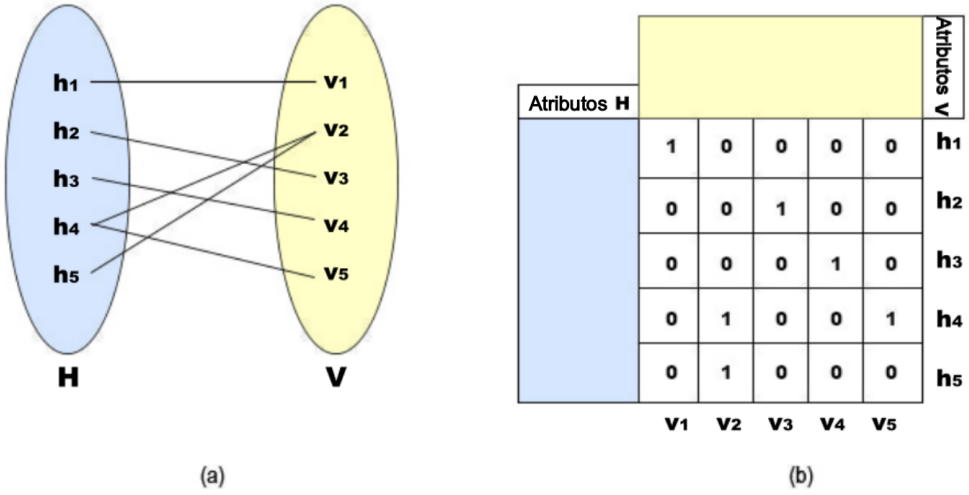


Figura 6 – (a) Representação de uma rede de interação, (b) A mesma rede representada como um dado de interação (matriz de interação).

Uma PBCT é capaz de fazer divisões horizontais e verticais (ao longo dos conjuntos H e V, respectivamente, como representado na Figura 6), predizendo as interações entre os nós. Como a PCT, a PBCT também é construída em um procedimento *top-down*.

Isso significa que, em cada iteração, um teste é aplicado a um dos atributos. O teste é escolhido considerando ambos os conjuntos de atributos (H e V), com base na heurística e no critério de parada. Mais detalhes do algoritmo serão apresentados no Capítulo 5 a seguir.

5 Proposta

Nesse capítulo, serão apresentados os diversos aspectos do algoritmo proposto, PBCT-HMC, como a representação de dados, heurísticas e critérios de parada.

5.1 Visão Geral: PBCT-HMC

Embora o método PBCT tenha sido aplicado a tarefas de classificação multirrótulo antes [Pliakos, Vens e Tsoumakas 2019], ele foi projetado para uma tarefa diferente (predição de interação) e, portanto, são necessárias algumas adaptações ao algoritmo de PBCT comum para tornar o método totalmente adaptado ao contexto do HMC. Nossa abordagem proposta é denominada PBCT-HMC.

Para ilustrar uma das vantagens do uso do PBCT-HMC frente ao uso de PBCT comuns em problemas HMC, a Figura 7 fornece um exemplo onde, para a matriz de rótulos mostrada na caixa superior esquerda, não existe uma divisão horizontal adequada. No entanto, se uma divisão vertical for realizada primeiro, é possível encontrar duas divisões horizontais posteriormente, resultando em quatro *bi-clusters* puros.

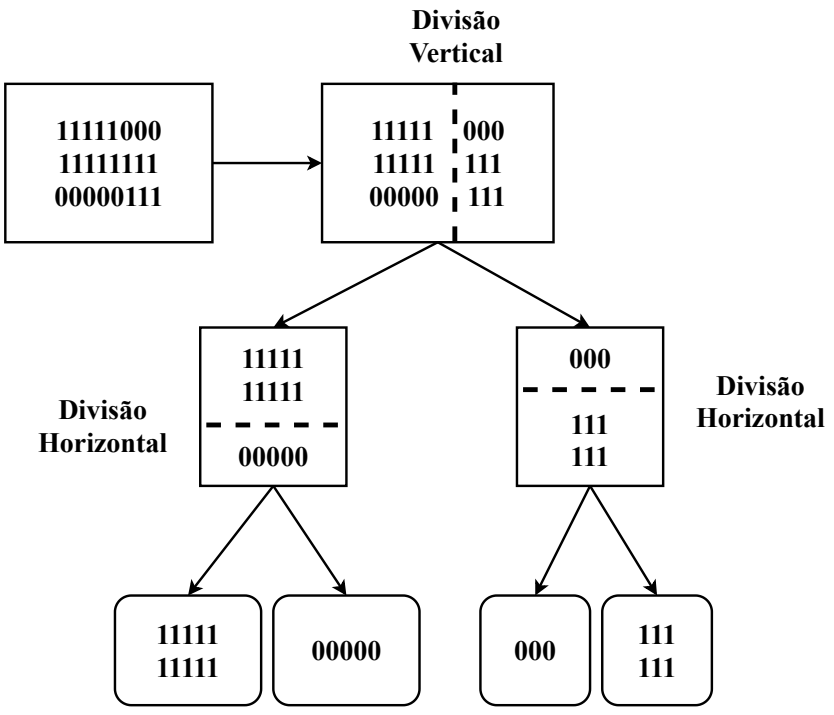


Figura 7 – Exemplo em que usar PBCT tem vantagens comparado a usar um PCT comum.

As seções a seguir detalham as diferentes etapas que abrangem a construção do método PBCT-HMC.

5.2 Representação dos Dados

Para modelagem em dados interativos, são construídos dois conjuntos de dados complementares: H e V (horizontal e vertical, respectivamente). Esses conjuntos são, cada um, representados como matrizes definidas da seguinte forma:

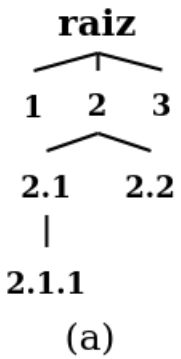
- Conjunto H: As linhas $0...n$ correspondem às n proteínas. As colunas $0...(k-1)$ representam os k atributos que descrevem as proteínas. As colunas $k...(k+m)$ representam as m classes da hierarquia, ou seja, as funções de proteínas. Se um objeto x pertence à classe y, então $H[X,Y]$ recebe o valor 1, onde X corresponde à linha em que x se encontra na matriz H, e Y, à coluna em que y se encontra. Caso contrário, $H[X,Y]$ recebe 0.
- Conjunto V: As linhas $0..m$ correspondem às m funções de proteínas. As colunas $0...(l-1)$ representam os l atributos que descrevem as funções de proteínas. As colunas $l...(l+n)$ representam as n proteínas. De maneira análoga ao Conjunto H, se uma proteína x pertence à classe y, então $V[Y,X]$ recebe o valor 1, onde Y corresponde à linha em que y se encontra na matriz V, e X, à coluna em que x se encontra.

