

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ  
ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**

**LUIZ RAFAEL SCHMITKE**

**ALGORITMO PARA CLASSIFICAÇÃO MULTIRRÓTULO BASEADO EM  
BICLUSTERIZAÇÃO**

**CURITIBA**

**2022**

**LUIZ RAFAEL SCHMITKE**

**ALGORITMO PARA CLASSIFICAÇÃO MULTIRRÓTULO BASEADO EM  
BICLUSTERIZAÇÃO**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica do Paraná, como requisito à obtenção do título de Doutor em Informática.

Área de Concentração: *Ciência da Computação*

Orientador: Prof. Dr. Júlio C. Nievola  
Coorientador: Prof. Dr. Emerson C. Paraiso

**CURITIBA**

**2022**

Dados da Catalogação na Publicação  
Pontifícia Universidade Católica do Paraná  
Sistema Integrado de Bibliotecas – SIBI/PUCPR  
Biblioteca Central  
Pamela Travassos de Freitas CRB/9 1960

S355a  
2022

Schmitke, Luiz Rafael  
Algoritmo para classificação multirrótulo baseado em biclusterização / Luiz Rafael Schmitke ; orientador: Júlio C. Nievola ; coorientador: Emerson C. Paraiso. –2022.  
122 f. : il. ; 30 cm

Tese (doutorado) – Pontifícia Universidade Católica do Paraná, Curitiba, 2022  
Bibliografia: f. 119-122

1. Informática. 2. Aprendizado do computador. 3. Algoritmos. 4. Mineração de dados (Computação). 5. Otimização combinatória. I. Nievola, Júlio César. II. Paraiso, Emerson Cabrera. III. Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática. IV. Título.

CDD 20. ed. – 004



Pontifícia Universidade Católica do Paraná  
Escola Politécnica  
Programa de Pós-Graduação em Informática

58-2022


### DECLARAÇÃO

Declaro para os devidos fins que o aluno **LUIZ RAFAEL SCHMITKE**, defendeu a tese de Doutorado intitulada “**Algoritmo para Classificação Multirrótulo baseado em Biclusterização**”, na área de concentração Ciência da Computação, no dia 01 de junho de 2022, no qual foi aprovado.

Declaro ainda que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade, firmo a presente declaração.

Curitiba, 06 de setembro de 2022.

  
\_\_\_\_\_  
Prof. Dr. Emerson Cabrera Paraiso  
Coordenador do Programa de Pós-Graduação em Informática  
Pontifícia Universidade Católica do Paraná

## AGRADECIMENTOS

Primeiramente, agradeço a Deus por estar sempre comigo, principalmente durante o nascimento do meu filho, Joaquim, e na passagem pela pandemia do COVID-19.

Ao meu pai Luiz Carlos Schmitke, minha mãe Joclena do Rocio Schmitke e meu irmão Guilherme Schmitke que desde a minha graduação estiveram sempre me apoiando e me lembrando de manter o foco nos trabalhos, mas sem esquecer da família.

À minha esposa, Kelly Cristina de Castro Schmitke, e meu filho que sempre me apoiaram nesta caminhada e que no meio dela me mostraram que ser pai é trabalhoso, mas recompensante.

Aos meus professores da PUCPR que contribuíram para meu crescimento profissional, em especial os professores Julio Cesar Nievola e Emerson Cabrera Paraiso que me ajudaram durante o desenvolvimento do trabalho.

À professora Helyane Borges que abriu a porta do doutorado para mim, permitindo com que essa etapa fosse realizada.

Por fim, agradeço a todos aqueles que de uma forma ou outra contribuíram para o meu desenvolvimento pessoal e profissional, durante o desenvolvimento do trabalho, e aos grandes colegas que jamais serão esquecidos.

Muito obrigado a todos vocês.

## LISTA DE FIGURAS

Figura 1: Correlação entre Tempo Total de Execução (Treino + Teste) e Acurácia .....	15
Figura 2: Exemplo aprendido indutivo .....	20
Figura 3: Grau de supervisão nos conjuntos de dados .....	20
Figura 4: Representação do conjunto de exemplos em cada grau de supervisão .....	21
Figura 5: Exemplos de conjunto de dados monorrótulo.....	22
Figura 6: Exemplo de conjunto de dados multirrótulo .....	22
Figura 7: Taxonomia dos algoritmos de classificação multirrótulo. ....	23
Figura 8: Taxonomia dos algoritmos de transformação de problema multirrótulo.....	24
Figura 9: Micro exemplo de conjunto de dados multirrótulo.....	24
Figura 10: Conjunto de dados transformado pelo algoritmo BR .....	25
Figura 11: Conjunto de dados transformado pelo algoritmo CC .....	25
Figura 12: Conjunto de dados transformado pelo algoritmo RPC .....	26
Figura 13: Conjunto de dados transformado pelo algoritmo LP .....	27
Figura 14: Conjunto de dados transformado pelos algoritmos de seleção de rótulos .....	27
Figura 15: Métricas para avaliação de classificadores multirrótulo.....	29
Figura 16: Tipos de biclusters .....	35
Figura 17: Tipos de estrutura de biclusters.....	38
Figura 18: Exemplo de execução do algoritmo Bimax .....	39
Figura 19: Etapas da pesquisa .....	41
Figura 20: Etapas do método de transformação de problema .....	51
Figura 21: Categorização da manutenção das relações entre rótulos .....	52
Figura 22: Fluxo de criação dos subproblemas no BicbPT .....	54
Figura 23: Fluxo de classificação de um novo exemplo no BicbPT .....	55
Figura 24: Micro conjunto de dados para exemplificação do BicbPT .....	55
Figura 25: Exemplo de biclusters encontrados com o Bimax .....	55
Figura 26: Algoritmo da camada de independência .....	57
Figura 27: Exemplo de subproblemas gerados na camada de independência.....	57
Figura 28: Algoritmo da camada de biclusterização da versão BicbPT $\alpha$ .....	58
Figura 29: Exemplo de subproblemas gerados na camada de biclusterização no BicbPT $\alpha$ .....	59
Figura 30: Algoritmo da camada de biclusterização da versão BicbPT $\beta$ .....	59
Figura 31: Exemplo de subproblemas gerados na camada de biclusterização no BicbPT $\beta$ .....	60
Figura 32: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – JI com C4.5 .....	69

Figura 33: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – HL com C4.5.....	70
Figura 34: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – F1 <sub>micro</sub> com C4.5.....	71
Figura 35: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – F1 <sub>macro</sub> com C4.5 .....	73
Figura 36: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – TT com C4.5 .....	74
Figura 37: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – JI com SVM .....	76
Figura 38: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – HL com SVM.....	77
Figura 39: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – F1 <sub>micro</sub> com SVM.....	78
Figura 40: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – TT com SVM.....	80
Figura 41: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – JI com NB .....	81
Figura 42: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – HL com NB.....	82
Figura 43: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – F1 <sub>micro</sub> com NB.....	83
Figura 44: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – F1 <sub>macro</sub> com NB .....	84
Figura 45: Gráfico de Nemenyi BicbPT $\alpha$ x Todos – TT com NB .....	86
Figura 46: Gráfico de Nemenyi BicbPT $\beta$ x Todos – JI com C4.5 .....	87
Figura 47: Gráfico de Nemenyi BicbPT $\beta$ x Todos – HL com C4.5.....	88
Figura 48: Gráfico de Nemenyi BicbPT $\beta$ x Todos – F1 <sub>micro</sub> com C4.5.....	90
Figura 49: Gráfico de Nemenyi BicbPT $\beta$ x Todos – F1 <sub>macro</sub> com C4.5 .....	91
Figura 50: Gráfico de Nemenyi BicbPT $\beta$ x Todos – TT com C4.5 .....	93
Figura 51: Gráfico de Nemenyi BicbPT $\beta$ x Todos – JI com SVM .....	94
Figura 52: Gráfico de Nemenyi BicbPT $\beta$ x Todos – HL com SVM.....	95
Figura 53: Gráfico de Nemenyi BicbPT $\beta$ x Todos – F1 <sub>micro</sub> com SVM.....	97
Figura 54: Gráfico de Nemenyi BicbPT $\beta$ x Todos – F1 <sub>macro</sub> com SVM .....	98
Figura 55: Gráfico de Nemenyi BicbPT $\beta$ x Todos – TT com SVM .....	99
Figura 56: Gráfico de Nemenyi BicbPT $\beta$ x Todos – JI com NB .....	101
Figura 57: Gráfico de Nemenyi BicbPT $\beta$ x Todos – HL com NB.....	102
Figura 58: Gráfico de Nemenyi BicbPT $\beta$ x Todos – F1 <sub>micro</sub> com NB.....	103
Figura 59: Gráfico de Nemenyi BicbPT $\beta$ x Todos – F1 <sub>macro</sub> com NB .....	104
Figura 60: Gráfico de Nemenyi BicbPT $\beta$ x Todos – TT com NB .....	106
Figura 61: Gráfico da relação $\rho$ x Jaccard Index .....	110
Figura 62: Gráfico da relação $\rho$ x HL.....	111
Figura 63: Gráfico da relação $\rho$ x F1 <sub>micro</sub> .....	112
Figura 64: Gráfico da relação $\rho$ x F1 <sub>macro</sub> .....	112
Figura 65: Gráfico da relação $\rho$ x Tempo Total (treino + teste).....	113

## LISTA DE QUADROS E TABELAS

Quadro 1: Diferenças entre agrupamento tradicional e <i>biclustering</i> .....	34
Quadro 2: Exemplos de algoritmos de biclusterização .....	38
Quadro 3: Trabalhos relacionados e suas características.....	46
Quadro 4: Forma de manutenção das relações nos algoritmos de transformação de problema .....	53
Quadro 5: Exemplos de classificação no BicbPT $\alpha$ com limiar igual a 0.5 .....	62
Quadro 6: Exemplos de classificações no BicbPT $\beta$ com limiar igual a 0.5.....	62
Quadro 7: Lista de parâmetros para pré-processamento das bases de texto.....	65
Tabela 1: Resultados da busca de trabalhos relacionados .....	42
Tabela 2: Bases de dados utilizadas nos experimentos do BicbPT .....	43
Tabela 3: Bases de dados selecionadas para a realização de experimentos .....	65
Tabela 4: BicbPT $\alpha$ x Todos – JI com C4.5 .....	68
Tabela 5: BicbPT $\alpha$ x Todos – HL com C4.5 .....	70
Tabela 6: BicbPT $\alpha$ x Todos – F1 <sub>micro</sub> com C4.5.....	71
Tabela 7: BicbPT $\alpha$ x Todos – F1 <sub>macro</sub> com C4.5 .....	72
Tabela 8: BicbPT $\alpha$ x Todos – TT com C4.5 .....	74
Tabela 9: BicbPT $\alpha$ x Todos – JI com SVM .....	75
Tabela 10: BicbPT $\alpha$ x Todos – HL com SVM.....	76
Tabela 11: BicbPT $\alpha$ x Todos – F1 <sub>micro</sub> com SVM.....	77
Tabela 12: BicbPT $\alpha$ x Todos – F1 <sub>macro</sub> com SVM .....	78
Tabela 13: BicbPT $\alpha$ x Todos – TT com SVM .....	79
Tabela 14: BicbPT $\alpha$ x Todos – JI com NB.....	80
Tabela 15: BicbPT $\alpha$ x Todos – HL com NB .....	82
Tabela 16: BicbPT $\alpha$ x Todos – F1 <sub>micro</sub> com NB .....	83
Tabela 17: BicbPT $\alpha$ x Todos – F1 <sub>macro</sub> com NB.....	84
Tabela 18: BicbPT $\alpha$ x Todos – TT com NB.....	85
Tabela 19: BicbPT $\beta$ x Todos – JI com C4.5 .....	87
Tabela 20: BicbPT $\beta$ x Todos – HL com C4.5 .....	88
Tabela 21: BicbPT $\beta$ x Todos – F1 <sub>micro</sub> com C4.5.....	89
Tabela 22: BicbPT $\beta$ x Todos – F1 <sub>macro</sub> com C4.5 .....	90
Tabela 23: BicbPT $\beta$ x Todos – TT com C4.5 .....	92



Tabela 24: BicbPT $\beta$ x Todos – JI com SVM .....	94
Tabela 25: BicbPT $\beta$ x Todos – HL com SVM .....	95
Tabela 26: BicbPT $\beta$ x Todos – F1 <sub>micro</sub> com SVM .....	96
Tabela 27: BicbPT $\beta$ x Todos – F1 <sub>macro</sub> com SVM .....	97
Tabela 28: BicbPT $\beta$ x Todos – TT com SVM .....	98
Tabela 29: BicbPT $\beta$ x Todos – JI com NB.....	100
Tabela 30: BicbPT $\beta$ x Todos – HL com NB .....	102
Tabela 31: BicbPT $\beta$ x Todos – F1 <sub>micro</sub> com NB .....	103
Tabela 32: BicbPT $\beta$ x Todos – F1 <sub>macro</sub> com NB.....	104
Tabela 33: BicbPT $\beta$ x Todos – TT com NB.....	105
Tabela 34: Resultados BicbPT $\alpha$ x BicbPT $\beta$ com C4.5 .....	107
Tabela 35: Resultados BicbPT $\alpha$ x BicbPT $\beta$ com SVM .....	108
Tabela 36: Resultados BicbPT $\alpha$ x BicbPT $\beta$ com Naive Bayes.....	109

## LISTA DE ABREVIATURAS E SIGLAS

ALA	<i>All Label Assignment</i>
AM	Aprendizagem de Máquina
BR	<i>Binary Relevance</i>
CC	<i>Classifier Chains</i>
CCC	<i>Circular Classifiers Chains</i>
CLR	<i>Calibrated Label Ranking</i>
CML	<i>Collective multi-label classifier</i>
ECC	<i>Ensemble of Classifiers Chains</i>
ELA	<i>Entropy based Label Assignment</i>
HL	<i>Hamming Loss</i>
IL	<i>Independent Labels</i>
JI	<i>Jaccard Index</i>
LC	<i>Labels Chain</i>
LLA	<i>Largest Label Assignment</i>
LP	<i>Label Powerset</i>
LPC-CC	<i>Label Priority Circular Classifier Chains</i>
ML-DFL	<i>Multi-label Learning with Discriminative Features for each Label</i>
NLA	<i>No Label Assignment</i>
PPT	<i>Pruned Problem Transformation</i>
RAkEL	<i>Random k-Labelsets</i>
RLA	<i>Random Label Assignment</i>
RPC	<i>Ranking by Pairwise Comparison</i>
SLA	<i>Smallest Label Assignment</i>
TSA	<i>Two Stage Architecture</i>
TT	Tempo Total

## RESUMO

Dentre as abordagens utilizadas na aprendizagem de máquina, a classificação se destaca principalmente na sua forma monorrótulo. Embora essa seja comum, em alguns domínios a presença de mais de um rótulo é característica inerente dos dados, sendo assim faz-se necessário a utilização de abordagens para classificação multirrótulo. Duas estratégias são possíveis para alcançar a classificação multirrótulo, uma é transformar o problema multirrótulo em um ou mais problemas monorrótulo, ou adaptar um algoritmo monorrótulo para que este possa lidar com a multirrotularidade dos dados. Embora a transformação de problema seja eficaz, alguns algoritmos possuem problemas como, parâmetros fixos para determinar a quantidade de subproblemas monorrótulo e a manutenção dos relacionamentos pré-existentes entre rótulos não usam medidas de correlação ou coocorrência. Dentre as categorias existentes de algoritmos para trabalhar com a transformação de problema, foi escolhida a que permite fazer a transformação de um problema multirrótulo para  $n$  problemas binários, pois esta possui como característica o baixo de tempo de execução, o que permite o uso de algoritmos monorrótulo mais complexos na fase de classificação, como por exemplo, redes neurais e *deep learning*, mas, proporcionalmente, também apresentam baixo desempenho nas métricas multirrótulo. Desta forma, neste trabalho é apresentado o algoritmo BicbPT, o qual utiliza a técnica de biclusterização combinada com a transformação de problema multirrótulo-binário a fim de minimizar aqueles problemas e melhorar o desempenho nas métricas multirrótulo sem perder a característica do baixo tempo de execução desta categoria. Para a avaliação do algoritmo proposto foram realizadas comparações com os algoritmos BR, CC, ECC, RAKEL e LP utilizando os algoritmos monorrótulo SVM, C4.5 e Naive Bayes durante a fase de classificação nos  $n$  problemas binários e 12 conjuntos de dados de diferentes domínios e complexidades. Os experimentos realizados demonstram que o BicbPT obtém melhor desempenho nas métricas multirrótulo que os demais algoritmos multirrótulo-binário comparados, sendo similar somente ao ECC, mas neste o tempo de execução é até 10 vezes mais alto, o que torna o algoritmo proposto melhor. Também, o algoritmo proposto consegue manter tempo de execução similar aos algoritmos da categoria multirrótulo-binário. Ainda, ao comparar as duas versões implementadas do BicbPT foi possível perceber que a maneira com a qual os rótulos se influenciam permite melhorar a classificação multirrótulo, e não somente considerar a manutenção das relações nos  $n$  problemas transformados.

**Palavras-chave:** Classificação multirrótulo. Transformação de problema. Biclusterização.

## ABSTRACT

Among the approaches used in machine learning, the classification stands out especially in its single label way. Although that is common, some domains have multiple labels that are such an intrinsic characteristic of the data, therefore it is necessary a multilabel classification approach. Two strategies are possible to get the multilabel classification, either to convert the multilabel problem into one or more single label problems, or to adapt a single label algorithm to deal with a multilabel data. Despite the fact that problem transformation is effective, some algorithms have issues, as fixed parameters to indicate the single label subproblem quantity and the maintenance of the preexistent relationship among the labels do not use correlation nor co-occurrence measures. Among the categories of algorithms to work with the problem transformation, it was chosen one that allows a transformation from a multilabel problem to  $n$  binary problems. That has a characteristic of having a low runtime, which allows to use more complex single label algorithms in the classification stage, like neural networks or deep learning, but it also shows a lower performance in multilabel metrics. Thus, this work shows the BicbPT algorithm that uses the biclustering and multilabel-binary problem transformation to minimize those problems and improve the multilabel metrics without losing the low execution time characteristic of this category. It was chosen the algorithms BR, CC, ECC, RAKEL and LP with SVM, C4.5 and Naïve Bayes to evaluate the proposed method and 12 datasets with distinct complexities and different domains. The experiments show that the BicbPT obtains better performance in the multilabel metrics than the multilabel-binary algorithms, being similar only to the ECC, but in this one the execution time is up to 10 times higher. The BicbPT also keeps the lower execution time, characteristic of the multilabel-binary category. Finally, comparing the two versions of the BicbPT is possible to realize that the way labels influence each other allows improving the multilabel classification, and not only considering the maintenance of relationships in the  $n$  transformed problems.

**Keywords:** Multilabel Classification. Problem Transformation. Biclustering.

## SUMÁRIO

<b>1 INTRODUÇÃO</b> .....	<b>14</b>
1.1 OBJETIVOS .....	16
1.2 HIPÓTESES .....	17
1.3 CONTRIBUIÇÕES .....	17
1.4 ESTRUTURA DO TRABALHO .....	17
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	<b>19</b>
2.1 CONCEITOS BÁSICOS DE APRENDIZAGEM .....	19
2.2 CLASSIFICAÇÃO MULTIRRÓTULO .....	22
2.2.1 Algoritmos de Transformação de Problema .....	24
2.2.2 Métricas para Avaliação de Classificadores Multirrótulo.....	28
2.2.3 Métricas para Avaliação de Conjunto de Dados .....	33
2.3 BICLUSTERING .....	34
2.3.1 Tipos de Biclusters.....	35
2.3.2 Estruturas de Biclusters.....	37
2.3.3 Bimax: <i>Binary Inclusion-Maximal Biclustering</i> .....	38
<b>3 PROCEDIMENTOS METODOLÓGICOS</b> .....	<b>41</b>
3.1 ETAPA 1: PLANEJAMENTO INICIAL E EXPLORAÇÃO .....	41
3.2 ETAPA 2: DESENVOLVIMENTO.....	44
3.3 ETAPA 3: AVALIAÇÃO .....	44
<b>4 TRABALHOS RELACIONADOS</b> .....	<b>46</b>
<b>5 ALGORITMO PROPOSTO</b> .....	<b>50</b>
5.1 MANUTENÇÃO DAS DEPENDÊNCIAS ENTRE RÓTULOS .....	51
5.2 BicbPT: <i>BICLUSTER-BASED PROBLEM TRANSFORMATION</i> .....	54
5.3 HIPER PARÂMETRO $\rho$ .....	56
5.4 CAMADA DE INDEPENDÊNCIA.....	56
5.5 CAMADA DE BICLUSTERIZAÇÃO: BicbPT $\alpha$ .....	58
5.6 CAMADA DE BICLUSTERIZAÇÃO: BicbPT $\beta$ .....	59
5.7 AGREGAÇÃO DAS PREDIÇÕES .....	61
5.8 LIMITAÇÕES .....	63
<b>6 AVALIAÇÃO EXPERIMENTAL</b> .....	<b>64</b>
6.1 PARÂMETROS DOS EXPERIMENTOS .....	64
6.2 EXPERIMENTO I: BicbPT $\alpha$ x {BR, CC, ECC, RAKEL $_1$ , RAKEL $_2$ , LP}.....	68
6.3 EXPERIMENTO II: BicbPT $\beta$ x {BR, CC, ECC, RAKEL $_1$ , RAKEL $_2$ , LP} .....	86
6.4 EXPERIMENTO III: BicbPT $\alpha$ x BicbPT $\beta$ .....	106
6.5 EXPERIMENTO IV: ANÁLISE DE SENSIBILIDADE DO HIPER PARÂMETRO $\rho$ .....	110
<b>7 CONCLUSÃO E TRABALHOS FUTUROS</b> .....	<b>115</b>
7.1 TRABALHOS FUTUROS .....	117
<b>REFERÊNCIAS</b> .....	<b>119</b>

## 1 INTRODUÇÃO

A quantidade de dados gerados e armazenados tem crescido, resultado em grande parte do avanço da Internet, chamando à atenção de especialistas em diversas áreas, as quais utilizam esses dados para analisar o comportamento humano, sugerir produtos, categorizar mídias e criar veículos autônomos.

A Aprendizagem de Máquina (AM) é uma das áreas que se utiliza desse volume de dados, sendo o seu objetivo aprender padrões a partir de exemplos pré-existentes, os quais podem estar rotulados (aprendizagem supervisionada), não rotulados (aprendizagem não-supervisionada) e apresentarem-se de forma híbrida (aprendizagem semisupervisionada), ou seja, alguns exemplos estão rotulados e outros não (WITTEN et al., 2016).

Dentre as abordagens utilizadas para aprender tais padrões, como a classificação, a associação e o agrupamento, a primeira é amplamente utilizada e seu uso supervisionado monorrótulo, onde um único rótulo é atribuído a cada exemplo, ainda representa boa parte do que é feito, como pode ser visto nos trabalhos de Lai e Tsai (2005), Bulbul e Unsal (2011), Ubik e Žejdl (2010) e Zhan et al. (2009).

Mas, existem contextos onde uma solução multirrótulo reflete melhor as características reais do problema, tais como, associar a quais assuntos concomitantes uma notícia pertence, como informática e medicina. Na categorização de filmes, que pode ser romance, comédia e ficção científica simultaneamente ou na bioinformática, onde a classificação multirrótulo permite predizer as funções de genes, visto que um gene pode estar associado a mais de uma função.

Nesse contexto, para realizar a classificação multirrótulo, é possível transformar os exemplos multirrotulados ou adaptar um algoritmo monorrótulo existente para realizar a tarefa, sendo que no primeiro o objetivo é obter um ou mais problemas monorrótulos derivados, onde posteriormente qualquer classificador tradicional pode ser utilizado, e no segundo são feitas adaptações em um algoritmo monorrótulo já existente de forma que este possa tratar exemplos com características multirrótulo (KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018).

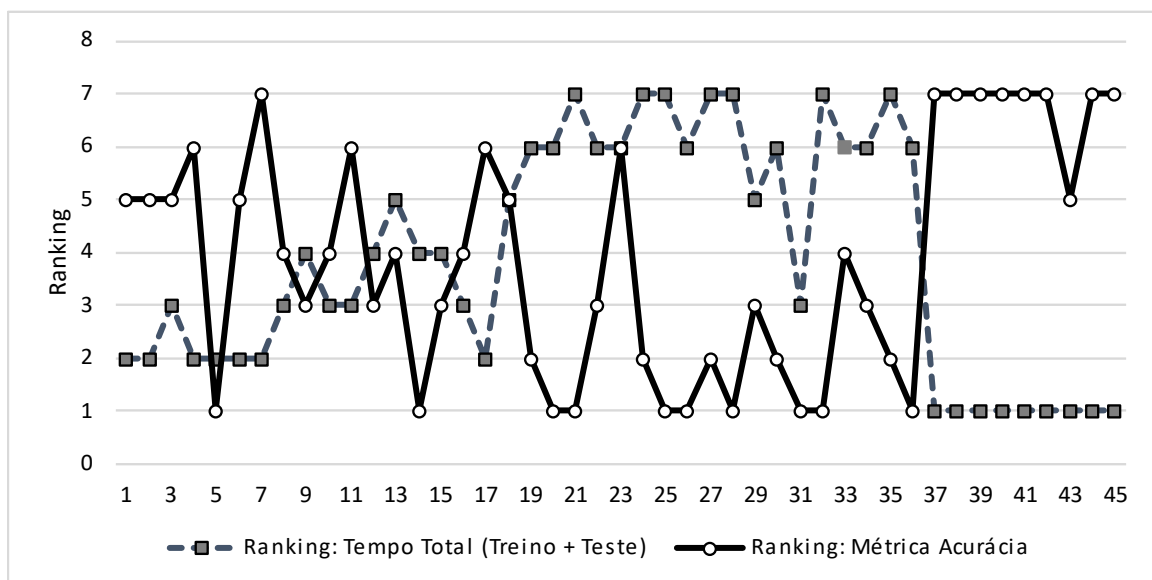
Alguns exemplos de algoritmos utilizados na transformação de problema são *Binary Relevance* (BR), *Calibrated Label Ranking* (CLR) e *Ensemble of Classifiers Chains* (ECC) e na adaptação de algoritmo tem-se ML-kNN, ML-NB, ML-C4.5 (KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018; ZHANG; ZHOU, 2014).

Em relação a classificação multirrótulo por transformação de problema, estratégia adotada neste trabalho, dois desafios se destacam: escalabilidade e manutenção da dependência entre rótulos. Como na transformação de problema são gerados  $n$  problemas monorrótulos derivados e, cada um está associado a um classificador, o tempo total de execução (treino + teste) do algoritmo acaba sendo maior do que em uma solução monorrótulo, o que torna a escalabilidade dos algoritmos de classificação multirrótulo algo importante de ser considerado.

Ainda, através de uma análise empírica inicial da classificação multirrótulo por transformação de problema, considerando as bases utilizadas no trabalho e os algoritmos testados é possível notar uma correlação negativa, obtida através de Pearson com valor de  $-0.78$ , entre a posição do algoritmo no tempo total de execução (treino + teste) e a posição do algoritmo no desempenho nas métricas, como ilustra a Figura 1.

Isso demonstra que algoritmos que escalam melhor, como os multirrótulo-binários BR, *Classifiers Chain* (CC), possuem baixo tempo total de execução (treino + teste), mas apresentam desempenho inferior nas métricas em relação a algoritmos que tem a escalabilidade inferior, como ECC e *Random k-Labelsets* (RAkEL).

Figura 1: Correlação entre Tempo Total de Execução (Treino + Teste) e Acurácia



Fonte: Autoria Própria (2022)

Assim, considerando-se a escalabilidade facilitada das soluções multirrótulo-binário, torna-se interessante o desenvolvimento de algoritmos nessa estratégia que consigam manter esta característica ao mesmo tempo em que se busca bons resultados nas métricas, de forma a tornar a correlação positiva.

Também, na classificação multirrótulo, a coexistência dos rótulos no problema permite que existam relacionamentos entre eles, algo que é considerado importante pela comunidade, visto que essa dependência entre os rótulos ajuda a melhorar a predição final, mas não é uma tarefa fácil identificar estas relações, visto que muitos algoritmos se utilizam apenas da aleatoriedade para tratar essa questão ou não a consideram, portanto, algoritmos que explorem essa natureza da dependência dos rótulos apresentam ganhos e novas estratégias para a comunidade.

Tais dependências podem ser identificadas e mantidas nos subproblemas monorrótulos derivados utilizando-se medidas de correlação, como os coeficientes de correlação de Pearson, Spearman, Kendall, entre outros ou através da coocorrência dos rótulos. Na forma de manutenção por coocorrência é possível utilizar algoritmos de agrupamento para tal fim, que é a abordagem utilizada neste trabalho através da biclusterização.

Na biclusterização, diferentemente do agrupamento tradicional, linhas e colunas são utilizadas ao mesmo tempo para definir *clusters* no conjunto de dados, o que possibilita que subespaços de exemplos e atributos possam ser utilizados, permitindo assim encontrar coocorrências entre os rótulos e também determinar em quais exemplos estas ocorrem.

Assim, neste trabalho é apresentado o algoritmo BicbPT (*Bicluster-based Problem Transformation*), o qual utiliza uma nova estratégia para identificar e manter as relações de dependência entre os rótulos, através da biclusterização, e que também se posiciona de forma a tornar positiva a correlação entre a posição no de tempo total de execução (treino + teste) e desempenho em métricas nas soluções multirrótulo-binário.

## 1.1 OBJETIVOS

O objetivo principal deste trabalho é apresentar um novo algoritmo para classificação multirrótulo combinando a transformação de problema multirrótulo-binário e a biclusterização.

Os objetivos específicos são:

- Caracterizar as formas de manutenção das dependências entre rótulos nos algoritmos de transformação de problema através de uma categorização
- Analisar o desempenho nas métricas do BicbPT com diferentes algoritmos de classificação multirrótulo por transformação de problema
- Analisar a escalabilidade do BicbPT através do tempo total (treino + teste) com diferentes algoritmos de classificação multirrótulo por transformação de problema



## 1.2 HIPÓTESES

As hipóteses investigadas neste trabalho são:

- H1. A combinação da biclusterização com a transformação de problema multirrótulo-binário permite obter resultados superiores ou similares aos algoritmos que não consideram nenhuma forma de dependência entre os rótulos ou utilizam-se de aleatoriedade
- H2. É possível manter a característica de alta escalabilidade da categoria de algoritmos multirrótulo-binários tratando a dependência real entre os rótulos sem degradar o tempo total (treino + teste)

## 1.3 CONTRIBUIÇÕES

As principais contribuições deste trabalho são:

- Um novo algoritmo de classificação multirrótulo que combina transformação de problema e biclusterização
- Uso de aprendizagem de máquina para construir subproblemas monorrótulos
- Disponibilização do algoritmo para o *framework* MEKA
- Categorização das formas de manutenção das dependências entre rótulos em algoritmos de classificação multirrótulo por transformação de problema

## 1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em sete capítulos, sendo no primeiro, o qual está contida esta subseção, apresenta-se a introdução, objetivos, hipóteses e contribuições do trabalho. Já no Capítulo 2 apresenta-se a fundamentação teórica com alguns conceitos sobre aprendizagem de máquina, classificação multirrótulo e o agrupamento bidirecional (biclusterização).

No Capítulo 3 são apresentados os procedimentos metodológicos e suas etapas, percorrendo sobre como foi realizada a pesquisa exploratória, o desenvolvimento do BicbPT e a pesquisa quantitativa, através de dos testes estatísticos de Friedman, Nememyi e Wilcoxon.

No Capítulo 4 são descritos os trabalhos relacionados e são apresentados os pontos na classificação multirrótulo por transformação de problema, relacionado ao BicbPT, que estes trabalhos não trabalham.

No Capítulo 5 é apresentado o algoritmo BicbPT, como funciona cada uma de suas camadas de transformação de problema, como é feita a classificação multirrótulo de um novo exemplo e suas limitações.

Por fim, nos Capítulos 6 e 7 são apresentados os resultados e discussões da avaliação experimental das duas versões do BicbPT,  $\alpha$  e  $\beta$ , a conclusão e alguns trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados conceitos básicos de aprendizagem, abordagens para classificação multirrótulo por transformação de problema, é descrita a biclusterização, o algoritmo Bimax, utilizado neste trabalho, e por fim são relacionadas as principais métricas utilizadas para avaliação de classificadores multirrótulo.

### 2.1 CONCEITOS BÁSICOS DE APRENDIZAGEM

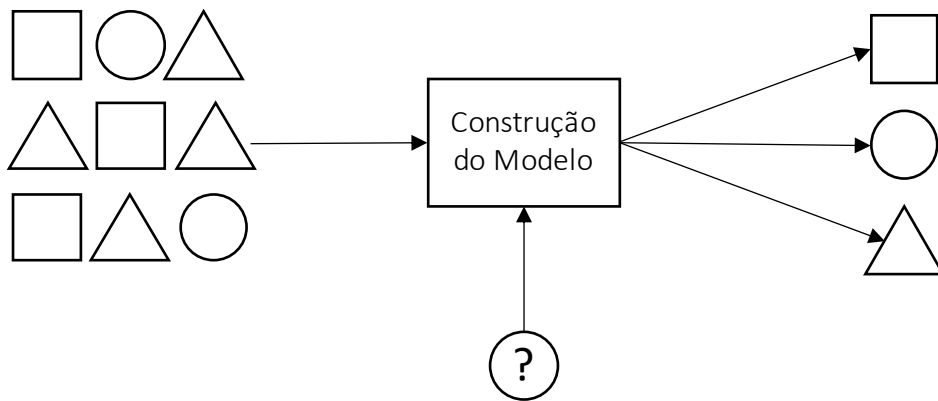
Alguns conceitos básicos presentes na literatura são necessários para o entendimento da aprendizagem de máquina, são eles, segundo Mitchell (1997), Rezende (2005) e Witten et al. (2016):

- Exemplo: representa um vetor de valores de atributos. Também pode ser chamado de instância ou objeto
- Atributo: refere-se a uma característica de um exemplo e seus valor pode ser nominal ou contínuo;
- Rótulo: também chamado de classe, é um atributo que tem como papel representar o exemplo no conjunto de dados, ou seja, indicar a que grupo ele pertence. Eles podem ser discretos (problemas de classificação) ou contínuos (problemas de regressão).

Na aprendizagem indutiva conceitos são generalizados a partir de exemplos e atributos, ou seja, parte-se de algo específico para o geral, o que permite que um conceito seja aprendido a partir de um conjunto de exemplos gerando hipóteses que podem ou não refletir a verdade, por isso deve-se ter atenção, pois caso a quantidade de exemplos seja insuficiente as hipóteses geradas podem não possuir valor (REZENDE, 2005).

De forma geral, como ilustra a Figura 2, na indução os exemplos (quadrado, círculo e triângulo) presentes no conjunto de dados são utilizados por algum algoritmo de indução para construir um modelo preditivo, que, ao receber um exemplo novo (?), ainda não rotulado, deverá apontar um rótulo para este novo exemplo (quadrado ou círculo ou triângulo).

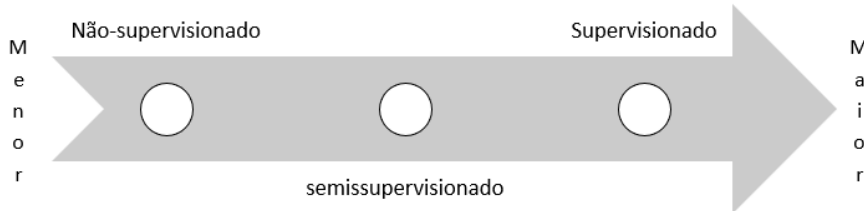
Figura 2: Exemplo aprendizado indutivo



Fonte: Autorial Própria (2022)

Segundo Mitchell (1997) a aprendizagem pode ser categorizada de acordo com o grau de supervisão, presença ou ausência de rótulos nos exemplos, presente no conjunto de exemplos, como mostra a Figura 3, sendo que quanto maior a quantidade de exemplos rotulados, maior é o grau de supervisão.

Figura 3: Grau de supervisão nos conjuntos de dados



Fonte: Autorial Própria (2022)

Na aprendizagem não-supervisionada os rótulos,  $Y$ , não estão presentes nos exemplos,  $T_i$ , como ilustra a Figura 4c, sendo assim, o foco aqui é tentar extrair algum significado ou informação dos dados utilizando para isso alguma medida de semelhança entre as características dos exemplos. O aprendizado não-supervisionado pode ser utilizado para construir agrupamentos, reduzir a dimensionalidade do conjunto, detectar anomalias, entre outros, sendo a sua tarefa mais comum a primeira situação (SARKAR; BALI; SHARMA, 2018; WITTEN et al., 2016; XU; WUNSCH II, 2005).

Figura 4: Representação do conjunto de exemplos em cada grau de supervisão

	$X_1$	$X_2$	...	$X_j$	$Y$
$T_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	$y_1$
$T_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	$y_2$
...	...	...	...	...	...
$T_i$	$x_{i1}$	$x_{i2}$	...	$x_{ij}$	$y_n$

a) Entrada aprendizagem supervisionada

	$X_1$	$X_2$	...	$X_j$	$Y$
$T_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	$y_1$
$T_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	?
...	...	...	...	...	...
$T_i$	$x_{i1}$	$x_{i2}$	...	$x_{ij}$	$y_n$

b) Entrada aprendizagem semissupervisionada

	$X_1$	$X_2$	...	$X_j$	
$T_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	
$T_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	
...	...	...	...	...	
$T_i$	$x_{i1}$	$x_{i2}$	...	$x_{ij}$	

c) Entrada aprendizagem não-supervisionada

Fonte: Autoria Própria (2022)

Já na aprendizagem supervisionada o objetivo é encontrar uma função  $f(x)$  que seja capaz de representar todos os exemplo  $T_i$ , rotulados e que possa prever corretamente um exemplo  $T_x$  não rotulado. Os exemplos são normalmente representados na forma de vetor de valores  $(x_{i1}, \dots, x_{ij})$ , onde  $x_{ij}$  refere-se ao valor do atributo  $X_j$  em relação ao exemplo  $T_i$  e  $y_n$  representa o rótulo do exemplo  $T_i$  como ilustra a Figura 4a. Quando  $y_n$  pertence a um conjunto discreto de rótulos, onde  $y \in L = \{y_1, \dots, y_{|L|}\}$ , e se  $|L| > 2$  o problema é dito monorrótulo multiclasse, caso contrário tem-se um problema monorrótulo binário (GAMA; BERNARDINI; ZADROZNY, 2012).

E entre a aprendizagem supervisionada e não-supervisionada tem-se a semissupervisão, a qual consiste de utilizar um conjunto com exemplos rotulados e não rotulados, como ilustra a Figura 4b (BRUCE, 2001; ZHU; GOLDBERG, 2009).

Também, é necessário distinguir as três maneiras pela qual um conjunto de dados é utilizado no processo de indução:

- Conjunto de treinamento: É o conjunto utilizado para a construção do modelo
- Conjunto de validação: Conjunto utilizado para ajustes finos no modelo construído, como alteração de parâmetros, alteração de algoritmo.
- Conjunto de teste: Utilizado para avaliar o modelo e, por ideal, os exemplos contidos aqui não devem fazer parte do conjunto de treinamento e nem de validação

Por fim, embora seja possível o uso da aprendizagem supervisionada em diversas situações, nem sempre é fácil conseguir exemplos rotulados, o que acaba gerando custo de tempo, humano e monetário. Nesse sentido, as aprendizagens não-supervisionada ou semissupervisionada podem ser utilizadas, visto que não necessitam de uma grande quantidade de exemplos rotulados para realização da aprendizagem (SANTOS, 2012; ZHU; GOLDBERG, 2009).

## 2.2 CLASSIFICAÇÃO MULTIRRÓTULO

Algoritmos de aprendizado supervisionado tradicionais, onde cada exemplo é representado por um vetor de características e um único rótulo, têm sido os mais estudados na aprendizagem de máquina. Esses são categorizados como aprendizado supervisionado monorrótulo.

Formalmente, segundo Tsoumakos e Katakis (2007), dado  $y$  como um único rótulo de um exemplo do conjunto de dados e  $L$  como o conjunto de rótulos do problema, se  $|L| = 2$  tem-se um problema de classificação monorrótulo binário e se  $|L| > 2$  tem-se um problema de classificação monorrótulo multiclasse, conforme representa a Figura 5 a e b, respectivamente.

Figura 5: Exemplos de conjunto de dados monorrótulo

#	X <sub>1</sub>	X <sub>2</sub>	...	X <sub>j</sub>	Y
T <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	y <sub>1</sub>
T <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	y <sub>2</sub>
T <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	y <sub>1</sub>
T <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	y <sub>2</sub>
T <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	y <sub>1</sub>

L = {y<sub>1</sub>, y<sub>2</sub>}  
a) binário

#	X <sub>1</sub>	X <sub>2</sub>	...	X <sub>j</sub>	Y
T <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	y <sub>1</sub>
T <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	y <sub>2</sub>
T <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	y <sub>3</sub>
T <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	y <sub>4</sub>
T <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	y <sub>5</sub>

L = {y<sub>1</sub>, y<sub>2</sub>, y<sub>3</sub>, y<sub>4</sub>, y<sub>5</sub>}  
b) multiclasse

Fonte: Aatoria Própria (2022)

Já na classificação multirrótulo, o que se tem é um conjunto de rótulos  $Y_i$  tal que  $Y_i \subseteq L$ , para cada exemplo, como mostra a Figura 6 (KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018; TSOUMAKAS; KATAKIS, 2007).