Os espaços de atributos das matrizes H e V serão denotados H^F e V^F, respectivamente. E os espaços de rótulos serão denotados H^T e V^T. Então é possível dizer que H^T é a matriz transposta de V^T. O próximo passo para ter as proteínas e suas funções representadas como dados interativos é determinar os atributos de V^F. Para tal, quatro diferentes abordagens foram inicialmente propostas:

- Caminho: Armazena vetores binários para cada classe, representando seu caminho a partir da raiz;
- Profundidade: Armazena vetores binários para cada classe, representando o nível da árvore a qual cada classe pertence;
- Descendentes: Armazena vetores binários para cada classe, representando os descendentes de cada nó;
- Subárvore: Armazena vetores binários para cada classe. Os componentes desse vetor equivalem aos rótulos que compõem o primeiro nível da hierarquia, isto é, o nível logo abaixo da raiz.

No entanto, das abordagens propostas, apenas a abordagem da Subárvore permitiu que fossem geradas divisões verticais significativas. As outras abordagens levaram a um espaço de atributos V^F muito esparso, ou até muito grande, o que degrada a performance do algoritmo. Desta forma, o estudo restringe-se apenas à abordagem Subárvore.

Para uma melhor performance do algoritmo, os atributos de V^F foram transformados em um único atributo categórico. Assim, essa abordagem pode ser descrita da seguinte maneira: Para cada rótulo l é determinado o rótulo que compõe o primeiro nível da hierarquia, isto é, o nível logo abaixo da raiz. Se o rótulo l é descendente do rótulo k (no primeiro nível), então o atributo categórico k se torna o atributo de l. Uma hierarquia de exemplo é mostrada na Fig. 8, juntamente com sua representação. O mapeamento proposto leva em consideração as propriedades estruturais da hierarquia de rótulos, garante produzir predições que atendam à chamada restrição de hierarquia (ou seja, a probabilidade predita para um rótulo filho não pode exceder a do rótulo pai) e tem as vantagens de ser pequeno (o que é benéfico para a abordagem *lookahead* - descrita mais adiante). Além disso, é muito simples de modelar e pode ser aplicada em qualquer domínio hierárquico.



	Atributo V
1	"1"
2	"2"
2.1	"2"
2.1.1	"2"
2.2	"2"
3	"3"

(b)

Figura 8 – (a) Exemplo de hierarquia de rótulos e (b) seu vetor de atributos resultante.

As Tabelas 4 e 5 a seguir representam os conjuntos H e V, respectivamente.

Tabela 4 – Ilustração do conjunto de dados H.

	Atr#1	**Atr#2**	**...**	**Atr#k**	**1**	**1.1**	**2**	**...**	**Classe m**
Proteína 1	0.98	0.12	...	0.87	1	1	0	...	1
Proteína 2	0.54	0.67	...	0.43	0	0	0	...	1
...	...	...	...	...	...	...	...	...	...
Proteína n	0.73	0.17	...	0.02	1	0	1	...	0

Tabela 5 – Ilustração do conjunto de dados V.

	Atr Vertical	**Proteína 1**	**Proteína 2**	**...**	**Proteína n**
1	"1"	1	0	...	1
1.1	"1"	1	0	...	0
2	"2"	0	0		1
...	...	...	...	...	...
Classe m	"m_0"	1	1	...	0

5.3 Indução da Árvore e Heurística de Divisão

Como no método PBCT, a indução da árvore é feita segundo o procedimento *top-down*. Dado um nó k da árvore, ele pode ser associado a um *bi-cluster* definido por um par (H_k, V_k) com $H_k \subseteq H^T$ e $V_k \subseteq V^T$. Os subconjuntos H_k e V_k podem ser obtidos seguindo o caminho de divisão de nós da raiz até o nó k. O nó raiz é associado a (H, V).

Para dividir o nó k, primeiro todos os atributos em H^F são percorridos, a fim de escolher o melhor teste horizontal. Aplica-se a heurística de redução de variância, definida pela Equação 4.1 (a mesma utilizada em PCTs convencionais), a H^T, visando avaliar a qualidade de cada divisão. Como podem ter havido divisões verticais anteriormente, responsáveis por dividir o espaço de funções, é preciso restringir o vetor de rótulos $\mathbf{v}_i$ aos objetos que estão em V_k. Denotamos então $\mathbf{v}[Z_k]$, em que Z_k corresponde aos índices das linhas de V_k (lembrando que ${H^T}_k$ é a matriz transposta de ${V^T}_k$).

Uma vez que trata-se de uma tarefa hierárquica, os pesos de cada classe são levados em consideração, como exposto anteriormente na Equação 4.2. Isso resulta na seguinte definição de variância:

$$Var(H_k, V_k) = \frac{\sum_i d(\mathbf{v}_i[Z_k], \overline{\mathbf{v}[Z_k]})^2}{|H_k|} \text{ with } \mathbf{v}_i \in H_k \tag{5.1}$$

onde d é a distância entre cada vetor de rótulos $\mathbf{v}_i$ e o vetor-protótipo do conjunto, $\overline{\mathbf{v}}$.

Assim, a função heurística para divisões horizontais é dada por:

$$h_h(s, H_k, V_k) = Var(H_k, V_k) - \left(\frac{|H_{kL}|}{|H_k|} \cdot Var(H_{kL}, V_k) + \frac{|H_{kR}|}{|H_k|} \cdot Var(H_{kR}, V_k) \right) \tag{5.2}$$

Na equação 5.2, L e R referem aos nós filhos esquerdo e direito criados para o nó k, após aplicar a divisão horizontal s.

Em uma PBCT regular [Pliakos, Geurts e Vens 2018], o mesmo procedimento seria aplicado aos atributos em V^F, e então o melhor teste geral seria selecionado para dividir o nó. No entanto, como o objetivo deste projeto é aplicar HMC, a redução de variância em V^T não é indicativo de que *clusters* com objetos mais fortemente relacionados estão sendo criados. Ao invés de medir diretamente a qualidade de uma divisão vertical, é preciso garantir que a escolha de uma divisão vertical beneficiará de fato o particionamento do espaço de objetos, já que o objetivo é fazer predições para novas (desconhecidas) linhas. Assim, o algoritmo percorre cada valor de atributo categórico em V^F, usando uma abordagem *lookahead* [Elomaa e Malinen 2003] (ilustrada na Figura 9): Para cada teste possível em V^F, a divisão vertical é simulada, assim como a próxima divisão horizontal (se houver) em ambos os nós filhos resultantes (H_k, V_{kL}) e (H_k, V_{kR}) de (H_k, V_k). Isso significa que não são feitas duas divisões verticais consecutivas, para evitar subconjuntos muito

pequenos e, consequentemente, um *overfitting* do modelo. Desta forma, a função heurística para divisões verticais é baseada no melhor valor da heurística de divisões horizontais subsequentemente aplicadas. Por razões computacionais, o *lookahead* só é aplicado até a profundidade um. Isso resulta na seguinte função para avaliar a qualidade de uma divisão vertical s:

$$h_v(s, H_k, V_k) = \frac{|V_{kL}|}{|V_k|} \cdot h_h(s_L, H_k, V_{kL}) + \frac{|V_{kR}|}{|V_k|} \cdot h_h(s_R, H_k, V_{kR}) \tag{5.3}$$

Divisões s_L e s_R são escolhidas para maximizar os valores de heurística horizontal nos nós filhos esquerdo e direito de k, respectivamente. A divisão s escolhida para dividir o nó k é aquela que fornece o maior valor geral para Equação 5.2 ou Equação 5.3.

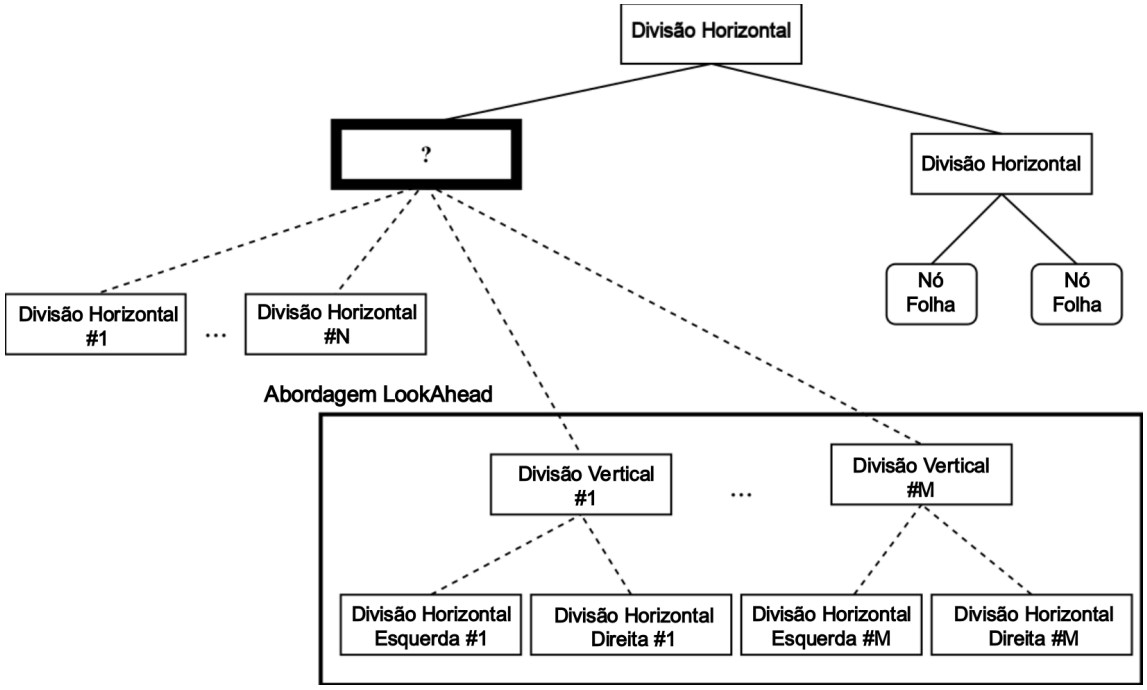


Figura 9 – Ilustração da abordagem *lookahead*.

Antes de aplicar a divisão, um F-test é usado para verificar se a redução de variância induzida pela divisão é estatisticamente significativa. Se a redução não for significativa, um nó-folha é criado. Senão, o teste é incluído na árvore e os subconjuntos de H_k ou V_k são criados para formar novos *bi-clusters*. A indução é recursivamente chamada até que um critério de parada seja atingido. Quando uma divisão vertical é incluída, as duas divisões horizontais subsequentes também são incluídas, isto é, uma divisão vertical leva a seis novos nós, ao invés de dois. Dois nós são filhos diretos do nó onde foi aplicada a divisão vertical, e cada um desses filhos fará uma divisão horizontal, gerando mais dois nós cada.

Cada nó-folha recebe um vetor protótipo. A Figura 10 apresenta um pequeno exemplo da árvore resultante do nosso método, usando a hierarquia de rótulos apresentada na Figura 8.

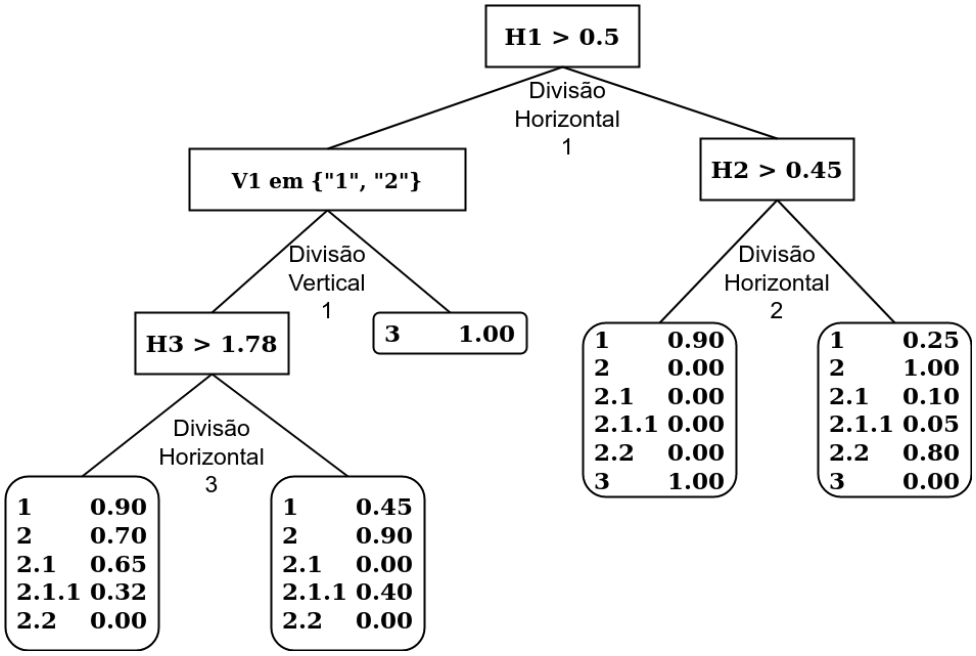


Figura 10 – Ilustração de uma árvore PBCT-HMC. H_n é o n-ésimo atributo de H^F, e V_1 é o (único) atributo de V^F.