Figura 6: Exemplo de conjunto de dados multirrótulo

#	X <sub>1</sub>	X <sub>2</sub>	...	X <sub>j</sub>	Y
T <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	y <sub>1</sub> , y <sub>3</sub> , y <sub>4</sub>
T <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	y <sub>2</sub> , y <sub>3</sub>
T <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	y <sub>1</sub> , y <sub>3</sub> , y <sub>4</sub>
T <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	y <sub>4</sub>
T <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	y <sub>3</sub> , y <sub>4</sub>

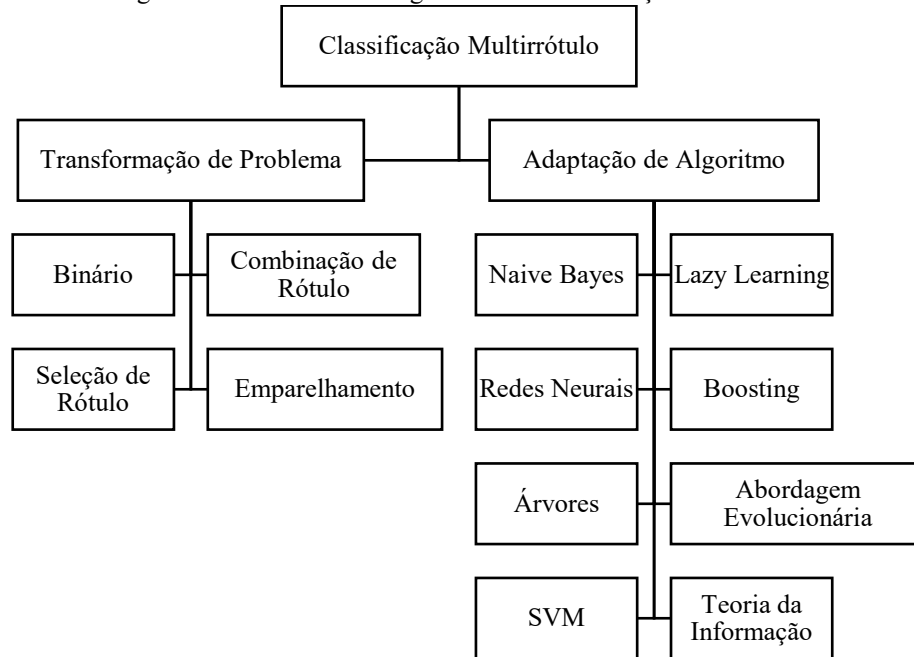
L = {y<sub>1</sub>, y<sub>2</sub>, y<sub>3</sub>, y<sub>4</sub>}

Fonte: Aatoria Própria (2022)

Diversos problemas no mundo real possuem essa configuração, como anotação de imagens e vídeos, classificação de função de proteínas, categorização de textos e emoções (KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018).

E, como a tarefa de classificação multirrótulo não é trivial, devido à presença simultânea de vários rótulos e, em alguns casos de alta dimensionalidade e esparsidade, duas abordagens para permitir a classificação neste cenário são propostas, a adaptação de algoritmo e a transformação de problema, como demonstra a Figura 7 (GONÇALVES; FREITAS; PLASTINO, 2018; KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018).

Figura 7: Taxonomia dos algoritmos de classificação multirrótulo.



Fonte: Adaptado de (GIBAJA; VENTURA, 2014; KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018)

Na estratégia de adaptação de algoritmo são feitas adaptações em algoritmos de aprendizagem monorrótulo de maneira que estes consigam lidar com a característica multirrótulo dos exemplos, como por exemplo, os algoritmos ML-KNN, Rank-SVM e *Collective Multi-Label Classifier* (CML) (ZHANG; ZHOU, 2014).

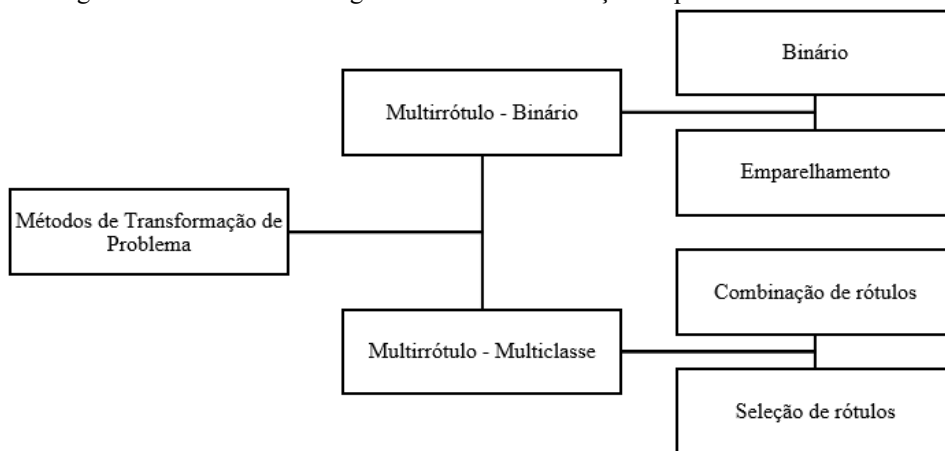
Já na transformação de problema o objetivo é ajustar os dados aos algoritmos, esta abordagem também pode ser dita independente de algoritmo, pois após a transformação qualquer algoritmo de aprendizagem monorrótulo pode ser utilizado para a tarefa de classificação. Alguns exemplos de algoritmos de transformação de problema são BR (GIBAJA; VENTURA, 2015), *Label Powerset* (LP) (READ et al., 2011) e RAKEL (TSOUMAKAS; VLAHAVAS, 2007).

Como este trabalho tem como base a classificação multirrótulo por transformação de problema, alguns desses algoritmos serão descritos mais detalhadamente em 2.2.1.

### 2.2.1 Algoritmos de Transformação de Problema

Como pode ser observado na Figura 8, os algoritmos de classificação multirrótulo por transformação de problema podem utilizar diversas estratégias: binário, emparelhamento, combinação de rótulos e seleção de rótulos para transformar o conjunto inicial multirrótulo em um ou mais conjuntos monorrótulos. De maneira geral, também pode-se dividi-los de acordo com o tipo de resultado esperado após a transformação do conjunto de dados multirrótulo original: um ou  $n$  conjuntos binários; um ou  $n$  conjuntos multiclasse.

Figura 8: Taxonomia dos algoritmos de transformação de problema multirrótulo



Fonte: Autoria Própria (2022)

Para exemplificação de alguns algoritmos de transformação de problema será utilizada a Figura 9 como um micro exemplo de um conjunto de dados multirrótulo.

Figura 9: Micro exemplo de conjunto de dados multirrótulo

T	X				Y			
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	1	1	1	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	1	0	1	0
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0	0	0	1
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	1	0	0	1

$$L = \{y_1, y_2, y_3, y_4\}$$

Fonte: Autoria Própria (2022)

Na transformação multirrótulo-binário/binário, o conjunto de dados multirrótulo original é decomposto em  $q$  conjuntos binários independentes, sendo um para cada rótulo  $y_i$  em



$L$ , como é possível observar na Figura 10. Após a decomposição é aplicado um algoritmo de aprendizagem monorrótulo a cada conjunto de dados transformado.

Quando um novo exemplo  $t$  precisa ser rotulado, a predição do conjunto multirrótulo é a união dos resultados positivos de todos os classificadores binários gerados.

Figura 10: Conjunto de dados transformado pelo algoritmo BR

T	Y	T	Y	T	Y	T	Y
	$y_1$		$y_2$		$y_3$		$y_4$
$t_1$	1	$t_1$	1	$t_1$	1	$t_1$	0
$t_2$	1	$t_2$	0	$t_2$	1	$t_2$	0
$t_3$	0	$t_3$	0	$t_3$	0	$t_3$	1
$t_4$	1	$t_4$	0	$t_4$	0	$t_4$	1

Fonte: Aatoria Própria (2022)

O algoritmo de transformação multirrótulo-binário mais representativo nesta categoria é o BR, mas o seu principal problema é ignorar a correlação dos rótulos, a qual é uma característica que ocorre com grande frequência em problemas multirrótulo, mas algoritmos baseados nesta transformação e que consideram a dependência entre os rótulos foram propostos, como por exemplo, o algoritmo CC (GIBAJA; VENTURA, 2014; ZHANG; ZHOU, 2014).

Figura 11: Conjunto de dados transformado pelo algoritmo CC

T	Y	T	X	Y	T	X	Y	T	X	Y			
	$y_1$		$y_1$	$y_2$		$y_1$	$y_2$	$y_3$		$y_1$	$y_2$	$y_3$	$y_4$
$t_1$	1	$t_1$	1	1	$t_1$	1	1	1	$t_1$	1	1	1	0
$t_2$	1	$t_2$	1	0	$t_2$	1	0	1	$t_2$	1	0	1	0
$t_3$	0	$t_3$	0	0	$t_3$	0	0	0	$t_3$	0	0	0	1
$t_4$	1	$t_4$	1	0	$t_4$	1	0	0	$t_4$	1	0	0	1

a)

b)

c)

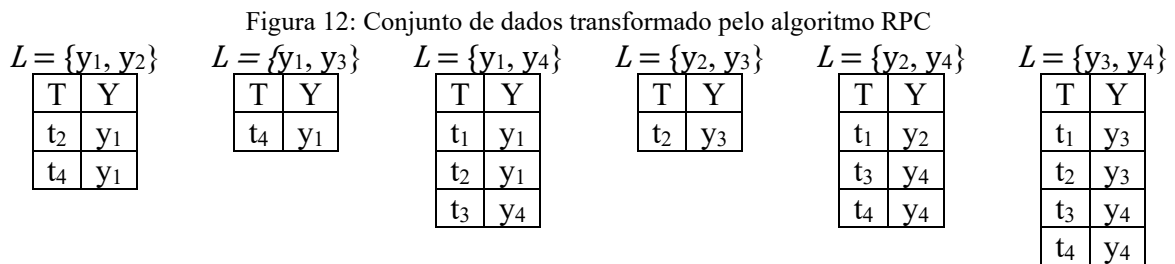
d)

Fonte: Aatoria Própria (2022)

A Figura 11 mostra a saída transformada do algoritmo CC, o qual tem execução semelhante ao BR, porque gera  $|L|$  conjuntos, a diferença no CC é que os conjuntos posteriores consideram os rótulos anteriores como características adicionais, ou seja, no conjunto da Figura 11b,  $y_1$  é uma característica adicional a todos os exemplos e  $Y$  é o rótulo binário, e assim sucessivamente para todos os rótulos.

Segundo Read et al. (2011) isto permite considerar a correlação dos rótulos durante a construção dos  $|L|$  classificadores sem aumentar o custo computacional e mantendo a simplicidade.

Na transformação multirrótulo-binário/emparelhamento, a ideia é transformar o conjunto original em no máximo  $|L|. (|L| - 1) / 2$  subproblemas, onde, em cada um é considerado um par de rótulos. Em cada um, os exemplos são somente considerados se pertencerem a um dos dois rótulos, como pode ser observado na Figura 12, e não a ambos. Após feita a transformação é gerado um classificador para cada conjunto não vazio (MADJAROV et al., 2012; TSOUMAKAS; KATAKIS; VLAHAVAS, 2009).



Fonte: Autoria Própria (2022)

Quando um novo exemplo  $t$  é submetido para ser multirrotulado, ele é atribuído a todos os classificadores e no final é construído um *ranking* com a soma de votos de cada rótulo, após é aplicada uma função de limiar capaz de separar os rótulos mais votados e atribui-los ao exemplo  $x$ .

A vantagem de algoritmos que trabalham com a abordagem de emparelhamento, como por exemplo, *Ranking by Pairwise Comparison* (RPC) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2009) e *Calibrated Label Ranking* (CLR) (GIBAJA; VENTURA, 2014), é que permitem encontrar relacionamentos entre os rótulos que não estão no conjunto de dados original. Suas principais desvantagens são o uso de um limiar, que deve ser calibrado para cada conjunto de dados, e a alta quantidade de conjuntos gerados, que no pior caso pode ser  $|L|. (|L| - 1) / 2$ .

Diferentemente da abordagem multirrótulo-binário, onde são produzidos vários classificadores, na multirrótulo-multiclasse é gerado apenas um classificador multiclasse a partir do conjunto de dados original, o que permite que algoritmos de aprendizagem mais sofisticados possam ser utilizados, como, por exemplo, algoritmos de *Deep Learning*.

Comparando-se com a transformação multirrótulo-binário, a desvantagem aqui é que os classificadores não conseguem rotular novos exemplos com um conjunto de classes que não aparece no conjunto de dados original, como ocorre com o BR e CC.

No caso da transformação de problema multirrótulo-multiclasse, o seu algoritmo mais representativo é o LP (TSOUMAKAS; KATAKIS; VLAHAVAS, 2009), o qual combina os

rótulos de cada exemplo no conjunto original para produzir novas classes no conjunto transformado, como pode ser observado na Figura 13.

Figura 13: Conjunto de dados transformado pelo algoritmo LP

T	Y
t <sub>1</sub>	y <sub>123</sub>
t <sub>2</sub>	y <sub>13</sub>
t <sub>3</sub>	y <sub>4</sub>
t <sub>4</sub>	y <sub>14</sub>

Fonte: Autoria Própria (2022)

Como é gerado apenas um único classificador nesta abordagem, quando é submetido um novo exemplo  $x$  a este, o rótulo atribuído será uma das classes compostas que corresponde a um conjunto de rótulos do conjunto de dados original.

Embora consiga lidar com a correlação dos rótulos, o LP pode gerar até  $2^{|L|}$  classes no pior caso, assim, ao considerar um conjunto de dados com 10 rótulos, pode-se obter 1024 classes, o que torna o algoritmo inviável para problemas reais. Soluções como *Pruned Problem Transformation* (PPT) (MADJAROV et al., 2012) tentam lidar com isso ao focar nas combinações de rótulos mais importantes e eliminar as menos frequentes.

Completando os algoritmos de transformação de problema multirrótulo apresentados na Figura 8, inclui-se a seleção de rótulos, a qual pode gerar conjuntos de dados multiclasse através dos algoritmos *All Label Assignment* (ALA), *Entropy based Label Assignment* (ELA), *Largest Label Assignment* (LLA), *Smallest Label Assignment* (SLA), *No Label Assignment* (NLA) (CHEN et al., 2007) e *Random Label Assignment* (RLA) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2009).

Figura 14: Conjunto de dados transformado pelos algoritmos de seleção de rótulos

a) ALA		b) ELA			c) LLA		d) SLA		e) RLA		f) NLA	
T	Y	T	Y	Peso	T	Y	T	Y	T	Y	T	Y
t <sub>1</sub>	y <sub>1</sub>	t <sub>1</sub>	y <sub>1</sub>	0.33	t <sub>1</sub>	y <sub>1</sub>	t <sub>1</sub>	y <sub>2</sub>	t <sub>1</sub>	y <sub>1</sub>	t <sub>3</sub>	y <sub>4</sub>
t <sub>1</sub>	y <sub>2</sub>	t <sub>1</sub>	y <sub>2</sub>	0.33	t <sub>2</sub>	y <sub>1</sub>	t <sub>2</sub>	y <sub>3</sub>	t <sub>2</sub>	y <sub>1</sub>		
t <sub>1</sub>	y <sub>3</sub>	t <sub>1</sub>	y <sub>3</sub>	0.33	t <sub>3</sub>	y <sub>4</sub>	t <sub>3</sub>	y <sub>4</sub>	t <sub>3</sub>	y <sub>3</sub>		
t <sub>2</sub>	y <sub>1</sub>	t <sub>2</sub>	y <sub>1</sub>	0.50	t <sub>4</sub>	y <sub>1</sub>	t <sub>4</sub>	y <sub>4</sub>	t <sub>4</sub>	y <sub>2</sub>		
t <sub>2</sub>	y <sub>3</sub>	t <sub>2</sub>	y <sub>3</sub>	0.50								
t <sub>3</sub>	y <sub>4</sub>	t <sub>3</sub>	y <sub>4</sub>	1.00								
t <sub>4</sub>	y <sub>1</sub>	t <sub>4</sub>	y <sub>1</sub>	0.50								
t <sub>4</sub>	y <sub>4</sub>	t <sub>4</sub>	y <sub>4</sub>	0.50								

Fonte: Autoria Própria (2022)

Na Figura 14 são apresentados os resultados das transformações do conjunto de dados multirrótulo da Figura 9 utilizando algoritmos multirrótulo-multiclasse/seleção. Basicamente cada algoritmo executa a transformação da seguinte maneira:

- ALA: faz uma cópia de cada exemplo até a quantidade de rótulos que foram atribuídos a cada exemplo. Como na Figura 9 o exemplo 1 possui três rótulos, ele é copiado três vezes e para cada cópia é atribuído um dos rótulos;
- ELA: o processo de cópia é o mesmo do ALA, mas aqui também são calculados pesos a partir de  $1/|Y_i|$ , ou seja, como no exemplo 1 existem três rótulos, então os pesos atribuídos a cada exemplo são iguais a  $1/3$ ;
- LLA: atribui o rótulo mais frequente no conjunto de dados ao exemplo. Como no exemplo 1 temos  $y_1 = 3$ ;  $y_2 = 1$ ;  $y_3 = 2$ , então o rótulo  $y_1$  é atribuído ao exemplo;
- SLA: atribui o rótulo menos frequente no conjunto de dados ao exemplo. Como no exemplo 1 temos  $y_1 = 3$ ;  $y_2 = 1$ ;  $y_3 = 2$ , então o rótulo  $y_2$  é atribuído ao exemplo;
- RLA: escolhe aleatoriamente um dos rótulos do exemplo para atribuí-lo no conjunto transformado;
- NLA: exemplos multirrótulo são descartados.

Algoritmos de aprendizagem probabilísticos são mais recomendados neste tipo de transformação, pois eles conseguem indicar estimativas para cada rótulo e assim, através de algum limiar é possível atribuir um conjunto multirrótulo para um novo exemplo (TSOUMAKAS; KATAKIS; VLAHAVAS, 2009).

Segundo Gibaja e Ventura (2014, p. 415) “apesar de simples, estes algoritmos produzem problemas de perda de informação em relação aos rótulos e seus relacionamentos e então eles podem não ser muito úteis”.

Como é possível observar, independente da abordagem de transformação de problema escolhida, binário ou multiclasse, a principal preocupação é a manter o relacionamento dos rótulos nos conjuntos de dados transformados, o que demonstra que novos algoritmos devem levar isto em consideração ao serem propostos.

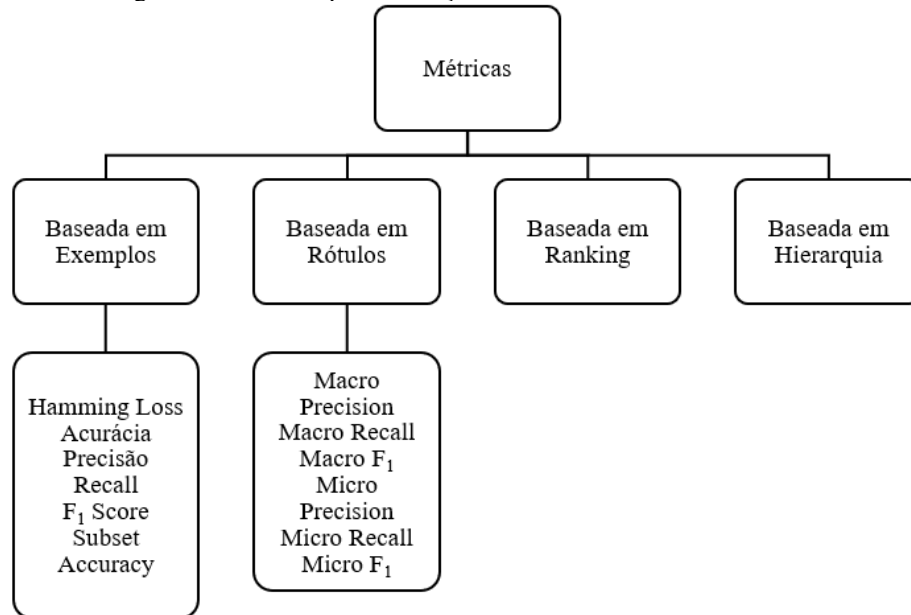
### 2.2.2 Métricas para Avaliação de Classificadores Multirrótulo

Quando é construído um classificador utilizando aprendizagem monorrótulo pode-se avaliá-lo utilizando métricas como, por exemplo, acurácia, *recall*, precisão e  $F_1$ , mas na classificação multirrótulo tais métricas necessitam de adaptações, visto que ocorrem acertos

parciais na abordagem multirrótulo, ou seja, o classificador pode atribuir somente alguns dos rótulos esperados e não todos (MADJAROV et al., 2012; ZHANG; ZHOU, 2014).

Em Zhang e Zhou (ZHANG; ZHOU, 2014) e Madjarov et al. (2012) é apresentada uma categorização para as métricas de avaliação de classificadores multirrótulo, como mostra a Figura 15.

Figura 15: Métricas para avaliação de classificadores multirrótulo.



Fonte: Adaptado de (MADJAROV et al., 2012, p. 3089)

Métricas baseadas em exemplos avaliam a qualidade do classificador no conjunto de dados, enquanto as baseadas em rótulos avaliam a qualidade do classificador em prever cada rótulo separadamente.

Quando os rótulos possuem uma ordem associada tem-se disponível métricas baseadas em *ranking* e para quando é considerado a existência de hierarquia entre os rótulos podem ser utilizadas métricas baseadas em hierarquia.

No decorrer da subseção serão apresentadas métricas baseadas em exemplos e rótulos, como descritas em Zhang e Zhou (ZHANG; ZHOU, 2014), Kashef, Nezamabadi-Pour e Nikpour (KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018) e Madjarov et al. (2012).

Detalhes sobre as métricas baseadas em *ranking* e hierarquia podem ser vistos em Gibaja e Ventura (2014).

Para o entendimento das métricas as seguintes notações são utilizadas nesta subseção e na subseção 2.2.3:

- T: quantidade de exemplos no conjunto

- L: quantidade de rótulos possíveis para atribuir a cada exemplo
- X: quantidade de atributos no conjunto de dados
- $Y_i$ : conjunto de rótulos de um determinado exemplo
- $Z_i$ : conjunto de rótulos predito para um determinado exemplo
- $\Delta$ : diferença simétrica entre dois conjuntos
- $\wedge$ : operador AND
- $\uparrow$ : quanto maior o resultado da métrica, melhor é o classificador
- $\downarrow$ : quanto menor o resultado da métrica, melhor é o classificador

A métrica *Hamming Loss*, definida na Equação 2.1, considera as predições incorretas do classificador, ou seja, quando um rótulo correto não é predito ou quando um incorreto é predito.

$$\downarrow \text{Hamming Loss} = \frac{1}{T} \sum_{i=1}^T \frac{|Y_i \Delta Z_i|}{L} \quad (2.1)$$

As métricas, acurácia, precisão, *recall* e  $F_1$ , apresentadas nas Equações 2.2, 2.3, 2.4, 2.5, respectivamente, são adaptações para avaliar classificadores multirrótulo das métricas utilizadas na classificação monorrótulo.

$$\uparrow \text{Acurácia} = \frac{1}{T} \sum_{i=1}^T \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \quad (2.2)$$

$$\uparrow \text{Precisão} = \frac{1}{T} \sum_{i=1}^T \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (2.3)$$

$$\uparrow \text{Recall} = \frac{1}{T} \sum_{i=1}^T \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (2.4)$$

$$\uparrow F_1 = \frac{1}{T} \sum_{i=1}^T \frac{2 \times |Y_i \cap Z_i|}{|Z_i| + |Y_i|} \quad (2.5)$$

E para finalizar as métricas baseadas em exemplos, apresenta-se a métrica *Subset Accuracy* na Equação 2.6, a qual tem como objetivo determinar a proporção de exemplos que tiveram todos os seus rótulos preditos corretamente, onde a saída da comparação  $I$  é 1 se os rótulos forem iguais ou 0 se forem diferentes. Esta métrica deve ser utilizada com cautela, já que é bastante rigorosa ao considerar somente corretos os acertos totais e não parciais.

$$\uparrow \textit{Subset Accuracy} = \frac{1}{T} \sum_{i=1}^T I(Z_i = Y_i) \quad (2.6)$$

Nas métricas baseadas em rótulos, os cálculos são feitos a partir dos valores de verdadeiro positivo (VP), falso positivo (FP), verdadeiro negativo (VN) e falso negativo (FN), os quais são apresentados nas Equações 2.7, 2.8, 2.9 e 2.10.

$$VP_j = |\{x_i \mid y_j \in Y_i \wedge y_j \in Z_i, 1 \leq i \leq T\}| \quad (2.7)$$

$$FP_j = |\{x_i \mid y_j \notin Y_i \wedge y_j \in Z_i, 1 \leq i \leq T\}| \quad (2.8)$$

$$VN_j = |\{x_i \mid y_j \notin Y_i \wedge y_j \notin Z_i, 1 \leq i \leq T\}| \quad (2.9)$$

$$FN_j = |\{x_i \mid y_j \in Y_i \wedge y_j \notin Z_i, 1 \leq i \leq T\}| \quad (2.10)$$

Com base nesses valores é possível calcular as métricas, acurácia, precisão, *recall* e  $F_1$  baseadas em rótulos, as quais são apresentadas nas Equações 2.11, 2.12, 2.13 e 2.14.

$$\uparrow \textit{Acurácia} = \frac{VP_j + VN_j}{VP_j + FP_j + VN_j + FN_j} \quad (2.11)$$

$$\uparrow \textit{Precisão} = \frac{VP_j}{VP_j + FP_j} \quad (2.12)$$

$$\uparrow \textit{Recall} = \frac{VP_j}{VP_j + FN_j} \quad (2.13)$$

$$\uparrow F_1 = \frac{2 \times \textit{Precisão} \times \textit{Recall}}{\textit{Precisão} + \textit{Recall}} \quad (2.14)$$

Embora sejam métricas úteis quando avaliamos o classificador pelos rótulos, utilizá-las desta maneira seria altamente custoso, visto que problemas multirrótulo podem ter um conjunto de classes na casa de centenas, e assim ter-se-ia um relatório grande para ser analisado. Para contornar esse problema, recomenda-se o uso dessas através de uma média macro e/ou micro, descritas nas Equações 2.15 e 2.16, respectivamente, onde  $B \in \{\text{Acurácia, Precisão, Recall, } F_1\}$ .

$$B_{macro} = \frac{1}{T} \sum_{j=1}^T B(VP_j, FP_j, VN_j, FN_j) \quad (2.15)$$

$$B_{micro} = B\left(\sum_{j=1}^T VP_j, \sum_{j=1}^T FP_j, \sum_{j=1}^T VN_j, \sum_{j=1}^T FN_j\right) \quad (2.16)$$

Na abordagem macro, é calculada a métrica escolhida para cada rótulo e então é feita uma média, enquanto que na abordagem micro são totalizados todos os VP, FP, VN e FN para todos os rótulos e então é calculada a métrica escolhida. As Equações 2.17, 2.18 e 2.19 apresentam as versões macro de precisão, *recall*, e  $F_1$  e as Equações 2.20, 2.21 e 2.22 as versões micro das mesmas métricas.

$$\uparrow \text{Precisão}_{macro} = \frac{1}{T} \sum_{j=1}^T \frac{VP_j}{VP_j + FP_j} \quad (2.17)$$

$$\uparrow \text{Recall}_{macro} = \frac{1}{T} \sum_{j=1}^T \frac{VP_j}{VP_j + FN_j} \quad (2.18)$$

$$\uparrow F_{1macro} = \frac{1}{T} \sum_{j=1}^T \frac{2 \times \text{Precisão}_j \times \text{Recall}_j}{\text{Precisão}_j + \text{Recall}_j} \quad (2.19)$$

$$\uparrow \text{Precisão}_{micro} = \frac{\sum_{j=1}^T VP_j}{\sum_{j=1}^T VP_j + \sum_{j=1}^T FP_j} \quad (2.20)$$

$$\uparrow \text{Recall}_{micro} = \frac{\sum_{j=1}^T VP_j}{\sum_{j=1}^T VP_j + \sum_{j=1}^T FN_j} \quad (2.21)$$



$$\uparrow F_{1_{micro}} = \frac{2 \times Precisão_{micro} \times Recall_{micro}}{Precisão_{micro} + Recall_{micro}} \quad (2.22)$$

### 2.2.3 Métricas para Avaliação de Conjunto de Dados

Além das métricas apresentadas na subseção 2.2.2 para avaliar um classificador, é possível também avaliar o conjunto de dados para verificar a multirrotularidade deste.

A métrica cardinalidade, apresentada na Equação 2.23, informa a quantidade média de rótulos por cada exemplo do conjunto de dados, sendo que quanto mais próximo de 1 for o resultado, menos multirrótulo é o conjunto de dados (KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018; ZHANG; ZHOU, 2014).

$$Cardinalidade = \frac{1}{T} \sum_{i=1}^T |Y_i| \quad (2.23)$$

Outra métrica utilizada é a densidade, apresentada na Equação 2.24, a qual é uma versão normalizada da cardinalidade e informa a proporção de rótulos disponíveis que estão sendo utilizados, onde, quanto mais próximo de 1/L for o resultado, menor é a quantidade de rótulos utilizados.

$$Densidade = \frac{1}{T} \frac{1}{L} \sum_{i=1}^T |Y_i| \quad (2.24)$$

Por fim, em Read et al. (2011) é apresentada a métrica complexidade, Equação 2.25, a qual considera a quantidade de exemplos, de atributos e rótulos para identificar qual o custo computacional que determinada base de dados pode ter.

$$Complexidade = T * X * L \quad (2.25)$$

## 2.3 BICLUSTERING

A biclusterização pode ser descrita da seguinte maneira, segundo Madeira e Oliveira (2004), dada uma matriz  $A^{n \times m} = (X, Y)$ , sendo  $X$  o conjunto de exemplos e  $Y$  o conjunto de atributos, o valor  $a_{ij}$  corresponde ao valor que representa a relação  $x_i$  e  $y_i$ .

Um bicluster  $B^{k \times s} = (G, C)$  é uma submatriz de  $A$  formada pelo subconjunto de exemplos  $G$ ,  $G \subseteq X$ , que possuem comportamentos similares com relação a um subconjunto de atributos  $C$ ,  $C \subseteq Y$  e vice-versa. O objetivo é identificar em  $A$  um conjunto de biclusters  $B = \{B_1, B_2, \dots, B_i\}$ , tal que cada bicluster satisfaça características de homogeneidade específicas do algoritmo escolhido.

Diferentemente dos algoritmos de agrupamento tradicionais que utilizam todo o espaço de atributos ou todo o espaço de exemplos para encontrar agrupamentos, na biclusterização isto é feito utilizando os dois espaços simultaneamente, sendo esta a sua principal característica.

Outras diferenças em relação ao agrupamento tradicional são apresentadas no Quadro 1, onde o modelo global da forma tradicional utiliza todo os exemplos ou atributos e a forma local na biclusterização utiliza somente um subconjunto desses.

Quadro 1: Diferenças entre agrupamento tradicional e *biclustering*

<b>Característica</b>	<b>Tradicional</b>	<b><i>Biclustering</i></b>
Forma de busca	Exemplos ou atributos	Exemplos e atributos
Modelo	Global	Local
Exemplos	Agrupados utilizando todo o espaço de atributos	Agrupados utilizando um subconjunto de atributos
Atributos	Agrupados utilizando todo o espaço de exemplos	Agrupados utilizando um subconjunto de exemplos

Fonte: Autoria Própria (2022)

Biclusters também podem ser descritos como um biclique de um grafo bipartido e encontra-los nesta situação é equivalente a encontrar o biclique com o máximo de arestas de um grafo bipartido, o qual é um problema NP-Completo, dessa forma boa parte dos algoritmos de biclusterização também são baseados em heurísticas ou meta-heurísticas (MADEIRA; OLIVEIRA, 2004; PEETERS, 2003).

Biclusters podem ser classificados segundo seu tipo e estrutura, os quais serão detalhados nas seções 2.3.1 e 2.3.2 seguintes.

### 2.3.1 Tipos de Biclusters

Segundo Madeira e Oliveira (2004) o tipo de bicluster que pode ser encontrado em um conjunto de dados é definido pelo tipo de homogeneidade que se espera ver ali, as quais podem ser:

- Com valores constantes (Figura 16a)
- Com valores constantes nas linhas ou colunas (Figura 16b)
- Com valores coerentes (Figura 16c)
- Com evoluções coerentes (Figura 16d)

Nos três primeiros é seguido algum modelo matemático, visto que são analisados os valores numéricos diretamente e no quarto a ideia é buscar por comportamentos coerentes, sem considerar nenhum modelo matemático, assim biclusters deste tipo enxergam os elementos da matriz como símbolos, como pode ser observado na Figura 16 (MADEIRA; OLIVEIRA, 2004; PONTES; GIRÁLDEZ; AGUILAR-RUIZ, 2015).

Figura 16: Tipos de biclusters

10	10	10
10	10	10
10	10	10
10	10	10

a)

10	10	10
20	20	20
30	30	30
40	40	40

b)

10	8	7
7	5	4
9	7	6
4	2	1

c)

2	5	10
3	5	8
1	2	3
1	5	6

d)

Fonte: Autoria Própria (2022)

A forma de busca de biclusters por valores constantes, Figura 16a, é a mais simples e tem como ideia revelar submatrizes em que todos os valores são iguais. Alguns algoritmos que objetivam encontrar esse tipo ordenam as linhas e colunas da matriz a fim de identificar tais biclusters (MADEIRA; OLIVEIRA, 2004) .

O bicluster perfeito nesse cenário é aquele onde todos os valores da submatriz (G, C) são iguais, mas como é difícil verificar isto em dados reais, porquanto normalmente existem ruídos associados aos valores, é utilizada a variância para avaliar a qualidade de cada bicluster encontrado, assim o bicluster perfeito é aquele que possui variância igual a zero (MADEIRA; OLIVEIRA, 2004).

De maneira similar é possível buscar por biclusters constantes somente nas linhas ou colunas, Figura 16b, os quais são representados por um valor base ( $\mu$ ) que pode ser extraído da

posição  $A_{ij}$  da matriz inicial ou ser informado manualmente e um ajuste aditivo ( $\alpha_i$ ) ou multiplicativo em relação a este valor.

Um bicluster com linhas constantes é aquele onde os valores são encontrados segundo a Equação 2.32 ou 2.33, onde  $\alpha_i$  representa o ajuste para o valor da linha  $i$ . Na Figura 16b,  $\alpha_i = \{0; 10; 20; 30\}$  representa o valor de cada linha e  $\mu = 10$  o valor base.

$$b_{ij} = \mu + \alpha_i \quad (2.32)$$

$$b_{ij} = \mu * \alpha_i \quad (2.33)$$

De maneira análoga, o bicluster perfeito com colunas constantes é aquele onde os valores são encontrados segundo a Equação 2.34 ou 2.35, onde  $\beta_j$  representa o ajuste para o valor da coluna  $j$ .

$$b_{ij} = \mu + \beta_j \quad (2.34)$$

$$b_{ij} = \mu * \beta_j \quad (2.35)$$

Como no caso anterior, pode ser difícil encontrar biclusters com valores perfeitamente constantes, visto a possibilidade de ruído em dados reais, mas diferentemente, aqui não é possível a utilização da variância. Assim, pode-se tentar resolver o problema normalizando as linhas ou colunas com suas respectivas médias, o que faz com que o bicluster apresentado na Figura 16b se torne o apresentado na Figura 16a.

Quando o tipo de bicluster esperado é o de valores coerentes, Figura 16c, pode-se avaliar a qualidade dos biclusters resultantes através de uma medida de coerência, onde calcula-se a diferença entre o bicluster encontrado e o modelo perfeito de coerência, chamada Resíduo Quadrático Médio (CHENG; CHURCH, 2000).

De maneira geral, um bicluster de valores coerentes perfeitos pode ser representado através de um modelo aditivo ou multiplicativo, representados nas Equações 2.36 e 2.37, onde  $\mu$  e  $\mu'$  representam o valor base do bicluster,  $\alpha_i$  e  $\alpha'_i$  são o ajuste para linha e  $\beta_j$  e  $\beta'_j$  são o ajuste da coluna.

$$b_{ij} = \mu + \alpha_i + \beta_j \quad (2.36)$$

$$b_{ij} = \mu' * \alpha'_i * \beta'_j \quad (2.37)$$

Na Figura 16c é apresentado um bicluster de valores coerentes com modelo aditivo com  $\mu = 10$ ,  $\alpha_i = \{0; -3; -1; -6\}$  e  $\beta_j = \{0; -2; -3\}$ , assim para determinar os valores de  $b_{01}$  e  $b_{11}$ , o seguinte cálculo é feito,  $b_{01} = 10 + 0 + (-2)$  e  $b_{11} = 10 + (-3) + (-2)$ .

É importante notar que se  $\alpha = 0$  e/ou  $\beta = 0$ , podem ser encontrados biclusters completamente constantes, constantes na linha ou constantes na coluna, desta forma a busca por biclusters de valores coerentes pode ser visto como uma forma geral para encontrar diversos tipos de biclusters (MADEIRA; OLIVEIRA, 2004).

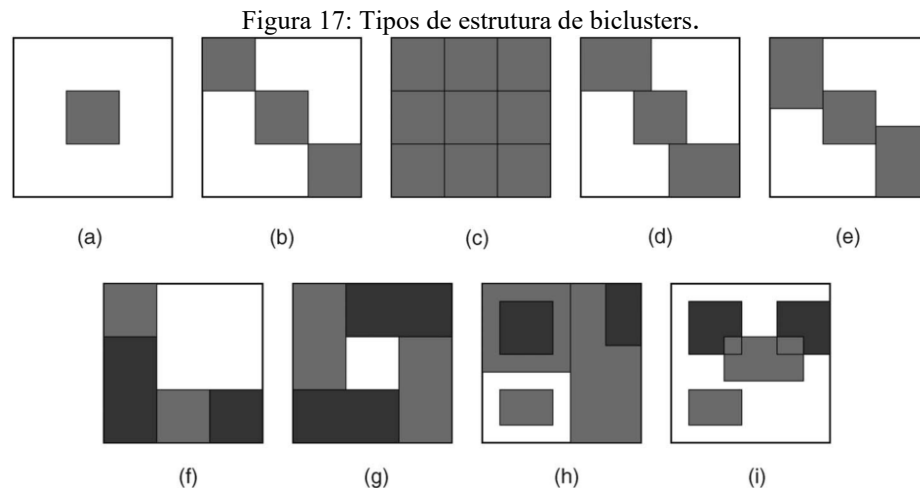
Por último tem-se o tipo de bicluster por evolução coerente, Figura 16d, o qual não considera os valores numéricos dentro do bicluster, mas sim propriedades coerentes nestes. Por exemplo, um bicluster com evolução coerente pode ser um no qual todos os valores são maiores que um limiar, ou ainda representados segundo uma ordem, crescente ou decrescente. Na Figura 16d é possível perceber que a propriedade utilizada é a de ordem crescente nas linhas.

As medidas de avaliação dos biclusters produzidos aqui podem ser o tamanho deste, dado um limiar ou o qual a proporção coerente do bicluster (MURALI; KASIF, 2003; TANAY; SHARAN; SHAMIR, 2002).

### 2.3.2 Estruturas de Biclusters

Algoritmos de bicluster, além de determinar um tipo de bicluster como saída também determinam como será a estrutura dos biclusters encontrados, sendo que pode ser produzido um único bicluster, Figura 17a, ou vários biclusters, e neste caso pode-se ter as seguintes estruturas, visualizadas na Figura 17 (MADEIRA; OLIVEIRA, 2004; PONTES; GIRÁLDEZ; AGUILAR-RUIZ, 2015):

- Linhas e colunas exclusivas (b)
- Sem sobreposição com estrutura xadrez (c)
- Linhas exclusivas (d)
- Colunas exclusivas (e)
- Sem sobreposição com mapa de árvore (f)
- Sem sobreposição e sem exclusividade (g)
- Sobreposição com hierarquia (h)
- Sobreposição arbitrária (i)



Fonte: (MADEIRA; OLIVEIRA, 2004, p. 34)

Alguns algoritmos presentes em Madeira e Oliveira(2004) e Prelic et al. (2006), com seus respectivos tipos e estruturas podem ser encontrados no Quadro 2.

Quadro 2: Exemplos de algoritmos de biclusterização

Algoritmo	Tipo de Bicluster	Estrutura de Bicluster
OPSM	Evolução Coerente	a ou i
PRM	Colunas constantes	b
Spectral Clustering	Valores coerentes	c
ITWC	Valores coerentes	d ou e
Block Clustering	Valores constantes	f
pClusters	Valores coerentes	g
Bimax	Valores constantes	a ou i

Fonte: Adaptado de Madeira e Oliveira(2004) e Prelic et al. (2006)

Neste trabalho foram escolhidas as características de sobreposição arbitrária e biclusters com valores constantes, ambas possíveis através do algoritmo Bimax (PRELIĆ et al., 2006), pois nelas um subconjunto de exemplos e/ou de atributos podem pertencer a mais de um bicluster simultaneamente, o que é útil para construção posterior de conjuntos de dados transformados, visto que um exemplo pode aparecer em mais de um conjunto binário e um subconjunto de atributos também pode estar presente em vários conjuntos binários transformados.