5.4 Critério de Parada

Quanto menor o vetor protótipo (depois das divisões verticais), mais fácil fica existir uma divisão horizontal subsequente estatisticamente significativa, usando um nível de significância fixo l_0. Isso resulta na árvore fazendo mais divisões do que o necessário e levando a um *overfitting*. Por esse motivo, uma correção no nível de significância do F-test é aplicada: Ao verificar a significância de uma divisão em um nó k definido por (H_k, V_k), $l = l_0 \times |V_k|/|V|$ é usado como nível de significância. Em outras palavras, o critério de significância se torna mais rígido à medida que os vetores de rótulos se tornam menores.

5.5 Fazendo Predições

Após a construção da árvore, é possível inserir um conjunto de testes e obter o vetor de probabilidades para cada objeto de teste. Para fazer isso, cada objeto de teste é classificado na árvore, começando com o nó raiz. Sempre que uma divisão horizontal é encontrada, um dos nós filhos é seguido, de acordo com o resultado do teste. Sempre que uma divisão vertical é encontrada, no entanto, os dois nós filhos são seguidos, pois é necessária uma predição para todos os rótulos. Como tal, o objeto de teste pode terminar em vários nós folha. A predição final é construída concatenando os protótipos desses nós folha. A Figura 11 ilustra o procedimento de predição.

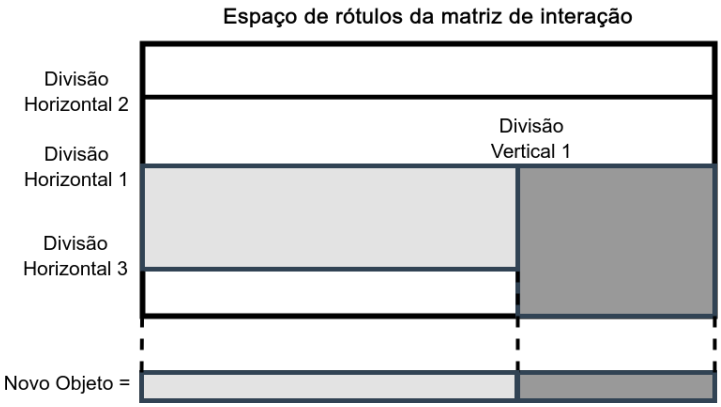


Figura 11 – Ilustração de um procedimento de predição para a árvore PBCT-HMC construída.

5.6 Pseudocódigo

O pseudocódigo da indução da árvore do PBCT-HMC proposto é apresentado no Algoritmo 1. Os principais aspectos são destacados a seguir:

- Linha 2: Cálculo da heurística horizontal (Equação 5.2).
- Linha 3: Cálculo da heurística vertical (Equação 5.3).
- Linha 4: Critério de parada do algoritmo de indução (Seção 5.4).
- Linha 7: Divisão horizontal.
- Linha 13: Divisão vertical.
- Linhas 8 e 14: Chamadas recursivas do algoritmo de indução.
- Linha 18: Geração de um nó-folha.

Algoritmo 1: PBCT-HMC: Indução Árvore

```
   Entrada: Dados H, Dados V, Nó raiz
 1 inicio
 2     Calcula heurísticaHorizontal a partir de H;
 3     Calcula heurísticaVertical a partir de V;
 4     enquanto Não atingir critério de parada faça
 5         se heurísticaHorizontal >= heurísticaVertical então
 6             para i em 0..2 faça
 7                 Faz divisão horizontal de raiz, armazena em nóFilho;
 8                 Chamada recursiva Indução Árvore(subconjuntoH[i], V, nóFilho);
 9             fim
10         senão
11             Ajusta F-Test;
12             para i em 0..2 faça
13                 Faz divisão vertical de raiz, armazena em nóFilho;
14                 Chamada recursiva Indução Árvore(H, subconjuntoV[i], nóFilho);
15             fim
16         fim
17     fim
18     Inclui nó-folha;
19 fim
```

6 Experimentos e Resultados

Os dados utilizados nos experimentos vêm do Functional Catalogue (FunCat), um sistema de classificação hierarquicamente estruturado que permite a descrição funcional de proteínas de praticamente qualquer organismo. A taxonomia FunCat pode ser aplicada para a anotação manual de procariontes, fungos, plantas e animais. O esquema de anotação do FunCat cobre funções como transporte celular, metabolismo e regulação da atividade de proteínas (a Tabela 6 mostra as principais categorias do FunCat) [A Zollner A 2004]. No total, a versão 2.1 do FunCat inclui 1362 categorias funcionais[1]. Cada um dos principais ramos funcionais é organizado como uma estrutura hierárquica semelhante a uma árvore. Este conceito básico foi mantido desde a anotação do genoma da levedura e provou ser adequado para a anotação de outros genomas. O FunCat fornece um esquema de anotação geral e estável e serve como um ambiente de recuperação de banco de dados.

Tabela 6 – Principais categorias do FunCat.

Metabolismo	
01	Metabolismo
02	Energia
04	Armazenamento
Vias de Informação	
11	Transcrição
12	Síntese de Proteínas
18	Regulação da atividade de proteínas
Transporte	
20	Transporte celular, facilitação de transportes e vias de transporte
Percepção e Resposta aos Estímulos	
32	Resgate celular, defesa e virulência
34	Interação com o ambiente celular
38	Elementos transponíveis, virais e plasmídicas
Processos de desenvolvimento	
40	Destino celular
42	Biogênese de componentes celulares
43	Diferenciação do tipo de célula
Localização	
70	Localização subcelular
73	Localização do tipo de célula
75	Localização de tecido
77	Localização de órgãos
Proteínas experimentalmente não caracterizadas	
98	Classificação ainda não bem definida
99	Proteínas não classificadas

No total, foram utilizados 16 conjuntos de dados de HMC, do campo da genômica

[1] www.helmholtz-muenchen.de/ibis/resourcesservices/genomics/funcat-the-functional-catalogue/index.html

funcional [Clare e King 2003], cuja hierarquia contém seis níveis. Esses conjuntos de dados estão disponíveis em duas versões: a versão de 2007[2] foi usada na publicação original do Clus-HMC [Vens et al. 2008] e a versão de 2018[3] vêm de um estudo recente que atualizou os rótulos de classes [Nakano, Lietaert e Vens 2019].

A Tabela 7 fornece um resumo dos conjuntos de dados usados. A seguir, são apresentados o número de objetos em cada subconjunto (Treino, Validação e Teste), o número de atributos de cada tipo (Categórico e Numérico) e o número de rótulos por nível da hierarquia.

Tabela 7 – Resumo dos dados.

	Treino	Validação	Teste	Categóricos	Numéricos	L1	L2	L3	L4	L5	L6
Cellcycle2007	1628	848	1281	0	77	18	80	178	142	77	4
Derisi2007	1608	842	1275	0	63	18	80	178	142	77	4
Eisen2007	1058	529	837	0	79	18	76	165	131	67	4
Expr2007	1639	849	1291	4	547	18	80	178	142	77	4
Gasch1_2007	1634	846	1284	0	173	18	80	178	142	77	4
Gasch2_2007	1639	849	1291	0	52	18	80	178	142	77	4
Seq2007	1701	879	1339	5	473	18	80	178	142	77	4
Spo2007	1600	837	1266	3	77	18	80	178	142	77	4
Cellcycle2018	1628	848	1281	0	77	20	86	210	171	92	6
Derisi2018	1608	842	1275	0	63	20	86	210	171	92	6
Eisen2018	1058	529	837	0	79	19	84	201	159	83	6
Expr2018	1639	849	1291	4	547	20	86	210	171	92	6
Gasch1_2018	1634	846	1284	0	173	20	86	210	171	92	6
Gasch2_2018	1639	849	1291	0	52	20	86	210	171	92	6
Seq2018	1701	879	1339	5	473	20	86	210	171	93	6
Spo2018	1600	837	1266	3	77	20	86	210	171	92	6

Conforme descrito na Seção 5.2, o primeiro procedimento foi gerar os conjuntos de dados complementares H e V. Então os algoritmos foram ajustados, encontrando o F-test ótimo. Os valores considerados foram 0.001, 0.005, 0.01, 0.05, 0.1, 0.125 para cada conjunto de dados. Esse ajuste foi realizado em um conjunto de validação. Em seguida, o conjunto de treinamento mais o conjunto de validação são combinados para formar o conjunto de treinamento final, que foi usado para executar os dois algoritmos (PBCT-HMC e Clus-HMC) com o nível de significância agora ótimo e o número mínimo fixo de objetos por nó-folha igual a 5.

A medida de avaliação considerada neste trabalho é a *Pooled Area Under Precision-Recall Curve* (Pooled AUPRC). Essa medida é globalmente adotada na literatura de HMC. Como geralmente dados HMC são muito desbalanceados, a porcentagem de classificações negativas para as classes menos frequentes tende a ser muito maior que a porcentagem de classificações positivas, podendo levar a falsos negativos. Portanto, utilizar apenas o

[2] <https://dtai.cs.kuleuven.be/clus/hmcdatasets/>
[3] <https://bit.ly/2YtCS1l>

AUPRC não é recomendado [Nakano, Lietaert e Vens 2019]. Pooled AUPRC corresponde à área ponderada abaixo da curva de precisão-revocação. Essa curva é gerada tomando-se a precisão e a revocação ponderadas de cada classe, para diferentes limiares. Esses limiares variam de 0 a 1 e são incrementados em passos de tamanho 0.02. Quanto mais próximo o valor estiver de 1, melhor o modelo é considerado.

As Equações 6.1 e 6.2 descrevem o cálculo de precisão e de revocação, respectivamente. A precisão busca identificar a proporção de classificações positivas que estão de fato corretas. A revocação busca identificar a proporção de verdadeiros positivos que foram identificados corretamente. Nas equações, TP refere-se à quantidade de verdadeiros positivos, FP refere-se à quantidade de falsos positivos, e FN, à quantidade de falsos negativos.

$$Precisao = \frac{TP}{TP + FP} \tag{6.1}$$

$$Revocacao = \frac{TP}{TP + FN} \tag{6.2}$$

Os resultados dos experimentos são apresentados na Tabela 8, onde os melhores valores estão destacados em negrito. As Tabelas 9 e 10 apresentam o número de nós e o tempo de indução dos algoritmos, respectivamente.

Tabela 8 – Pooled AUPRC usando valores ótimos para o F-test.

	PBCT-HMC	Clus-HMC
Cellcycle2007	0.165	**0.172**
Derisi2007	**0.177**	0.175
Eisen2007	0.195	**0.205**
Expr2007	0.207	**0.210**
Gasch1_2007	**0.206**	0.205
Gasch2_2007	0.189	**0.195**
Seq2007	0.191	0.211
Spo2007	0.184	**0.186**
Cellcycle2018	**0.192**	**0.192**
Derisi2018	0.192	**0.195**
Eisen2018	0.218	**0.229**
Expr2018	0.216	**0.218**
Gasch1_2018	**0.212**	**0.212**
Gasch2_2018	**0.205**	**0.205**
Seq2018	0.219	**0.229**
Spo2018	**0.208**	0.205
Média	0.197	**0.202**

De acordo com a Tabela 8, o PBCT-HMC obteve maior ou igual valor de Pooled AUPRC em 6 dos 16 conjuntos de dados, em comparação ao Clus-HMC. Para uma com-

Tabela 9 – Tamanho do modelo: Nós (Nós-Folha).

	PBCT-HMC	Clus-HMC
Cellcycle2007	45 (23)	**41 (21)**
Derisi2007	23 (12)	**7 (4)**
Eisen2007	209 (105)	**11 (6)**
Expr2007	163 (82)	**75 (38)**
Gasch1_007	73 (37)	**67 (34)**
Gasch2_007	**31 (16)**	53 (27)
Seq2007	269 (135)	**95 (48)**
Spo2007	25 (13)	**11 (6)**
Cellcycle2018	63 (32)	**35 (18)**
Derisi2018	**15 (8)**	19 (10)
Eisen2018	**33 (17)**	55 (28)
Expr2018	119 (60)	**19 (10)**
Gasch1_018	153 (77)	**81 (41)**
Gasch2_018	81 (41)	**41 (21)**
Seq2018	153 (77)	**23 (12)**
Spo2018	**25 (13)**	39 (20)
Média	91 (47)	**41 (22)**

Tabela 10 – Tempo de indução do modelo, em segundos.