### 2.3.3 Bimax: *Binary Inclusion-Maximal Biclustering*

O Bimax é um algoritmo de biclusterização e seu principal uso é relacionar um subconjunto de genes com um subconjunto de expressões que estes genes desencadeiam. O

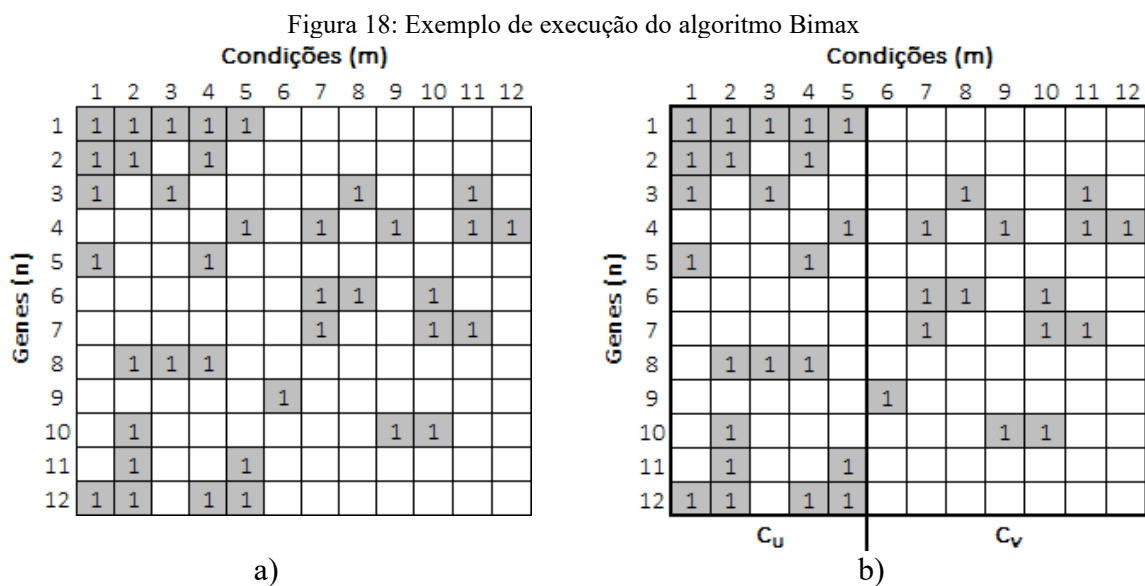
algoritmo assume, basicamente, dois níveis de expressões por gene: sem mudança ou com mudança, em relação a um experimento de controle.

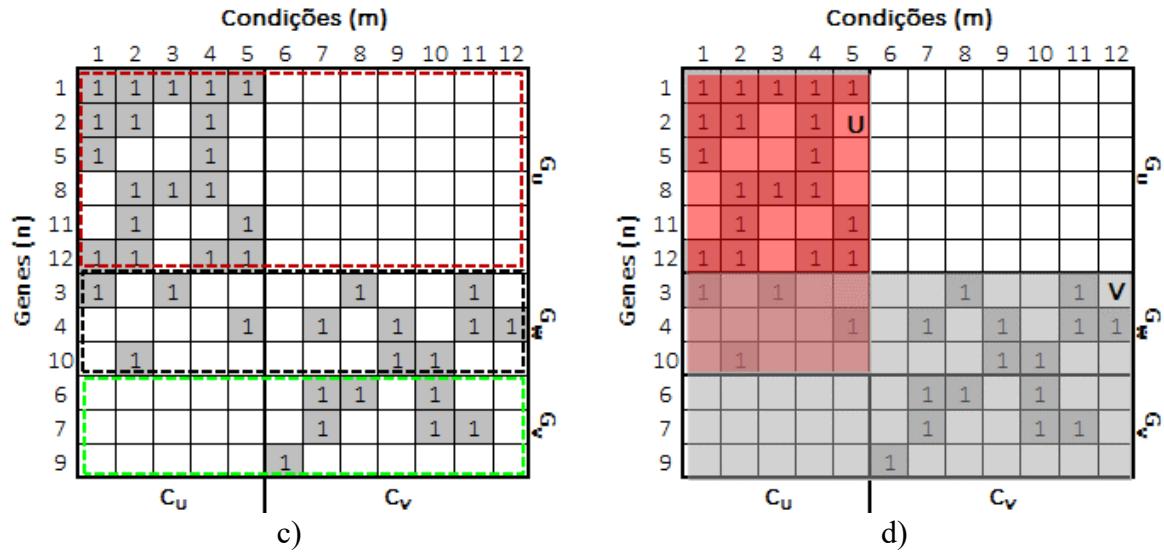
Assim, segundo Prelic (2006), um conjunto  $m$  de experimentos para  $n$  genes pode ser representado através de uma matriz binária  $E^{n \times m}$ , onde  $e_{ij} = 1$  se o gene  $i$  expressa algo na condição  $j$  ou 0, caso contrário. De maneira similar, um bicluster  $(G, C)$  corresponde a um subconjunto de genes  $G$  que respondem a um subconjunto de condições  $C$ , representados por uma submatriz, a qual é um subconjunto maximal com relação de inclusão.

Segundo Rodriguez-Baena et al. (2011), um bicluster maximal é assim definido se, e somente se, ele não está inteiramente contido em qualquer outro bicluster, ou seja, as linhas e colunas devem ser específicas de cada bicluster. É importante notar que, o Bimax permite a sobreposição de algumas linhas e/ou colunas, mas não a sobreposição total de um bicluster em outro, o que criaria uma relação do tipo  $A \subset B$ .

A execução do Bimax inicia escolhendo uma linha,  $r^*$ , na matriz  $E$  que contenha 0 e 1, caso nenhuma linha em  $E$  tenha essa característica, significa que toda a matriz  $E$  é um único bicluster, todas as linhas possuem valor 1, ou não existe bicluster em  $E$ .

Para exemplificar, na Figura 18a é escolhida a linha 1 para iniciar a execução, de forma que após a escolha a matriz  $E$  é dividida em duas submatrizes,  $C_U$  e  $C_V$ , onde a primeira é composta por todas as linhas em  $E$  onde as colunas em  $r^*$  são iguais a 1 e a segunda por todas as linhas em  $E$  onde as colunas em  $r^*$  são iguais a 0, alcançando assim o resultado ilustrado na Figura 18b.





Fonte: Adaptado de Prelic (2006)

Após dividir  $E$  em  $C_U$  e  $C_V$ , as linhas são reorganizadas em três submatrizes,  $G_U$ ,  $G_W$  e  $G_V$ , Figura 18c, as quais possuem a seguinte composição:

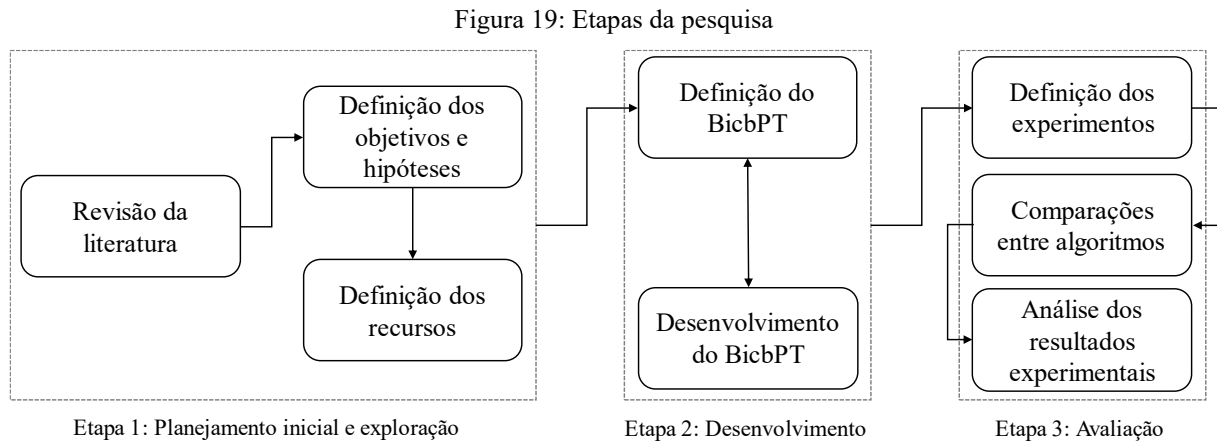
- $G_U$ : linhas que possuem algum valor 1 somente em  $C_U$
- $G_W$ : linhas que possuem algum valor 1 em  $C_U$  e  $C_V$
- $G_V$ : linhas que possuem algum valor 1 somente em  $C_V$

Por fim, a combinação das submatrizes  $C_U$ ,  $C_V$ ,  $G_U$ ,  $G_W$  e  $G_V$  resultam nas submatrizes  $U$  e  $V$ , sendo  $U = (G_U \cup G_W, C_U)$  e  $V = (G_W \cup G_V, C_U \cup C_V)$ , as quais são processadas recursivamente, Figura 18d, e a submatriz formada por  $(G_U, C_V)$  é descartada, pois não é possível encontrar biclusters nela.



### 3 PROCEDIMENTOS METODOLÓGICOS

Neste capítulo é apresentado como a pesquisa foi estruturada, as ferramentas, bases de dados multirrótulo e o *hardware* utilizado para desenvolvimento do algoritmo BicbPT, proposto nesta tese, e a execução dos experimentos. De forma geral, as etapas do trabalho podem ser visualizadas na Figura 19.



Fonte: Autoria Própria (2022)

#### 3.1 ETAPA 1: PLANEJAMENTO INICIAL E EXPLORAÇÃO

Inicialmente, na Etapa 1, foi realizada uma pesquisa exploratória a fim de buscar familiaridade com o problema, classificação multirrótulo por transformação de problema, e também para dar suporte as hipóteses de pesquisa.

Foram feitas duas revisões de literatura para identificar lacunas e definir o trabalho, e também para proporcionar a compreensão do problema e identificar trabalhos relacionados.

As buscas foram feitas nas bases de dados ACM *Digital Library* (ACM *Full-Text Collection*), IEEE Xplore, ScienceDirect, Wiley *Online Library* e SpringerLink considerando o período de 2008 à 2022 e todos os resultados retornados da busca, ou seja, artigos, capítulos de livros e revistas e foram selecionados aqueles que continham os termos buscados no título e/ou resumo e/ou palavras-chave.

Uma primeira revisão foi feita utilizando a *string* de busca: *(multilabel OR "multi-label" OR "multi label") AND (classification OR learning) AND "problem transformation"*; a qual foi utilizada para compreender a classificação multirrótulo por transformação de problema e identificar algoritmos neste contexto.

Posteriormente, uma segunda *string* de busca foi utilizada acrescentando-se o termo *binary*, ficando *(multilabel OR "multi-label" OR "multi label") AND (classification OR learning) AND "problem transformation" AND binary*, que foi utilizada para definir o trabalho e encontrar trabalhos relacionados a esta tese. Os resultados desta segunda revisão são apresentados na Tabela 1.

Tabela 1: Resultados da busca de trabalhos relacionados

<b>Base</b>	<b>Retornados</b>	<b>Selecionados</b>
ACM Digital Library	87	30
IEEE Xplore	11	7
ScienceDirect	9	9
Wiley Online Library	26	7
SpringerLink	170	47
Total	303	100

Fonte: Autoria Própria (2022)

Ao todo, foram selecionados 100 trabalhos, os quais foram classificados em:

- Aplicação: o trabalho aplica algoritmos de transformação de problema multirrótulo-binário em algum contexto, a fim de demonstrar que seu uso para classificação ou para alguma outra tarefa pode ter bons resultados
- Revisão: trabalhos que realizaram revisão da literatura
- Algoritmo: trabalhos que propõe novos algoritmos para classificação multirrótulo através da transformação de problema multirrótulo-binário

Após esta pesquisa exploratória, identificou-se duas questões:

- I. como tratar a dependência real entre rótulos em algoritmos de classificação multirrótulo por transformação de problema, visto que a maior parte dos algoritmos identificados pressupõe a dependência sem verificá-la através de correlação, coocorrência ou outro método que permita identificar relações reais;
- II. a existência de uma possível correlação negativa entre a posição do algoritmo no tempo total de execução (treino + teste) e a posição do algoritmo no desempenho nas métricas, visto que algoritmos que utilizam a abordagem multirrótulo-binário escalam com mais facilidade do que algoritmos multirrótulo-multiclasse ou *ensembles*, mas apresentam desempenho inferior nas métricas.

A questão I levou a definição da hipótese H1 definida na Seção 1.2: *A combinação da biclusterização com a transformação de problema multirrótulo-binário permite obter resultados superiores ou similares aos algoritmos que não consideram nenhuma forma de dependência entre os rótulos ou utilizam-se de aleatoriedade*; e a questão II ajudou a definir a hipótese H2, também apresentada na Seção 1.2: *É possível manter a característica de alta escalabilidade da categoria de algoritmos multirrótulo-binários tratando a dependência real entre os rótulos sem degradar o tempo total (treino + teste)*.

Assim, o objetivo geral do trabalho foi apresentar um novo algoritmo de classificação multirrótulo por transformação de problema na categoria multirrótulo-binário que construísse subproblemas monorrótulos binários utilizando AM, de forma a manter as relações reais entre os rótulos nos subproblemas e, que também pudesse ser escalado como os algoritmos multirrótulo-binário, mas alcançando um desempenho superior nas métricas em relação aos algoritmos desta categoria.

Por fim, foram definidos os recursos necessários para desenvolvimento do BicbPT. A Tabela 2 apresenta as bases de dados multirrótulo utilizadas nos experimentos, as quais estão disponíveis no website do framework Mulan (TSOUMAKAS et al., 2011) e para execução dos experimentos foi utilizado um computador com 8GB de RAM, CPU Intel Core i7 3537-U com Windows 10 Pro 21H2.

Tabela 2: Bases de dados utilizadas nos experimentos do BicbPT

<b>Bases</b>	<b>Domínio</b>	<b>n</b>	<b>X</b>	<b>L</b>
<b>Birds</b>	Áudio	645	260	19
<b>Emotions</b>	Áudio	593	71	6
<b>Enron</b>	Texto	1702	1001	53
<b>Flags</b>	Imagem	194	19	7
<b>Genbase</b>	Biologia	662	1186	27
<b>Medical</b>	Texto	978	1449	45
<b>Scene</b>	Imagem	2407	294	6
<b>Yeast</b>	Biologia	2417	103	14
<b>Bibtex</b>	Texto	7395	1836	159
<b>G1</b>	Texto	2000	1013	7
<b>GP</b>	Texto	668	997	6
<b>BFRC</b>	Texto	7879	994	8

n: quantidade de exemplos X: quantidade de características L: quantidade de rótulos

Fonte: Autoria Própria (2022)

### 3.2 ETAPA 2: DESENVOLVIMENTO

Para o desenvolvimento do BicbPT foi utilizado o framework Meka 1.9.3 (READ et al., 2016), pois, como identificado na pesquisa exploratória, este é utilizado em diversos trabalhos da área e para utilização da biclusterização no algoritmo proposto foram utilizados os pacotes biclust 2.0.2 (KAISER et al., 2021) e rJava 0.9 (URBANEK, 2021) disponíveis na linguagem R, sendo que o primeiro contém o algoritmo Bimax e o segundo contém mecanismos para comunicação entre a linguagem Java e o R.

Nesta fase também foi identificado que tipo de bicluster seria utilizado, optando-se pelo tipo de valores constantes com sobreposição, como mencionado em 2.3, pois a sobreposição apresenta-se útil para a transformação multirrótulo-binário, visto que um exemplo do conjunto de dados original pode aparecer em vários conjuntos binários e, de forma análoga, um conjunto de atributos também.

Também, durante o desenvolvimento foram definidas e implementadas duas versões do BicbPT,  $\alpha$  e  $\beta$ , sendo que a versão  $\alpha$  considera cada bicluster encontrado pelo Bimax como um subproblema monorrótulo binário e a versão  $\beta$  considera cada rótulo do bicluster como um subproblema monorrótulo binário, de forma que um rótulo seja o  $Y$  do subproblema binário e os demais rótulos do bicluster façam parte do conjunto  $X$  de características.

### 3.3 ETAPA 3: AVALIAÇÃO

Na Etapa 3 foi realizada uma pesquisa quantitativa, a fim de, através de experimentos e testes estatísticos, validar as hipóteses H1 e H2, verificar a sensibilidade do hiper parâmetro  $\rho$  do BicbPT e comparar ambas as versões do algoritmo proposto. Para tanto foram definidos os seguintes experimentos:

- I. BicbPT $\alpha$  x {BR, CC, ECC, RAKEL, LP}
- II. BicbPT $\beta$  x {BR, CC, ECC, RAKEL, LP}
- III. BicbPT $\alpha$  x BicbPT $\beta$
- IV. Análise de sensibilidade do hiper parâmetro  $\rho$  do BicbPT

Os experimentos I e II foram utilizados para avaliar as hipóteses, de forma que pudesse ser verificado se o uso da biclusterização, identificando as relações reais entre os rótulos, melhorou a classificação multirrótulo ou se obteve resultados similares aos demais algoritmos multirrótulo-binários. Também, através do tempo total (treino + teste) coletado nos experimentos foi possível analisar se a escalabilidade da categoria multirrótulo-binário foi

mantida enquanto o desempenho nas métricas era similar ou superior aos demais algoritmos desta categoria.

Para comparação estatística dos resultados dos experimentos I e II foi utilizado o teste estatístico não-paramétrico de Friedman com intervalo de confiança de 95% para identificar se existiam diferenças entre os algoritmos comparados e, em caso positivo, foi realizado o pós-teste de Nemenyi para identificar onde estavam as diferenças e a posição dos algoritmos no *ranking* construído com o Nemenyi (DEMSAR, 2006).

O experimento III foi realizado para comparar as duas versões do BicbPT e identificar se a estratégia adotada para criação dos subproblemas a partir dos biclusters influenciava no resultado final da classificação multirrótulo. Posteriormente foi realizado o teste estatístico não-paramétrico de Wilcoxon (DEMSAR, 2006) com intervalo de confiança de 95%.

E por fim, o experimento IV permitiu verificar como o hiper parâmetro  $\rho$ , responsável por indicar o percentual mínimo de exemplos por bicluster no BicbPT, afetava o resultado da classificação multirrótulo considerando os valores de 70%, 50%, 30%, 10%, 5%, 2%, 1%, 0,5% e 0,2% para ele, visto que quanto menor o valor de  $\rho$ , mais biclusters são gerados.

As métricas multirrótulo coletadas em todos os experimentos foram *Jaccard Index*, *Hamming Loss*,  $F_1$ micro,  $F_1$ macro e Tempo Total (treino + teste) em segundos. Todas essas métricas foram identificadas na pesquisa exploratória como sendo as mais comumente utilizadas para avaliar algoritmos de classificação multirrótulo por transformação de problema.

#### 4 TRABALHOS RELACIONADOS

Neste capítulo serão apresentados e discutidos os principais estudos da literatura que foram classificados como algoritmo durante a pesquisa exploratória, apresentada no Capítulo 3, e que possuem relação com este trabalho.

O Quadro 3 apresenta os trabalhos relacionados encontrados e quais as características de cada um em relação ao BicbPT, as quais são:

- Relações Sintéticas: algoritmos que utilizam aleatoriedade ou a simples ordem de ocorrência dos rótulos para tratar a dependência entre os rótulos;
- Relações Orgânicas: algoritmos que verificam a dependência entre os rótulos utilizando correlação ou coocorrência;
- Sem Relações: algoritmos que consideram a existência de independência entre os rótulos;
- AM: uso de aprendizagem de máquina para construir os subproblemas monorrótulo;
- Subproblemas fixos: algoritmos que usam parâmetros para indicar, *a priori*, a quantidade de subproblemas monorrótulo a serem construídos.

Quadro 3: Trabalhos relacionados e suas características

Trabalho	Relações Sintéticas	Relações Orgânicas	Sem Relações	AM*	Subproblemas Fixos
Lotf e Ramdani (2020)		X			X
Wever et al. (2020)	X		X		X
Sun e Kudo (2019)		X	X		X
Soonsiripanichkul e Murata (2016)	X		X		X
Glinka e Zakrzewska (2016)			X		X
Glinka e Zakrzewska (2016)	X				X
Zhang, Fang e Li (2015)			X		X
Madjarov, Gjorgjevikj e Džeroski (2012)	X		X		X
Gjorgjevikj e Madjarov (2011)	X		X		X
Read et al. (2011)	X		X		X
BicbPT		X	X	X	

\*AM: utiliza AM para gerar subproblemas

Fonte: Autoria Própria (2022)

Lotf e Ramdani (2020) apresentam o 3RC (*Relations Recognition and Removing Cycles*), o qual encontra as relações entre os rótulos comparando a acurácia de um classificador multiclasse baseado em árvore de decisão e de um monorrótulo binário para cada rótulo no conjunto multirrótulo. Após escolher qual classificador utilizar para representar o rótulo é construído um grafo de dependências entre os classificadores calculando-se a força da relação dos classificadores através de uma função  $S$  definida pelos autores.

No 3RC, a principal característica ao descobrir as relações entre os rótulos é eliminar relações cíclicas, ou seja, se existe  $A \rightarrow B$ , não existe  $B \leftarrow A$ , mas isto pode ser uma desvantagem visto que o algoritmo não verifica a força desta relação de retorno antes de eliminá-la.

Diferentemente da maioria dos algoritmos de classificação multirrótulo por transformação de problema que utiliza um único algoritmo monorrótulo como base para todos os subproblemas, o LiBRe, apresentado por Wever et al. (2020) permite associar um algoritmo monorrótulo diferente para cada subproblema.

Esta estratégia tem a vantagem de se obter o melhor classificador monorrótulo para cada subproblema, mas o LiBRe ainda possui a principal desvantagem do BR, que é não tratar a dependência entre os rótulos.

Em Sun e Kudo (2019) é apresentado o algoritmo PACC (*Polytree-Augmented Classifier Chains*) o qual tem como objetivo resolver o problema da dependência aleatória entre os rótulos, que é vista no CC. Para isso o PACC constrói uma poli-árvore e calcula a informação mútua entre os nós para identificar qual a probabilidade de existir uma relação de dependência entre eles.

Embora o PACC consiga encontrar relações de dependência entre os rótulos de forma orgânica, ou seja, utilizando a informação mútua entre os nós e não apenas a aleatoriedade, ele ainda esbarra no problema de construir subproblemas monorrótulos fixos, ou seja, limitados pela quantidade de rótulos no problema multirrótulo.

Em Soonsiripanichkul e Murata (2016) é apresentada uma variação do algoritmo CC, a qual os autores chamam de *Label Priority Circular Classifier Chains* (LP-CCC). O algoritmo proposto pelos autores utiliza o algoritmo *Circular Classifiers Chains* (CCC) combinado com a acurácia de cada classificador binário para determinar a ordem com que a cadeia de classificadores será percorrida, visto que no CC tradicional a ordem é dada de forma aleatória.

No CCC após a execução do último classificador na cadeia a saída é adicionada como atributo ao primeiro classificador e o processo reinicia-se até uma quantidade pré-determinada

de ciclos, mas a ordem dos classificadores não muda. Com a adição da análise da acurácia proposta, a ideia é mudar a ordem a cada ciclo, visto que segundo os autores, ela afeta o resultado final de classificação.

Assim como no CC, o LPC-CC não garante que na ordem escolhida existam relações reais entre os rótulos e até uma possível existência de independência, desta forma as relações são pressupostas, ou seja, nenhuma medida de correlação é utilizada para avaliar se dois ou mais rótulos deveriam ou não ser encadeados.

Glinka e Zakrzewska (2016) propõem em seu trabalho dois novos algoritmos de transformação de problema, nomeados *Independent Labels* (IL) e *Labels Chain* (LC). Os dois são similares aos algoritmos BR e CC, respectivamente, mas diferente destes, onde a transformação produz vários conjuntos binários, em IL e LC são produzidos vários conjuntos multiclasse. Desta forma a saída de cada classificador é um rótulo possível, e não um binário 0/1 como em BR e CC. A quantidade de classificadores gerados em IL e LC é igual ao maior subconjunto de rótulos que aparece nos exemplos, já em BR e CC a quantidade de classificadores é igual a quantidade de rótulos disponíveis. Também, o algoritmo IL não considera o relacionamento entre os rótulos e o LC, apesar de manter os relacionamentos, o faz sem nenhuma medida de correlação ou coocorrência.

Em Zhang, Fang e Li (2015), é apresentado o algoritmo *Multi-label Learning with Discriminative Features for each Label* (ML-DFL), o qual tem como principal característica construir os classificadores binários com um subconjunto de características para cada rótulo. Para a realização desta tarefa o ML-DFL utiliza a abordagem *Spectral Clustering* em cada rótulo, separadamente, onde nas linhas da matriz estão as instâncias que pertencem ao rótulo e nas colunas as instâncias que não pertencem ao rótulo.

Desta forma são criados grupos contendo instâncias positivas e negativas que se correlacionam em relação ao rótulo analisado, e é esta correlação que permite a seleção de um subconjunto de características. Ao criar subconjuntos de características para cada rótulo, o ML-DFL ignora uma característica comum aos algoritmos de transformação de problema, a manutenção de relações entre rótulos, igualmente ao BR.

Em Madjarov, Gjorgjevikj e Džeroski (2012) e Gjorgjevikj e Madjarov (2011) é proposta uma nova arquitetura para transformação de problema quando é utilizada a abordagem multirrótulo-binário/emparelhamento, *Two Stage Architecture* (TSA). Nesta arquitetura são construídos os classificadores binários na primeira camada utilizando o algoritmo BR e então estes são relacionados com os classificadores binários emparelhados na segunda camada.



A saída 0/1 de um classificador na primeira camada determina se um classificador da segunda camada deve ou não ser consultado, permitindo assim que menos comparações devam ser realizadas. Nos dois trabalhos além da arquitetura são apresentados algoritmos que a implementam.

Embora consiga lidar com a independência e a dependência entre rótulos, o TSA alcança somente a dependência binária, ou seja, relacionando somente dois rótulos. O pressuposto de existência de relacionamentos entre rótulos na classificação multirrótulo não é limitado, dessa forma podem existir dois ou mais rótulos na relação, até o limite do conjunto de rótulos presente nos dados.

No trabalho de Read et al. (2011) é apresentado o algoritmo CC, o qual já foi descrito com mais detalhes anteriormente em 2.2.1.

Durante esta busca por trabalhos relacionados foram identificados que (a) os algoritmos existentes não tratam ou tratam parcialmente as relações reais pré-existentes entre os rótulos, partindo somente de pressupostos e sem utilizar nenhuma medida de correlação ou alguma forma para identificar coocorrências; (b) não existe uma categorização para a forma como a dependência entre os rótulos é mantida nos algoritmos, o que prejudica o entendimento de como esse processo realmente ocorre *in situ*; (c) os algoritmos existentes não utilizam AM para construir os subproblemas, mas somente para construção dos classificadores monorrótulo; (d) os algoritmos utilizam parâmetros fixos ou se baseiam na quantidade de rótulos do problema multirrótulo original para construir os subproblemas monorrótulos derivados.

## 5 ALGORITMO PROPOSTO

Este capítulo apresenta o algoritmo proposto, BicbPT, para realização da tarefa de classificação multirrótulo, o qual combina a abordagem de transformação de problema multirrótulo-binário/binário e a biclusterização de valores constantes com sobreposição através do algoritmo Bimax, o qual consegue tratar a característica binária dos rótulos nos problemas multirrótulo, visto que ele busca por biclusters em matrizes binárias.

A escolha pela abordagem de transformação de problema deu-se pelas características apresentadas em (GIBAJA; VENTURA, 2014; KASHEF; NEZAMABADI-POUR; NIKPOUR, 2018; ZHANG; ZHOU, 2014; ZHANG et al., 2018), as quais são:

- Não é restrita a um único algoritmo de aprendizagem
- Permite paralelização, caso necessário
- Permite manter o relacionamento dos rótulos

A escolha pela biclusterização se deu pelo fato de que ela consegue lidar com subconjuntos de exemplos e rótulos simultaneamente, o que ajuda a identificar coocorrências no conjunto de dados e permite gerar subproblemas onde os rótulos possuem uma relação real e não pressuposta, como é feito por outros algoritmos. Ainda, a sobreposição alcançada pelo Bimax apresenta-se útil para este tipo de transformação, visto que um exemplo do conjunto de dados pode aparecer em vários subproblemas binários e, de forma análoga, um conjunto de atributos também.

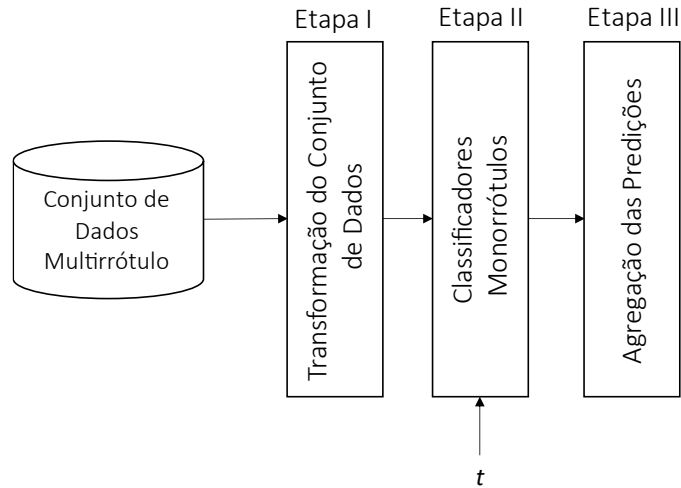
Também, no algoritmo proposto é possível flexibilizar a quantidade de subproblemas gerados, visto que o algoritmo de biclusterização vai aprender quais relações existem nos dados e quais são os rótulos que coocorrem em cada relação, diferentemente de outros algoritmos que baseiam a criação de subproblemas na quantidade de rótulos ou em parâmetros fixos.

É importante ressaltar que, assim como em outros algoritmos de classificação multirrótulo por transformação de problema, em nenhum momento é utilizado o espaço de características  $X$  do conjunto de dados para gerar os subproblemas, somente o espaço  $L$  de rótulos é utilizado nesta etapa.

Dessa forma, funções para pré-processamento no espaço de características devem ser aplicadas antes da transformação de problema ocorrer de fato, sendo o espaço de características apenas copiado para os subproblemas, pois este é necessário para a 2ª etapa do processo, onde são gerados os  $n$  classificadores monorrótulo.

De forma geral a Figura 20 ilustra o método de classificação multirrótulo por transformação de problema, no qual o algoritmo BicbPT está inserido. Cada etapa do método será detalhada ao longo desta seção, ilustrando como o BicbPT realiza cada uma.

Figura 20: Etapas do método de transformação de problema



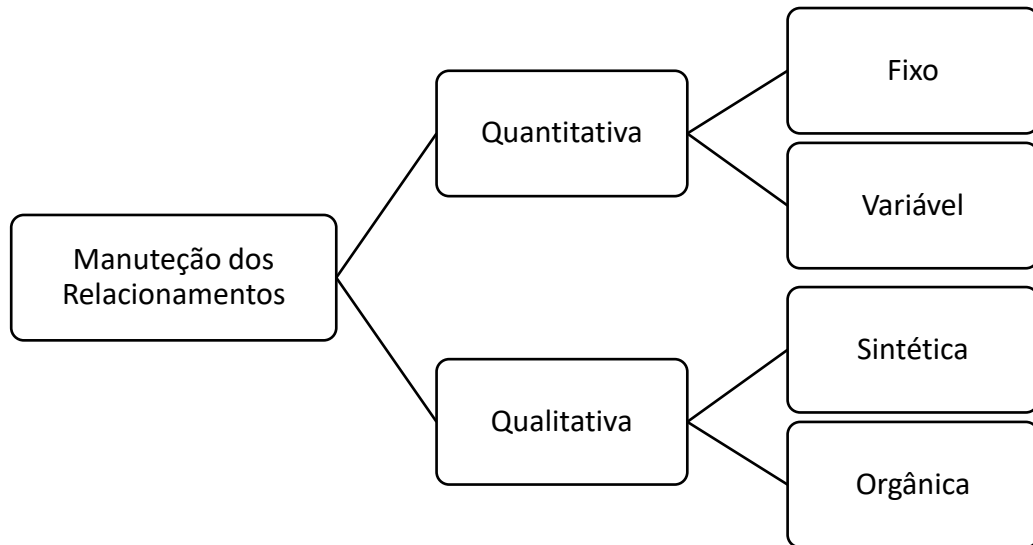
Fonte: Autoria Própria (2022)

## 5.1 MANUTENÇÃO DAS DEPENDÊNCIAS ENTRE RÓTULOS

Durante a fundamentação teórica e as etapas 1 e 2, apresentadas no Capítulo 3, verificou-se a inexistência de uma categorização para a forma como é feita a manutenção dos relacionamentos entre rótulos nos algoritmos de classificação multirrótulo por transformação de problema.

Isso é importante porque ajuda a entender o processo de manutenção das dependências entre rótulos sem a necessidade de ter de rastrear o código fonte de um algoritmo, deixando mais claro se o mesmo pressupõe ou não a existência de relacionamentos e sua forma de manutenção. A Figura 21 apresenta a categorização proposta neste trabalho.

Figura 21: Categorização da manutenção das relações entre rótulos



Fonte: Autoria Própria (2022)

Quando um algoritmo presume que os rótulos apresentam independência nos subproblemas, ou seja, não é considerado a existência de relações, somente é possível classificá-lo sob o aspecto quantitativo. Em contrapartida, quando se pressupõe a existência de relacionamentos entre dois ou mais rótulos é possível classificar de forma quantitativa e/ou qualitativa.

A manutenção na forma quantitativa diz respeito a quantidade de rótulos utilizada em cada subproblema, sendo dividida em:

- Fixo: a mesma quantidade de rótulos é utilizada em todos os subproblemas, mesmo que existam variações nos rótulos utilizados
- Variável: a quantidade de rótulos utilizada é variável nos subproblemas

Já a manutenção qualitativa considera a maneira como as relações entre os rótulos são mantidas nos subproblemas a partir do conjunto de dados multirrótulo, podendo ser:

- Orgânica: quando as relações são mantidas utilizando correlação, coocorrência ou outras medidas que permitam verificar se dois ou mais rótulos possuem uma relação no conjunto de dados
- Sintética: quando as relações são mantidas através de aleatoriedade, pela ordem com que os rótulos aparecem no conjunto multirrótulo ou outras formas onde não se verifique o relacionamento entre dois ou mais rótulos

O Quadro 4 demonstra a categorização dos algoritmos utilizados para comparação com o BicbPT e dele próprio.

Quadro 4: Forma de manutenção das relações nos algoritmos de transformação de problema

Algoritmo	Quantitativa	Qualitativa
BicbPT	Total	Orgânica
BR	Parcial	-
CC	Total	Sintética
ECC	Total	Sintética
RAkEL	Parcial	Sintética
LP	-	Sintética

Fonte: Autoria Própria (2022)

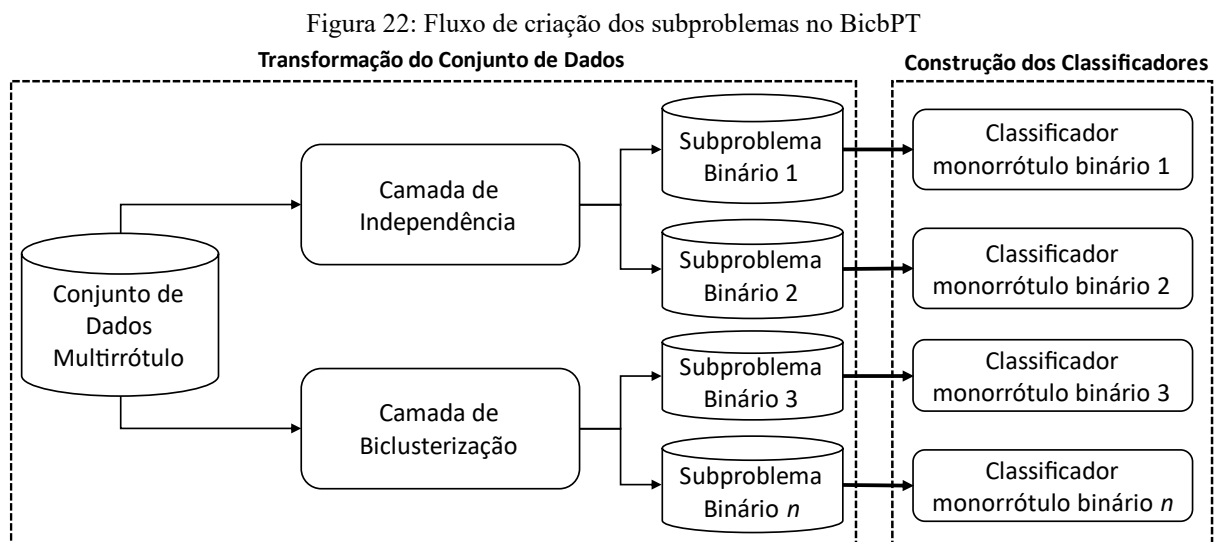
É importante observar que, como no LP a transformação é de multirrótulo para um único multiclasse, não existe a manutenção quantitativa, mas é possível considerar a manutenção qualitativa, visto que no processo de transformação de problema é pressuposto a existência de relacionamentos.

Já no BR, como os rótulos são tratados de maneira independente, ou seja, não se presume a existência de relacionamentos, somente é possível classificar quanto a manutenção quantitativa.

## 5.2 BicbPT: *BICLUSTER-BASED PROBLEM TRANSFORMATION*

Foram implementadas duas versões do BicbPT,  $\alpha$  e  $\beta$ , ambas com o mesmo fluxo, Figura 22, mas diferindo-se na camada de biclusterização, onde, na versão  $\alpha$  cada bicluster dá origem a um subproblema binário; e na versão  $\beta$  cada rótulo no bicluster dá origem a um subproblema binário, sendo os demais rótulos pertencentes ao espaço de características do subproblema.

Em linhas gerais, o conjunto de dados multirrótulo é fornecido para as camadas de independência, onde os rótulos de cada subproblema binário são tratados de forma independente, e biclusterização, onde é considerado a dependência entre os rótulos nos subproblemas. Posteriormente, Figura 22.II, é aplicado um algoritmo de classificação monorrótulo a cada subproblema binário construído na Figura 22.I.



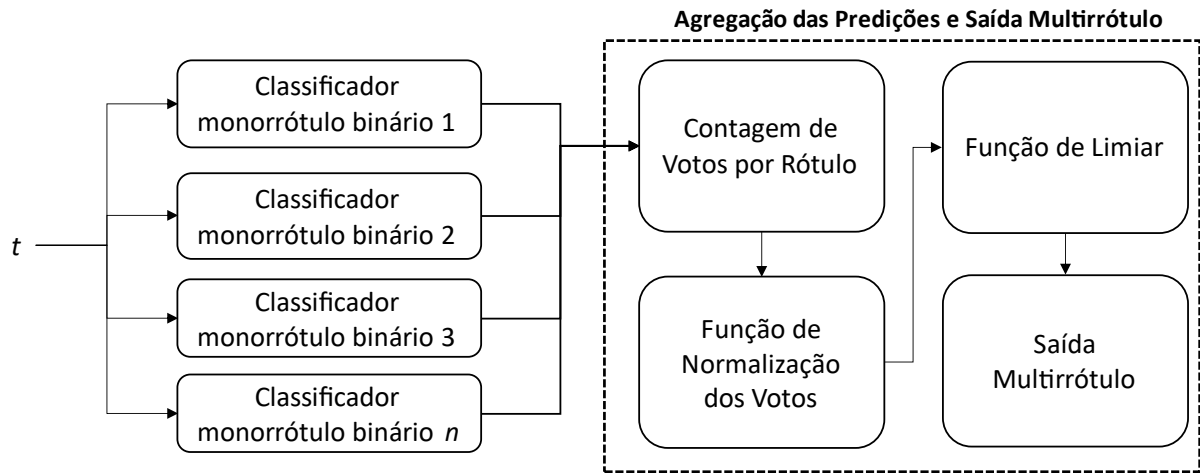
Fonte: Autoria Própria (2022)

Ao receber um novo exemplo  $t$  para classificação, Figura 23.I, ele é submetido a todos os classificadores binários construídos, onde cada classificador irá gerar uma saída 0 ou 1, indicando se  $t$  pode ou não possuir o rótulo do subproblema.

No final, para gerar a saída multirrótulo, Figura 23.II, são contabilizados os votos que cada rótulo em  $L$  recebeu, então é feita uma normalização para transformar os valores dos votos para o intervalo entre 0 e 1.

Com os votos normalizados é aplicado uma função de limiar, comum a todos os algoritmos de transformação de problema, que é responsável por indicar quais rótulos irão compor a saída multirrótulo final, a depender se o valor normalizado dos votos de cada rótulo está acima ou abaixo do limiar estabelecido.

Figura 23: Fluxo de classificação de um novo exemplo no BicbPT



Fonte: Autoria Própria (2022)

Para demonstração posterior do funcionamento das versões  $\alpha$  e  $\beta$  do BicbPT, será utilizado o micro conjunto de dados  $D$ , apresentado na Figura 24, o qual possui 6 rótulos e 6 exemplos e os biclusters apresentados na Figura 25, ao utilizar  $\rho = 0.3$ . O espaço de atributos  $X$  não é detalhado, pois como já comentado anteriormente, ele não é utilizado durante a etapa de construção dos subproblemas.

Figura 24: Micro conjunto de dados para exemplificação do BicbPT

T	X				Y					
	$x_1$	$x_2$	...	$x_j$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	0	0	0	0	0	1
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	1	0	1	0	0	1
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	0	0	0	0	1	1
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	0	1	1	1	0	0
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	1	0	1	1	0	1
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	1	1	0	0	0	0

$$L = \{y_1, y_2, y_3, y_4, y_5, y_6\}$$

Fonte: Autoria Própria (2022)

Figura 25: Exemplo de biclusters encontrados com o Bimax

T	X	
	$y_3$	$y_4$
$t_4$	1	1
$t_5$	1	1

Bicluster 0

T	X		
	$y_1$	$y_3$	$y_6$
$t_2$	1	1	1
$t_5$	1	1	1

Bicluster 1

Fonte: Autoria Própria (2022)

Ainda, a Figura 25 demonstra os rótulos  $y_3$  e  $y_4$  coocorrendo no bicluster 0 e os rótulos  $y_1$ ,  $y_3$  e  $y_6$  coocorrendo no bicluster 1. Assim, através da biclusterização é possível assumir que tais rótulos ocorrem sempre juntos nesse conjunto de dados multirrótulo, o que implica na existência de um relacionamento real entre eles, o que é explorado por ambas as versões do BicbPT.

### 5.3 HIPER PARÂMETRO $\rho$

O BicbPT possui um hiper parâmetro utilizado durante a biclusterização que deve ser informado, chamado de  $\rho$ . Este hiper parâmetro representa o percentual mínimo de exemplos que uma submatriz de uns deve ter para ser considerada um bicluster, e assim possa dar origem a subproblemas binários no BicbPT durante o processo de transformação de problema.

Valores baixos para  $\rho$  permitem encontrar um maior número de biclusters, e por consequência, uma quantidade maior de coocorrências, o que é benéfico, visto que permite um melhor uso dos relacionamentos pré-existentes entre os rótulos durante o processo de criação dos subproblemas binários e na classificação multirrótulo.