	PBCT-HMC	Clus-HMC
Cellcycle2007	10.64	**2.62**
Derisi2007	8.97	**1.6**
Eisen2007	15.22	**4.26**
Expr2007	94.61	**29.74**
Gasch1_2007	29.17	**20.13**
Gasch2_2007	4.17	**3.1**
Seq2007	75.56	**20.09**
Spo2007	14.56	**2.41**
Cellcycle2018	18.29	**3.05**
Derisi2018	10.47	**2.98**
Eisen2018	7.75	**2.17**
Expr2018	93.41	**18.09**
Gasch1_2018	25.82	**8.85**
Gasch2_2018	24.07	**10.98**
Seq2018	37.52	**17.37**
Spo2018	17.89	**4.37**
Média	30.51	**9.49**

paração ainda melhor entre os resultados do PBCT-HMC e do Clus-HMC, foi executado o teste de Friedman [Friedman 1937]. O teste de Friedman é um teste estatístico não paramétrico, semelhante ao ANOVA, utilizado para identificar diferenças estatisticamente significantes entre um conjunto de observações. Quanto menor o *p-value* obtido, maior a certeza em relação à significância das diferenças estatísticas. Para os experimentos aqui descritos, foi obtido um valor de *p-value* igual a 0.0522. Traçando-se o diagrama crítico (Figura 12) a partir do teste *post-hoc* de Nemenyi [Nemenyi 1963], é possível constatar que PBCT-HMC e Clus-HMC não apresentam diferença estatisticamente significante (representada por algoritmos conectados por uma mesma linha no diagrama).

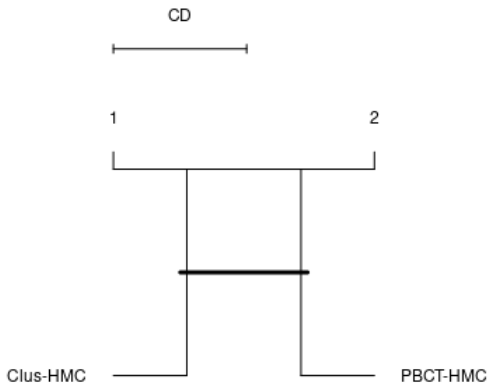


Figura 12 – Teste Friedman-Nemenyi comparando Pooled AUPRC para os dois algoritmos.

Levando-se em consideração o tamanho do modelo (Tabela 9), Clus-HMC induziu modelos consideravelmente menores. Esse resultado já era esperado, uma vez que o Clus-HMC faz somente divisões horizontais e, para o PBCT-HMC, cada divisão vertical levou a seis novos nós comparado com apenas dois nós para as divisões horizontais.

Em relação ao tempo de indução, o PBCT-HMC leva mais tempo para induzir a árvore e predizer as funções proteicas do que o Clus-HMC devido a alguns fatores, como a estratégia *lookahead*. O desempenho do tempo de indução depende do número de classes do conjunto de dados V. Além disso, outro fator é o número de nós de cada árvore, que deve ser maior para o PBCT-HMC do que para o Clus-HMC, pois o primeiro faz não apenas divisões horizontais, mas também divisões verticais, conforme explicado anteriormente.

7 Conclusão

Neste trabalho foi proposto um modelo preditivo que faz uso de *Predictive Bi-Clustering Trees* para classificação hierárquica multirrótulo (HMC) de funções de proteínas. Ao contrário das abordagens tradicionais para HMC, o modelo proposto, denominado PBCT-HMC, particiona automaticamente o espaço de rótulos durante seu processo de indução. Para conseguir isso, uma abordagem de *lookahead* é incorporada na construção da árvore, de modo que uma partição de espaço de rótulos seja introduzida, se levar a uma melhor partição de espaço de objetos em um nível mais profundo da árvore.

Experimentos demonstraram que o PBCT-HMC obteve resultados competitivos em comparação ao seu concorrente Clus-HMC, sem diferença estatisticamente significante.

Em domínios biológicos, como é o caso da tarefa de predição de funções de proteínas, a interpretabilidade de um modelo é um fator relevante. Nesse sentido, *bi-clusters* podem prover subconjuntos mais específicos e com maior correlação entre os objetos/rótulos em comparação com *clusters* regulares.

Esse trabalho foi estendido para outros domínios de HMC além das funções de proteínas. Os resultados obtidos foram submetidos em forma de artigo para a Conferência Europeia "*on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*" (ECML-PKDD), *qualis* A1. O artigo foi aceito e tem previsão de publicação para setembro de 2020.

Trabalhos futuros podem abordar a aplicação do PBCT-HMC em outros domínios de dados, especialmente conjuntos de dados cuja hierarquia de rótulos está estruturada como GAD, e também a aplicação de nossa abordagem para tarefas de classificação não hierárquica multirrótulo, onde o maior desafio é encontrar uma representação adequada de atributos para o espaço de rótulos. Por fim, pretende-se estudar novas abordagens para representação do espaço de atributos das funções.

Referências

A ZOLLNER A, M. D. R. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic Acids Research*, 2004. Citado na página 49.

BARANIUK, R. et al. Model-based compressive sensing. *IEEE Transactions on Information Theory*, v. 56, n. 4, p. 1982–2001, April 2010. Citado na página 34.

BI, W.; KWOK, J. Mandatory leaf node prediction in hierarchical multilabel classification. *IEEE Transactions on Neural Networks and Learning Systems*, v. 25, n. 12, p. 2275–2287, Dec 2014. Citado na página 34.

BOBADILLA, J. et al. Recommender systems survey. *Knowledge-Based Systems*, v. 46, p. 109 – 132, 2013. Citado na página 35.

BORGES, H.; NIEVOLA, J. Multi-label hierarchical classification using a competitive neural network for protein function prediction. In: *International Joint Conference on Neural Networks*. [S.l.: s.n.], 2012. p. 1–8. Citado na página 33.

CERRI, R. et al. Reduction strategies for hierarchical multi-label classification in protein function prediction. *BMC Bioinformatics*, v. 17, n. 1, p. 373, 2016. ISSN 1471-2105. Citado na página 34.

CESA-BIANCHI, N.; RE, M.; VALENTINI, G. Synergy of multi-label hierarchical ensembles, data fusion, and cost-sensitive methods for gene functional inference. *Machine Learning*, Springer Netherlands, p. 1–33, 2011. ISSN 0885-6125. Citado na página 33.

CESA-BIANCHI, N.; VALENTINI, G. Hierarchical cost-sensitive algorithms for genome-wide gene function prediction. *Journal of Machine Learning Research*, v. 8, p. 14–29, 2010. Citado na página 33.

CHENG, W.; H'ULLERMEIER, E. Combining instance-based learning and logistic regression for multilabel classification (resubmission). *LWA 2009 - Workshop-Woche: Lernen-Wissen-Adaptivitat - Learning, Knowledge, and Adaptivity*, p. 22–29, 01 2009. Citado na página 26.

CHERMAN, E. A.; MONARD, M. C.; METZ, J. Multi-label problem transformation methods: a case study. *CLEI Electron. J.*, v. 14, 2011. Citado 2 vezes nas páginas 24 e 25.

CLARE, A.; KING, R. D. Knowledge discovery in multi-label phenotype data. In: RAEDT, L. D.; SIEBES, A. (Ed.). *Principles of Data Mining and Knowledge Discovery*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. p. 42–53. Citado na página 25.

CLARE, A.; KING, R. D. Predicting gene function in saccharomyces cerevisiae. *Bioinformatics*, v. 19, n. suppl2, p. ii42–ii49, 2003. Citado na página 50.

COSTA, E. P. et al. Comparing several approaches for hierarchical classification of proteins with decision trees. *Advances in Bioinformatics and Computational Biology:*

Second Brazilian Symposium on Bioinformatics, BSB 2007, Angra dos Reis, Brazil, August 29-31, 2007. Proceedings, Springer Berlin Heidelberg, p. 126–137, 2007. Citado na página 19.

DIETTERICH, T. G. Ensemble methods in machine learning. In: *Multiple Classifier Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. p. 1–15. Citado na página 27.

DING, Y.; TANG, J.; GUO, F. Predicting protein-protein interactions via multivariate mutual information of protein sequences. *BMC Bioinformatics*, v. 17, n. 1, p. 398, 2016. ISSN 1471-2105. Disponível em: <http://dx.doi.org/10.1186/s12859-016-1253-9>. Citado na página 20.

ELISSEEFF, A.; WESTON, J. A kernel method for multi-labelled classification. In: *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*. Cambridge, MA, USA: MIT Press, 2001. (NIPS01), p. 681687. Citado na página 26.

ELKAFRAWY, P.; MAUSAD, A.; ESMAIL, H. Experimental comparison of methods for multi-label classification in different application domains. *International Journal of Computer Applications*, v. 114, p. 1–9, 03 2015. Citado na página 24.

ELOMAA, T.; MALINEN, T. On lookahead heuristics in decision tree learning. *Foundations of Intelligent Systems: 14th International Symposium, ISMIS 2003, Maebashi City, Japan, October 28-31, 2003. Proceedings*, Springer Berlin Heidelberg, Berlin, Heidelberg, p. 445–453, 2003. Citado na página 44.

FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. [S.l.]: LTC, 2011. ISBN 9788521618805. Citado na página 30.

FIELDS, O.-k. S. S. A novel genetic system to detect protein-protein interactions. *Nature*, v. 340, 1989. Citado na página 35.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, Taylor & Francis, v. 32, n. 200, p. 675–701, 1937. Citado na página 53.

HAMP, T.; ROST, B. Evolutionary profiles improve proteinprotein interaction prediction from sequence. *Bioinformatics*, v. 31, n. 12, p. 1945–1950, 2015. Citado na página 35.

JEFFERY, C. J. Protein moonlighting: what is it, and why is it important? *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, v. 373, n. 1738, Jan 2018. Citado na página 31.

KAMADA M., S.-Y. H. M. A. T. Prediction of protein-protein interaction strength using domain features with supervised regression. *The Scientific World Journal*, 2014. Citado na página 35.

KOCEV, D. et al. Ensembles of multi-objective decision trees. In: *ECML*. [S.l.: s.n.], 2007. Citado na página 27.

KOLLER, D.; SAHAMI, M. Hierarchically classifying documents using very few words. In: *Proceedings of the 14th International Conference on Machine Learning (ICML)*. Nashville, Tennessee: [s.n.], 1997. p. 170–178. Citado na página 29.

LETOVSKY, S.; KASIF, S. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, v. 19, p. i197, 2003. Citado 2 vezes nas páginas 19 e 20.

LI, H. C. X. Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach. *Decision Support Systems*, v. 54, p. 880–890, 2013. Citado na página 35.

LIBEN-NOWELL, D.; KLEINBERG, J. The link prediction problem for social networks. In: *Proceedings of the Twelfth International Conference on Information and Knowledge Management*. [S.l.: s.n.], 2003. p. 556–559. Citado na página 35.

LÜ, L. et al. Toward link predictability of complex networks. *Proceedings of the National Academy of Sciences*, National Academy of Sciences, v. 112, n. 8, p. 2325–2330, 2015. Citado na página 35.

MADJAROV G., D. T. D. I. G. D. Evaluation of differentdata-derived label hierarchies in multi-label classification. In: *Proc. of the 3rd International Workshop on New Frontiers in Mining Complex Patterns held in conjunction with ECML/PKDD2014*. [S.l.: s.n.], 2014. p. 124135. Citado na página 30.

Min-Ling Zhang; Zhi-Hua Zhou. A k-nearest neighbor based algorithm for multi-label classification. In: *2005 IEEE International Conference on Granular Computing*. [S.l.: s.n.], 2005. v. 2, p. 718–721 Vol. 2. Citado na página 25.

MITCHELL, T. M. *Machine Learning*. 1. ed. USA: McGraw-Hill, Inc., 1997. ISBN 0070428077. Citado 2 vezes nas páginas 37 e 38.

MOYANO, J. M. et al. Review of ensembles of multi-label classifiers: Models, experimental study and prospects. *Information Fusion*, v. 44, p. 33 – 45, 2018. ISSN 1566-2535. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1566253517307169>. Citado na página 26.

NAKANO, F. K.; LIETAERT, M.; VENS, C. Machine learning for discovering missing or wrong protein function annotations. *BMC bioinformatics*, Springer, v. 20, n. 1, p. 485, 2019. Citado 2 vezes nas páginas 50 e 51.