Já, valores altos restringem a quantidade de biclusters que podem ser encontrados nos dados multirrótulo, afetando diretamente a classificação, de forma que se  $\rho$  for muito elevado e nenhum bicluster for encontrado, o BicbPT se comportará de maneira análoga ao algoritmo BR.

Uma análise da sensibilidade deste hiper parâmetro no resultado da classificação multirrótulo é feita na Seção 6.5.

### 5.4 CAMADA DE INDEPENDÊNCIA

Esta camada, apresentada na Figura 22.I, é responsável por gerar subproblemas binários partindo-se da premissa de que todos os rótulos no conjunto original são independentes, ou seja, não existem relacionamentos entre eles. Isso é importante, porque deve-se considerar não somente a existência de relacionamentos entre os rótulos, mas também a possibilidade de que estes não existam.

Assim, um conjunto de dados multirrótulo,  $D$ , submetido ao BicbPT passa pela camada de independência, onde então é particionado em  $n$  subproblemas binários, de tal forma que o  $Y$  em cada subproblema possua somente um único rótulo  $y_i$  dos possíveis em  $L$ , de forma que  $Y = y_i$ .



Por fim, o espaço  $X$  de características é copiado de  $D$  para cada subproblema. Todo o processo é ilustrado no algoritmo apresentado na Figura 26.

Figura 26: Algoritmo da camada de independência

---

**Algoritmo 1** Camada de independência

---

**Entrada:**  $D$

```

1: subproblema  $\leftarrow \{\}$ 
2: for  $i \leftarrow 0$  to  $L$  do
3:   subproblema $_{iX}$   $\leftarrow D_X$ 
4:   subproblema $_{iY}$   $\leftarrow y_i$ 
5: end for
6: return subproblema

```

---

Fonte: Autoria Própria (2022)

Utilizando-se do micro conjunto de dados apresentado na Figura 24, os subproblemas binários resultantes da camada de independência serão os apresentados na Figura 27.

Figura 27: Exemplo de subproblemas gerados na camada de independência

T	X				Y
	$x_1$	$x_2$	...	$x_j$	$y_1$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	0
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	1
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	0
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	0
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	1
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	1

Subproblema 0

T	X				Y
	$x_1$	$x_2$	...	$x_j$	$y_2$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	0
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	0
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	0
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	1
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	0
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	1

Subproblema 1

T	X				Y
	$x_1$	$x_2$	...	$x_j$	$y_3$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	0
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	1
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	0
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	1
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	1
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	0

Subproblema 2

T	X				Y
	$x_1$	$x_2$	...	$x_j$	$y_4$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	0
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	0
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	0
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	1
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	1
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	0

Subproblema 3

T	X				Y
	$x_1$	$x_2$	...	$x_j$	$y_5$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	0
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	0
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	1
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	0
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	0
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	0

Subproblema 4

T	X				Y
	$x_1$	$x_2$	...	$x_j$	$y_6$
$t_1$	$x_{11}$	$x_{12}$	...	$x_{1j}$	1
$t_2$	$x_{21}$	$x_{22}$	...	$x_{2j}$	1
$t_3$	$x_{31}$	$x_{32}$	...	$x_{3j}$	1
$t_4$	$x_{41}$	$x_{42}$	...	$x_{4j}$	0
$t_5$	$x_{51}$	$x_{52}$	...	$x_{5j}$	1
$t_6$	$x_{61}$	$x_{62}$	...	$x_{6j}$	0

Subproblema 5

Fonte: Autoria Própria (2022)

## 5.5 CAMADA DE BICLUSTERIZAÇÃO: BicbPT $\alpha$

A Figura 28 apresenta o algoritmo da versão  $\alpha$  do BicbPT, onde o conjunto multirrótulo  $D$  é submetido a camada de biclusterização e é particionado em  $n$  biclusters, onde, cada bicluster representa um subproblema binário que tem como  $Y$  a combinação do conjunto de atributos  $X$  do bicluster.

Os exemplos positivos, ou seja, com valor 1 em  $Y$  são aqueles encontrados no bicluster, indicados no algoritmo por  $bicluster_{iT}$ , e os exemplos negativos, ou seja, com valor 0 em  $Y$  são aqueles que estão nos dados originais, mas não estão no bicluster, expresso por  $D_T - bicluster_{iT}$ .

Figura 28: Algoritmo da camada de biclusterização da versão BicbPT $\alpha$

---

**Algoritmo 2** Camada de biclusterização - BicbPT $\alpha$

---

**Entrada:**  $D, \rho$

```

subproblema  $\leftarrow$  {}
1: biclusters  $\leftarrow$  Bimax( $D, \rho$ )
2: for  $i \leftarrow 0$  to biclusters do
3:   subproblema $_{iY}$   $\leftarrow$  concatenar( $bicluster_{iX}$ )
4:   subproblema $_{iX}$   $\leftarrow$   $D_X$ 
5:   subproblema $_{iT}$   $\leftarrow$   $bicluster_{iT} \cup (D_T - bicluster_{iT})$ 
6: end for
7: return subproblema

```

---

Fonte: Aatoria Própria (2022)

Utilizando o micro conjunto de dados, Figura 24, e os biclusters encontrados pelo algoritmo Bimax, Figura 25, onde tem-se a coocorrência dos rótulos  $y_3$  e  $y_4$  no Bicluster 0 e  $y_1$ ,  $y_3$  e  $y_6$  no Bicluster 1 é possível obter os subproblemas binários, Figura 29, de forma que o espaço  $X$  de atributos do bicluster tornar-se-á a classe  $Y$  do subproblema, os exemplos positivos serão aqueles que estão no bicluster e os exemplos negativos são aqueles que estão fora do bicluster.

Figura 29: Exemplo de subproblemas gerados na camada de biclusterização no BicbPT $\alpha$

T	X				Y
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>3</sub> .y <sub>4</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	0
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	1
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	0

Subproblema 6

T	X				Y
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>1</sub> .y <sub>3</sub> .y <sub>6</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	1
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	0
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	0

Subproblema 7

Fonte: Autoria Própria (2022)

## 5.6 CAMADA DE BICLUSTERIZAÇÃO: BicbPT $\beta$

Na Figura 30 é apresentado o algoritmo da versão  $\beta$  do BicbPT, onde o conjunto  $D$  é submetido a camada de biclusterização e é particionado em  $n$  biclusters. Cada atributo  $X$  do bicluster da origem a um subproblema binário, de forma que os atributos restantes do bicluster componham o espaço de atributos  $X$  do subproblema que tem  $Y$  como classe.

Os exemplos positivos, ou seja, com valor 1 para  $Y$  são aqueles encontrados no bicluster, e os exemplos negativos, ou seja, com valor 0 para  $Y$  são aqueles que estão no conjunto de dados multirrótulo, mas não estão no bicluster, expresso por  $D_T - \text{bicluster}_T$ .

É importante destacar que os exemplos positivos e negativos são os mesmos em todos os subproblemas binários gerados a partir de cada bicluster.

Figura 30: Algoritmo da camada de biclusterização da versão BicbPT $\beta$

---

### Algoritmo 3 Camada de biclusterização - BicbPT $\beta$

---

**Entrada:**  $D, \rho$

```

1: subproblema  $\leftarrow \{\}$ ,  $i \leftarrow 0$ 
2: biclusters  $\leftarrow \text{Bimax}(D, \rho)$ 
3: for each bicluster  $\subset$  biclusters do
4:   for each  $x \in \text{bicluster}_x$  do
5:     subproblema $iY$  =  $x$ 
6:     subproblema $x$  =  $D_x \cup (\text{bicluster}_x - x)$ 
7:     subproblema $iT$   $\leftarrow \text{bicluster}_T \cup (D_T - \text{bicluster}_T)$ 
8:      $i \leftarrow i + 1$ 
9:   end for each
10: end for each
11: return subproblema

```

---

Fonte: Autoria Própria (2022)

Utilizando o micro conjunto de dados, Figura 24, e os biclusters apresentados na Figura 25 para a construção dos subproblemas na versão  $\beta$  do BicbPT, produzir-se-ão um total de cinco biclusters e não somente dois, como na versão  $\alpha$ , visto que aqui cada atributo do bicluster será transformado em um subproblema binário, como pode ser visto na Figura 31.

Figura 31: Exemplo de subproblemas gerados na camada de biclusterização no BicbPT $\beta$

T	X				Y	
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>4</sub>	y <sub>3</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	0	0
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	1	1
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	0	0

Subproblema 6

T	X				Y	
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>3</sub>	y <sub>4</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	1	0
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	1	1
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	0	0

Subproblema 7

T	X				Y		
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>3</sub>	y <sub>6</sub>	y <sub>1</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0	1	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	1	1	1
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0	1	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	1	0	0
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1	1	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	0	0	1

Subproblema 8

T	X				Y		
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>1</sub>	y <sub>6</sub>	y <sub>3</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0	1	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	1	1	1
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0	1	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	0	0	0
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1	1	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	1	0	0

Subproblema 9

T	X				Y		
	x <sub>1</sub>	x <sub>2</sub>	...	x <sub>j</sub>	y <sub>1</sub>	y <sub>3</sub>	y <sub>6</sub>
t <sub>1</sub>	x <sub>11</sub>	x <sub>12</sub>	...	x <sub>1j</sub>	0	0	0
t <sub>2</sub>	x <sub>21</sub>	x <sub>22</sub>	...	x <sub>2j</sub>	1	1	1
t <sub>3</sub>	x <sub>31</sub>	x <sub>32</sub>	...	x <sub>3j</sub>	0	0	0
t <sub>4</sub>	x <sub>41</sub>	x <sub>42</sub>	...	x <sub>4j</sub>	0	1	0
t <sub>5</sub>	x <sub>51</sub>	x <sub>52</sub>	...	x <sub>5j</sub>	1	1	1
t <sub>6</sub>	x <sub>61</sub>	x <sub>62</sub>	...	x <sub>6j</sub>	1	0	0

Subproblema 10

Fonte: Autoria Própria (2022)

Diferentemente da versão  $\alpha$ , onde a coocorrência dos rótulos em cada bicluster dá origem a um subproblema, na versão  $\beta$ , a coocorrência encontrada é utilizada para construir subproblemas considerando como a relação de dependência ocorre, de forma que seja possível avaliar a relação no formato  $AB \rightarrow C$ , assim os valores dos rótulos A e B no espaço de atributos  $X$  do subproblema influenciarão se a saída será  $C$  ou  $\sim C$  durante a classificação.

## 5.7 AGREGAÇÃO DAS PREDIÇÕES

Esta etapa, Figura 23, presente nos algoritmos de classificação multirrótulo por transformação de problema, é idêntica em ambas as versões do BicbPT, de maneira que, após gerados os subproblemas binários é aplicado um único algoritmo de classificação monorrótulo em cada um e, ao final, as saídas previstas de cada classificador, Figura 23.I, são agrupadas utilizando a estratégia de voto majoritário.

Vale destacar que, outras estratégias de votos para as saídas dos classificadores monorrótulos foram testadas, como voto ponderado e o valor absoluto das probabilidades retornadas pelos classificadores, mas nestas o desempenho nas métricas do BicbPT, em ambas as versões, era degradado.

Além dos votos, para poder gerar a saída multirrótulo é necessário normalizar os votos, Equação 5.1, deixando-os em um intervalo entre 0 e 1, para que seja possível posteriormente utilizar a função de limiar, Equação 5.2, que é responsável por indicar quais rótulos irão compor a saída multirrótulo.

Por exemplo, dado um conjunto  $X = \{4, 2, 2, 4, 6, 1\}$  de votos por rótulo,  $\max(X)$  será 6, então os votos de cada rótulo em  $X$  serão normalizados para os valores  $X = \{0.66, 0.33, 0.33, 0.66, 1, 0.16\}$  através da Equação 5.1.

A saída multirrótulo dada pela Equação 5.2, depende do limiar adotado, mas para exemplificação, adotar-se-á o valor de 0.5. Dessa forma a saída multirrótulo gerada no conjunto  $X$  será  $\{1, 0, 0, 1, 1, 0\}$ .

$$x_i = \frac{x_i}{\max(X)} \quad 5.1$$

$$f(x_i) = \begin{cases} 1, & x_i \geq \text{limiar} \\ 0, & x_i < \text{limiar} \end{cases} \quad 5.2$$

O Quadro 5 exemplifica a classificação de um novo exemplo  $t$  no BicbPT $\alpha$  considerando os subproblemas binários presentes na Figura 27 (Camada de Independência) e na Figura 29 (Camada de Biclusterização - BicbPT $\alpha$ ).

Os classificadores monorrótulo que geraram como saída o valor 1 para  $t$  foram os classificadores  $y_2, y_3, y_4$  e  $y_6$ , produzidos pela camada de independência, e os classificadores  $\{y_3.y_4\}$  e  $\{y_1.y_3.y_6\}$ , produzidos pela camada de biclusterização no BicbPT $\alpha$ .

Quadro 5: Exemplos de classificação no BicbPT $\alpha$  com limiar igual a 0.5

Predições	Votos	Normalização	Saída
$y_2$	$y_1=1$	$y_1=0,3$	y <sub>3</sub> y <sub>4</sub> y <sub>6</sub>
$y_3$	$y_2=1$	$y_2=0,3$	
$y_4$	$y_3=3$	$y_3=1$	
$y_6$	$y_4=2$	$y_4=0,6$	
{ $y_3.y_4$ }	$y_6=2$	$y_6=0,6$	
{ $y_1.y_3.y_6$ }			

Fonte: Autoria Própria (2022)

Então são contabilizados os votos para cada rótulo, sendo que, no caso dos classificadores oriundos da camada de biclusterização no BicbPT $\alpha$  é contabilizado um voto para rótulo na relação de dependência, por isso, por exemplo  $y_3$  recebeu 3 votos, um do classificador da camada de independência e dois nas relações { $y_3.y_4$ } e { $y_1.y_3.y_6$ }.

Após é feita a normalização, como explicado na Equação 5.1, e adotando-se o limiar de 0.5, é produzido a saída multirrótulo, como explicado na Equação 5.2.

Já o Quadro 6 exemplifica a classificação no BicbPT $\beta$  considerando os subproblemas apresentados na Figura 27 (Camada de Independência) e na Figura 31 (Camada de Biclusterização - BicbPT $\beta$ ).

A única diferença no BicbPT $\beta$  nesta etapa é durante a contabilização dos votos, pois, diferentemente do BicbPT $\alpha$  que considera cada rótulo da relação para contabilizar os votos, aqui apenas o rótulo que representa  $Y$  no classificador monorrótulo recebe o voto, pois os demais rótulos da relação fazem parte do espaço  $X$  de características.

No exemplo ilustrado no Quadro 6, os classificadores que produziram a saída 1 para  $t$  foram  $y_2$ ,  $y_4$  e  $y_5$ , produzidos pela camada de independência, e { $y_4$ }, produzidos pela camada de biclusterização no BicbPT $\beta$ .

Quadro 6: Exemplos de classificações no BicbPT $\beta$  com limiar igual a 0.5

Predições	Votos	Normalização	Saída
$y_2$	$y_2=1$	$y_2=0,5$	y <sub>2</sub> y <sub>4</sub> y <sub>5</sub>
$y_4$	$y_4=2$	$y_4=1$	
$y_5$	$y_5=1$	$y_5=0,5$	
{ $y_4$ }			

Fonte: Autoria Própria (2022)

## 5.8 LIMITAÇÕES

O BicbPT, em ambas as versões, não realiza nenhuma tarefa de seleção ou extração de características internamente, ou seja, caso necessário, o usuário deve utilizar algum algoritmo de seleção ou extração de características no conjunto de dados multirrótulo antes de submeter ao BicbPT.

O algoritmo proposto realiza uma tarefa de classificação binária internamente, ou seja, todos os rótulos devem possuir os valores 1 ou 0. Isso deve-se a característica de funcionamento do algoritmo Bimax, utilizado na biclusterização, o qual busca por bicluster em matrizes binárias. Tal característica do BicbPT não impede o uso de rótulos não-binários, visto que é possível transformar dados não-binários para binários de várias maneiras, sendo a mais simples utilizar uma função de limiar, onde caso o valor esteja acima do limiar ele se torna 1 e abaixo torna-se 0.

Também, não há suporte para valores faltantes nos rótulos, dessa forma, exemplos onde os valores dos rótulos estão ausentes precisam ser excluídos ou receber algum tratamento antes de serem submetidos ao BicbPT. O suporte para valores faltantes no espaço  $X$  de características depende do algoritmo monorrótulo utilizado, visto que algoritmos de classificação multirrótulo por transformação de problema não alteram esse espaço, assim como o BicbPT.

## 6 AVALIAÇÃO EXPERIMENTAL

Neste capítulo são apresentados os conjuntos de dados multirrótulo utilizados para realização dos experimentos com ambas as versões do BicbPT, os parâmetros dos outros algoritmos de classificação multirrótulo por transformação de problema comparados e as métricas multirrótulo coletadas.

Também são apresentados os resultados e discussões dos experimentos descritos em 3.3, os quais são:

- I. BicbPT $\alpha$  x {BR, CC, ECC, RAKEL, LP}
- II. BicbPT $\beta$  x {BR, CC, ECC, RAKEL, LP}
- III. BicbPT $\alpha$  x BicbPT $\beta$
- IV. Análise de sensibilidade do hiper parâmetro  $\rho$  do BicbPT

### 6.1 PARÂMETROS DOS EXPERIMENTOS

Para a realização dos experimentos foram utilizados os conjuntos de dados multirrótulo disponíveis no website do framework Mulan (TSOUMAKAS et al., 2011), os quais são apresentados na Tabela 3 e para realização dos testes optou-se por utilizar a estratégia de *5 x 2-folds Cross-Validation* (DIETTERICH, 1998).

Na Tabela 3, os cabeçalhos das colunas indicam, para cada conjunto de dados multirrótulo:

- n: quantidade de exemplos
- X: quantidade de atributos (características)
- L: quantidade de rótulos
- Cardinalidade: quantidade média de rótulos/exemplo
- Densidade: proporção de rótulos disponíveis que estão sendo utilizados
- Complexidade: custo computacional de se utilizar a base de dados

A coluna complexidade da Tabela 3 é uma adaptação da métrica complexidade apresentada anteriormente em 2.2.3, sendo obtida multiplicando-se o seu valor pela métrica densidade.

Essa adaptação foi necessária, pois, notou-se durante o desenvolvimento do trabalho que a métrica complexidade, em 2.2.3, ao não considerar a densidade no seu cálculo tratava de maneira distinta bases similares, como, por exemplo, Bibtext e BFRC.



Tabela 3: Bases de dados selecionadas para a realização de experimentos

Bases	Domínio	n	X	L	Cardinalidade	Densidade	Complexidade
<b>Birds</b>	Áudio	645	260	19	1,01	0,05	0,17 M
<b>Emotions</b>	Áudio	593	71	6	1,87	0,31	0,08 M
<b>Enron</b>	Texto	1702	1001	53	3,38	0,06	5,78 M
<b>Flags</b>	Imagem	194	19	7	3,39	0,49	0,01 M
<b>Genbase</b>	Biologia	662	1186	27	1,25	0,05	0,98 M
<b>Medical</b>	Texto	978	1449	45	1,25	0,03	1,79 M
<b>Scene</b>	Imagem	2407	294	6	1,07	0,18	0,76 M
<b>Yeast</b>	Biologia	2417	103	14	4,24	0,30	1,06 M
<b>Bibtex</b>	Texto	7395	1836	159	2,40	0,02	32,61 M
<b>G1</b>	Texto	2000	1013	7	1,96	0,28	3,97 M
<b>GP</b>	Texto	668	997	6	2,16	0,36	1,44 M
<b>BFRC</b>	Texto	7879	994	8	3,93	0,49	30,78 M

Fonte: Autoria Própria (2022)

Também, vale destacar que as três últimas bases de dados listadas, G1, GP e BFRC, pertencem ao grupo de pesquisa de Descoberta de Conhecimento e Aprendizagem de Máquina (DCAM) do Programa de Pós-Graduação em Informática da PUCPR (PPGIA) (CURI; BRITTO JR; PARAISO, 2018; CURI; BRITTO JR.; PARAISO, 2019; DOSCIATTI; FERREIRA; PARAISO, 2015).

Estas três bases foram transformadas para o formato ARFF, para serem utilizadas no MEKA, após foi feito um pré-processamento com o filtro *StringToWordVector* (*weka – filters – unsupervised – attribute*) com os parâmetros listados no Quadro 7 e também foram removidas *stop words*.

Quadro 7: Lista de parâmetros para pré-processamento das bases de texto

Parâmetro	Valor
<i>IDFTransform</i>	True
<i>TFTransform</i>	True
<i>lowerCaseTokens</i>	True
<i>stemmer</i>	<i>SnowballStemmer{Portuguese}</i>
<i>tokenizer</i>	{.,:;!?)[]{}'"%/+-*@}

Fonte: Autoria Própria (2022)

Os parâmetros *IDFTransform* e *TFTransform* compõe uma métrica chamada TF-IDF (*Term Frequency-Inverse Document Frequency*), que é responsável por indicar a importância de uma palavra de um documento em relação a uma coleção de documentos. Ambos estão

marcados como *true*, pois TF-IDF é uma métrica amplamente utilizada no pré-processamento de textos.

*LowerCaseTokens* indica que todas as palavras serão deixadas em minúsculo antes de executar o filtro, o que é pertinente, pois evita problemas onde uma mesma palavra pode estar em maiúsculo e minúsculo no decorrer do texto.

*Stemmer* indica qual algoritmo de estemização será utilizado, sendo que na estemização a ideia é reduzir as palavras aos seus radicais, assim, as palavras *amigos* e *amigas* são reduzidas para *amig*. O algoritmo utilizado para esta tarefa foi o *SnowballStemmer* com a língua portuguesa.

Por fim, *tokenizer* indica quais caracteres foram utilizados para quebrar o texto em pedaços menores, chamados *tokens*.

As métricas multirrótulo escolhidas para serem coletadas dos experimentos são Jaccard Index (JI), F<sub>1</sub>-macro e micro, Hamming Loss (HL) e Tempo Total (TT) (treinamento + teste), as quais são apresentadas em 2.2.2 e são as mais frequentemente utilizadas, como foi possível observar na pesquisa exploratória explanada em 3.1. É importante destacar que para todas as métricas é calculada a média das 10 execuções, 5 x 2 *folds*, para realizar as comparações e testes estatísticos.

Os algoritmos escolhidos como base para os subproblemas foram J48 (C4.5), Naive Bayes e SVM com as parametrizações padrão, pois são os mais frequentemente escolhidos quando é realizada a classificação multirrótulo por transformação de problema, como foi possível observar na pesquisa exploratória.

Os algoritmos de transformação de problema multirrótulo-binário escolhidos para comparação foram BR, CC e ECC; e os multirrótulo não-binários foram RAKEL<sub>n</sub> e LP, todos com as configurações padrão.

O *n* em RAKEL<sub>n</sub> indica qual a configuração dos hiper parâmetros *m* (quantidade de subproblemas) e *k* (quantidade de rótulos por subproblema) do algoritmo, sendo que, quando *n* = 1, *m* = 10 e *k* = L/2; e para *n* = 2, *m* = L\*2, *k* = 3.

No Meka foi utilizado o módulo *Experimenter* para realização dos experimentos, sendo os parâmetros do *Setup (expert)* definidos como: *evaluator* = {*RepeatedRuns* [*CrossValidation*; *numFolds*=2; *numThreads*=1; *seed*=1; *threshold*=0.5]; *lowerRuns*=1; *upperRuns*=5}.

- *evaluator*: indica como a avaliação deve ser executada
- *RepeatedRuns*: indica que a avaliação será repetida *n* vezes

- *CrossValidation*: indica que será utilizada a validação cruzada
- *numFolds*: indica a quantidade de *folders* na validação cruzada
- *numThreads*: indica a quantidade de *threads* de processamento foram utilizadas
- *seed*: indica o valor da semente para trabalhar com a aleatoriedade
- *threshold*: indica o valor do limiar utilizado
- *lowerRuns*: quantidade mínima de execuções
- *upperRuns*: quantidade máxima de execuções

Após coletadas as métricas, foram utilizados os testes estatísticos não-paramétricos de Friedman e Nemenyi para comparar o algoritmo proposto com outros algoritmos de transformação de problema escolhidos. Também foi utilizado o teste estatístico não-paramétrico Wilcoxon *Signed-Rank* para comparar as duas versões do BicbPT.

6.2 EXPERIMENTO I: BicbPT $\alpha$  x {BR, CC, ECC, RAKEL<sub>1</sub>, RAKEL<sub>2</sub>, LP}

Nesta seção são apresentados os resultados com os experimentos comparando o BicbPT $\alpha$  com os algoritmos descritos anteriormente no Capítulo 2.2 apresentando-se os classificadores monorrótulo base na seguinte ordem, C4.5, SVM e por último o Naive Bayes.

Como observa-se na Tabela 4, JI com C4.5, ao considerar somente os valores absolutos alcançados por algoritmos de transformação multirrótulo-binário, o BicbPT $\alpha$  foi melhor somente em 2 de 12 bases nesta métrica, e no geral somente em 1 de 12.

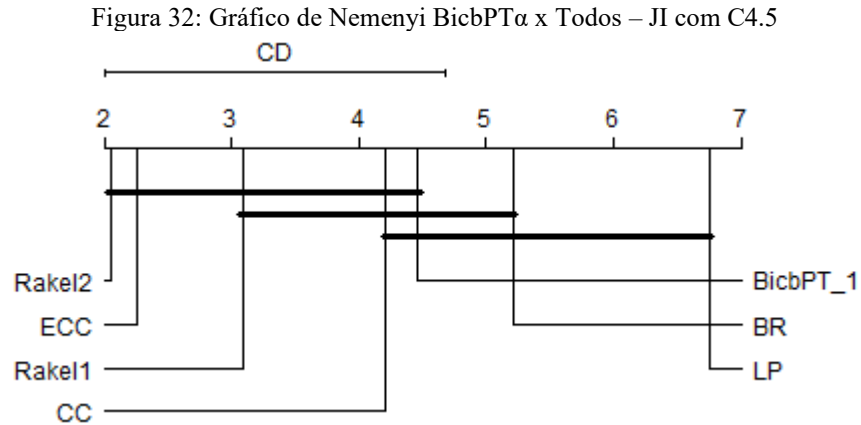
Tabela 4: BicbPT $\alpha$  x Todos – JI com C4.5

Bases	JI $\uparrow$						
	BicbPT $\alpha$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	0,548 $\pm$ 0,03	0,548 $\pm$ 0,03	0,553 $\pm$ 0,02	0,565 $\pm$ 0,01	0,561 $\pm$ 0,02	0,572 $\pm$ 0,02	0,491 $\pm$ 0,03
<b>Emotions</b>	0,465 $\pm$ 0,01	0,437 $\pm$ 0,01	0,437 $\pm$ 0,02	0,5 $\pm$ 0,01	0,492 $\pm$ 0,02	0,479 $\pm$ 0,02	0,427 $\pm$ 0,02
<b>Enron</b>	0,396 $\pm$ 0,01	0,392 $\pm$ 0,01	0,401 $\pm$ 0,01	0,45 $\pm$ 0,01	0,378 $\pm$ 0,01	0,416 $\pm$ 0,02	0,189 $\pm$ 0,02
<b>Flags</b>	0,569 $\pm$ 0,02	0,561 $\pm$ 0,03	0,574 $\pm$ 0,02	0,598 $\pm$ 0,03	0,603 $\pm$ 0,02	0,613 $\pm$ 0,01	0,436 $\pm$ 0,06
<b>Genbase</b>	0,976 $\pm$ 0,01	0,976 $\pm$ 0,01	0,976 $\pm$ 0,01	0,973 $\pm$ 0,01	0,974 $\pm$ 0,01	0,976 $\pm$ 0,01	0,97 $\pm$ 0,01
<b>Medical</b>	0,72 $\pm$ 0,03	0,721 $\pm$ 0,03	0,725 $\pm$ 0,03	0,743 $\pm$ 0,03	0,75 $\pm$ 0,02	0,723 $\pm$ 0,03	0,701 $\pm$ 0,02
<b>Scene</b>	0,515 $\pm$ 0,01	0,514 $\pm$ 0,01	0,58 $\pm$ 0,01	0,609 $\pm$ 0,01	0,608 $\pm$ 0,01	0,602 $\pm$ 0,01	0,566 $\pm$ 0,01
<b>Yeast</b>	0,419 $\pm$ 0,02	0,419 $\pm$ 0,02	0,417 $\pm$ 0,01	0,486 $\pm$ 0,01	0,46 $\pm$ 0,01	0,471 $\pm$ 0,01	0,382 $\pm$ 0,01
<b>Bibtex</b>	0,287 $\pm$ 0,01	0,287 $\pm$ 0,01	0,285 $\pm$ 0,01	0,287 $\pm$ 0,01	0,283 $\pm$ 0	0,29 $\pm$ 0,01	0,181 $\pm$ 0,01
<b>G1</b>	0,312 $\pm$ 0,01	0,307 $\pm$ 0,01	0,357 $\pm$ 0,01	0,343 $\pm$ 0,01	0,38 $\pm$ 0,01	0,387 $\pm$ 0,01	0,309 $\pm$ 0,01
<b>GP</b>	0,413 $\pm$ 0,02	0,399 $\pm$ 0,02	0,4 $\pm$ 0,01	0,482 $\pm$ 0,02	0,466 $\pm$ 0,01	0,455 $\pm$ 0,02	0,385 $\pm$ 0,02
<b>BFRC</b>	0,45 $\pm$ 0	0,448 $\pm$ 0	0,448 $\pm$ 0	0,486 $\pm$ 0	0,477 $\pm$ 0	0,489 $\pm$ 0	0,421 $\pm$ 0,01

Fonte: Autoria Própria (2022)

O p-valor com 95% de confiança no teste de Friedman é de 4.779e-08, o que indica que existe diferença significativa entre os algoritmos comparados, a qual é apontada no teste de Nemenyi, Figura 32, onde é possível observar que o BicbPT $\alpha$  apresenta-se em vários grupos, o que indica que, mesmo não alcançando valores absolutos para JI tão bons quantos o ECC, estatisticamente eles são semelhantes.

Essa semelhança estatística é positiva para o BicbPT $\alpha$ , pois quanto trata-se do tempo total de execução, o ECC possui um tempo muito maior que este.



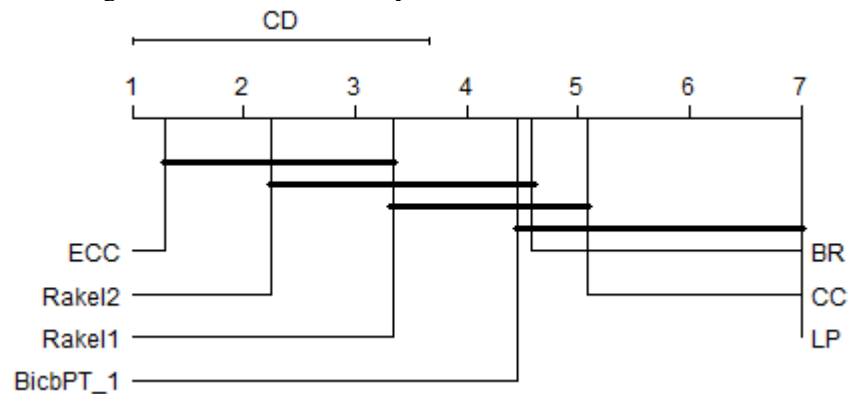
Fonte: Autoria Própria (2022)

Para a métrica HL com o C4.5, Tabela 5, o BicbPT $\alpha$  conseguiu bons resultados somente em 1 das 12 bases, mas ao analisar o gráfico, Figura 33, do teste de Nemenyi é possível verificar que o algoritmo proposto se posiciona de maneira intermediária entre os algoritmos comparados, indicando ser uma alternativa escalável quando pensa-se no tempo de execução e no desempenho na métrica, visto que ECC, RAKEL<sub>1</sub> e RAKEL<sub>2</sub> possuem tempo mais elevado. O p-valor com 95% de confiança em Friedman obtido aqui foi de 2.131e-11.

Tabela 5: BicbPT $\alpha$  x Todos – HL com C4.5

Bases	HL ↓						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
Birds	0,054 ± 0	0,054 ± 0	0,054 ± 0	0,045 ± 0	0,052 ± 0	0,048 ± 0	0,281 ± 0,03
Emotions	0,265 ± 0,01	0,26 ± 0,01	0,275 ± 0,01	0,208 ± 0,01	0,24 ± 0,01	0,232 ± 0,01	0,334 ± 0,02
Enron	0,054 ± 0	0,054 ± 0	0,054 ± 0	0,048 ± 0	0,057 ± 0	0,05 ± 0	0,657 ± 0,03
Flags	0,271 ± 0,01	0,279 ± 0,02	0,281 ± 0,02	0,261 ± 0,02	0,263 ± 0,02	0,252 ± 0,01	0,466 ± 0,06
Genbase	0,002 ± 0	0,002 ± 0	0,002 ± 0	0,002 ± 0	0,003 ± 0	0,002 ± 0	0,005 ± 0
Medical	0,012 ± 0	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,087 ± 0,02
Scene	0,143 ± 0	0,14 ± 0	0,146 ± 0	0,097 ± 0	0,109 ± 0	0,107 ± 0	0,175 ± 0,01
Yeast	0,255 ± 0,01	0,255 ± 0,01	0,274 ± 0	0,209 ± 0	0,242 ± 0	0,232 ± 0	0,394 ± 0,01
Bibtex	0,015 ± 0	0,015 ± 0	0,015 ± 0	0,013 ± 0	0,014 ± 0	0,014 ± 0	0,578 ± 0,01
G1	0,273 ± 0,01	0,273 ± 0,01	0,286 ± 0,01	0,253 ± 0	0,266 ± 0,01	0,258 ± 0,01	0,495 ± 0,01
GP	0,329 ± 0,01	0,332 ± 0,01	0,332 ± 0,01	0,26 ± 0,01	0,295 ± 0,01	0,285 ± 0,01	0,428 ± 0,02
BFRC	0,382 ± 0	0,383 ± 0	0,384 ± 0	0,336 ± 0	0,368 ± 0	0,358 ± 0	0,461 ± 0

Fonte: Autoria Própria (2022)

Figura 33: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – HL com C4.5

Fonte: Autoria Própria (2022)

Para a métrica  $F1_{\text{micro}}$ , Tabela 6, o algoritmo proposto conseguiu bom desempenho em 3 de 12 bases em relação aos algoritmos multirrótulo-binário, BR, CC e ECC, e somente em 1 de 12 no geral.

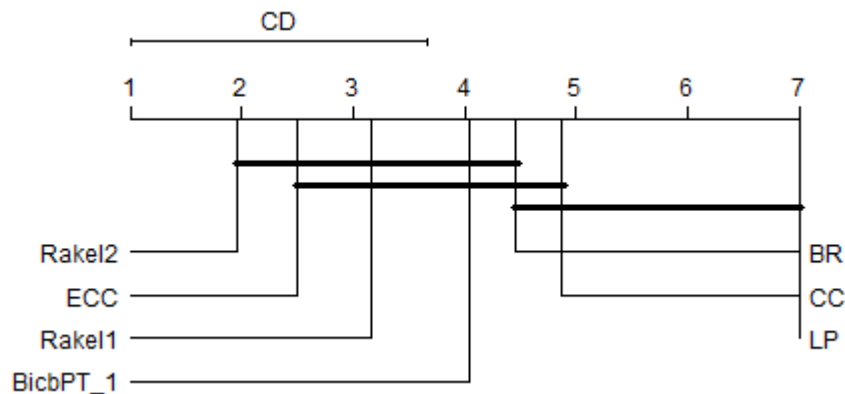
De maneira análoga a métrica JI, aqui o ECC se saiu melhor na maior parte das bases também, mas deve-se lembrar que seu tempo de execução é maior que o do BicbPT $\alpha$ , sendo assim um ponto positivo a ser considerado.

Tabela 6: BicbPT $\alpha$  x Todos – F1<sub>micro</sub> com C4.5

Bases	F1 <sub>micro</sub> ↑						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,378 ± 0,04	0,378 ± 0,04	0,382 ± 0,03	0,36 ± 0,02	0,398 ± 0,03	0,394 ± 0,03	0,191 ± 0,01
<b>Emotions</b>	0,604 ± 0,01	0,584 ± 0,01	0,558 ± 0,02	0,639 ± 0,01	0,623 ± 0,02	0,614 ± 0,02	0,535 ± 0,02
<b>Enron</b>	0,524 ± 0,01	0,519 ± 0,01	0,516 ± 0,01	0,566 ± 0,01	0,49 ± 0,01	0,55 ± 0,01	0,123 ± 0
<b>Flags</b>	0,716 ± 0,02	0,712 ± 0,02	0,71 ± 0,02	0,733 ± 0,02	0,742 ± 0,01	0,75 ± 0,01	0,577 ± 0,06
<b>Genbase</b>	0,979 ± 0,01	0,979 ± 0,01	0,979 ± 0,01	0,977 ± 0,01	0,972 ± 0,01	0,979 ± 0,01	0,951 ± 0,03
<b>Medical</b>	0,788 ± 0,02	0,789 ± 0,02	0,788 ± 0,02	0,8 ± 0,02	0,795 ± 0,01	0,79 ± 0,02	0,334 ± 0,04
<b>Scene</b>	0,602 ± 0,01	0,605 ± 0,01	0,596 ± 0,01	0,691 ± 0,01	0,676 ± 0,01	0,678 ± 0,01	0,556 ± 0,01
<b>Yeast</b>	0,566 ± 0,01	0,566 ± 0,01	0,545 ± 0,01	0,622 ± 0,01	0,599 ± 0,01	0,611 ± 0,01	0,493 ± 0,01
<b>Bibtex</b>	0,389 ± 0,01	0,389 ± 0,01	0,384 ± 0,01	0,386 ± 0	0,374 ± 0	0,394 ± 0	0,036 ± 0
<b>G1</b>	0,431 ± 0,01	0,425 ± 0,01	0,455 ± 0,01	0,44 ± 0,01	0,491 ± 0,01	0,497 ± 0,01	0,421 ± 0,01
<b>GP</b>	0,551 ± 0,01	0,539 ± 0,01	0,54 ± 0,01	0,603 ± 0,01	0,594 ± 0,01	0,579 ± 0,01	0,523 ± 0,02
<b>BFRC</b>	0,613 ± 0	0,611 ± 0	0,611 ± 0	0,641 ± 0	0,638 ± 0	0,648 ± 0	0,593 ± 0

Fonte: Autoria Própria (2022)

Ainda, após o teste de Friedman, com 95% de confiança e p-valor de 4.789e-08, o pós-teste de Nemenyi, Figura 34, indica similaridade estatística entre o BicbPT $\alpha$ , RAkEL<sub>2</sub>, ECC e RAkEL<sub>1</sub>, algoritmos que geralmente apresentam bom desempenho nas métricas, mas tempo de execução alto. Assim, o algoritmo proposto entre neste grupo, mas como um que consegue obter um tempo de execução menor e com desempenho estatisticamente similar.

Figura 34: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – F1<sub>micro</sub> com C4.5

Fonte: Autoria Própria (2022)

Na Tabela 7, onde é apresentado o resultado da métrica F1<sub>macro</sub>, ao considerar somente os algoritmos de transformação multirrótulo-binário, diferentemente das outras métricas já apresentadas, o BicbPT $\alpha$  possui bom desempenho em 9 de 12 bases. Na comparação geral, o desempenho é melhor em 3 de 12 bases, o que indica bom desempenho do algoritmo proposto,

tanto na categoria de multirrótulo-binário, quanto na comparação com os demais algoritmos.

Tabela 7: BicbPT $\alpha$  x Todos – F1<sub>macro</sub> com C4.5

Bases	F1 <sub>macro</sub> ↑						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,249 ± 0,03	0,249 ± 0,03	0,25 ± 0,03	0,188 ± 0,03	0,257 ± 0,02	0,255 ± 0,02	0,172 ± 0,01
<b>Emotions</b>	0,592 ± 0,01	0,573 ± 0,02	0,549 ± 0,02	0,621 ± 0,01	0,616 ± 0,02	0,605 ± 0,02	0,529 ± 0,02
<b>Enron</b>	0,144 ± 0,01	0,144 ± 0,01	0,143 ± 0,01	0,134 ± 0,01	0,138 ± 0,01	0,146 ± 0,01	0,103 ± 0
<b>Flags</b>	0,592 ± 0,03	0,606 ± 0,04	0,625 ± 0,01	0,622 ± 0,02	0,67 ± 0,02	0,66 ± 0,02	0,539 ± 0,04
<b>Genbase</b>	0,713 ± 0,04	0,713 ± 0,04	0,713 ± 0,04	0,683 ± 0,04	0,699 ± 0,05	0,713 ± 0,04	0,668 ± 0,06
<b>Medical</b>	0,353 ± 0,03	0,353 ± 0,03	0,352 ± 0,03	0,334 ± 0,03	0,352 ± 0,02	0,353 ± 0,03	0,18 ± 0,01
<b>Scene</b>	0,614 ± 0,01	0,615 ± 0,01	0,611 ± 0,01	0,698 ± 0,01	0,68 ± 0,01	0,688 ± 0,01	0,564 ± 0,01
<b>Yeast</b>	0,38 ± 0,01	0,38 ± 0,01	0,39 ± 0,01	0,372 ± 0,01	0,386 ± 0,01	0,392 ± 0	0,396 ± 0,01
<b>Bibtex</b>	0,26 ± 0	0,26 ± 0	0,254 ± 0	0,21 ± 0	0,216 ± 0	0,257 ± 0	0,035 ± 0
<b>G1</b>	0,361 ± 0,01	0,353 ± 0,01	0,374 ± 0,02	0,335 ± 0,01	0,412 ± 0,01	0,416 ± 0,01	0,4 ± 0,01
<b>GP</b>	0,399 ± 0,02	0,393 ± 0,02	0,393 ± 0,02	0,357 ± 0,02	0,392 ± 0,01	0,371 ± 0,01	0,427 ± 0,02
<b>BFRC</b>	0,532 ± 0	0,531 ± 0	0,528 ± 0	0,516 ± 0	0,546 ± 0	0,542 ± 0	0,556 ± 0,01

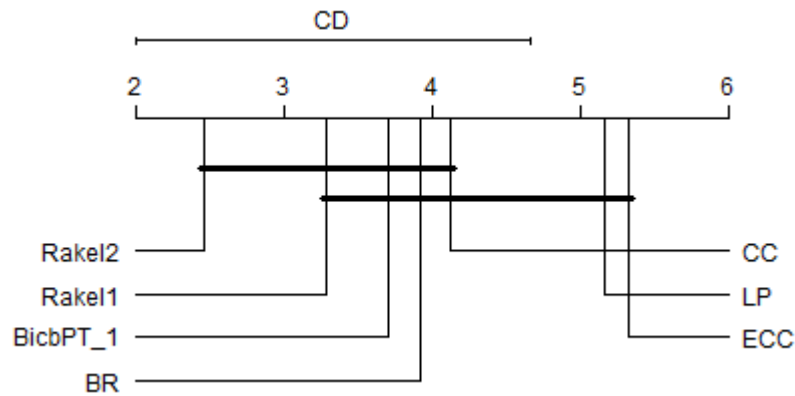
Fonte: Autoria Própria (2022)

Em uma análise estatística com 95% de confiança, o teste de Friedman retornou um p-valor de 0.01256, indicando haver diferença significativa entre os algoritmos comparados, com o pós-teste de Nemenyi, Figura 35, demonstrando que o BicbPT $\alpha$  é melhor, estatisticamente, que o ECC para esta métrica, mas apresenta resultados similares ao BR e CC, embora estes dois se posicionem em 4<sup>o</sup> e 5<sup>o</sup> lugar, respectivamente.

É interessante notar aqui, também, que nenhum algoritmo se sobressai em relação a F1<sub>macro</sub> como aconteceu nas métricas anteriores, mas isso pode ser pelo fato de que as métricas macros são uma média aritmética, compensando assim valores baixos com valores altos para cada rótulo.



Figura 35: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – F1<sub>macro</sub> com C4.5



Fonte: Autoria Própria (2022)

A Tabela 8 apresenta o resultado obtido com o tempo total de execução, somando o treino e teste, ao utilizar o C4.5 como classificador base para todos os algoritmos. Ao considerar somente os algoritmos de transformação de problema multirrótulo-binário, o BicbPT $\alpha$  possui, em média, tempo semelhante ao BR e CC, e é 6 vezes mais rápido que o ECC, onde em alguns casos a diferença pode chegar a 7 vezes, como ocorre com as bases Enron e Bibtex.

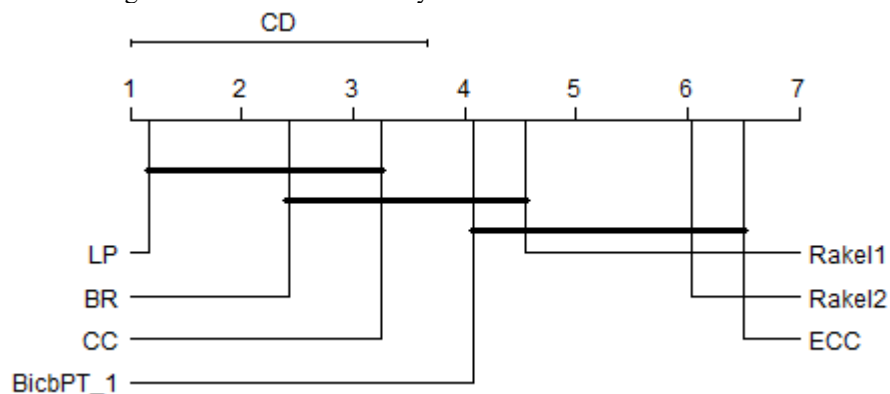
Já em relação aos algoritmos não-binários, o BicbPT $\alpha$  é em média 4 vezes mais rápido que o RAKEL<sub>2</sub> e 1,5 vezes mais rápido que o RAKEL<sub>1</sub>, podendo chegar a 6 e 3 vezes, respectivamente. Todos os algoritmos são mais lentos que o LP, mas isto é devido ao fato de que o LP não gera subproblemas, sua transformação é 1:1, ou seja, um multirrótulo para um único multiclasse.

Tabela 8: BicbPT $\alpha$  x Todos – TT com C4.5

Bases	TT ↓						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
Birds	1,691 ± 0,41	0,93 ± 0,27	1,166 ± 0,25	5,188 ± 0,21	3,921 ± 0,46	6,026 ± 0,19	0,422 ± 0,02
Emotions	1,034 ± 0,03	0,158 ± 0,03	0,223 ± 0,05	0,858 ± 0,04	0,525 ± 0,04	0,583 ± 0,02	0,119 ± 0,01
Enron	91,277 ± 1,73	95,275 ± 1,93	111,647 ± 3,67	626,61 ± 13,27	179,023 ± 4,67	543,048 ± 10,37	13,147 ± 0,55
Flags	0,333 ± 0,03	0,023 ± 0,01	0,045 ± 0,01	0,091 ± 0,01	0,041 ± 0,01	0,04 ± 0,01	0,013 ± 0,01
Genbase	1,686 ± 0,04	1,036 ± 0,2	1,841 ± 0,5	4,704 ± 0,4	2,354 ± 0,27	5,22 ± 0,25	0,632 ± 0,04
Medical	6,177 ± 0,34	5,086 ± 0,25	6,348 ± 0,26	38,711 ± 1,03	11,454 ± 0,39	26,894 ± 0,87	1,417 ± 0,09
Scene	6,779 ± 0,28	3,196 ± 0,11	3,796 ± 0,13	18,585 ± 0,24	11,8 ± 0,29	14,025 ± 0,3	2,178 ± 0,07
Yeast	5,978 ± 0,48	3,346 ± 0,17	3,039 ± 0,06	16,698 ± 0,22	12,839 ± 0,27	19,822 ± 0,46	2,485 ± 0,06
Bibtex	1183,599 ± 19,74	1316,432 ± 17,88	1318,304 ± 35,32	8401,92 ± 53,9	855,178 ± 11,28	5990,728 ± 41,73	90,707 ± 2,68
G1	137,578 ± 3,77	36,433 ± 0,78	31,646 ± 0,72	184,298 ± 5,92	162,256 ± 6,02	216,617 ± 6,62	67,801 ± 4,66
GP	10,731 ± 0,41	8,665 ± 0,35	9,209 ± 0,31	53,235 ± 1,17	32,315 ± 1,14	38,461 ± 1,15	4,32 ± 0,96
BFRC	567,366 ± 6,68	517,272 ± 26,08	427,25 ± 4,12	2668,001 ± 37,16	1371,874 ± 21,97	1853,073 ± 30,05	102,221 ± 1,9

Fonte: Autoria Própria (2022)

O p-valor com 95% de confiança obtido aqui foi de 2.675e-10 e o teste de Nemenyi, Figura 36, demonstra não haver diferença significativa, estatisticamente, entre o BicbPT $\alpha$  e BR e CC, assim o algoritmo proposto consegue, ao utilizar o C4.5 como classificador base, manter a característica de baixo tempo de execução dos algoritmos multirrótulo-binário, mas apresentar bom desempenho nas métricas JI, HL, F1<sub>micro</sub> e F1<sub>macro</sub>, diferentemente destes algoritmos.

Figura 36: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – TT com C4.5

Fonte: Autoria Própria (2022)

Na Tabela 9 apresentam-se os resultados obtidos na métrica Jaccard Index ao utilizar o SVM como classificador base. Nessa configuração o BicbPT $\alpha$  obteve os melhores resultados em apenas 1 de 12 bases, ao considerar todos os algoritmos e em 2 de 12 bases ao considerar os algoritmos de transformação multirrótulo-binário.

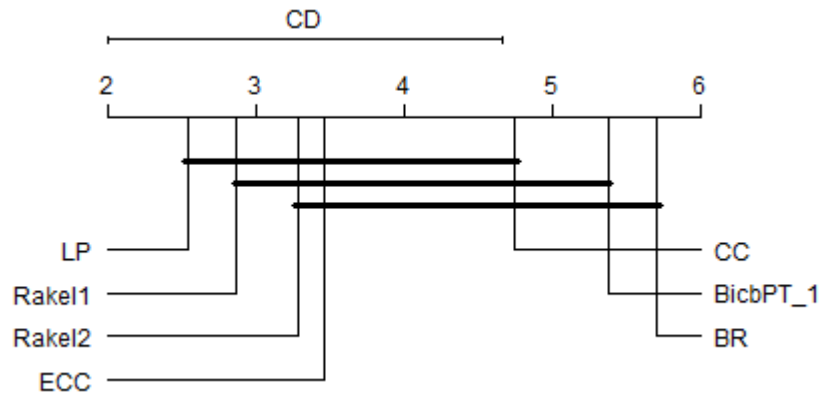
É interessante observar que ao utilizar o SVM, os resultados obtidos são melhores do que com o C4.5 para todos os algoritmos, isto demonstra que a escolha do classificador base influencia diretamente no resultado final da classificação multirrótulo gerado pelo algoritmo de transformação de problema.

Tabela 9: BicbPT $\alpha$  x Todos – JI com SVM

Bases	JI $\uparrow$						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,603 $\pm$ 0,01	0,603 $\pm$ 0,01	0,607 $\pm$ 0,01	0,604 $\pm$ 0,01	0,617 $\pm$ 0,01	0,611 $\pm$ 0,01	0,612 $\pm$ 0,01
<b>Emotions</b>	0,498 $\pm$ 0,01	0,497 $\pm$ 0,01	0,525 $\pm$ 0,02	0,559 $\pm$ 0,02	0,569 $\pm$ 0,02	0,564 $\pm$ 0,02	0,568 $\pm$ 0,01
<b>Enron</b>	0,388 $\pm$ 0,01	0,387 $\pm$ 0,01	0,394 $\pm$ 0,01	0,433 $\pm$ 0,01	0,438 $\pm$ 0,01	0,397 $\pm$ 0,01	0,408 $\pm$ 0,01
<b>Flags</b>	0,587 $\pm$ 0,01	0,59 $\pm$ 0,01	0,577 $\pm$ 0,02	0,596 $\pm$ 0,01	0,585 $\pm$ 0,01	0,59 $\pm$ 0,01	0,554 $\pm$ 0,03
<b>Genbase</b>	0,982 $\pm$ 0,01	0,982 $\pm$ 0,01	0,982 $\pm$ 0,01	0,98 $\pm$ 0,01	0,98 $\pm$ 0,01	0,981 $\pm$ 0,01	0,982 $\pm$ 0,01
<b>Medical</b>	0,73 $\pm$ 0,01	0,73 $\pm$ 0,01	0,746 $\pm$ 0,01	0,761 $\pm$ 0,01	0,752 $\pm$ 0,01	0,731 $\pm$ 0,01	0,75 $\pm$ 0,01
<b>Scene</b>	0,587 $\pm$ 0,01	0,587 $\pm$ 0,01	0,677 $\pm$ 0,01	0,69 $\pm$ 0,01	0,675 $\pm$ 0,01	0,668 $\pm$ 0,01	0,724 $\pm$ 0,01
<b>Yeast</b>	0,498 $\pm$ 0,01	0,498 $\pm$ 0,01	0,482 $\pm$ 0,02	0,527 $\pm$ 0,01	0,532 $\pm$ 0,01	0,524 $\pm$ 0,01	0,52 $\pm$ 0,01
<b>Bibtex</b>	0,313 $\pm$ 0,01	0,313 $\pm$ 0,01	0,313 $\pm$ 0,01	0,311 $\pm$ 0,01	0,307 $\pm$ 0,01	0,316 $\pm$ 0,01	0,323 $\pm$ 0,01
<b>G1</b>	0,372 $\pm$ 0	0,368 $\pm$ 0,01	0,376 $\pm$ 0,01	0,401 $\pm$ 0,01	0,416 $\pm$ 0,01	0,416 $\pm$ 0,01	0,417 $\pm$ 0,01
<b>GP</b>	0,445 $\pm$ 0,02	0,444 $\pm$ 0,02	0,446 $\pm$ 0,02	0,474 $\pm$ 0,02	0,495 $\pm$ 0,01	0,491 $\pm$ 0,02	0,504 $\pm$ 0,02
<b>BFRC</b>	0,469 $\pm$ 0	0,468 $\pm$ 0	0,468 $\pm$ 0	0,488 $\pm$ 0	0,504 $\pm$ 0	0,504 $\pm$ 0	0,489 $\pm$ 0

Fonte: Autoria Própria (2022)

Nas bases onde o BicbPT $\alpha$  tem desempenho menor que os demais algoritmos, como o ECC, RAkEL<sub>1</sub> e RAkEL<sub>2</sub>, isto é compensado pelo tempo de execução baixo do BicbPT $\alpha$ , assim como observou-se com o C4.5. O p-valor para a métrica JI foi de 0.0002816 com 95% de confiança, e na Figura 37, o gráfico com o teste de Nemenyi demonstra, que entre os algoritmos de transformação multirrótulo-binário, o BicbPT $\alpha$  é similar, estatisticamente, a todos os outros algoritmos, apresentando diferença significativa somente com o algoritmo LP, o qual não cria subproblemas.

Figura 37: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – JI com SVM

Fonte: Autoria Própria (2022)

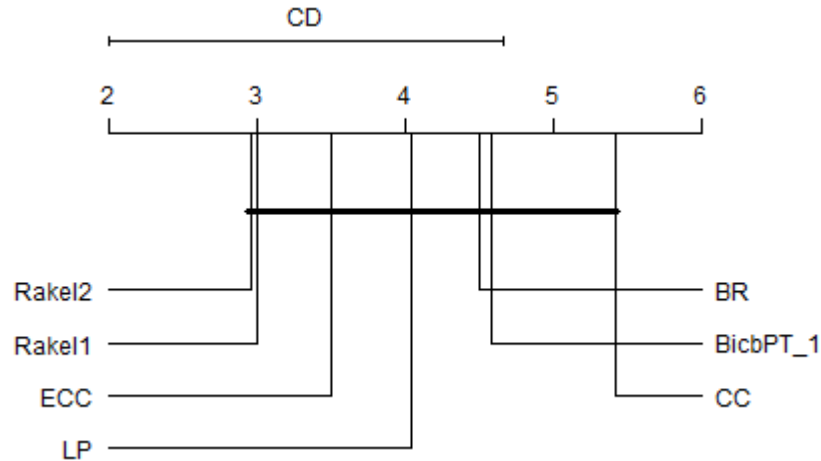
Para a métrica HL, Tabela 10, utilizando o SVM como classificador base, diferentemente do resultado apresentado pela mesma métrica utilizando o C4.5 como classificador base, Tabela 5, aqui o BicbPT $\alpha$  se saiu melhor em 3 de 12 bases na comparação geral e foi melhor em 4 de 12 bases ao considerar somente os algoritmos multirrotulo-binário.

Também se nota pelo pós-teste de Nemenyi, Figura 38, que não foram encontradas diferenças estatisticamente significativas entre os algoritmos comparados, indicando que todos possuem desempenho similar nesta métrica, mesmo o p-valor obtido no teste de Friedman para HL com 95% de confiança sendo de 0.02913.

Tabela 10: BicbPT $\alpha$  x Todos – HL com SVM

Bases	HL ↓						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,052 ± 0	0,052 ± 0	0,051 ± 0	0,051 ± 0	0,047 ± 0	0,048 ± 0	0,05 ± 0
<b>Emotions</b>	0,203 ± 0	0,204 ± 0	0,222 ± 0,01	0,206 ± 0,01	0,196 ± 0,01	0,195 ± 0,01	0,207 ± 0,01
<b>Enron</b>	0,06 ± 0	0,06 ± 0	0,06 ± 0	0,053 ± 0	0,052 ± 0	0,058 ± 0	0,057 ± 0
<b>Flags</b>	0,265 ± 0,01	0,265 ± 0,01	0,278 ± 0,02	0,265 ± 0,01	0,274 ± 0,01	0,268 ± 0,01	0,291 ± 0,02
<b>Genbase</b>	0,001 ± 0	0,001 ± 0	0,001 ± 0	0,002 ± 0	0,002 ± 0	0,002 ± 0	0,002 ± 0
<b>Medical</b>	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,01 ± 0	0,011 ± 0	0,011 ± 0	0,012 ± 0
<b>Scene</b>	0,111 ± 0	0,11 ± 0	0,11 ± 0	0,097 ± 0	0,095 ± 0	0,096 ± 0	0,094 ± 0
<b>Yeast</b>	0,201 ± 0	0,201 ± 0	0,213 ± 0,01	0,204 ± 0	0,204 ± 0	0,201 ± 0	0,212 ± 0
<b>Bibtex</b>	0,015 ± 0	0,015 ± 0	0,015 ± 0	0,013 ± 0	0,013 ± 0	0,015 ± 0	0,016 ± 0
<b>G1</b>	0,29 ± 0	0,287 ± 0	0,286 ± 0	0,265 ± 0	0,259 ± 0	0,257 ± 0	0,256 ± 0,01
<b>GP</b>	0,294 ± 0,01	0,295 ± 0,01	0,295 ± 0,01	0,275 ± 0,01	0,264 ± 0,01	0,262 ± 0,01	0,257 ± 0,01
<b>BFRC</b>	0,359 ± 0	0,359 ± 0	0,361 ± 0	0,354 ± 0	0,343 ± 0	0,342 ± 0	0,34 ± 0

Fonte: Autoria Própria (2022)

Figura 38: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – HL com SVM

Fonte: Autoria Própria (2022)

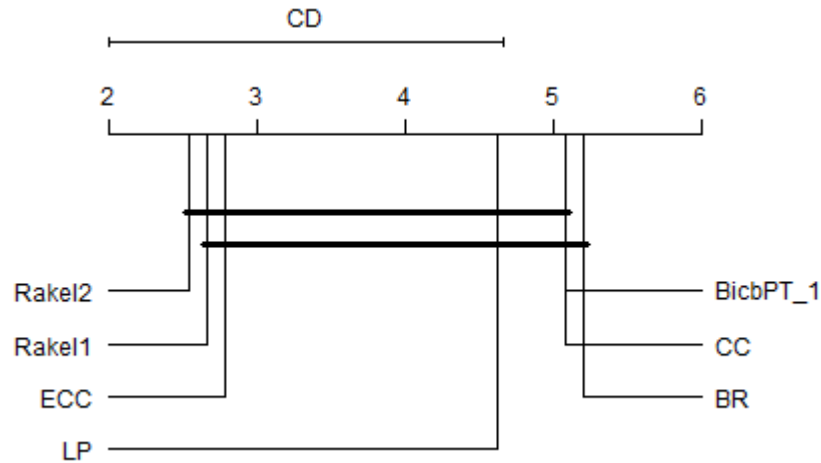
Na Tabela 11 são apresentados os resultados da métrica  $F1_{\text{micro}}$  com o SVM, onde, como também é observado nas outras métricas ao utilizar o SVM, os resultados tem uma melhora significativa em relação ao uso do C4.5, embora o BicbPT $\alpha$  tenha se saído melhor em apenas 1 de 12 bases.

O p-valor de 0.0003048 no teste de Friedman indicou a existência de diferenças entre os algoritmos, e após com o teste de Nemenyi, Figura 39, foi possível verificar que o BicbPT $\alpha$  obteve desempenho estatisticamente similar a todos os outros algoritmos de transformação de problema, sendo a diferença apontada por Friedman residindo entre BR e RAKEL<sub>2</sub>.

Tabela 11: BicbPT $\alpha$  x Todos –  $F1_{\text{micro}}$  com SVM

Bases	$F1_{\text{micro}} \uparrow$						
	BicbPT $\alpha$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
Birds	0,456 $\pm$ 0,02	0,456 $\pm$ 0,02	0,463 $\pm$ 0,02	0,466 $\pm$ 0,02	0,493 $\pm$ 0,03	0,477 $\pm$ 0,03	0,474 $\pm$ 0,02
Emotions	0,638 $\pm$ 0,01	0,638 $\pm$ 0,01	0,639 $\pm$ 0,02	0,673 $\pm$ 0,01	0,683 $\pm$ 0,01	0,679 $\pm$ 0,01	0,673 $\pm$ 0,01
Enron	0,505 $\pm$ 0,01	0,503 $\pm$ 0,01	0,502 $\pm$ 0,01	0,547 $\pm$ 0,01	0,547 $\pm$ 0,01	0,516 $\pm$ 0,01	0,5 $\pm$ 0,01
Flags	0,727 $\pm$ 0,01	0,732 $\pm$ 0,01	0,716 $\pm$ 0,02	0,737 $\pm$ 0,01	0,725 $\pm$ 0,01	0,729 $\pm$ 0,01	0,694 $\pm$ 0,03
Genbase	0,985 $\pm$ 0,01	0,985 $\pm$ 0,01	0,985 $\pm$ 0,01	0,983 $\pm$ 0,01	0,982 $\pm$ 0,01	0,983 $\pm$ 0,01	0,981 $\pm$ 0,01
Medical	0,792 $\pm$ 0,01	0,792 $\pm$ 0,01	0,798 $\pm$ 0,01	0,811 $\pm$ 0,01	0,798 $\pm$ 0,01	0,793 $\pm$ 0,01	0,769 $\pm$ 0,01
Scene	0,673 $\pm$ 0,01	0,674 $\pm$ 0,01	0,688 $\pm$ 0,01	0,722 $\pm$ 0,01	0,725 $\pm$ 0,01	0,723 $\pm$ 0,01	0,734 $\pm$ 0,01
Yeast	0,632 $\pm$ 0,01	0,632 $\pm$ 0,01	0,615 $\pm$ 0,01	0,649 $\pm$ 0,01	0,652 $\pm$ 0,01	0,65 $\pm$ 0,01	0,633 $\pm$ 0,01
Bibtex	0,41 $\pm$ 0,01	0,41 $\pm$ 0,01	0,41 $\pm$ 0	0,416 $\pm$ 0	0,392 $\pm$ 0	0,414 $\pm$ 0,01	0,353 $\pm$ 0
G1	0,496 $\pm$ 0	0,491 $\pm$ 0	0,492 $\pm$ 0,01	0,518 $\pm$ 0,01	0,53 $\pm$ 0,01	0,531 $\pm$ 0,01	0,513 $\pm$ 0,01
GP	0,575 $\pm$ 0,02	0,573 $\pm$ 0,02	0,575 $\pm$ 0,02	0,599 $\pm$ 0,01	0,61 $\pm$ 0,01	0,605 $\pm$ 0,02	0,612 $\pm$ 0,01
BFRC	0,629 $\pm$ 0	0,629 $\pm$ 0	0,628 $\pm$ 0	0,648 $\pm$ 0	0,661 $\pm$ 0	0,662 $\pm$ 0	0,641 $\pm$ 0

Fonte: Autoria Própria (2022)

Figura 39: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – F1<sub>micro</sub> com SVM

Fonte: Autoria Própria (2022)

Na métrica F1<sub>macro</sub>, Tabela 12, o algoritmo proposto apresentou-se melhor em 2 de 12 bases, quando comparado com BR, CC e ECC, os quais são algoritmos de transformação de problema multirrótulo-binário, enquanto que o p-valor obtido pelo teste de Friedman com 95% de confiança foi de 0.4953, indicando não haver diferença estatisticamente significativa entre os algoritmos comparados, portanto, não havendo a necessidade de executar o pós-teste de Nemenyi.

Tabela 12: BicbPT $\alpha$  x Todos – F1<sub>macro</sub> com SVM

Bases	F1 <sub>macro</sub> ↑						
	BicbPT $\alpha$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	0,349 ± 0,03	0,349 ± 0,03	0,368 ± 0,04	0,361 ± 0,03	0,377 ± 0,03	0,373 ± 0,04	0,375 ± 0,02
<b>Emotions</b>	0,6 ± 0,01	0,599 ± 0,01	0,596 ± 0,02	0,663 ± 0,02	0,668 ± 0,01	0,661 ± 0,01	0,666 ± 0,01
<b>Enron</b>	0,201 ± 0,02	0,201 ± 0,02	0,201 ± 0,02	0,203 ± 0,02	0,198 ± 0,02	0,203 ± 0,02	0,185 ± 0,02
<b>Flags</b>	0,631 ± 0,03	0,652 ± 0,02	0,64 ± 0,03	0,658 ± 0,02	0,637 ± 0,02	0,635 ± 0,02	0,604 ± 0,03
<b>Genbase</b>	0,756 ± 0,04	0,756 ± 0,04	0,758 ± 0,03	0,753 ± 0,04	0,751 ± 0,04	0,753 ± 0,04	0,757 ± 0,04
<b>Medical</b>	0,363 ± 0,02	0,363 ± 0,02	0,368 ± 0,02	0,357 ± 0,02	0,363 ± 0,02	0,36 ± 0,02	0,343 ± 0,02
<b>Scene</b>	0,68 ± 0,01	0,681 ± 0,01	0,699 ± 0,01	0,731 ± 0,01	0,732 ± 0,01	0,732 ± 0,01	0,743 ± 0,01
<b>Yeast</b>	0,327 ± 0,01	0,327 ± 0,01	0,359 ± 0,01	0,357 ± 0,01	0,399 ± 0,01	0,363 ± 0,01	0,407 ± 0,01
<b>Bibtex</b>	0,299 ± 0,01	0,299 ± 0,01	0,296 ± 0,01	0,265 ± 0,01	0,23 ± 0,01	0,301 ± 0,01	0,217 ± 0
<b>G1</b>	0,458 ± 0,01	0,454 ± 0,01	0,453 ± 0,01	0,471 ± 0,01	0,483 ± 0,01	0,486 ± 0,01	0,462 ± 0,02
<b>GP</b>	0,388 ± 0,02	0,387 ± 0,02	0,388 ± 0,02	0,377 ± 0,02	0,379 ± 0,02	0,372 ± 0,02	0,356 ± 0,01
<b>BFRC</b>	0,553 ± 0	0,553 ± 0	0,555 ± 0	0,573 ± 0	0,586 ± 0,01	0,579 ± 0,01	0,541 ± 0,01

Fonte: Autoria Própria (2022)

Na Tabela 13 apresentam-se os resultados obtidos em relação ao tempo total de execução dos algoritmos comparados ao utilizar o SVM como classificador base, sendo

interessante notar que ao comparar o uso do SVM com o C4.5, o primeiro conseguiu reduzir o tempo de execução em média em 45% dos resultados, o que demonstra ser um bom algoritmo monorrótulo para ser utilizado em conjunto com algoritmos de transformação de problema.

Tabela 13: BicbPT $\alpha$  x Todos – TT com SVM

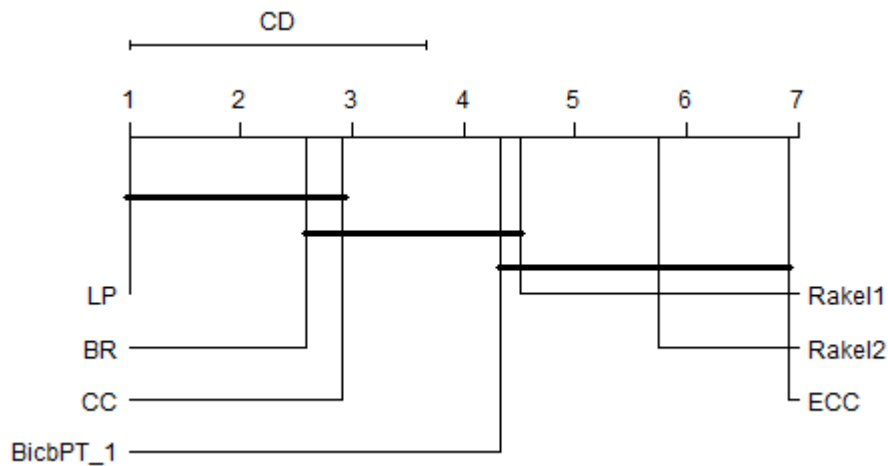
Bases	TT ↓						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
Birds	1,79 ± 0,07	0,955 ± 0,08	1,044 ± 0,07	7,144 ± 0,32	2,575 ± 0,13	4,462 ± 0,31	0,527 ± 0,01
Emotions	1,385 ± 0,05	0,213 ± 0,02	0,265 ± 0,04	1,478 ± 0,03	0,742 ± 0,02	0,88 ± 0,01	0,138 ± 0,01
Enron	23,697 ± 0,57	21,065 ± 0,71	26,645 ± 0,64	187,518 ± 2,56	47,112 ± 1,31	116,448 ± 2,6	7,865 ± 0,37
Flags	0,657 ± 0,19	0,053 ± 0,02	0,034 ± 0,01	0,177 ± 0,02	0,071 ± 0,01	0,095 ± 0,01	0,028 ± 0,01
Genbase	17,255 ± 0,5	15,149 ± 0,41	14,488 ± 0,55	94,138 ± 3,48	21,374 ± 1,38	64,551 ± 2,29	2,906 ± 0,13
Medical	5,664 ± 0,15	4,897 ± 0,17	6,35 ± 0,1	50,727 ± 0,56	3,675 ± 0,14	17,374 ± 0,36	0,638 ± 0,03
Scene	14,781 ± 0,92	8,299 ± 0,17	7,35 ± 0,18	44,837 ± 1,8	29,467 ± 0,44	35,69 ± 0,51	3,596 ± 0,26
Yeast	17,847 ± 0,87	13,331 ± 0,23	8,866 ± 0,15	55,251 ± 0,74	24,98 ± 0,25	49,738 ± 0,5	3,86 ± 0,06
Bibtex	598,755 ± 5,6	622,407 ± 4,91	665,048 ± 5,85	4830,659 ± 36,46	1830,896 ± 28,66	3149,844 ± 33,54	412,93 ± 6,17
G1	13,997 ± 0,23	2,878 ± 0,1	3,441 ± 0,06	23,038 ± 0,24	8,809 ± 0,04	12,312 ± 0,06	1,499 ± 0,02
GP	8,159 ± 0,71	3,572 ± 0,07	4,074 ± 0,04	25,18 ± 0,33	12,41 ± 0,26	14,911 ± 0,27	1,927 ± 0,09
BFRC	644,639 ± 15,2	553,036 ± 12,62	530,381 ± 10,14	1823,57 ± 20,25	734,466 ± 6,25	1093,021 ± 10,97	111,461 ± 1,1

Fonte: Autoria Própria (2022)

Ainda, ao analisar o BicbPT $\alpha$ , ele apresentou tempo semelhante ao BR e CC e consegue ser 5 vezes mais rápido que o ECC. Para as bases Enron, Medical e Bibtex a diferença alcançada é de até 9 vezes, o que demonstra que o algoritmo proposto consegue manter a característica de baixo tempo de execução dos algoritmos multirrótulo-binário.

Comparando-se com os outros algoritmos, o BicbPT $\alpha$  é em média 2 vezes mais rápido que o RAkEL<sub>1</sub> e 3,5 vezes mais rápido que o RAkEL<sub>2</sub>, sendo que esta diferença pode chegar a 5 vezes a depender dos dados, e assim como ocorreu com o C4.5, o LP é o algoritmo mais rápido, mas como já mencionado, isto deve-se por conta de sua transformação 1:1.

O p-valor obtido aqui no teste de Friedman foi de 1.764e-11 e no gráfico de Nemenyi apresentado na Figura 40 é possível observar os resultados descritos anteriormente em relação ao tempo total com SVM.

Figura 40: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – TT com SVM

Fonte: Autoria Própria (2022)

Por fim, para finalizar a apresentação e análise dos resultados da comparação entre o BicbPT $\alpha$  e os demais algoritmos de transformação de problema, são tratados os dados obtidos ao utilizar o Naive Bayes como classificador base para todos os algoritmos e bases.

Na Tabela 14 são exibidos os resultados obtidos com a métrica JI utilizando o NB como classificador base e algo que chama bastante atenção aqui e nas demais métricas que serão apresentadas, é o fato de o NB ter sido o pior classificador base entre os escolhidos para execução dos experimentos.

Tabela 14: BicbPT $\alpha$  x Todos – JI com NB

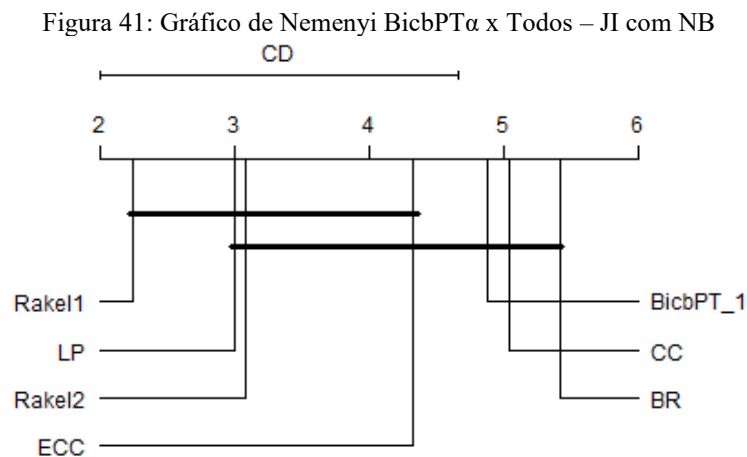
Bases	JI $\uparrow$						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,151 $\pm$ 0,01	0,151 $\pm$ 0,01	0,15 $\pm$ 0,01	0,17 $\pm$ 0,02	0,198 $\pm$ 0,02	0,159 $\pm$ 0,01	0,189 $\pm$ 0,02
<b>Emotions</b>	0,523 $\pm$ 0,02	0,522 $\pm$ 0,01	0,528 $\pm$ 0,01	0,527 $\pm$ 0,01	0,515 $\pm$ 0,02	0,513 $\pm$ 0,02	0,508 $\pm$ 0,02
<b>Enron</b>	0,239 $\pm$ 0,01	0,239 $\pm$ 0,01	0,239 $\pm$ 0,01	0,251 $\pm$ 0,01	0,413 $\pm$ 0,01	0,263 $\pm$ 0,01	0,397 $\pm$ 0,01
<b>Flags</b>	0,517 $\pm$ 0,03	0,49 $\pm$ 0,04	0,5 $\pm$ 0,04	0,512 $\pm$ 0,04	0,522 $\pm$ 0,02	0,537 $\pm$ 0,02	0,525 $\pm$ 0,02
<b>Genbase</b>	0,025 $\pm$ 0,06	0,025 $\pm$ 0,06	0,07 $\pm$ 0,1	0 $\pm$ 0	0,268 $\pm$ 0,03	0,027 $\pm$ 0,06	0,285 $\pm$ 0,04
<b>Medical</b>	0,238 $\pm$ 0,02	0,238 $\pm$ 0,02	0,244 $\pm$ 0,02	0,143 $\pm$ 0,02	0,193 $\pm$ 0,04	0,234 $\pm$ 0,03	0,387 $\pm$ 0,02
<b>Scene</b>	0,452 $\pm$ 0	0,452 $\pm$ 0	0,457 $\pm$ 0	0,458 $\pm$ 0	0,558 $\pm$ 0,01	0,55 $\pm$ 0,01	0,614 $\pm$ 0,02
<b>Yeast</b>	0,421 $\pm$ 0,01	0,421 $\pm$ 0,01	0,42 $\pm$ 0,02	0,427 $\pm$ 0,01	0,472 $\pm$ 0,01	0,448 $\pm$ 0,01	0,46 $\pm$ 0,01
<b>Bibtex</b>	0,203 $\pm$ 0	0,203 $\pm$ 0	0,191 $\pm$ 0	0,196 $\pm$ 0	0,236 $\pm$ 0	0,205 $\pm$ 0	0,191 $\pm$ 0,01
<b>G1</b>	0,431 $\pm$ 0,01	0,428 $\pm$ 0	0,433 $\pm$ 0,01	0,433 $\pm$ 0	0,434 $\pm$ 0	0,438 $\pm$ 0	0,405 $\pm$ 0,01
<b>GP</b>	0,463 $\pm$ 0,02	0,461 $\pm$ 0,02	0,461 $\pm$ 0,02	0,475 $\pm$ 0,02	0,499 $\pm$ 0,01	0,495 $\pm$ 0,01	0,509 $\pm$ 0,01
<b>BFRC</b>	0,473 $\pm$ 0	0,473 $\pm$ 0	0,471 $\pm$ 0	0,476 $\pm$ 0	0,494 $\pm$ 0	0,487 $\pm$ 0,01	0,476 $\pm$ 0

Fonte: Autoria Própria (2022)



É possível que esse resultado seja por conta da característica de presumir independência do NB, e, mesmo os algoritmos de transformação de problema tratando de alguma forma a relação entre os rótulos, isso não ajuda a minimizar o efeito negativo do NB no resultado da classificação multirrótulo, o que, portanto, sugere-se que este não seja o algoritmo monorrótulo rótulo ideal para utilizar como classificador base ao utilizar essa estratégia de classificação multirrótulo.

Analisando os resultados de forma geral para esta métrica, o BicbPT $\alpha$  não se sobressaiu em nenhuma das bases, apresentando diferença estatística significativa somente com o RAKEL1, como demonstra o teste de Nemenyi, Figura 41, realizado após obter-se o p-valor de 0.0006922 para JI em Friedman com 95% de confiança.



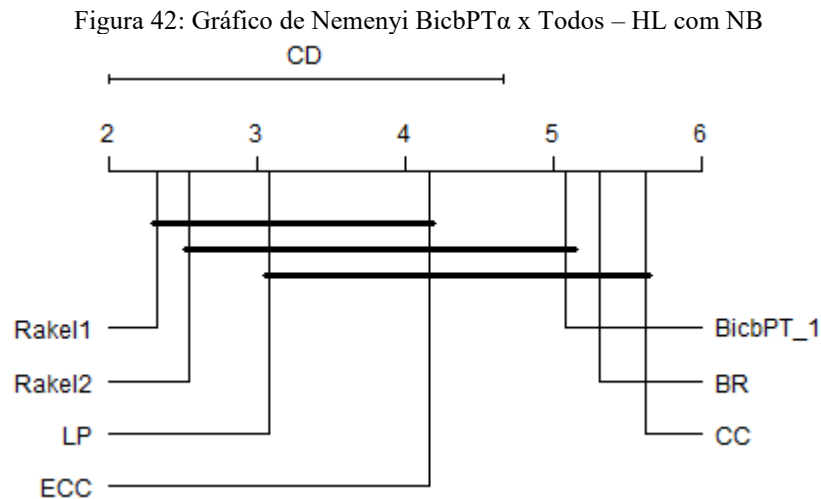
Na Tabela 15 são exibidos os resultados obtidos com a métrica HL em conjunto com o NB, onde o algoritmo proposto, assim como na métrica JI, não se sobressaiu em nenhuma base de forma geral, mas apresentando-se melhor em 2 de 12 ao considerar somente os algoritmos binários BR, CC e ECC.

O p-valor com 95% de confiança obtido foi de 4.766e-05 e no gráfico da Figura 42 nota-se que existe diferença estatisticamente significativa somente entre o BicbPT $\alpha$  e o RAKEL<sub>1</sub>, sendo similar em desempenho ao demais algoritmos, principalmente quando observa-se o ECC, que, como já ocorreu com outras métricas, apesar de estar em um mesmo grupo do BicbPT $\alpha$ , apresenta tempo de execução mais alto que este.

Tabela 15: BicbPT $\alpha$  x Todos – HL com NB

Bases	HL ↓						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,295 ± 0,03	0,295 ± 0,03	0,303 ± 0,04	0,247 ± 0,04	0,104 ± 0,01	0,193 ± 0,02	0,08 ± 0
<b>Emotions</b>	0,258 ± 0,02	0,256 ± 0,01	0,251 ± 0,01	0,249 ± 0,01	0,252 ± 0,02	0,247 ± 0,02	0,238 ± 0,01
<b>Enron</b>	0,174 ± 0,02	0,174 ± 0,02	0,182 ± 0,02	0,169 ± 0,02	0,055 ± 0	0,132 ± 0,01	0,074 ± 0
<b>Flags</b>	0,318 ± 0,02	0,334 ± 0,04	0,334 ± 0,03	0,319 ± 0,03	0,327 ± 0,01	0,312 ± 0,01	0,322 ± 0,02
<b>Genbase</b>	0,045 ± 0	0,045 ± 0	0,044 ± 0	0,046 ± 0	0,036 ± 0	0,045 ± 0	0,062 ± 0
<b>Medical</b>	0,025 ± 0	0,025 ± 0	0,024 ± 0	0,025 ± 0	0,024 ± 0	0,025 ± 0	0,032 ± 0
<b>Scene</b>	0,242 ± 0	0,24 ± 0	0,236 ± 0	0,234 ± 0	0,157 ± 0,01	0,158 ± 0	0,135 ± 0,01
<b>Yeast</b>	0,297 ± 0,01	0,297 ± 0,01	0,299 ± 0,01	0,293 ± 0,01	0,245 ± 0,01	0,268 ± 0	0,263 ± 0,01
<b>Bibtex</b>	0,06 ± 0	0,06 ± 0	0,098 ± 0	0,078 ± 0	0,015 ± 0	0,055 ± 0	0,025 ± 0
<b>G1</b>	0,278 ± 0,01	0,274 ± 0	0,275 ± 0	0,271 ± 0	0,272 ± 0	0,268 ± 0	0,313 ± 0,01
<b>GP</b>	0,295 ± 0,02	0,296 ± 0,02	0,297 ± 0,02	0,277 ± 0,02	0,258 ± 0,01	0,257 ± 0,01	0,253 ± 0,01
<b>BFRC</b>	0,369 ± 0	0,369 ± 0	0,373 ± 0	0,366 ± 0	0,351 ± 0	0,355 ± 0	0,355 ± 0

Fonte: Autoria Própria (2022)



Fonte: Autoria Própria (2022)

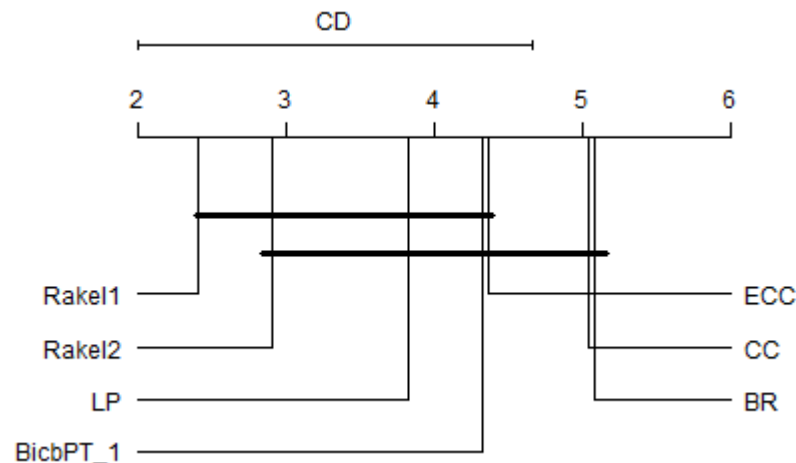
Na Tabela 16 apresentam-se os resultados obtidos com a métrica  $F1_{\text{micro}}$ , onde o BicbPT $\alpha$  conseguiu se sair melhor em 3 das 12 bases utilizadas nos experimentos, considerando os algoritmos de transformação de problema multirrótulo-binário.