NEMENYI, P. *Distribution-free Multiple Comparisons*. Princeton University, 1963. Disponível em: <https://books.google.com.br/books?id=nhDMtgAACAAJ>. Citado na página 53.

O. NUSSINOV R., G. A. K. Prism: Protein-protein interaction prediction by structural matching. *Methods in molecular biology*, n. 484, p. 505–521, 2008. Citado na página 35.

PAKRASHI, A.; NAMEE, B. M. KFHE-HOMER: kalman filter-based heuristic ensemble of HOMER for multi-label classification. *CoRR*, abs/1904.10552, 2019. Citado na página 27.

PLIAKOS, K.; GEURTS, P.; VENS, C. Global multi-output decision trees for interaction prediction. *Machine Learning*, v. 107, n. 8-10, p. 1257–1281, 2018. Disponível em: <https://doi.org/10.1007/s10994-018-5700-x>. Citado na página 20.

PLIAKOS, K.; GEURTS, P.; VENS, C. Global multi-output decision trees for interaction prediction. *Machine Learning*, v. 107, n. 8, p. 1257–1281, Sep 2018. ISSN 1573-0565. Citado 2 vezes nas páginas 35 e 44.

PLIAKOS, K.; VENS, C.; TSOUMAKAS, G. Predicting drug-target interactions with multi-label classification and label partitioning. *IEEE-ACM Transactions On Computational Biology And Bioinformatics*, p. 1–11, 2019. Citado na página 41.

QUINLAN, J. R. Induction of decision trees. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 1, n. 1, p. 81106, mar. 1986. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1022643204877>. Citado na página 38.

QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Citado na página 38.

READ, J.; PFAHRINGER, B.; HOLMES, G. Multi-label classification using ensembles of pruned sets. In: *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*. USA: IEEE Computer Society, 2008. (ICDM 08), p. 9951000. ISBN 9780769535029. Disponível em: <https://doi.org/10.1109/ICDM.2008.74>. Citado 2 vezes nas páginas 26 e 27.

READ, J. et al. Classifier chains for multi-label classification. In: BUNTINE, W. et al. (Ed.). *Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 254–269. ISBN 978-3-642-04174-7. Citado 2 vezes nas páginas 26 e 27.

ROBERTS, R. J. Identifying protein functiona call for community action. *PLOS Biology*, Public Library of Science, v. 2, n. 3, 03 2004. Citado na página 19.

SCHIETGAT, L. et al. Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics*, v. 11, p. 2, 2010. Citado na página 33.

SHI, C. et al. Multi-label ensemble learning. In: . [S.l.: s.n.], 2011. v. 6913, p. 223–239. Citado na página 26.

SILLA, C.; FREITAS, A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, Springer Netherlands, v. 22, p. 31–72, 2010. ISSN 1384-5810. Citado 3 vezes nas páginas 28, 29 e 30.

STOJANOVA, D. et al. Using ppi network autocorrelation in hierarchical multi-label classification trees for gene function prediction. *BMC Bioinformatics*, v. 14, n. 1, p. 285, 2013. Citado na página 33.

SUN, T. et al. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. *BMC Bioinformatics*, v. 18, p. 277, 2017. Citado na página 20.

SUN, Z. et al. Hierarchical multilabel classification with optimal path prediction. *Neural Processing Letters*, p. 1–15, 2016. Citado na página 34.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. USA: Addison-Wesley Longman Publishing Co., Inc., 2005. ISBN 0321321367. Citado na página 23.

Thabtah, F. A.; Cowling, P.; Yonghong Peng. Mmac: a new multi-class, multi-label associative classification approach. In: *Fourth IEEE International Conference on Data Mining (ICDM'04)*. [S.l.: s.n.], 2004. p. 217–224. Citado na página 26.

TRIGUERO, I.; VENS, C. Labelling strategies for hierarchical multi-label classification techniques. *Pattern Recognition*, Elsevier Science Inc., New York, NY, USA, v. 56, n. C, p. 170–183, ago. 2016. ISSN 0031-3203. Citado na página 34.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Effective and efficient multilabel classification in domains with large number of labels. 01 2008. Citado na página 27.

TSOUMAKAS, G.; VLAHAVAS, I. Random k-labelsets: An ensemble method for multilabel classification. In: KOK, J. N. et al. (Ed.). *Machine Learning: ECML 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. p. 406–417. ISBN 978-3-540-74958-5. Citado na página 27.

VALENTINI, G. True path rule hierarchical ensembles. In: *International Workshop on Multiple Classifier Systems*. [S.l.: s.n.], 2009. p. 232–241. ISBN 978-3-642-02325-5. Citado na página 33.

VALENTINI, G. True path rule hierarchical ensembles for genome-wide gene function prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, IEEE Computer Society Press, Los Alamitos, CA, USA, v. 8, n. 3, p. 832–847, maio 2011. ISSN 1545-5963. Citado na página 33.

VALENTINI, G.; RE, M. Weighted true path rule: a multilabel hierarchical algorithm for gene function prediction. In: *1st Workshop on Learning from Multi-Label Data (MLD) held in conjunction with ECML/PKDD*. [S.l.: s.n.], 2009. p. 132–145. Citado na página 33.

VENS, C. et al. Decision trees for hierarchical multi-label classification. *Machine Learning*, v. 73, n. 2, p. 185, 2008. ISSN 1573-0565. Citado 6 vezes nas páginas 20, 22, 33, 38, 39 e 50.

WANG H., H. H.; DING, C. Function-function correlated multi-label protein function prediction over interaction networks. *J Comput Biol*, 2013. Citado na página 20.

YOU, Z.-h.; CHAN, K. C. C.; HU, P. Predicting protein-protein interactions from primary protein sequences using a novel multi-scale local feature representation scheme and the random forest. *PLoS ONE*, v. 10, p. 5, 2015. Citado na página 35.

YU, G.; ZHU, H.; DOMENICONI, C. Predicting protein functions using incomplete hierarchical labels. *BMC Bioinformatics*, v. 16, n. 1, p. 1, 2015. Citado na página 34.

ZHANG W., C. Y. L. F. L. F. T. G.; LI, X. Predicting potential drug-drug interactions by integrating chemical, biological, phenotypic and network data. *BMC Bioinformatics*, v. 18, n. 1, p. 18, 2017. Citado na página 35.

ZIMEK, A. et al. A study of hierarchical and flat classification of proteins. IEEE/ACM Trans Comput Biol Bioinform, v. 7, n. 3, p. 563–71, 2010. Citado na página 29.