Tabela 16: BicbPT $\alpha$  x Todos – F1<sub>micro</sub> com NB

Bases	F1 <sub>micro</sub> ↑						
	BicbPT $\alpha$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,169 ± 0,02	0,169 ± 0,02	0,166 ± 0,02	0,189 ± 0,03	0,306 ± 0,03	0,217 ± 0,01	0,252 ± 0,03
<b>Emotions</b>	0,651 ± 0,02	0,65 ± 0,01	0,653 ± 0,01	0,65 ± 0,01	0,638 ± 0,01	0,634 ± 0,01	0,616 ± 0,02
<b>Enron</b>	0,307 ± 0,02	0,307 ± 0,02	0,297 ± 0,02	0,308 ± 0,02	0,535 ± 0,01	0,36 ± 0,01	0,444 ± 0,02
<b>Flags</b>	0,688 ± 0,02	0,666 ± 0,04	0,667 ± 0,03	0,679 ± 0,03	0,683 ± 0,02	0,694 ± 0,02	0,668 ± 0,02
<b>Genbase</b>	0,036 ± 0,08	0,036 ± 0,08	0,095 ± 0,14	0 ± 0	0,353 ± 0,04	0,038 ± 0,08	0,256 ± 0,04
<b>Medical</b>	0,379 ± 0,03	0,38 ± 0,03	0,388 ± 0,02	0,263 ± 0,02	0,309 ± 0,05	0,377 ± 0,03	0,393 ± 0,03
<b>Scene</b>	0,555 ± 0	0,556 ± 0	0,56 ± 0	0,561 ± 0	0,638 ± 0,01	0,636 ± 0,01	0,634 ± 0,02
<b>Yeast</b>	0,55 ± 0,01	0,55 ± 0,01	0,544 ± 0,01	0,552 ± 0,01	0,604 ± 0,01	0,584 ± 0,01	0,571 ± 0,01
<b>Bibtex</b>	0,214 ± 0,01	0,214 ± 0,01	0,152 ± 0,01	0,151 ± 0,01	0,299 ± 0	0,228 ± 0	0,15 ± 0,01
<b>G1</b>	0,557 ± 0,01	0,553 ± 0	0,556 ± 0	0,557 ± 0	0,552 ± 0	0,557 ± 0	0,488 ± 0,01
<b>GP</b>	0,584 ± 0,02	0,581 ± 0,02	0,581 ± 0,02	0,591 ± 0,02	0,611 ± 0,02	0,606 ± 0,02	0,617 ± 0,01
<b>BFRC</b>	0,629 ± 0	0,629 ± 0	0,626 ± 0	0,631 ± 0	0,65 ± 0	0,644 ± 0	0,627 ± 0

Fonte: Autoria Própria (2022)

Já, ao analisar a Figura 43, observa-se que não existem diferenças estatisticamente significativas entre o algoritmo proposto e os demais algoritmos comparados, embora, em relação ao ECC, CC e BR, o BicbPT $\alpha$  apresente-se na frente no ranking criado pelo teste de Nemenyi com confiança de 95%, após obtenção do p-valor de 0.01232 no teste de Friedman.

Figura 43: Gráfico de Nemenyi BicbPT $\alpha$  x Todos – F1<sub>micro</sub> com NB

Fonte: Autoria Própria (2022)

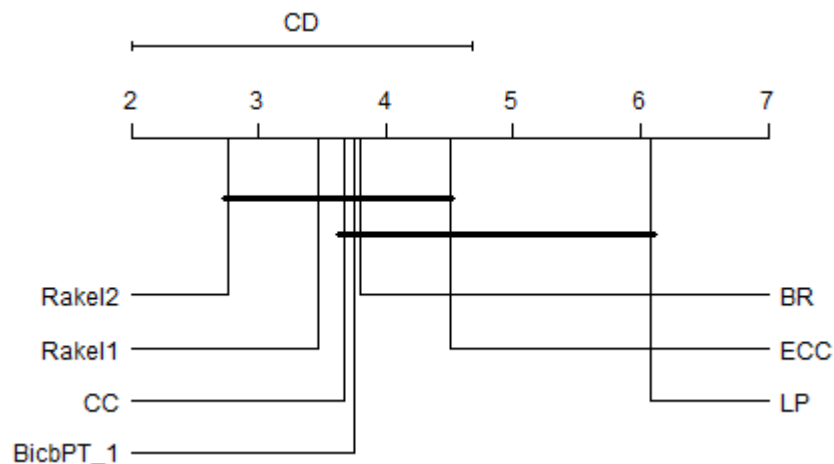
Na Tabela 17 apresentam-se os resultados da métrica F1<sub>macro</sub>, sendo o BicbPT $\alpha$  melhor em 5 de 12 bases nos algoritmos de transformação multirrótulo-binário e na comparação geral, em apenas 1 de 12 bases.

O p-valor com 95% de confiança obtido para a métrica  $F1_{macro}$  foi de 0.006689 no teste de Friedman, indicando que existem diferenças significativas entre os algoritmos comparados, justificando assim o uso do teste de Nemenyi, Figura 44, posteriormente. Embora tenham sido indicadas diferenças, estas não residem no BicbPT $\alpha$ , e dizem respeito somente ao LP e ao RAKEL<sub>1</sub> e RAKEL<sub>2</sub>, dessa forma o algoritmo proposto apresenta desempenho estatisticamente similar aos demais algoritmos comparados.

Tabela 17: BicbPT $\alpha$  x Todos –  $F1_{macro}$  com NB

Bases	$F1_{macro} \uparrow$						
	BicbPT $\alpha$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
Birds	0,16 ± 0,02	0,16 ± 0,02	0,156 ± 0,02	0,162 ± 0,03	0,223 ± 0,03	0,195 ± 0,02	0,144 ± 0,02
Emotions	0,645 ± 0,01	0,645 ± 0,01	0,647 ± 0,01	0,641 ± 0,01	0,634 ± 0,01	0,627 ± 0,02	0,603 ± 0,02
Enron	0,164 ± 0,01	0,164 ± 0,01	0,162 ± 0,01	0,156 ± 0,01	0,197 ± 0,02	0,177 ± 0,01	0,148 ± 0,01
Flags	0,61 ± 0,03	0,619 ± 0,03	0,62 ± 0,03	0,628 ± 0,03	0,63 ± 0,02	0,638 ± 0,02	0,59 ± 0,02
Genbase	0,006 ± 0,01	0,006 ± 0,01	0,012 ± 0,02	0 ± 0	0,044 ± 0,01	0,006 ± 0,01	0,03 ± 0,01
Medical	0,051 ± 0	0,051 ± 0	0,052 ± 0	0,029 ± 0	0,04 ± 0,01	0,051 ± 0	0,049 ± 0,01
Scene	0,573 ± 0	0,574 ± 0,01	0,578 ± 0	0,579 ± 0,01	0,656 ± 0,01	0,659 ± 0,01	0,642 ± 0,02
Yeast	0,45 ± 0,01	0,45 ± 0,01	0,447 ± 0,01	0,448 ± 0,01	0,447 ± 0,01	0,449 ± 0,01	0,424 ± 0,01
Bibtex	0,17 ± 0	0,17 ± 0	0,138 ± 0	0,118 ± 0	0,094 ± 0	0,179 ± 0	0,055 ± 0
G1	0,523 ± 0,01	0,52 ± 0,01	0,521 ± 0,01	0,523 ± 0,01	0,519 ± 0	0,525 ± 0,01	0,455 ± 0,01
GP	0,434 ± 0,02	0,434 ± 0,01	0,435 ± 0,02	0,413 ± 0,02	0,402 ± 0,02	0,398 ± 0,02	0,371 ± 0,01
BFRC	0,587 ± 0	0,587 ± 0	0,587 ± 0	0,587 ± 0	0,592 ± 0	0,589 ± 0	0,542 ± 0,01

Fonte: Autoria Própria (2022)

Figura 44: Gráfico de Nemenyi BicbPT $\alpha$  x Todos –  $F1_{macro}$  com NB

Fonte: Autoria Própria (2022)

Em relação a métrica Tempo Total, Tabela 18, utilizando o NB como classificador base é possível perceber que o uso deste algoritmo monorrótulo fez com que o tempo total de execução fosse o mais rápido entre os algoritmos utilizados, SVM, C4.5 e NB.

Sendo que a diferença média do tempo de execução para o SVM e o C4.5 foi de 58%, o que indica que, talvez, explorar o NB para conseguir bons resultados nas métricas JI, HL e F1 seria interessante, visto a sua capacidade de diminuir o tempo de execução na classificação multirrótulo por transformação de problema.

Ainda, ao analisar os tempos do BicbPT $\alpha$  com os outros algoritmos, este se apresenta semelhante ao BR e CC e em média 9 vezes mais rápido que o ECC. Já em relação aos algoritmos multirrótulo não-binários, o algoritmo proposto é 8 vezes mais rápido que o RAKEL<sub>1</sub> e 3 vezes mais rápido que o RAKEL<sub>2</sub>.

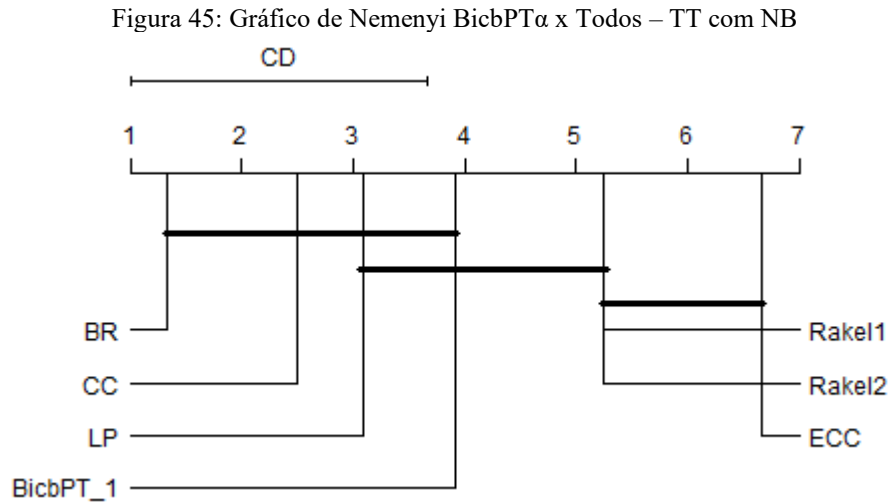
Tabela 18: BicbPT $\alpha$  x Todos – TT com NB

Bases	TT ↓						
	BicbPT $\alpha$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	1,999 ± 0,35	1,34 ± 0,18	1,344 ± 0,18	10,459 ± 0,25	5,98 ± 0,26	5,175 ± 0,28	2,257 ± 0,27
<b>Emotions</b>	1,078 ± 0,03	0,106 ± 0,01	0,162 ± 0,03	0,901 ± 0,09	0,503 ± 0,07	0,579 ± 0,07	0,183 ± 0,01
<b>Enron</b>	50,686 ± 0,33	50,635 ± 0,34	53,221 ± 0,2	459,596 ± 2,45	444,502 ± 11,28	191,561 ± 2,54	114,618 ± 3,54
<b>Flags</b>	0,554 ± 0,19	0,008 ± 0,01	0,034 ± 0,01	0,102 ± 0,01	0,051 ± 0,01	0,059 ± 0,01	0,022 ± 0,01
<b>Genbase</b>	3,092 ± 0,11	2,785 ± 0,05	2,818 ± 0,08	24,92 ± 0,52	4,482 ± 0,34	7,891 ± 0,3	1,125 ± 0,06
<b>Medical</b>	9,908 ± 0,46	9,242 ± 0,25	11,1 ± 0,11	102,751 ± 0,41	17,808 ± 0,87	31,14 ± 0,51	4,297 ± 0,21
<b>Scene</b>	5,441 ± 0,21	1,308 ± 0,03	1,844 ± 0,01	12,843 ± 0,08	5,466 ± 0,1	6,41 ± 0,14	1,6 ± 0,07
<b>Yeast</b>	3,732 ± 0,04	1,149 ± 0,06	1,364 ± 0,02	10,839 ± 0,11	12,658 ± 0,27	6,396 ± 0,06	4,246 ± 0,14
<b>Bibtex</b>	372,778 ± 3,05	382,12 ± 7 ± 1,57	423,466 ± 5,95	3907,149 ± 6,82	3974,958 ± 43,96	1396,591 ± 19,61	959,336 ± 17,3
<b>G1</b>	59,247 ± 2,1	8,757 ± 1,83	7,917 ± 0,34	64,16 ± 2,04	27,698 ± 0,5	38,509 ± 0,75	19,403 ± 0,69
<b>GP</b>	5,696 ± 0,52	2,408 ± 0,12	2,68 ± 0,04	22,084 ± 0,13	10,61 ± 0,09	12,651 ± 0,13	4,677 ± 0,18
<b>BFRC</b>	51,192 ± 2,72	37,108 ± 0,33	39,374 ± 0,28	324,72 ± 1,7	205,391 ± 2,38	188,515 ± 3,43	214,247 ± 3,82

Fonte: Autoria Própria (2022)

Chama atenção, também, o fato de o BicbPT $\alpha$  ser, em média, 2 vezes mais rápido que o LP ao utilizar o NB como classificador base, o que contrasta com os resultados obtidos com os algoritmos C4.5 e SVM.

O p-valor obtido aqui foi de  $1.431e-09$  com 95% de confiança no teste de Friedman, e na Figura 45, no teste de Nemenyi observa-se diferença significativa no tempo total somente em relação ao algoritmo proposto e o algoritmo multirrótulo-binário ECC.



### 6.3 EXPERIMENTO II: BicbPT $\beta$ x {BR, CC, ECC, RAKEL<sub>1</sub>, RAKEL<sub>2</sub>, LP}

Nesta seção são apresentados os resultados com os experimentos comparando o BicbPT $\beta$  com os algoritmos descritos anteriormente no Capítulo 2.2, apresentando-se na seguinte ordem, o C4.5 como classificador base, SVM e por último o Naive Bayes.

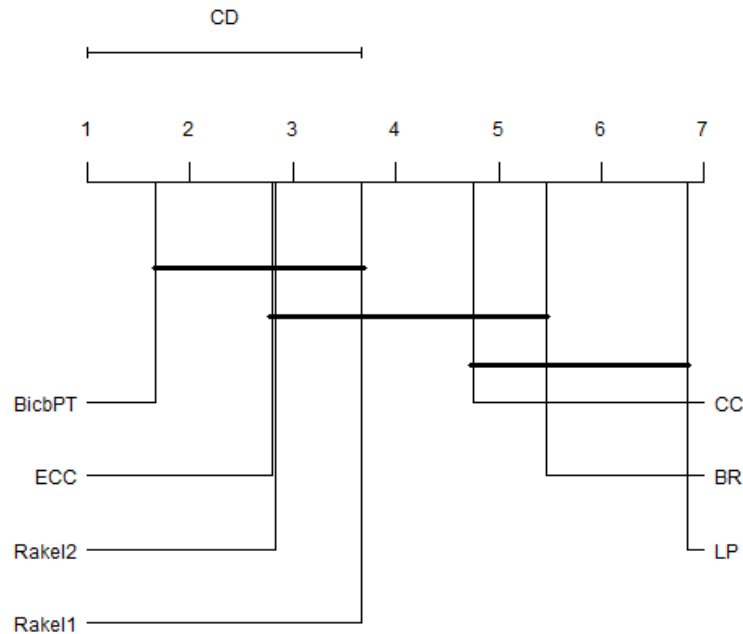
Como observa-se na Tabela 19, JI com C4.5, ao considerar somente os algoritmos de transformação multirrótulo-binário, o BicbPT $\beta$  foi melhor em 8 de 12 bases nesta métrica, não conseguindo bater o ECC, que também é multirrótulo-binário, em 4 bases, mas como pode ser visto na Tabela 23, o ECC possui um tempo total bem mais alto que o BicbPT $\beta$ , e assim, a relação JI x TT é positiva para o algoritmo proposto.

Tabela 19: BicbPT $\beta$  x Todos – JI com C4.5

Bases	JI $\uparrow$						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,597 $\pm$ 0,03	0,548 $\pm$ 0,03	0,553 $\pm$ 0,02	0,565 $\pm$ 0,01	0,561 $\pm$ 0,02	0,572 $\pm$ 0,02	0,491 $\pm$ 0,03
<b>Emotions</b>	0,562 $\pm$ 0,02	0,437 $\pm$ 0,01	0,437 $\pm$ 0,02	0,5 $\pm$ 0,01	0,492 $\pm$ 0,02	0,479 $\pm$ 0,02	0,427 $\pm$ 0,02
<b>Enron</b>	0,435 $\pm$ 0,01	0,392 $\pm$ 0,01	0,401 $\pm$ 0,01	0,45 $\pm$ 0,01	0,378 $\pm$ 0,01	0,416 $\pm$ 0,02	0,189 $\pm$ 0,02
<b>Flags</b>	0,631 $\pm$ 0,02	0,561 $\pm$ 0,03	0,574 $\pm$ 0,02	0,598 $\pm$ 0,03	0,603 $\pm$ 0,02	0,613 $\pm$ 0,01	0,436 $\pm$ 0,06
<b>Genbase</b>	0,985 $\pm$ 0,01	0,976 $\pm$ 0,01	0,976 $\pm$ 0,01	0,973 $\pm$ 0,01	0,974 $\pm$ 0,01	0,976 $\pm$ 0,01	0,97 $\pm$ 0,01
<b>Medical</b>	0,767 $\pm$ 0,03	0,721 $\pm$ 0,03	0,725 $\pm$ 0,03	0,743 $\pm$ 0,03	0,75 $\pm$ 0,02	0,723 $\pm$ 0,03	0,701 $\pm$ 0,02
<b>Scene</b>	0,628 $\pm$ 0,02	0,514 $\pm$ 0,01	0,58 $\pm$ 0,01	0,609 $\pm$ 0,01	0,608 $\pm$ 0,01	0,602 $\pm$ 0,01	0,566 $\pm$ 0,01
<b>Yeast</b>	0,476 $\pm$ 0,01	0,419 $\pm$ 0,02	0,417 $\pm$ 0,01	0,486 $\pm$ 0,01	0,46 $\pm$ 0,01	0,471 $\pm$ 0,01	0,382 $\pm$ 0,01
<b>Bibtex</b>	0,297 $\pm$ 0,01	0,287 $\pm$ 0,01	0,285 $\pm$ 0,01	0,287 $\pm$ 0,01	0,283 $\pm$ 0	0,29 $\pm$ 0,01	0,181 $\pm$ 0,01
<b>G1</b>	0,448 $\pm$ 0,01	0,307 $\pm$ 0,01	0,357 $\pm$ 0,01	0,343 $\pm$ 0,01	0,38 $\pm$ 0,01	0,387 $\pm$ 0,01	0,309 $\pm$ 0,01
<b>GP</b>	0,433 $\pm$ 0,02	0,399 $\pm$ 0,02	0,4 $\pm$ 0,01	0,482 $\pm$ 0,02	0,466 $\pm$ 0,01	0,455 $\pm$ 0,02	0,385 $\pm$ 0,02
<b>BFRC</b>	0,455 $\pm$ 0	0,448 $\pm$ 0	0,448 $\pm$ 0	0,486 $\pm$ 0	0,477 $\pm$ 0	0,489 $\pm$ 0	0,421 $\pm$ 0,01

Fonte: Autoria Própria (2022)

O p-valor com 95% de confiança no teste de Friedman é de 5.615e-09, o que indica que existe diferença significativa entre os algoritmos comparados. No teste de Nemenyi, apresentado na Figura 46 é possível observar que o BicbPT $\beta$  apresenta desempenho semelhante, estatisticamente, ao ECC, RAkEL<sub>1</sub> e RAkEL<sub>2</sub>, mas posicionando-se em primeiro lugar no ranking.

Figura 46: Gráfico de Nemenyi BicbPT $\beta$  x Todos – JI com C4.5

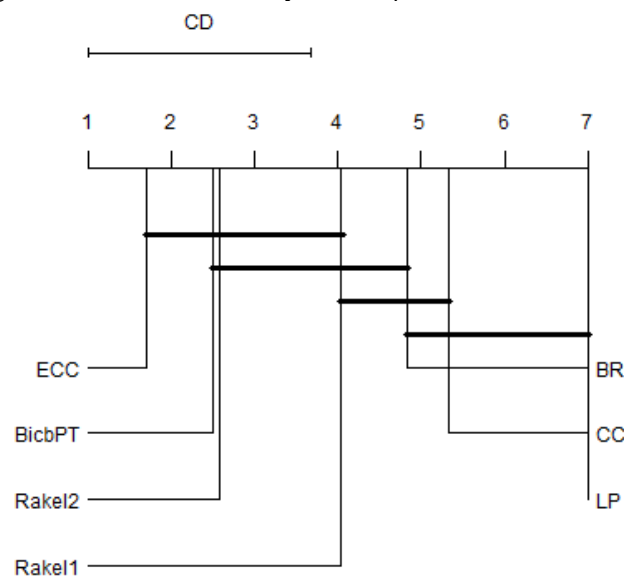
Fonte: Autoria Própria (2022)

Para a métrica HL com o C4.5, Tabela 20, o BicbPT $\beta$  conseguiu bons resultados somente em 4 das 12 bases, mas ao analisar o gráfico da Figura 47, gerado pelo teste de Nemenyi, é possível verificar que o algoritmo proposto se posiciona em 2º lugar para esta métrica, ao utilizar o C4.5 como classificador base, e ainda, apresenta resultados estatisticamente semelhantes a ECC, RAKEL<sub>1</sub> e RAKEL<sub>2</sub>. O p-valor para 95% de confiança em Friedman obtido aqui foi de 1.673e-10.

Tabela 20: BicbPT $\beta$  x Todos – HL com C4.5

Bases	HL ↓						
	BicbPT $\beta$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	0,050 ± 0	0,054 ± 0	0,054 ± 0	0,045 ± 0	0,052 ± 0	0,048 ± 0	0,281 ± 0,03
<b>Emotions</b>	0,216 ± 0,01	0,260 ± 0,01	0,275 ± 0,01	0,208 ± 0,01	0,240 ± 0,01	0,232 ± 0,01	0,334 ± 0,02
<b>Enron</b>	0,051 ± 0	0,054 ± 0	0,054 ± 0	0,048 ± 0	0,057 ± 0	0,050 ± 0	0,657 ± 0,03
<b>Flags</b>	0,222 ± 0,01	0,279 ± 0,02	0,281 ± 0,02	0,261 ± 0,02	0,263 ± 0,02	0,252 ± 0,01	0,466 ± 0,06
<b>Genbase</b>	0,001 ± 0	0,002 ± 0	0,002 ± 0	0,002 ± 0	0,003 ± 0	0,002 ± 0	0,005 ± 0
<b>Medical</b>	0,010 ± 0	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,011 ± 0	0,087 ± 0,02
<b>Scene</b>	0,128 ± 0,01	0,140 ± 0	0,146 ± 0	0,097 ± 0	0,109 ± 0	0,107 ± 0	0,175 ± 0,01
<b>Yeast</b>	0,227 ± 0,01	0,255 ± 0,01	0,274 ± 0	0,209 ± 0	0,242 ± 0	0,232 ± 0	0,394 ± 0,01
<b>Bibtex</b>	0,015 ± 0	0,015 ± 0	0,015 ± 0	0,013 ± 0	0,014 ± 0	0,014 ± 0	0,578 ± 0,01
<b>G1</b>	0,240 ± 0	0,273 ± 0,01	0,286 ± 0,01	0,253 ± 0	0,266 ± 0,01	0,258 ± 0,01	0,495 ± 0,01
<b>GP</b>	0,294 ± 0,01	0,332 ± 0,01	0,332 ± 0,01	0,260 ± 0,01	0,295 ± 0,01	0,285 ± 0,01	0,428 ± 0,02
<b>BFRC</b>	0,380 ± 0	0,383 ± 0	0,384 ± 0	0,336 ± 0	0,368 ± 0	0,358 ± 0	0,461 ± 0

Fonte: Autoria Própria (2022)

Figura 47: Gráfico de Nemenyi BicbPT $\beta$  x Todos – HL com C4.5

Fonte: Autoria Própria (2022)



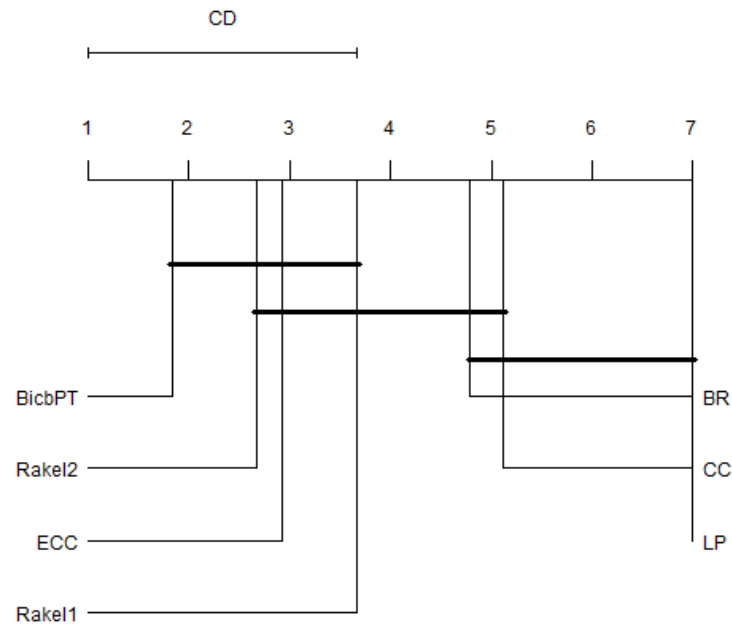
Para a métrica  $F1_{\text{micro}}$ , Tabela 21, o algoritmo proposto conseguiu bom desempenho em 8 de 12 bases. De maneira análoga a métrica JI, onde, entre os algoritmos multirrótulo-binários o ECC se saiu melhor em 4 bases, a Tabela 23 indica que o BicbPT $\beta$  possui tempo de execução menor, sendo assim um ponto positivo a ser considerado.

Tabela 21: BicbPT $\beta$  x Todos –  $F1_{\text{micro}}$  com C4.5

Bases	$F1_{\text{micro}} \uparrow$						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,443 ± 0,04	0,378 ± 0,04	0,382 ± 0,03	0,360 ± 0,02	0,398 ± 0,03	0,394 ± 0,03	0,191 ± 0,01
<b>Emotions</b>	0,671 ± 0,02	0,584 ± 0,01	0,558 ± 0,02	0,639 ± 0,01	0,623 ± 0,02	0,614 ± 0,02	0,535 ± 0,02
<b>Enron</b>	0,563 ± 0,01	0,519 ± 0,01	0,516 ± 0,01	0,566 ± 0,01	0,490 ± 0,01	0,550 ± 0,01	0,123 ± 0
<b>Flags</b>	0,763 ± 0,02	0,712 ± 0,02	0,710 ± 0,02	0,733 ± 0,02	0,742 ± 0,01	0,750 ± 0,01	0,577 ± 0,06
<b>Genbase</b>	0,987 ± 0,01	0,979 ± 0,01	0,979 ± 0,01	0,977 ± 0,01	0,972 ± 0,01	0,979 ± 0,01	0,951 ± 0,03
<b>Medical</b>	0,818 ± 0,02	0,789 ± 0,02	0,788 ± 0,02	0,800 ± 0,02	0,795 ± 0,01	0,790 ± 0,02	0,334 ± 0,04
<b>Scene</b>	0,674 ± 0,01	0,605 ± 0,01	0,596 ± 0,01	0,691 ± 0,01	0,676 ± 0,01	0,678 ± 0,01	0,556 ± 0,01
<b>Yeast</b>	0,623 ± 0,01	0,566 ± 0,01	0,545 ± 0,01	0,622 ± 0,01	0,599 ± 0,01	0,611 ± 0,01	0,493 ± 0,01
<b>Bibtex</b>	0,398 ± 0,01	0,389 ± 0,01	0,384 ± 0,01	0,386 ± 0	0,374 ± 0	0,394 ± 0	0,036 ± 0
<b>G1</b>	0,551 ± 0,01	0,425 ± 0,01	0,455 ± 0,01	0,440 ± 0,01	0,491 ± 0,01	0,497 ± 0,01	0,421 ± 0,01
<b>GP</b>	0,569 ± 0,03	0,539 ± 0,01	0,540 ± 0,01	0,603 ± 0,01	0,594 ± 0,01	0,579 ± 0,01	0,523 ± 0,02
<b>BFRC</b>	0,618 ± 0	0,611 ± 0	0,611 ± 0	0,641 ± 0	0,638 ± 0	0,648 ± 0	0,593 ± 0

Fonte: Autoria Própria (2022)

Ainda, após o teste de Friedman, com 95% de confiança e p-valor de 1.022e-08, o pós-teste de Nemenyi, Figura 48, indica similaridade estatística entre o BicbPT $\beta$ , RAkEL<sub>2</sub>, ECC e RAkEL<sub>1</sub>, algoritmos que geralmente apresentam bom desempenho em métricas, mas tempo de execução alto. Assim, o algoritmo proposto entre neste grupo, mas como um que consegue obter um tempo de execução menor e com desempenho similar na métrica.

Figura 48: Gráfico de Nemenyi BicbPT $\beta$  x Todos – F1<sub>micro</sub> com C4.5

Fonte: Autoria Própria (2022)

Na Tabela 22, onde é apresentado o resultado da métrica F1<sub>macro</sub>, ao considerar somente os algoritmos de transformação multirrótulo-binário, diferentemente das outras métricas já apresentadas, o BicbPT $\beta$  possui bom desempenho em 11 de 12 bases. Na comparação geral, o desempenho é melhor em 8 de 12 bases, o que indica bom desempenho do algoritmo proposto, tanto na categoria de multirrótulo-binário, quanto na comparação com os demais algoritmos.

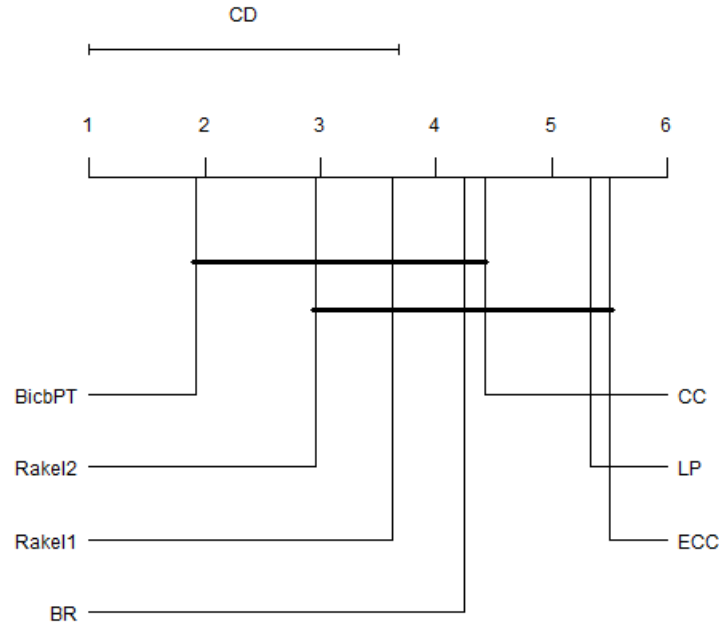
Tabela 22: BicbPT $\beta$  x Todos – F1<sub>macro</sub> com C4.5

Bases	F1 <sub>macro</sub> ↑						
	BicbPT $\beta$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	0,299 ± 0,03	0,249 ± 0,03	0,250 ± 0,03	0,188 ± 0,03	0,257 ± 0,02	0,255 ± 0,02	0,172 ± 0,01
<b>Emotions</b>	0,659 ± 0,02	0,573 ± 0,02	0,549 ± 0,02	0,621 ± 0,01	0,616 ± 0,02	0,605 ± 0,02	0,529 ± 0,02
<b>Enron</b>	0,161 ± 0,01	0,144 ± 0,01	0,143 ± 0,01	0,134 ± 0,01	0,138 ± 0,01	0,146 ± 0,01	0,103 ± 0
<b>Flags</b>	0,634 ± 0,03	0,606 ± 0,04	0,625 ± 0,01	0,622 ± 0,02	0,670 ± 0,02	0,660 ± 0,02	0,539 ± 0,04
<b>Genbase</b>	0,766 ± 0,04	0,713 ± 0,04	0,713 ± 0,04	0,683 ± 0,04	0,699 ± 0,05	0,713 ± 0,04	0,668 ± 0,06
<b>Medical</b>	0,400 ± 0,03	0,353 ± 0,03	0,352 ± 0,03	0,334 ± 0,03	0,352 ± 0,02	0,353 ± 0,03	0,180 ± 0,01
<b>Scene</b>	0,690 ± 0,02	0,615 ± 0,01	0,611 ± 0,01	0,698 ± 0,01	0,680 ± 0,01	0,688 ± 0,01	0,564 ± 0,01
<b>Yeast</b>	0,410 ± 0,01	0,380 ± 0,01	0,390 ± 0,01	0,372 ± 0,01	0,386 ± 0,01	0,392 ± 0	0,396 ± 0,01
<b>Bibtex</b>	0,267 ± 0	0,260 ± 0	0,254 ± 0	0,210 ± 0	0,216 ± 0	0,257 ± 0	0,035 ± 0
<b>G1</b>	0,458 ± 0,01	0,353 ± 0,01	0,374 ± 0,02	0,335 ± 0,01	0,412 ± 0,01	0,416 ± 0,01	0,400 ± 0,01
<b>GP</b>	0,365 ± 0,04	0,393 ± 0,02	0,393 ± 0,02	0,357 ± 0,02	0,392 ± 0,01	0,371 ± 0,01	0,427 ± 0,02
<b>BFRC</b>	0,536 ± 0	0,531 ± 0	0,528 ± 0	0,516 ± 0	0,546 ± 0	0,542 ± 0	0,556 ± 0,01

Fonte: Autoria Própria (2022)

Em uma análise estatística com 95% de confiança, o teste de Friedman retornou um p-valor de 0.0002708, indicando haver diferença significativa entre os algoritmos comparados, com o pós-teste de Nemenyi, Figura 49, demonstrando que o BicbPT $\beta$  é melhor, estatisticamente, que o ECC para esta métrica, mas apresenta resultados similares ao BR e CC, embora estes dois se posicionem em 4º e 5º lugar.

Figura 49: Gráfico de Nemenyi BicbPT $\beta$  x Todos – F1<sub>macro</sub> com C4.5



Fonte: Autoria Própria (2022)

A Tabela 23 apresenta o resultado obtido com o tempo total de execução ao utilizar o C4.5 como classificador base para todos os algoritmos. Ao considerar somente os algoritmos de transformação de problema multirrótulo-binário, o BicbPT $\beta$  possui, em média, tempo semelhante ao BR e CC, e é 5 vezes mais rápido que o ECC. Em alguns casos a diferença com o ECC pode chegar a 7 vezes, como ocorre com as bases Enron, Medical e Bibtex.

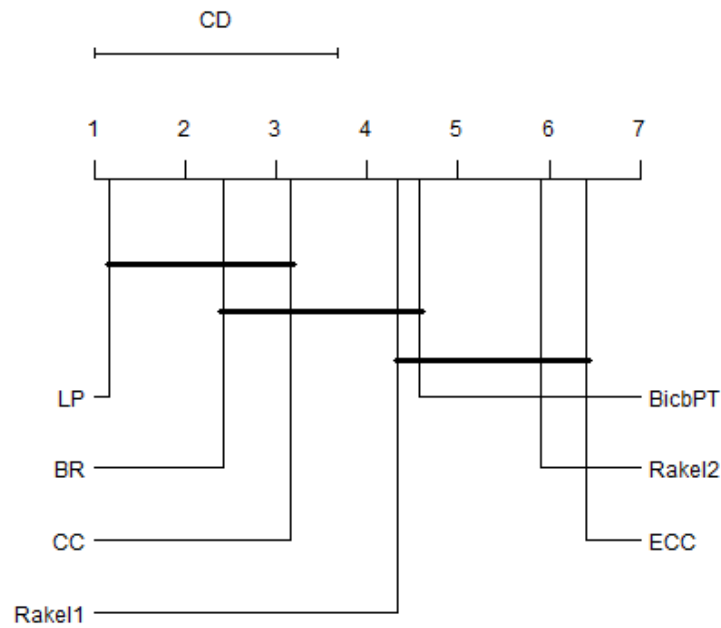
Já em relação ao algoritmos não-binários, o BicbPT $\beta$  é em média 3 vezes mais rápido que o RAKEL<sub>2</sub>. Todos os algoritmos são mais lentos que o LP, mas isto é devido ao fato de que o LP não gera subproblemas, sua transformação é 1:1, ou seja, um multirrótulo para um multiclasse.

Tabela 23: BicbPT $\beta$  x Todos – TT com C4.5

Bases	TT ↓						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	1,610 ± 0,29	0,930 ± 0,27	1,166 ± 0,25	5,188 ± 0,21	3,921 ± 0,46	6,026 ± 0,19	0,422 ± 0,02
<b>Emotions</b>	1,704 ± 0,09	0,158 ± 0,03	0,223 ± 0,05	0,858 ± 0,04	0,525 ± 0,04	0,583 ± 0,02	0,119 ± 0,01
<b>Enron</b>	93,213 ± 11,12	95,275 ± 1,93	111,647 ± 3,67	626,610 ± 13,27	179,023 ± 4,67	543,048 ± 10,37	13,147 ± 0,55
<b>Flags</b>	0,484 ± 0,05	0,023 ± 0,01	0,045 ± 0,01	0,091 ± 0,01	0,041 ± 0,01	0,040 ± 0,01	0,013 ± 0,01
<b>Genbase</b>	2,422 ± 0,17	1,036 ± 0,2	1,841 ± 0,5	4,704 ± 0,4	2,354 ± 0,27	5,220 ± 0,25	0,632 ± 0,04
<b>Medical</b>	5,516 ± 0,25	5,086 ± 0,25	6,348 ± 0,26	38,711 ± 1,03	11,454 ± 0,39	26,894 ± 0,87	1,417 ± 0,09
<b>Scene</b>	12,111 ± 1,64	3,196 ± 0,11	3,796 ± 0,13	18,585 ± 0,24	11,800 ± 0,29	14,025 ± 0,3	2,178 ± 0,07
<b>Yeast</b>	5,610 ± 0,19	3,346 ± 0,17	3,039 ± 0,06	16,698 ± 0,22	12,839 ± 0,27	19,822 ± 0,46	2,485 ± 0,06
<b>Bibtex</b>	1186,331 ± 15,12	1316,432 ± 17,88	1318,304 ± 35,32	8401,920 ± 53,9	855,178 ± 11,28	5990,728 ± 41,73	90,707 ± 2,68
<b>G1</b>	700,238 ± 46,72	36,433 ± 0,78	31,646 ± 0,72	184,298 ± 5,92	162,256 ± 6,02	216,617 ± 6,62	67,801 ± 4,66
<b>GP</b>	20,986 ± 6,42	8,665 ± 0,35	9,209 ± 0,31	53,235 ± 1,17	32,315 ± 1,14	38,461 ± 1,15	4,320 ± 0,96
<b>BFRC</b>	613,837 ± 10,4	517,272 ± 26,08	427,250 ± 4,12	2668,001 ± 37,16	1371,874 ± 21,97	1853,073 ± 30,05	102,221 ± 1,9

Fonte: Autoria Própria (2022)

O p-valor com 95% de confiança obtido aqui foi de 1.022e-08 e o teste de Nemenyi, Figura 50, demonstra não haver diferença significativa, estatisticamente, entre o BicbPT $\beta$  e BR e CC, assim o algoritmo proposto consegue, ao utilizar o C4.5 como classificador base, manter a característica de baixo tempo de execução dos algoritmos multirrótulo-binário, mas apresentar bom desempenho nas métricas JI, HL,  $F1_{\text{micro}}$  e  $F1_{\text{macro}}$ , diferentemente destes.

Figura 50: Gráfico de Nemenyi BicbPT $\beta$  x Todos – TT com C4.5

Fonte: Autoria Própria (2022)

Na Tabela 24 apresentam-se os resultados obtidos na métrica Jaccard Index ao utilizar o SVM como classificador base. Nessa configuração o BicbPT $\beta$  obteve os melhores resultados em apenas 5 de 12 bases, ao considerar todos os algoritmos e em 7 de 12 bases ao considerar os algoritmos de transformação multirrótulo-binário.

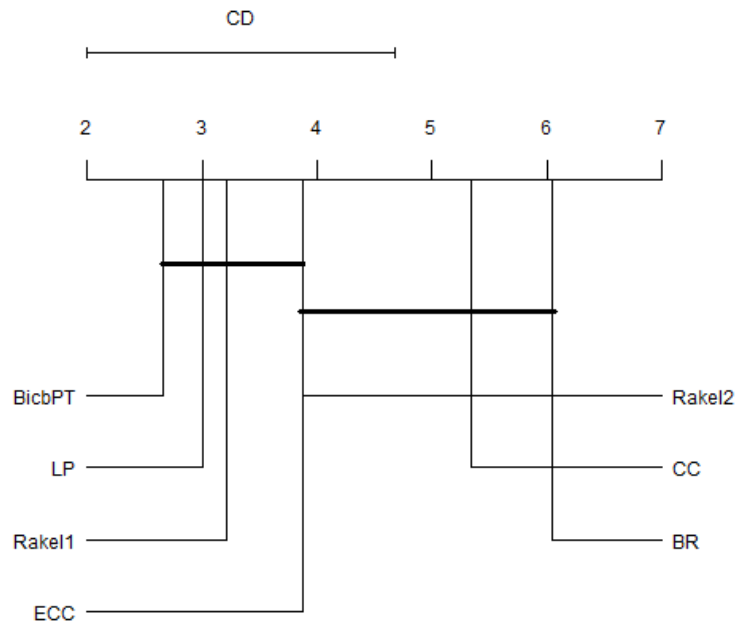
É interessante observar que ao utilizar o SVM, os resultados obtidos são melhores que com o C4.5 para todos os algoritmos, isto demonstra que a escolha do classificador base influencia diretamente no resultado final da classificação multirrótulo gerado pelo algoritmo de transformação de problema.

Tabela 24: BicbPT $\beta$  x Todos – JI com SVM

Bases	JI $\uparrow$						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,626 $\pm$ 0,01	0,603 $\pm$ 0,01	0,607 $\pm$ 0,01	0,604 $\pm$ 0,01	0,617 $\pm$ 0,01	0,611 $\pm$ 0,01	0,612 $\pm$ 0,01
<b>Emotions</b>	0,589 $\pm$ 0,01	0,497 $\pm$ 0,01	0,525 $\pm$ 0,02	0,559 $\pm$ 0,02	0,569 $\pm$ 0,02	0,564 $\pm$ 0,02	0,568 $\pm$ 0,01
<b>Enron</b>	0,414 $\pm$ 0,01	0,387 $\pm$ 0,01	0,394 $\pm$ 0,01	0,433 $\pm$ 0,01	0,438 $\pm$ 0,01	0,397 $\pm$ 0,01	0,408 $\pm$ 0,01
<b>Flags</b>	0,641 $\pm$ 0,01	0,590 $\pm$ 0,01	0,577 $\pm$ 0,02	0,596 $\pm$ 0,01	0,585 $\pm$ 0,01	0,590 $\pm$ 0,01	0,554 $\pm$ 0,03
<b>Genbase</b>	0,991 $\pm$ 0,01	0,982 $\pm$ 0,01	0,982 $\pm$ 0,01	0,980 $\pm$ 0,01	0,980 $\pm$ 0,01	0,981 $\pm$ 0,01	0,982 $\pm$ 0,01
<b>Medical</b>	0,754 $\pm$ 0,01	0,730 $\pm$ 0,01	0,746 $\pm$ 0,01	0,761 $\pm$ 0,01	0,752 $\pm$ 0,01	0,731 $\pm$ 0,01	0,750 $\pm$ 0,01
<b>Scene</b>	0,676 $\pm$ 0,02	0,587 $\pm$ 0,01	0,677 $\pm$ 0,01	0,690 $\pm$ 0,01	0,675 $\pm$ 0,01	0,668 $\pm$ 0,01	0,724 $\pm$ 0,01
<b>Yeast</b>	0,507 $\pm$ 0,01	0,498 $\pm$ 0,01	0,482 $\pm$ 0,02	0,527 $\pm$ 0,01	0,532 $\pm$ 0,01	0,524 $\pm$ 0,01	0,520 $\pm$ 0,01
<b>Bibtex</b>	0,326 $\pm$ 0,01	0,313 $\pm$ 0,01	0,313 $\pm$ 0,01	0,311 $\pm$ 0,01	0,307 $\pm$ 0,01	0,316 $\pm$ 0,01	0,323 $\pm$ 0,01
<b>G1</b>	0,404 $\pm$ 0	0,368 $\pm$ 0,01	0,376 $\pm$ 0,01	0,401 $\pm$ 0,01	0,416 $\pm$ 0,01	0,416 $\pm$ 0,01	0,417 $\pm$ 0,01
<b>GP</b>	0,475 $\pm$ 0,02	0,444 $\pm$ 0,02	0,446 $\pm$ 0,02	0,474 $\pm$ 0,02	0,495 $\pm$ 0,01	0,491 $\pm$ 0,02	0,504 $\pm$ 0,02
<b>BFRC</b>	0,475 $\pm$ 0	0,468 $\pm$ 0	0,468 $\pm$ 0	0,488 $\pm$ 0	0,504 $\pm$ 0	0,504 $\pm$ 0	0,489 $\pm$ 0

Fonte: Autoria Própria (2022)

Nas bases onde o BicbPT tem desempenho pior que o ECC, isto é compensado pelo tempo de execução, assim como com o C4.5, como é possível observar na Tabela 28. O p-valor para a métrica JI foi de 0.0004243 com 95% de confiança, e na Figura 51, o gráfico com Nemenyi demonstra, que entre os algoritmos de transformação multirrótulo-binário, o BicbPT $\beta$  é melhor, estatisticamente, que BR e CC, e não apresenta diferença em relação ao ECC.

Figura 51: Gráfico de Nemenyi BicbPT $\beta$  x Todos – JI com SVM

Fonte: Autoria Própria (2022)

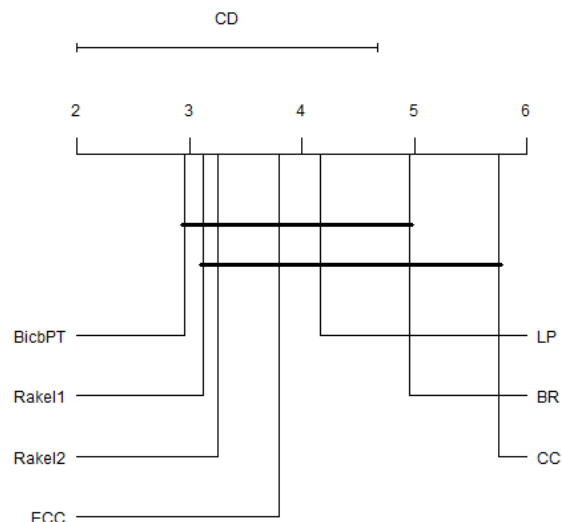
Para a métrica HL, Tabela 25, utilizando o SVM como classificador base, diferentemente do resultado apresentado pela mesma métrica utilizando o C4.5 como classificador base, Tabela 20, aqui o BicbPT $\beta$  se saiu melhor em 5 de 12 bases na comparação geral e foi melhor em 8 de 12 bases ao considerar somente os algoritmos multirrótulo-binário.

Também se nota um equilíbrio nos resultados desta métrica entre os diversos algoritmos ao utilizar o SVM, o que se confirma no gráfico apresentado pela Figura 52, onde pela análise de Nemenyi, não existe diferença significativa entre os algoritmos comparados. O p-valor para HL aqui foi de 0.006627.

Tabela 25: BicbPT $\beta$  x Todos – HL com SVM

Bases	HL ↓						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,049 ± 0	0,052 ± 0	0,051 ± 0	0,051 ± 0	0,047 ± 0	0,048 ± 0	0,050 ± 0
<b>Emotions</b>	0,180 ± 0	0,204 ± 0	0,222 ± 0,01	0,206 ± 0,01	0,196 ± 0,01	0,195 ± 0,01	0,207 ± 0,01
<b>Enron</b>	0,057 ± 0	0,060 ± 0	0,060 ± 0	0,053 ± 0	0,052 ± 0	0,058 ± 0	0,057 ± 0
<b>Flags</b>	0,219 ± 0,01	0,265 ± 0,01	0,278 ± 0,02	0,265 ± 0,01	0,274 ± 0,01	0,268 ± 0,01	0,291 ± 0,02
<b>Genbase</b>	0,001 ± 0	0,001 ± 0	0,001 ± 0	0,002 ± 0	0,002 ± 0	0,002 ± 0	0,002 ± 0
<b>Medical</b>	0,010 ± 0	0,011 ± 0	0,011 ± 0	0,010 ± 0	0,011 ± 0	0,011 ± 0	0,012 ± 0
<b>Scene</b>	0,102 ± 0	0,110 ± 0	0,110 ± 0	0,097 ± 0	0,095 ± 0	0,096 ± 0	0,094 ± 0
<b>Yeast</b>	0,196 ± 0	0,201 ± 0	0,213 ± 0,01	0,204 ± 0	0,204 ± 0	0,201 ± 0	0,212 ± 0
<b>Bibtex</b>	0,015 ± 0	0,015 ± 0	0,015 ± 0	0,013 ± 0	0,013 ± 0	0,015 ± 0	0,016 ± 0
<b>G1</b>	0,270 ± 0	0,287 ± 0	0,286 ± 0	0,265 ± 0	0,259 ± 0	0,257 ± 0	0,256 ± 0,01
<b>GP</b>	0,264 ± 0,01	0,295 ± 0,01	0,295 ± 0,01	0,275 ± 0,01	0,264 ± 0,01	0,262 ± 0,01	0,257 ± 0,01
<b>BFRC</b>	0,354 ± 0	0,359 ± 0	0,361 ± 0	0,354 ± 0	0,343 ± 0	0,342 ± 0	0,340 ± 0

Fonte: Autoria Própria (2022)

Figura 52: Gráfico de Nemenyi BicbPT $\beta$  x Todos – HL com SVM

Fonte: Autoria Própria (2022)

Na Tabela 26 são apresentados os resultados da métrica  $F1_{\text{micro}}$  com o SVM. Aqui, como também é observado nas outras métricas ao utilizar o SVM, os resultados tem uma melhora significativa em relação ao uso do C4.5. O BicbPT $\beta$  se saiu melhor em 8 de 12 bases ao considerar os algoritmos multirrótulo-binário.

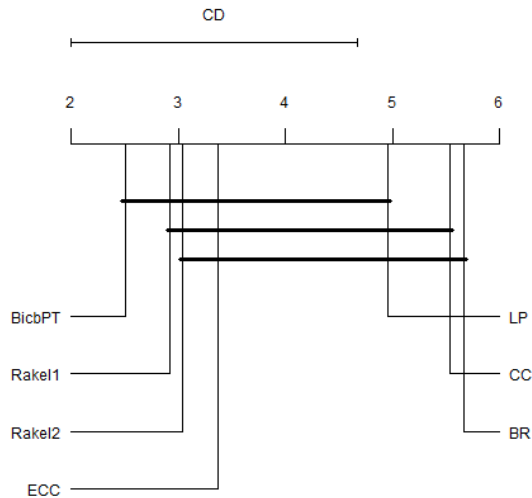
Com um p-valor de 9.251e-05 no teste de Friedman, indicou-se a existência de diferenças entre os algoritmos, e após com o teste de Nemenyi, Figura 53, foi possível verificar que o BicbPT $\beta$  se saiu melhor que os algoritmos BR e CC, mas não apresentou diferenças significativas com o ECC, embora, como já mencionado, o ECC possui um tempo de execução mais elevado.

Tabela 26: BicbPT $\beta$  x Todos –  $F1_{\text{micro}}$  com SVM

Bases	$F1_{\text{micro}} \uparrow$						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,497 ± 0,02	0,456 ± 0,02	0,463 ± 0,02	0,466 ± 0,02	0,493 ± 0,03	0,477 ± 0,03	0,474 ± 0,02
<b>Emotions</b>	0,705 ± 0,01	0,638 ± 0,01	0,639 ± 0,02	0,673 ± 0,01	0,683 ± 0,01	0,679 ± 0,01	0,673 ± 0,01
<b>Enron</b>	0,534 ± 0,01	0,503 ± 0,01	0,502 ± 0,01	0,547 ± 0,01	0,547 ± 0,01	0,516 ± 0,01	0,500 ± 0,01
<b>Flags</b>	0,771 ± 0,01	0,732 ± 0,01	0,716 ± 0,02	0,737 ± 0,01	0,725 ± 0,01	0,729 ± 0,01	0,694 ± 0,03
<b>Genbase</b>	0,991 ± 0,01	0,985 ± 0,01	0,985 ± 0,01	0,983 ± 0,01	0,982 ± 0,01	0,983 ± 0,01	0,981 ± 0,01
<b>Medical</b>	0,814 ± 0,01	0,792 ± 0,01	0,798 ± 0,01	0,811 ± 0,01	0,798 ± 0,01	0,793 ± 0,01	0,769 ± 0,01
<b>Scene</b>	0,721 ± 0,01	0,674 ± 0,01	0,688 ± 0,01	0,722 ± 0,01	0,725 ± 0,01	0,723 ± 0,01	0,734 ± 0,01
<b>Yeast</b>	0,639 ± 0,01	0,632 ± 0,01	0,615 ± 0,01	0,649 ± 0,01	0,652 ± 0,01	0,650 ± 0,01	0,633 ± 0,01
<b>Bibtex</b>	0,418 ± 0,01	0,410 ± 0,01	0,410 ± 0	0,416 ± 0	0,392 ± 0	0,414 ± 0,01	0,353 ± 0
<b>G1</b>	0,523 ± 0	0,491 ± 0	0,492 ± 0,01	0,518 ± 0,01	0,530 ± 0,01	0,531 ± 0,01	0,513 ± 0,01
<b>GP</b>	0,602 ± 0,03	0,573 ± 0,02	0,575 ± 0,02	0,599 ± 0,01	0,610 ± 0,01	0,605 ± 0,02	0,612 ± 0,01
<b>BFRC</b>	0,635 ± 0	0,629 ± 0	0,628 ± 0	0,648 ± 0	0,661 ± 0	0,662 ± 0	0,641 ± 0

Fonte: Autoria Própria (2022)



Figura 53: Gráfico de Nemenyi BicbPT $\beta$  x Todos – F1<sub>micro</sub> com SVM

Fonte: Autoria Própria (2022)

Na métrica  $F1_{macro}$ , Tabela 27, o algoritmo proposto apresentou-se melhor em 8 de 12 bases, quando comparado com BR, CC e ECC, os quais são algoritmos de transformação de problema multirrótulo-binário. O p-valor obtido aqui pelo teste de Friedman com 95% de confiança foi de 0.04897.

Tabela 27: BicbPT $\beta$  x Todos – F1<sub>macro</sub> com SVM

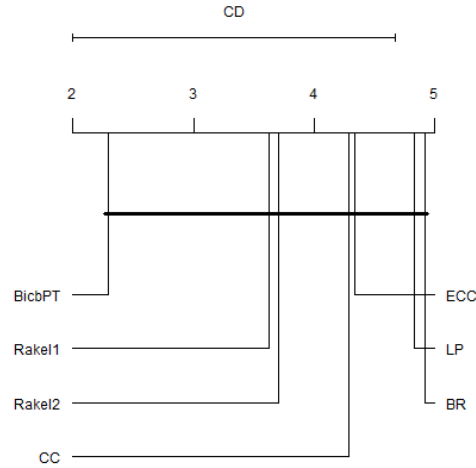
Bases	F1 <sub>macro</sub> ↑						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,411 ± 0,03	0,349 ± 0,03	0,368 ± 0,04	0,361 ± 0,03	0,377 ± 0,03	0,373 ± 0,04	0,375 ± 0,02
<b>Emotions</b>	0,686 ± 0,02	0,599 ± 0,01	0,596 ± 0,02	0,663 ± 0,02	0,668 ± 0,01	0,661 ± 0,01	0,666 ± 0,01
<b>Enron</b>	0,233 ± 0,02	0,201 ± 0,02	0,201 ± 0,02	0,203 ± 0,02	0,198 ± 0,02	0,203 ± 0,02	0,185 ± 0,02
<b>Flags</b>	0,685 ± 0,04	0,652 ± 0,02	0,640 ± 0,03	0,658 ± 0,02	0,637 ± 0,02	0,635 ± 0,02	0,604 ± 0,03
<b>Genbase</b>	0,813 ± 0,04	0,756 ± 0,04	0,758 ± 0,03	0,753 ± 0,04	0,751 ± 0,04	0,753 ± 0,04	0,757 ± 0,04
<b>Medical</b>	0,403 ± 0,02	0,363 ± 0,02	0,368 ± 0,02	0,357 ± 0,02	0,363 ± 0,02	0,360 ± 0,02	0,343 ± 0,02
<b>Scene</b>	0,729 ± 0,01	0,681 ± 0,01	0,699 ± 0,01	0,731 ± 0,01	0,732 ± 0,01	0,732 ± 0,01	0,743 ± 0,01
<b>Yeast</b>	0,340 ± 0,01	0,327 ± 0,01	0,359 ± 0,01	0,357 ± 0,01	0,399 ± 0,01	0,363 ± 0,01	0,407 ± 0,01
<b>Bibtex</b>	0,310 ± 0,01	0,299 ± 0,01	0,296 ± 0,01	0,265 ± 0,01	0,230 ± 0,01	0,301 ± 0,01	0,217 ± 0
<b>G1</b>	0,483 ± 0,01	0,454 ± 0,01	0,453 ± 0,01	0,471 ± 0,01	0,483 ± 0,01	0,486 ± 0,01	0,462 ± 0,02
<b>GP</b>	0,381 ± 0,05	0,387 ± 0,02	0,388 ± 0,02	0,377 ± 0,02	0,379 ± 0,02	0,372 ± 0,02	0,356 ± 0,01
<b>BFRC</b>	0,558 ± 0	0,553 ± 0	0,555 ± 0	0,573 ± 0	0,586 ± 0,01	0,579 ± 0,01	0,541 ± 0,01

Fonte: Autoria Própria (2022)

Mesmo o p-valor indicando a existência de diferença entre os algoritmos, ele está bem próximo do limite entre aceitar ou rejeitar a hipótese nula, assim, quando se realizou o teste de

Nemenyi, Figura 54, foi possível confirmar que não existem diferenças significativas entre nenhum dos algoritmos comparados.

Figura 54: Gráfico de Nemenyi BicbPT $\beta$  x Todos – F1<sub>macro</sub> com SVM



Fonte: Autoria Própria (2022)

Tabela 28: BicbPT $\beta$  x Todos – TT com SVM

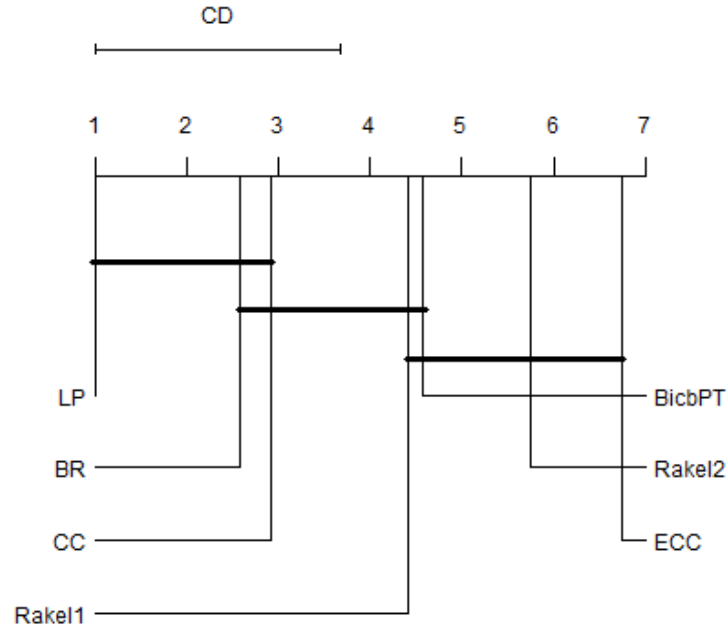
Bases	TT ↓						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	1,969 ± 0,09	0,955 ± 0,08	1,044 ± 0,07	7,144 ± 0,32	2,575 ± 0,13	4,462 ± 0,31	0,527 ± 0,01
<b>Emotions</b>	2,812 ± 0,65	0,213 ± 0,02	0,265 ± 0,04	1,478 ± 0,03	0,742 ± 0,02	0,880 ± 0,01	0,138 ± 0,01
<b>Enron</b>	24,159 ± 0,76	21,065 ± 0,71	26,645 ± 0,64	187,518 ± 2,56	47,112 ± 1,31	116,448 ± 2,6	7,865 ± 0,37
<b>Flags</b>	0,969 ± 0,56	0,053 ± 0,02	0,034 ± 0,01	0,177 ± 0,02	0,071 ± 0,01	0,095 ± 0,01	0,028 ± 0,01
<b>Genbase</b>	18,313 ± 1,05	15,149 ± 0,41	14,488 ± 0,55	94,138 ± 3,48	21,374 ± 1,38	64,551 ± 2,29	2,906 ± 0,13
<b>Medical</b>	5,079 ± 0,34	4,897 ± 0,17	6,350 ± 0,1	50,727 ± 0,56	3,675 ± 0,14	17,374 ± 0,36	0,638 ± 0,03
<b>Scene</b>	30,738 ± 4,3	8,299 ± 0,17	7,350 ± 0,18	44,837 ± 1,8	29,467 ± 0,44	35,690 ± 0,51	3,596 ± 0,26
<b>Yeast</b>	15,408 ± 1,14	13,331 ± 0,23	8,866 ± 0,15	55,251 ± 0,74	24,980 ± 0,25	49,738 ± 0,5	3,860 ± 0,06
<b>Bibtex</b>	603,520 ± 5,63	622,407 ± 4,91	665,048 ± 5,85	4830,659 ± 36,46	1830,896 ± 28,66	3149,844 ± 33,54	412,930 ± 6,17
<b>G1</b>	78,945 ± 4	2,878 ± 0,1	3,441 ± 0,06	23,038 ± 0,24	8,809 ± 0,04	12,312 ± 0,06	1,499 ± 0,02
<b>GP</b>	9,438 ± 2,17	3,572 ± 0,07	4,074 ± 0,04	25,180 ± 0,33	12,410 ± 0,26	14,911 ± 0,27	1,927 ± 0,09
<b>BFRC</b>	612,750 ± 11,27	553,036 ± 12,62	530,381 ± 10,14	1823,570 ± 20,25	734,466 ± 6,25	1093,021 ± 10,97	111,461 ± 1,1

Fonte: Autoria Própria (2022)

Na Tabela 28 apresentam-se os resultados obtidos em relação ao tempo total de execução dos algoritmos comparados ao utilizar o SVM como classificador base. Comparando

o uso do SVM com o C4.5, o primeiro conseguiu reduzir o tempo de execução em 46% dos resultados, o que demonstra ser um bom algoritmo para ser utilizado em conjunto com algoritmos de transformação de problema.

Figura 55: Gráfico de Nemenyi BicbPT $\beta$  x Todos – TT com SVM



Fonte: Autoria Própria (2022)

Ainda, ao analisar o BicbPT $\beta$ , ele apresentou tempo semelhante ao BR e CC e consegue ser 5 vezes mais rápido que o ECC. Para as bases Enron, Medical e Bibtex a diferença alcançada é de até 10 vezes, o que demonstra que o algoritmo proposto consegue manter a característica de baixo tempo de execução dos algoritmos multirrótulo-binário.

Comparando-se com os outros algoritmos, o BicbPT $\beta$  é em média 2 vezes mais rápido que o RAKEL<sub>1</sub> e 3 vezes mais rápido que o RAKEL<sub>2</sub>, sendo que em esta diferença pode chegar a 5 vezes a depender dos dados, e assim como ocorreu com o C4.5, o LP é o algoritmo mais rápido, mas como já mencionado, por conta de sua transformação 1:1.

O p-valor obtido aqui foi de 4.577e-11 e no gráfico de Nemenyi apresentado na Figura 55 é possível observar os resultados descritos anteriormente em relação ao tempo total com SVM.

Por fim, para finalizar a apresentação e análise dos resultados da comparação entre o BicbPT $\beta$  e os demais algoritmos de transformação de problema, são tratados os dados obtidos ao utilizar o Naive Bayes como classificador base para todos os algoritmos e bases.

Na Tabela 29 são exibidos os resultados obtidos com a métrica JI utilizando o NB como classificador base e algo que chama bastante atenção aqui e nas demais métricas que serão apresentadas, é o fato de o NB ter sido o pior classificador base entre os escolhidos para execução dos experimentos, de maneira que enquanto com o SVM o JI médio é de 0,56 e com o C4.5 é de 0,52, com o NB o valor médio em JI é de 0,37.

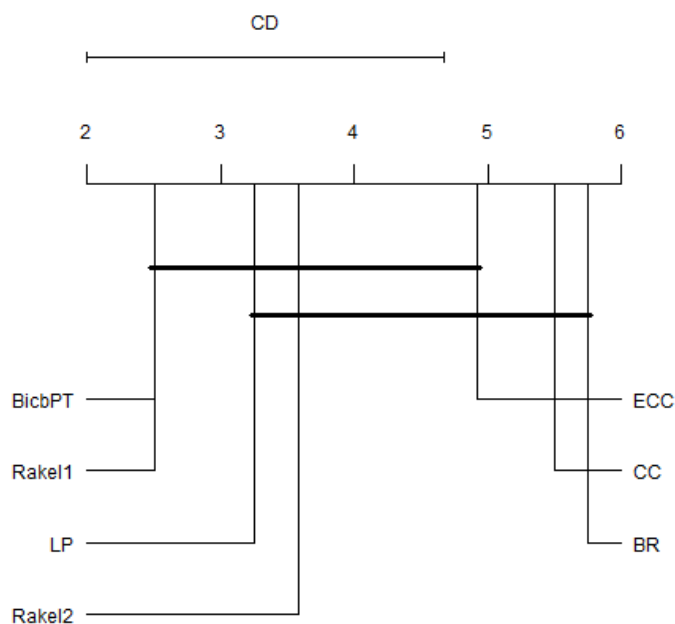
Tabela 29: BicbPT $\beta$  x Todos – JI com NB

Bases	JI $\uparrow$						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,168 $\pm$ 0,01	0,151 $\pm$ 0,01	0,150 $\pm$ 0,01	0,170 $\pm$ 0,02	0,198 $\pm$ 0,02	0,159 $\pm$ 0,01	0,189 $\pm$ 0,02
<b>Emotions</b>	0,553 $\pm$ 0,01	0,522 $\pm$ 0,01	0,528 $\pm$ 0,01	0,527 $\pm$ 0,01	0,515 $\pm$ 0,02	0,513 $\pm$ 0,02	0,508 $\pm$ 0,02
<b>Enron</b>	0,254 $\pm$ 0,01	0,239 $\pm$ 0,01	0,239 $\pm$ 0,01	0,251 $\pm$ 0,01	0,413 $\pm$ 0,01	0,263 $\pm$ 0,01	0,397 $\pm$ 0,01
<b>Flags</b>	0,577 $\pm$ 0,04	0,490 $\pm$ 0,04	0,500 $\pm$ 0,04	0,512 $\pm$ 0,04	0,522 $\pm$ 0,02	0,537 $\pm$ 0,02	0,525 $\pm$ 0,02
<b>Genbase</b>	0,190 $\pm$ 0,06	0,025 $\pm$ 0,06	0,070 $\pm$ 0,1	0,000 $\pm$ 0	0,268 $\pm$ 0,03	0,027 $\pm$ 0,06	0,285 $\pm$ 0,04
<b>Medical</b>	0,271 $\pm$ 0,02	0,238 $\pm$ 0,02	0,244 $\pm$ 0,02	0,143 $\pm$ 0,02	0,193 $\pm$ 0,04	0,234 $\pm$ 0,03	0,387 $\pm$ 0,02
<b>Scene</b>	0,459 $\pm$ 0,01	0,452 $\pm$ 0	0,457 $\pm$ 0	0,458 $\pm$ 0	0,558 $\pm$ 0,01	0,550 $\pm$ 0,01	0,614 $\pm$ 0,02
<b>Yeast</b>	0,493 $\pm$ 0,01	0,421 $\pm$ 0,01	0,420 $\pm$ 0,02	0,427 $\pm$ 0,01	0,472 $\pm$ 0,01	0,448 $\pm$ 0,01	0,460 $\pm$ 0,01
<b>Bibtex</b>	0,210 $\pm$ 0	0,203 $\pm$ 0	0,191 $\pm$ 0	0,196 $\pm$ 0	0,236 $\pm$ 0	0,205 $\pm$ 0	0,191 $\pm$ 0,01
<b>G1</b>	0,452 $\pm$ 0	0,428 $\pm$ 0	0,433 $\pm$ 0,01	0,433 $\pm$ 0	0,434 $\pm$ 0	0,438 $\pm$ 0	0,405 $\pm$ 0,01
<b>GP</b>	0,490 $\pm$ 0,03	0,461 $\pm$ 0,02	0,461 $\pm$ 0,02	0,475 $\pm$ 0,02	0,499 $\pm$ 0,01	0,495 $\pm$ 0,01	0,509 $\pm$ 0,01
<b>BFRC</b>	0,480 $\pm$ 0	0,473 $\pm$ 0	0,471 $\pm$ 0	0,476 $\pm$ 0	0,494 $\pm$ 0	0,487 $\pm$ 0,01	0,476 $\pm$ 0

Fonte: Autoria Própria (2022)

É possível que esse resultado seja por conta da característica de presumir independência do NB, e, mesmo os algoritmos de transformação de problema tratando de alguma forma a relação entre os rótulos, isso não ajuda a minimizar o efeito negativo do NB no resultado da classificação multirrótulo, o que, portanto, sugere-se que este não seja o algoritmo monorrótulo rótulo ideal para utilizar como classificador base ao utilizar essa estratégia de classificação multirrótulo.

Analisando os resultados dos algoritmos multirrótulo-binário, o BicbPT $\beta$  se saiu melhor em 11 de 12 bases e na comparação geral em 4 de 12 bases, sendo bem equilibrado com RAkEL<sub>1</sub> e LP. O p-valor para JI em Friedman com 95% de confiança foi de 4.88e-05, e o teste Nemenyi, Figura 56, demonstra que o BicbPT $\beta$  apresenta diferenças significativas com o BR e CC, e é similar ao ECC, embora, como já mencionado anteriormente, o ECC apresente um tempo de execução bem mais elevado. No geral, o algoritmo proposto possui JI similar aos algoritmos RAkEL<sub>1</sub>, RAkEL<sub>2</sub>, LP e ECC.

Figura 56: Gráfico de Nemenyi BicbPT $\beta$  x Todos – JI com NB

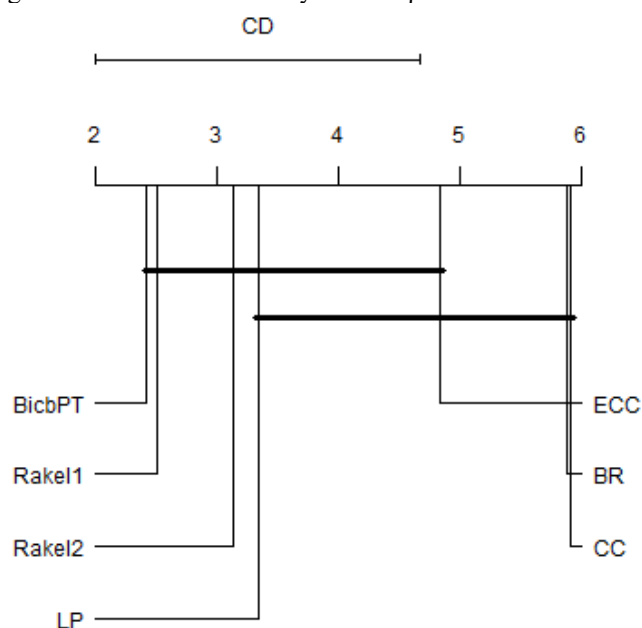
Fonte: Autoria Própria (2022)

Na Tabela 30 são exibidos os resultados obtidos com a métrica HL em conjunto com o NB, onde o algoritmo proposto se saiu melhor em todas as bases ao olhar para os algoritmos multirrótulo-binário e em 5 de 12 ao comparar todos os algoritmos de transformação de problema. O p-valor com 95% de confiança obtido foi de  $6.584e-06$  e no gráfico na Figura 57 nota-se que existe diferença significativa somente entre o BicbPT $\beta$  e os algoritmos BR e CC. O ECC, como já ocorreu com outras métricas, apesar de estar no mesmo grupo do BicbPT $\beta$ , apresenta tempo de execução mais alto que este.

Tabela 30: BicbPT $\beta$  x Todos – HL com NB

Bases	HL ↓						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,237 ± 0,03	0,295 ± 0,03	0,303 ± 0,04	0,247 ± 0,04	0,104 ± 0,01	0,193 ± 0,02	0,080 ± 0
<b>Emotions</b>	0,235 ± 0,01	0,256 ± 0,01	0,251 ± 0,01	0,249 ± 0,01	0,252 ± 0,02	0,247 ± 0,02	0,238 ± 0,01
<b>Enron</b>	0,153 ± 0,02	0,174 ± 0,02	0,182 ± 0,02	0,169 ± 0,02	0,055 ± 0	0,132 ± 0,01	0,074 ± 0
<b>Flags</b>	0,271 ± 0,03	0,334 ± 0,04	0,334 ± 0,03	0,319 ± 0,03	0,327 ± 0,01	0,312 ± 0,01	0,322 ± 0,02
<b>Genbase</b>	0,041 ± 0	0,045 ± 0	0,044 ± 0	0,046 ± 0	0,036 ± 0	0,045 ± 0	0,062 ± 0
<b>Medical</b>	0,024 ± 0	0,025 ± 0	0,024 ± 0	0,025 ± 0	0,024 ± 0	0,025 ± 0	0,032 ± 0
<b>Scene</b>	0,232 ± 0	0,240 ± 0	0,236 ± 0	0,234 ± 0	0,157 ± 0,01	0,158 ± 0	0,135 ± 0,01
<b>Yeast</b>	0,260 ± 0,01	0,297 ± 0,01	0,299 ± 0,01	0,293 ± 0,01	0,245 ± 0,01	0,268 ± 0	0,263 ± 0,01
<b>Bibtex</b>	0,057 ± 0	0,060 ± 0	0,098 ± 0	0,078 ± 0	0,015 ± 0	0,055 ± 0	0,025 ± 0
<b>G1</b>	0,259 ± 0	0,274 ± 0	0,275 ± 0	0,271 ± 0	0,272 ± 0	0,268 ± 0	0,313 ± 0,01
<b>GP</b>	0,250 ± 0,02	0,296 ± 0,02	0,297 ± 0,02	0,277 ± 0,02	0,258 ± 0,01	0,257 ± 0,01	0,253 ± 0,01
<b>BFRC</b>	0,365 ± 0	0,369 ± 0	0,373 ± 0	0,366 ± 0	0,351 ± 0	0,355 ± 0	0,355 ± 0

Fonte: Aatoria Própria (2022)

Figura 57: Gráfico de Nemenyi BicbPT $\beta$  x Todos – HL com NB

Fonte: Aatoria Própria (2022)

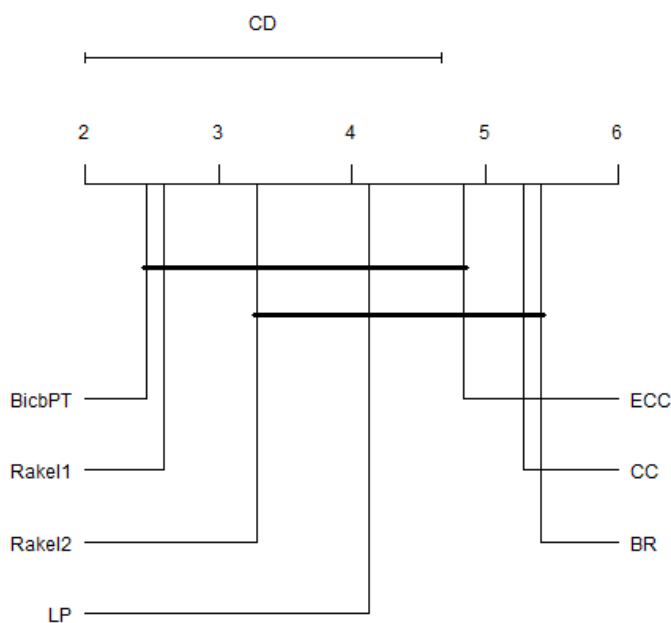
Na Tabela 31 apresentam-se os resultados obtidos com a métrica  $F1_{\text{micro}}$ , onde observa-se um equilíbrio entre BicbPT $\beta$  e RAkEL<sub>1</sub>, sendo que cada um conseguiu se sair melhor em 6 das 12 bases utilizadas nos experimentos. Este é um comportamento interessante, visto que os dois algoritmos estão em categorias diferentes de algoritmos de transformação de problema, sendo o algoritmo proposto um multirrótulo-binário e o RAkEL<sub>1</sub> um multirrótulo-multiclasse.

Tabela 31: BicbPT $\beta$  x Todos – F1<sub>micro</sub> com NB

Bases	F1 <sub>micro</sub> ↑						
	BicbPT $\beta$	BR	CC	ECC	RAkEL <sub>1</sub>	RAkEL <sub>2</sub>	LP
<b>Birds</b>	0,211 ± 0,02	0,169 ± 0,02	0,166 ± 0,02	0,189 ± 0,03	0,306 ± 0,03	0,217 ± 0,01	0,252 ± 0,03
<b>Emotions</b>	0,678 ± 0,01	0,650 ± 0,01	0,653 ± 0,01	0,650 ± 0,01	0,638 ± 0,01	0,634 ± 0,01	0,616 ± 0,02
<b>Enron</b>	0,328 ± 0,02	0,307 ± 0,02	0,297 ± 0,02	0,308 ± 0,02	0,535 ± 0,01	0,360 ± 0,01	0,444 ± 0,02
<b>Flags</b>	0,734 ± 0,03	0,666 ± 0,04	0,667 ± 0,03	0,679 ± 0,03	0,683 ± 0,02	0,694 ± 0,02	0,668 ± 0,02
<b>Genbase</b>	0,256 ± 0,08	0,036 ± 0,08	0,095 ± 0,14	0,000 ± 0	0,353 ± 0,04	0,038 ± 0,08	0,256 ± 0,04
<b>Medical</b>	0,418 ± 0,02	0,380 ± 0,03	0,388 ± 0,02	0,263 ± 0,02	0,309 ± 0,05	0,377 ± 0,03	0,393 ± 0,03
<b>Scene</b>	0,547 ± 0,01	0,556 ± 0	0,560 ± 0	0,561 ± 0	0,638 ± 0,01	0,636 ± 0,01	0,634 ± 0,02
<b>Yeast</b>	0,621 ± 0,01	0,550 ± 0,01	0,544 ± 0,01	0,552 ± 0,01	0,604 ± 0,01	0,584 ± 0,01	0,571 ± 0,01
<b>Bibtex</b>	0,221 ± 0,01	0,214 ± 0,01	0,152 ± 0,01	0,151 ± 0,01	0,299 ± 0	0,228 ± 0	0,150 ± 0,01
<b>G1</b>	0,576 ± 0	0,553 ± 0	0,556 ± 0	0,557 ± 0	0,552 ± 0	0,557 ± 0	0,488 ± 0,01
<b>GP</b>	0,618 ± 0,04	0,581 ± 0,02	0,581 ± 0,02	0,591 ± 0,02	0,611 ± 0,02	0,606 ± 0,02	0,617 ± 0,01
<b>BFRC</b>	0,633 ± 0	0,629 ± 0	0,626 ± 0	0,631 ± 0	0,650 ± 0	0,644 ± 0	0,627 ± 0

Fonte: Autoria Própria (2022)

Ao analisar a Figura 58, observa-se que não existem diferenças significativas entre os dois algoritmos, e, inclusive, se apresentam bem próximos no ranking criado pelo teste de Nemenyi com confiança de 95%.

Figura 58: Gráfico de Nemenyi BicbPT $\beta$  x Todos – F1<sub>micro</sub> com NB

Fonte: Autoria Própria (2022)

Ao olhar para os algoritmos multirrótulo-binário, o BicbPT $\beta$  se saiu melhor em 11 das 12 bases, sendo superior sempre ao BR e ao CC e não apresentando diferenças significativas com o ECC. O p-valor obtido com 95% de confiança foi de 0.0005262.

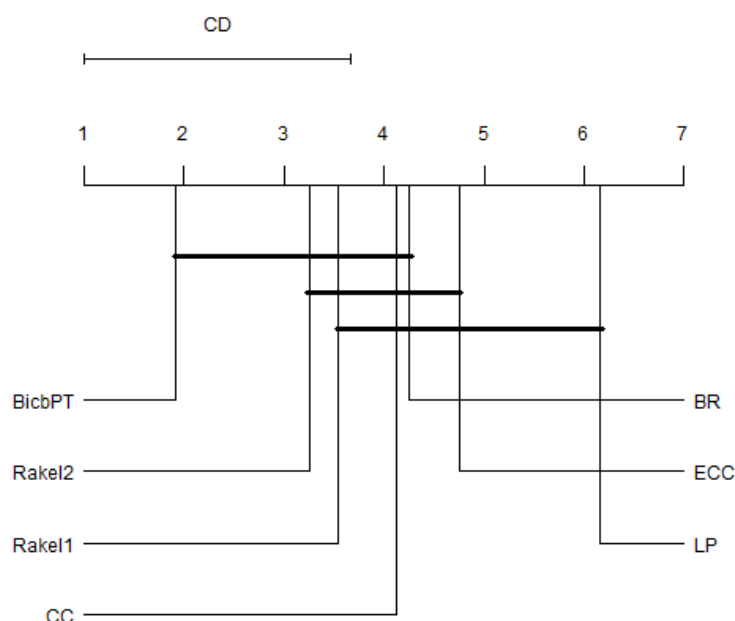
Na Tabela 32 apresenta-se os resultados da métrica  $F1_{macro}$ , sendo o BicbPT $\beta$  melhor em 11 de 12 bases nos algoritmos de transformação multirrótulo-binário e na comparação geral, em 6 de 12 bases.

Tabela 32: BicbPT $\beta$  x Todos –  $F1_{macro}$  com NB

Bases	$F1_{macro}$ $\uparrow$						
	BicbPT $\beta$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	0,206 $\pm$ 0,02	0,160 $\pm$ 0,02	0,156 $\pm$ 0,02	0,162 $\pm$ 0,03	0,223 $\pm$ 0,03	0,195 $\pm$ 0,02	0,144 $\pm$ 0,02
<b>Emotions</b>	0,675 $\pm$ 0,01	0,645 $\pm$ 0,01	0,647 $\pm$ 0,01	0,641 $\pm$ 0,01	0,634 $\pm$ 0,01	0,627 $\pm$ 0,02	0,603 $\pm$ 0,02
<b>Enron</b>	0,180 $\pm$ 0,01	0,164 $\pm$ 0,01	0,162 $\pm$ 0,01	0,156 $\pm$ 0,01	0,197 $\pm$ 0,02	0,177 $\pm$ 0,01	0,148 $\pm$ 0,01
<b>Flags</b>	0,668 $\pm$ 0,04	0,619 $\pm$ 0,03	0,620 $\pm$ 0,03	0,628 $\pm$ 0,03	0,630 $\pm$ 0,02	0,638 $\pm$ 0,02	0,590 $\pm$ 0,02
<b>Genbase</b>	0,035 $\pm$ 0,01	0,006 $\pm$ 0,01	0,012 $\pm$ 0,02	0,000 $\pm$ 0	0,044 $\pm$ 0,01	0,006 $\pm$ 0,01	0,030 $\pm$ 0,01
<b>Medical</b>	0,060 $\pm$ 0	0,051 $\pm$ 0	0,052 $\pm$ 0	0,029 $\pm$ 0	0,040 $\pm$ 0,01	0,051 $\pm$ 0	0,049 $\pm$ 0,01
<b>Scene</b>	0,551 $\pm$ 0,01	0,574 $\pm$ 0,01	0,578 $\pm$ 0	0,579 $\pm$ 0,01	0,656 $\pm$ 0,01	0,659 $\pm$ 0,01	0,642 $\pm$ 0,02
<b>Yeast</b>	0,485 $\pm$ 0,01	0,450 $\pm$ 0,01	0,447 $\pm$ 0,01	0,448 $\pm$ 0,01	0,447 $\pm$ 0,01	0,449 $\pm$ 0,01	0,424 $\pm$ 0,01
<b>Bibtex</b>	0,175 $\pm$ 0	0,170 $\pm$ 0	0,138 $\pm$ 0	0,118 $\pm$ 0	0,094 $\pm$ 0	0,179 $\pm$ 0	0,055 $\pm$ 0
<b>G1</b>	0,544 $\pm$ 0	0,520 $\pm$ 0,01	0,521 $\pm$ 0,01	0,523 $\pm$ 0,01	0,519 $\pm$ 0	0,525 $\pm$ 0,01	0,455 $\pm$ 0,01
<b>GP</b>	0,437 $\pm$ 0,03	0,434 $\pm$ 0,01	0,435 $\pm$ 0,02	0,413 $\pm$ 0,02	0,402 $\pm$ 0,02	0,398 $\pm$ 0,02	0,371 $\pm$ 0,01
<b>BFRC</b>	0,591 $\pm$ 0	0,587 $\pm$ 0	0,587 $\pm$ 0	0,587 $\pm$ 0	0,592 $\pm$ 0	0,589 $\pm$ 0	0,542 $\pm$ 0,01

Fonte: Autoria Própria (2022)

Figura 59: Gráfico de Nemenyi BicbPT $\beta$  x Todos –  $F1_{macro}$  com NB



Fonte: Autoria Própria (2022)



O p-valor com 95% de confiança obtido para a métrica  $F1_{macro}$  foi de 0.0001358 no teste de Friedman, indicando que existiam diferenças significativas entre os algoritmos comparados, justificando assim o uso do teste de Nemenyi, Figura 59, posteriormente, o qual chama atenção aqui pelo fato de não ter sido encontrado diferença significativa entre o algoritmo proposto e os algoritmos BR e CC, havendo somente diferença com o ECC, contrastando com os resultados das demais métricas analisadas até este momento.

Tabela 33: BicbPT $\beta$  x Todos – TT com NB

Bases	TT ↓						
	BicbPT $\beta$	BR	CC	ECC	RAKEL <sub>1</sub>	RAKEL <sub>2</sub>	LP
<b>Birds</b>	1,765 ± 0,31	1,340 ± 0,18	1,344 ± 0,18	10,459 ± 0,25	5,980 ± 0,26	5,175 ± 0,28	2,257 ± 0,27
<b>Emotions</b>	1,281 ± 0,07	0,106 ± 0,01	0,162 ± 0,03	0,901 ± 0,09	0,503 ± 0,07	0,579 ± 0,07	0,183 ± 0,01
<b>Enron</b>	51,442 ± 0,27	50,635 ± 0,34	53,221 ± 0,2	459,596 ± 2,45	444,502 ± 11,28	191,561 ± 2,54	114,618 ± 3,54
<b>Flags</b>	0,500 ± 0,31	0,008 ± 0,01	0,034 ± 0,01	0,102 ± 0,01	0,051 ± 0,01	0,059 ± 0,01	0,022 ± 0,01
<b>Genbase</b>	3,891 ± 0,32	2,785 ± 0,05	2,818 ± 0,08	24,920 ± 0,52	4,482 ± 0,34	7,891 ± 0,3	1,125 ± 0,06
<b>Medical</b>	9,016 ± 0,58	9,242 ± 0,25	11,100 ± 0,11	102,751 ± 0,41	17,808 ± 0,87	31,140 ± 0,51	4,297 ± 0,21
<b>Scene</b>	6,610 ± 0,58	1,308 ± 0,03	1,844 ± 0,01	12,843 ± 0,08	5,466 ± 0,1	6,410 ± 0,14	1,600 ± 0,07
<b>Yeast</b>	3,735 ± 0,41	1,149 ± 0,06	1,364 ± 0,02	10,839 ± 0,11	12,658 ± 0,27	6,396 ± 0,06	4,246 ± 0,14
<b>Bibtex</b>	421,169 ± 5,76	382,127 ± 1,57	423,466 ± 5,95	3907,149 ± 6,82	3974,958 ± 43,96	1396,591 ± 19,61	959,336 ± 17,3
<b>G1</b>	156,095 ± 7,91	8,757 ± 1,83	7,917 ± 0,34	64,160 ± 2,04	27,698 ± 0,5	38,509 ± 0,75	19,403 ± 0,69
<b>GP</b>	5,922 ± 1,28	2,408 ± 0,12	2,680 ± 0,04	22,084 ± 0,13	10,610 ± 0,09	12,651 ± 0,13	4,677 ± 0,18
<b>BFRC</b>	53,724 ± 4,03	37,108 ± 0,33	39,374 ± 0,28	324,720 ± 1,7	205,391 ± 2,38	188,515 ± 3,43	214,247 ± 3,82

Fonte: Autoria Própria (2022)

Em relação a métrica Tempo Total, Tabela 33, utilizando o NB como classificador base é possível perceber que o uso deste algoritmo monorrótulo fez com que o tempo total de execução fosse o mais rápido entre os algoritmos utilizados, SVM, C4.5 e NB.

Sendo que a diferença média do tempo de execução para o SVM foi de 129% e com o C4.5 de 207%, o que indica que, talvez, explorar o NB para conseguir bons resultados nas métricas JI, HL e F1 seria interessante, visto a sua capacidade de diminuir o tempo de execução na classificação multirrótulo por transformação de problema.

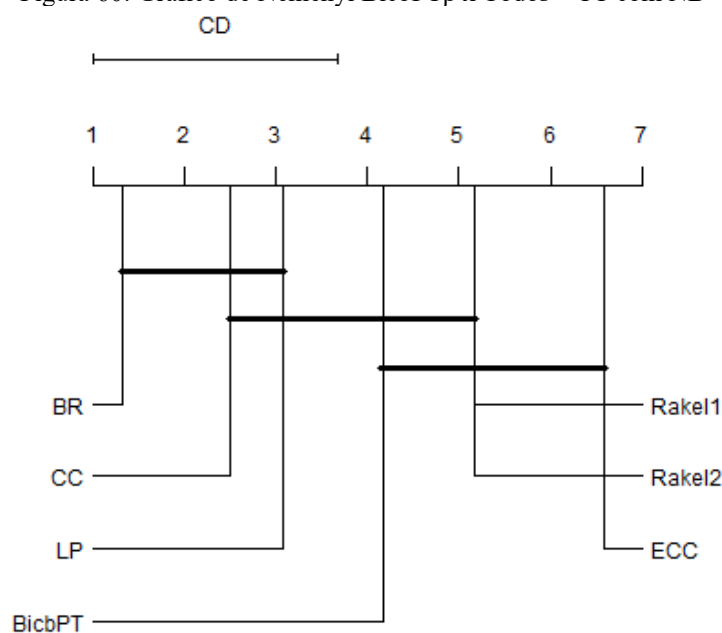
Ainda, ao analisar os tempos do BicbPT $\beta$  com os outros algoritmos, este se apresenta semelhante ao BR e CC e em média 7 vezes mais rápido que o ECC, podendo chegar a 11 vezes a depender dos dados utilizados.

Já em relação aos algoritmos multirrótulo não-binários, o algoritmo proposto é 7 vezes mais rápido que o RAKEL<sub>1</sub> e 3 vezes mais rápido que o RAKEL<sub>2</sub>, podendo ser até 9 vezes mais rápido, como, por exemplo, nas bases Enron e Bibtex.

Chama atenção, também, o fato de o BicbPT $\beta$  ser, em média, 2 vezes mais rápido que o LP ao utilizar o NB como classificador base, o que contrasta com os resultados obtidos com os algoritmos C4.5 e SVM.

O p-valor obtido aqui foi de 3.794e-09 com 95% de confiança no teste de Friedman, e na Figura 60, no teste de Nemenyi observa-se diferença significativa no tempo total somente em relação aos algoritmos BR e ECC.

Figura 60: Gráfico de Nemenyi BicbPT $\beta$  x Todos – TT com NB



Fonte: Autoria Própria (2022)

#### 6.4 EXPERIMENTO III: BicbPT $\alpha$ x BicbPT $\beta$

Nesta seção é apresentado o comparativo entre as duas versões do algoritmo proposto, BicbPT $\alpha$  e BicbPT $\beta$ , o qual tem seus resultados apresentados na Tabela 34, sendo utilizado o C4.5 como classificador base. O p-valor com confiança de 95% para JI, HL, F1-micro, F1-macro e TT, respectivamente 0.0004883, 0.00384, 0.0004883, 0.004883 e 0.01221. Com

exceção do tempo total de execução, a versão  $\beta$  se saiu melhor em todas as outras métricas para o classificador base escolhido.

Tabela 34: Resultados BicbPT $\alpha$  x BicbPT $\beta$  com C4.5

Bases	JI $\uparrow$		HL $\downarrow$		F1-micro $\uparrow$		F1-macro $\uparrow$		TT $\downarrow$	
	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$
<b>Birds</b>	0,548 $\pm$ 0,03	0,597 $\pm$ 0,03	0,054 $\pm$ 0	0,050 $\pm$ 0	0,378 $\pm$ 0,04	0,443 $\pm$ 0,04	0,249 $\pm$ 0,03	0,299 $\pm$ 0,03	1,691 $\pm$ 0,41	1,610 $\pm$ 0,29
<b>Emotions</b>	0,465 $\pm$ 0,01	0,562 $\pm$ 0,02	0,265 $\pm$ 0,01	0,216 $\pm$ 0,01	0,604 $\pm$ 0,01	0,671 $\pm$ 0,02	0,592 $\pm$ 0,01	0,659 $\pm$ 0,02	1,034 $\pm$ 0,03	1,704 $\pm$ 0,09
<b>Enron</b>	0,396 $\pm$ 0,01	0,435 $\pm$ 0,01	0,054 $\pm$ 0	0,051 $\pm$ 0	0,524 $\pm$ 0,01	0,563 $\pm$ 0,01	0,144 $\pm$ 0,01	0,161 $\pm$ 0,01	91,277 $\pm$ 1,73	93,213 $\pm$ 11,12
<b>Flags</b>	0,569 $\pm$ 0,02	0,631 $\pm$ 0,02	0,271 $\pm$ 0,01	0,222 $\pm$ 0,01	0,716 $\pm$ 0,02	0,763 $\pm$ 0,02	0,592 $\pm$ 0,03	0,634 $\pm$ 0,03	0,333 $\pm$ 0,03	0,484 $\pm$ 0,05
<b>Genbase</b>	0,976 $\pm$ 0,01	0,985 $\pm$ 0,01	0,002 $\pm$ 0	0,001 $\pm$ 0	0,979 $\pm$ 0,01	0,987 $\pm$ 0,01	0,713 $\pm$ 0,04	0,766 $\pm$ 0,04	1,686 $\pm$ 0,04	2,422 $\pm$ 0,17
<b>Medical</b>	0,720 $\pm$ 0,03	0,767 $\pm$ 0,03	0,012 $\pm$ 0	0,010 $\pm$ 0	0,788 $\pm$ 0,02	0,818 $\pm$ 0,02	0,353 $\pm$ 0,03	0,400 $\pm$ 0,03	6,177 $\pm$ 0,34	5,516 $\pm$ 0,25
<b>Scene</b>	0,515 $\pm$ 0,01	0,628 $\pm$ 0,02	0,143 $\pm$ 0	0,128 $\pm$ 0,01	0,602 $\pm$ 0,01	0,674 $\pm$ 0,01	0,614 $\pm$ 0,01	0,690 $\pm$ 0,02	6,779 $\pm$ 0,28	12,111 $\pm$ 1,64
<b>Yeast</b>	0,419 $\pm$ 0,02	0,476 $\pm$ 0,01	0,255 $\pm$ 0,01	0,227 $\pm$ 0,01	0,566 $\pm$ 0,01	0,623 $\pm$ 0,01	0,380 $\pm$ 0,01	0,410 $\pm$ 0,01	5,978 $\pm$ 0,48	5,610 $\pm$ 0,19
<b>Bibtex</b>	0,287 $\pm$ 0,01	0,297 $\pm$ 0,01	0,015 $\pm$ 0	0,015 $\pm$ 0	0,389 $\pm$ 0,01	0,398 $\pm$ 0,01	0,260 $\pm$ 0	0,267 $\pm$ 0	1183,59 9 $\pm$ 19,74	1186,33 1 $\pm$ 15,12
<b>G1</b>	0,312 $\pm$ 0,01	0,448 $\pm$ 0,01	0,273 $\pm$ 0,01	0,240 $\pm$ 0	0,431 $\pm$ 0,01	0,551 $\pm$ 0,01	0,361 $\pm$ 0,01	0,458 $\pm$ 0,01	137,578 $\pm$ 3,77	700,238 $\pm$ 46,72
<b>GP</b>	0,413 $\pm$ 0,02	0,433 $\pm$ 0,02	0,329 $\pm$ 0,01	0,294 $\pm$ 0,01	0,551 $\pm$ 0,01	0,569 $\pm$ 0,03	0,399 $\pm$ 0,02	0,365 $\pm$ 0,04	10,731 $\pm$ 0,41	20,986 $\pm$ 6,42
<b>BFRC</b>	0,450 $\pm$ 0	0,455 $\pm$ 0	0,382 $\pm$ 0	0,380 $\pm$ 0	0,613 $\pm$ 0	0,618 $\pm$ 0	0,532 $\pm$ 0	0,536 $\pm$ 0	567,366 $\pm$ 6,68	613,837 $\pm$ 10,4

Fonte: Autoria Própria (2022)

Na Tabela 35 são exibidos os resultados da comparação entre as duas versões, utilizando o SVM como classificador base. O valor de p-valor com confiança de 95% para cada métrica é 0.002516, 0.005857, 0.002507, 0.001465 e 0.2334. Assim como, ao utilizar o C4.5 como classificador base, com o SVM a versão  $\beta$  também se saiu melhor. Chama a atenção neste caso somente o tempo total, que obteve um p-valor de 0.2334, indicando que não existe diferença significativa entre ambas as versões para esta métrica.

Tabela 35: Resultados BicbPT $\alpha$  x BicbPT $\beta$  com SVM

Bases	JI $\uparrow$		HL $\downarrow$		F1-micro $\uparrow$		F1-macro $\uparrow$		TT $\downarrow$	
	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$
<b>Birds</b>	0,603 $\pm$ 0,01	0,626 $\pm$ 0,01	0,052 $\pm$ 0	0,049 $\pm$ 0	0,456 $\pm$ 0,02	0,497 $\pm$ 0,02	0,349 $\pm$ 0,03	0,411 $\pm$ 0,03	1,790 $\pm$ 0,07	1,969 $\pm$ 0,09
<b>Emotions</b>	0,498 $\pm$ 0,01	0,589 $\pm$ 0,01	0,203 $\pm$ 0	0,180 $\pm$ 0	0,638 $\pm$ 0,01	0,705 $\pm$ 0,01	0,600 $\pm$ 0,01	0,686 $\pm$ 0,02	1,385 $\pm$ 0,05	2,812 $\pm$ 0,65
<b>Enron</b>	0,388 $\pm$ 0,01	0,414 $\pm$ 0,01	0,060 $\pm$ 0	0,057 $\pm$ 0	0,505 $\pm$ 0,01	0,534 $\pm$ 0,01	0,201 $\pm$ 0,02	0,233 $\pm$ 0,02	23,697 $\pm$ 0,57	24,159 $\pm$ 0,76
<b>Flags</b>	0,587 $\pm$ 0,01	0,641 $\pm$ 0,01	0,265 $\pm$ 0,01	0,219 $\pm$ 0,01	0,727 $\pm$ 0,01	0,771 $\pm$ 0,01	0,631 $\pm$ 0,03	0,685 $\pm$ 0,04	0,657 $\pm$ 0,19	0,969 $\pm$ 0,56
<b>Genbase</b>	0,982 $\pm$ 0,01	0,991 $\pm$ 0,01	0,001 $\pm$ 0	0,001 $\pm$ 0	0,985 $\pm$ 0,01	0,991 $\pm$ 0,01	0,756 $\pm$ 0,04	0,813 $\pm$ 0,04	17,255 $\pm$ 0,5	18,313 $\pm$ 1,05
<b>Medical</b>	0,730 $\pm$ 0,01	0,754 $\pm$ 0,01	0,011 $\pm$ 0	0,010 $\pm$ 0	0,792 $\pm$ 0,01	0,814 $\pm$ 0,01	0,363 $\pm$ 0,02	0,403 $\pm$ 0,02	5,664 $\pm$ 0,15	5,079 $\pm$ 0,34
<b>Scene</b>	0,587 $\pm$ 0,01	0,676 $\pm$ 0,02	0,111 $\pm$ 0	0,102 $\pm$ 0	0,673 $\pm$ 0,01	0,721 $\pm$ 0,01	0,680 $\pm$ 0,01	0,729 $\pm$ 0,01	14,781 $\pm$ 0,92	30,738 $\pm$ 4,3
<b>Yeast</b>	0,498 $\pm$ 0,01	0,507 $\pm$ 0,01	0,201 $\pm$ 0	0,196 $\pm$ 0	0,632 $\pm$ 0,01	0,639 $\pm$ 0,01	0,327 $\pm$ 0,01	0,340 $\pm$ 0,01	17,847 $\pm$ 0,87	15,408 $\pm$ 1,14
<b>Bibtex</b>	0,313 $\pm$ 0,01	0,326 $\pm$ 0,01	0,015 $\pm$ 0	0,015 $\pm$ 0	0,410 $\pm$ 0,01	0,418 $\pm$ 0,01	0,299 $\pm$ 0,01	0,310 $\pm$ 0,01	598,755 $\pm$ 5,6	603,520 $\pm$ 5,63
<b>G1</b>	0,372 $\pm$ 0	0,404 $\pm$ 0	0,290 $\pm$ 0	0,270 $\pm$ 0	0,496 $\pm$ 0	0,523 $\pm$ 0	0,458 $\pm$ 0,01	0,483 $\pm$ 0,01	13,997 $\pm$ 0,23	78,945 $\pm$ 4
<b>GP</b>	0,445 $\pm$ 0,02	0,475 $\pm$ 0,02	0,294 $\pm$ 0,01	0,264 $\pm$ 0,01	0,575 $\pm$ 0,02	0,602 $\pm$ 0,03	0,388 $\pm$ 0,02	0,381 $\pm$ 0,05	8,159 $\pm$ 0,71	9,438 $\pm$ 2,17
<b>BFRC</b>	0,469 $\pm$ 0	0,475 $\pm$ 0	0,359 $\pm$ 0	0,354 $\pm$ 0	0,629 $\pm$ 0	0,635 $\pm$ 0	0,553 $\pm$ 0	0,558 $\pm$ 0	644,639 $\pm$ 15,2	612,750 $\pm$ 11,27

Fonte: Autoria Própria (2022)

Observando os resultados na Tabela 34 e Tabela 35, é possível verificar que esse resultado foi alcançado pela redução do tempo total nas bases bibtex, gp e bfrc ao utilizar o SVM e pela forma que a versão  $\beta$  gera os subproblemas, ao considerar como os rótulos se influenciam.

Na Tabela 36 são apresentados os resultados da comparação entre as duas versões ao utilizar o algoritmo Naive Bayes como classificador base. O p-valor obtido para JI, HL, F1-micro, F1-macro e TT foi, respectivamente 0.002516, 0.0004883, 0.002441, 0.009277, 0.06396.

Aqui a versão  $\beta$  também apresentou melhores resultados, mas dois pontos chamam a atenção, um em relação ao tempo total e outro em relação aos algoritmos base escolhidos. Assim como com o SVM, o p-valor do tempo total, 0.06396, também indicou não haver diferença significativa entre as duas versões e, ainda, este foi o mais rápido entre os classificadores base utilizados.

Tabela 36: Resultados BicbPT $\alpha$  x BicbPT $\beta$  com Naive Bayes

Bases	JI $\uparrow$		HL $\downarrow$		F1-micro $\uparrow$		F1-macro $\uparrow$		TT $\downarrow$	
	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$	$\alpha$	$\beta$
<b>Birds</b>	0,151 $\pm$ 0,01	0,168 $\pm$ 0,01	0,295 $\pm$ 0,03	0,237 $\pm$ 0,03	0,169 $\pm$ 0,02	0,211 $\pm$ 0,02	0,160 $\pm$ 0,02	0,206 $\pm$ 0,02	1,999 $\pm$ 0,35	1,765 $\pm$ 0,31
<b>Emotions</b>	0,523 $\pm$ 0,02	0,553 $\pm$ 0,01	0,258 $\pm$ 0,02	0,235 $\pm$ 0,01	0,651 $\pm$ 0,02	0,678 $\pm$ 0,01	0,645 $\pm$ 0,01	0,675 $\pm$ 0,01	1,078 $\pm$ 0,03	1,281 $\pm$ 0,07
<b>Enron</b>	0,239 $\pm$ 0,01	0,254 $\pm$ 0,01	0,174 $\pm$ 0,02	0,153 $\pm$ 0,02	0,307 $\pm$ 0,02	0,328 $\pm$ 0,02	0,164 $\pm$ 0,01	0,180 $\pm$ 0,01	50,686 $\pm$ 0,33	51,442 $\pm$ 0,27
<b>Flags</b>	0,517 $\pm$ 0,03	0,577 $\pm$ 0,04	0,318 $\pm$ 0,02	0,271 $\pm$ 0,03	0,688 $\pm$ 0,02	0,734 $\pm$ 0,03	0,610 $\pm$ 0,03	0,668 $\pm$ 0,04	0,554 $\pm$ 0,19	0,500 $\pm$ 0,31
<b>Genbase</b>	0,025 $\pm$ 0,06	0,190 $\pm$ 0,06	0,045 $\pm$ 0	0,041 $\pm$ 0	0,036 $\pm$ 0,08	0,256 $\pm$ 0,08	0,006 $\pm$ 0,01	0,035 $\pm$ 0,01	3,092 $\pm$ 0,11	3,891 $\pm$ 0,32
<b>Medical</b>	0,238 $\pm$ 0,02	0,271 $\pm$ 0,02	0,025 $\pm$ 0	0,024 $\pm$ 0	0,379 $\pm$ 0,03	0,418 $\pm$ 0,02	0,051 $\pm$ 0	0,060 $\pm$ 0	9,908 $\pm$ 0,46	9,016 $\pm$ 0,58
<b>Scene</b>	0,452 $\pm$ 0	0,459 $\pm$ 0,01	0,242 $\pm$ 0	0,232 $\pm$ 0	0,555 $\pm$ 0	0,547 $\pm$ 0,01	0,573 $\pm$ 0	0,551 $\pm$ 0,01	5,441 $\pm$ 0,21	6,610 $\pm$ 0,58
<b>Yeast</b>	0,421 $\pm$ 0,01	0,493 $\pm$ 0,01	0,297 $\pm$ 0,01	0,260 $\pm$ 0,01	0,550 $\pm$ 0,01	0,621 $\pm$ 0,01	0,450 $\pm$ 0,01	0,485 $\pm$ 0,01	3,732 $\pm$ 0,04	3,735 $\pm$ 0,41
<b>Bibtex</b>	0,203 $\pm$ 0	0,210 $\pm$ 0	0,060 $\pm$ 0	0,057 $\pm$ 0	0,214 $\pm$ 0,01	0,221 $\pm$ 0,01	0,170 $\pm$ 0	0,175 $\pm$ 0	372,778 $\pm$ 3,05	421,169 $\pm$ 5,76
<b>G1</b>	0,431 $\pm$ 0,01	0,452 $\pm$ 0	0,278 $\pm$ 0,01	0,259 $\pm$ 0	0,557 $\pm$ 0,01	0,576 $\pm$ 0	0,523 $\pm$ 0,01	0,544 $\pm$ 0	59,247 $\pm$ 2,1	156,095 $\pm$ 7,91
<b>GP</b>	0,463 $\pm$ 0,02	0,490 $\pm$ 0,03	0,295 $\pm$ 0,02	0,250 $\pm$ 0,02	0,584 $\pm$ 0,02	0,618 $\pm$ 0,04	0,434 $\pm$ 0,02	0,437 $\pm$ 0,03	5,696 $\pm$ 0,52	5,922 $\pm$ 1,28
<b>BFRC</b>	0,473 $\pm$ 0	0,480 $\pm$ 0	0,369 $\pm$ 0	0,365 $\pm$ 0	0,629 $\pm$ 0	0,633 $\pm$ 0	0,587 $\pm$ 0	0,591 $\pm$ 0	51,192 $\pm$ 2,72	53,724 $\pm$ 4,03

Fonte: Autoria Própria (2022)

Em relação aos algoritmos base, de forma geral, embora o NB tenha apresentado um tempo de execução baixo, seu desempenho nas outras métricas é muito pior que o C4.5 e o SVM, o que pode ter ocorrido pelo fato deste tratar as variáveis do problema de forma independente, sendo este um cenário possível na classificação multirrotulo, mas não o mais comum.

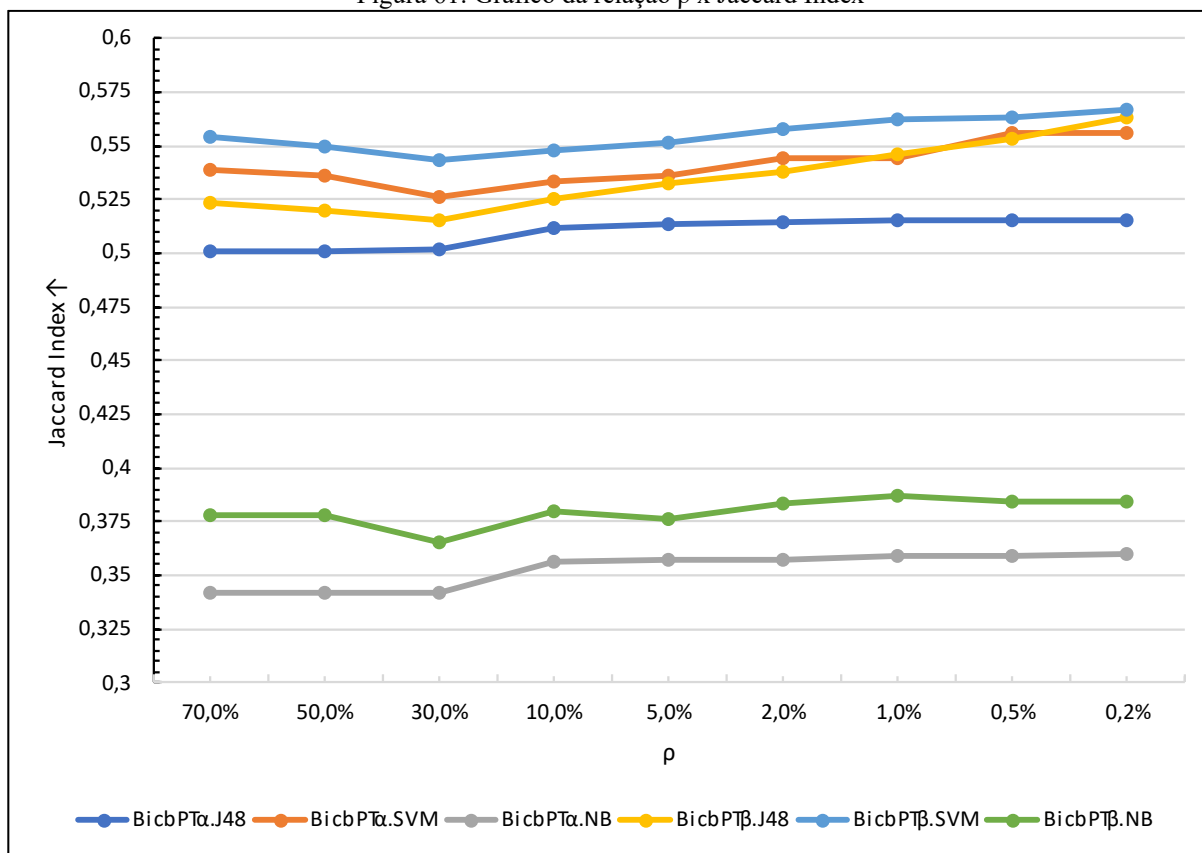
## 6.5 EXPERIMENTO IV: ANÁLISE DE SENSIBILIDADE DO HIPER PARÂMETRO $\rho$

Este experimento foi realizado com o objetivo de verificar o impacto do parâmetro  $\rho$ , percentual mínimo de exemplos por bicluster, em relação as métricas escolhidas e também para sugerir um valor ou intervalo padrão para ele ao utilizar o BicbPT $\alpha$  ou  $\beta$ .

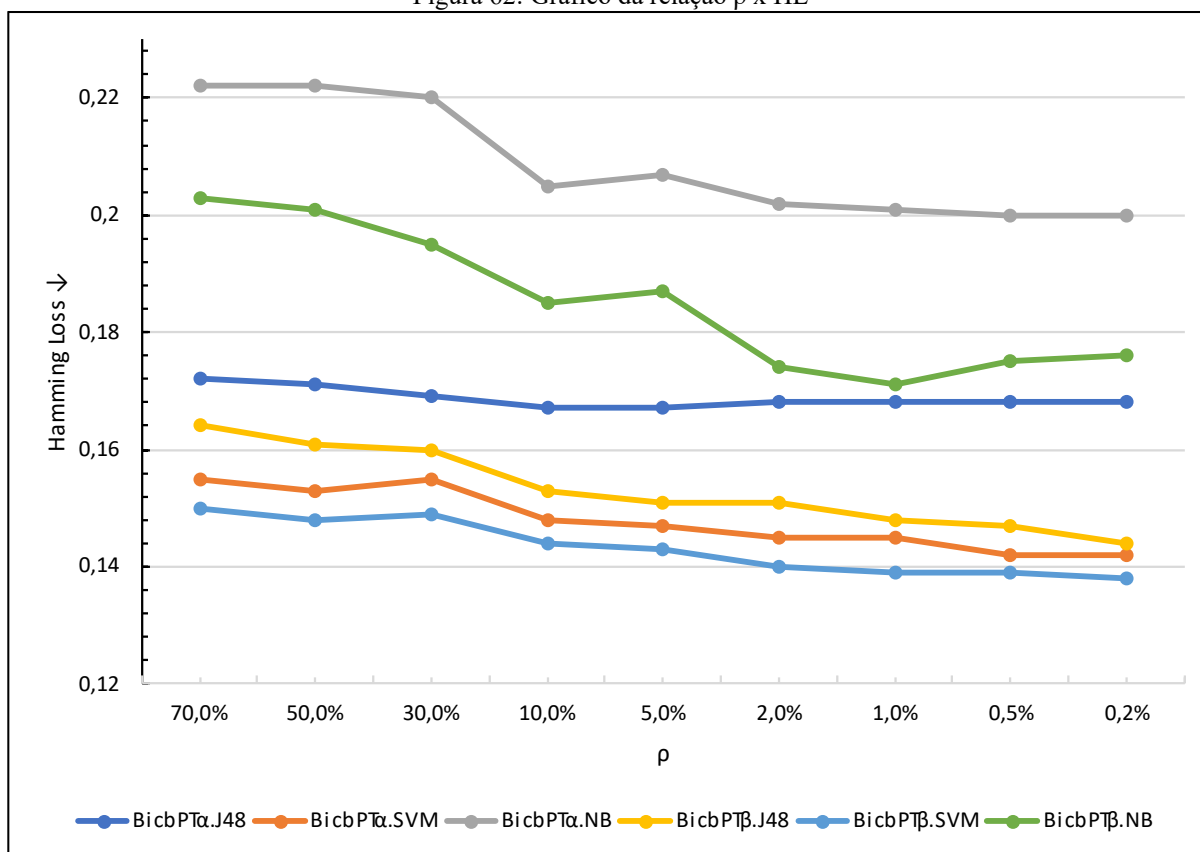
Para este teste foram considerados os valores para  $\rho$  de 70%, 50%, 30%, 10%, 5%, 2%, 1%, 0,5% e 0,2% para cada base de dados com cada classificador base, e após, foi calculada uma média aritmética com os valores obtidos.

É possível notar na Figura 61 que quanto menor o valor de  $\rho$ , maior é valor alcançado para a métrica Jaccard Index, isto porque, quando  $\rho$  é pequeno a quantidade de subproblemas aumenta, permitindo cobrir mais relacionamentos entre rótulos e, como consequência, obtendo melhores resultados. Também, nota-se que o J48 como classificador base no BicbPT $\beta$  possui comportamento semelhante ao SVM, diferentemente do BicbPT $\alpha$ , onde o J48 tem um resultado mais linear, sem muita variação.

Figura 61: Gráfico da relação  $\rho$  x Jaccard Index



Fonte: Aatoria Própria (2022)

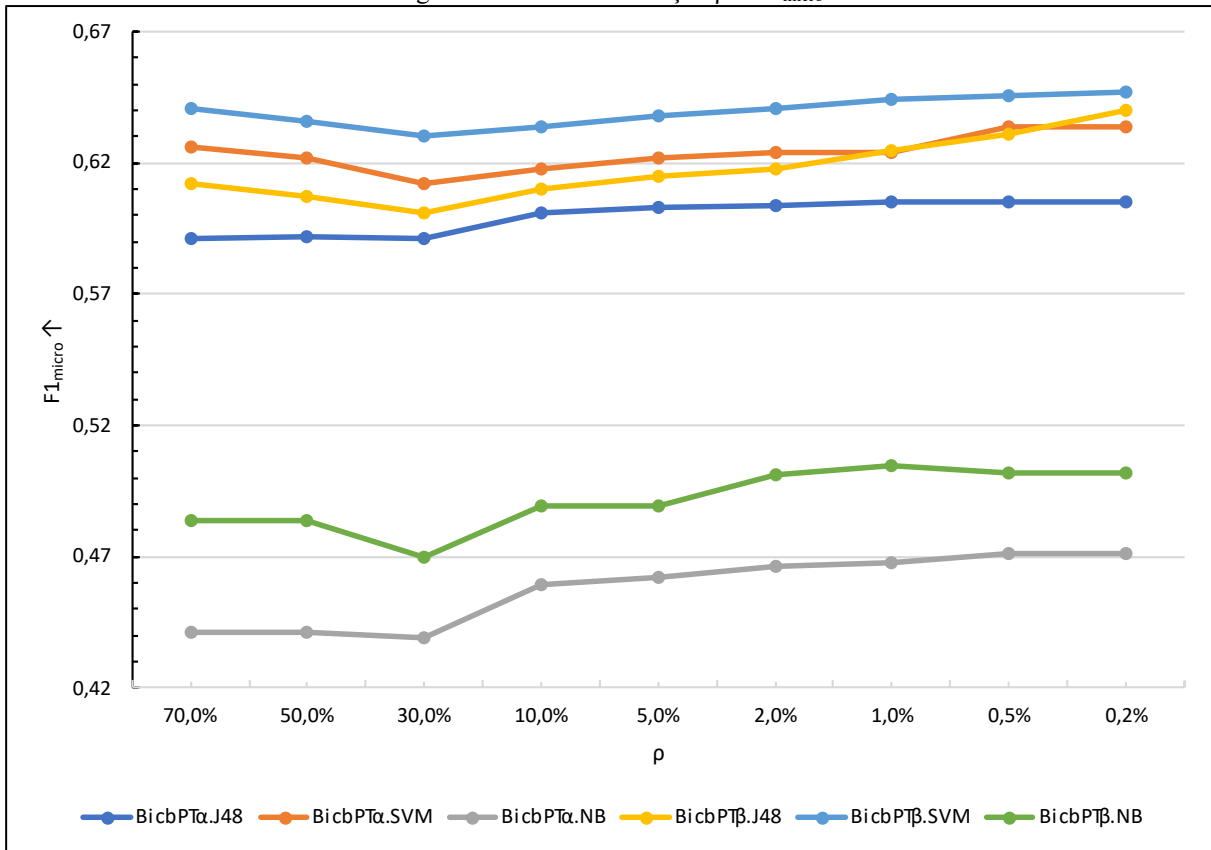
Figura 62: Gráfico da relação  $\rho$  x HL

Fonte: Autoria Própria (2022)

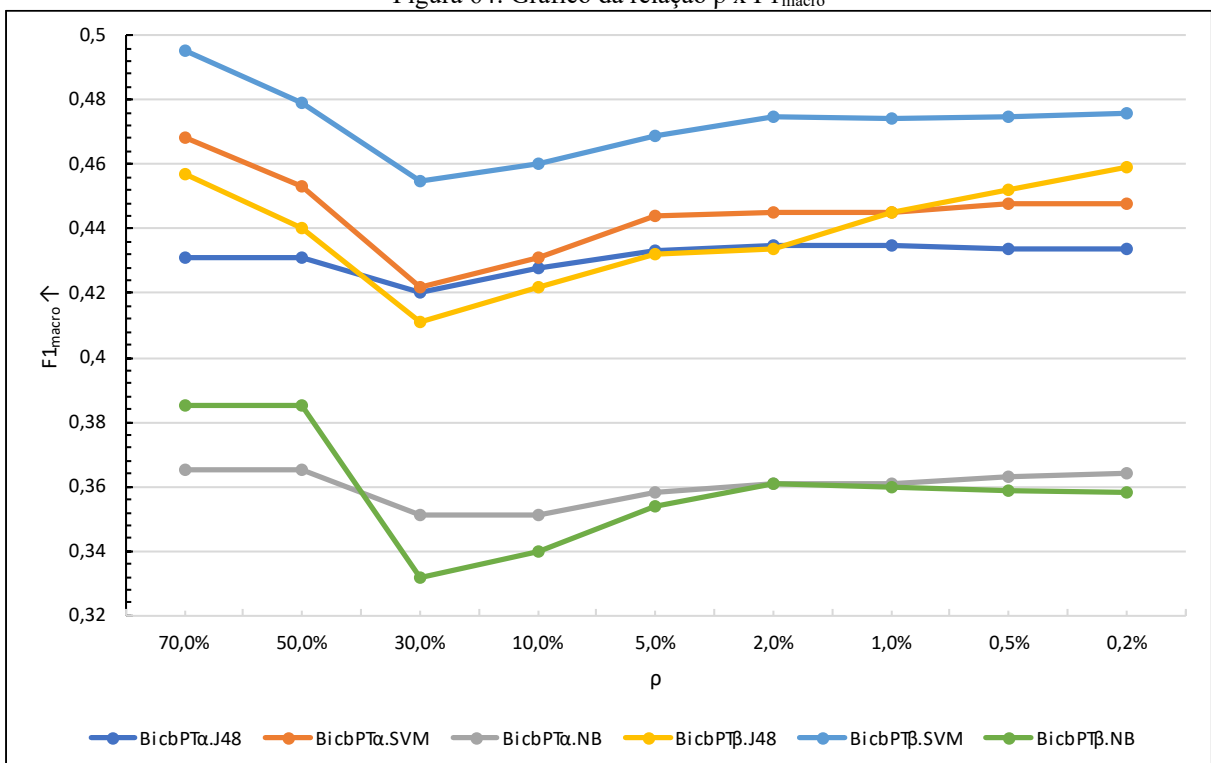
Na Figura 62 é obtido comportamento semelhante ao apresentado na métrica Jaccard Index, sendo que, aumentar a quantidade de subproblemas permite alcançar um HL menor. Importante lembrar que, para esta métrica, quanto menor o valor alcançado, melhor é o resultado.

Já para as métricas F1, enquanto na  $F1_{\text{micro}}$ , Figura 63, é observado comportamento similar às outras métricas, quanto a quantidade de subproblemas e resultado alcançado, na métrica  $F1_{\text{macro}}$ , Figura 64, o comportamento observado difere de todas as métricas apresentadas até então, de forma que quando a quantidade de subproblemas aumenta, diminuindo o valor de  $\rho$ , o desempenho na  $F1_{\text{macro}}$  cai consideravelmente, principalmente ao utilizar o SVM como classificador base, e volta a subir, mas sem ultrapassar os resultados obtidos quando  $\rho$  possui um valor alto.

Como as métricas F1 são uma relação entre precisão e revocação (*recall*), isso pode indicar que, aumentar a quantidade de subproblemas degrada uma ou ambas as métricas.

Figura 63: Gráfico da relação  $\rho$  x  $F1_{micro}$ 

Fonte: Autoria Própria (2022)

Figura 64: Gráfico da relação  $\rho$  x  $F1_{macro}$ 

Fonte: Autoria Própria (2022)

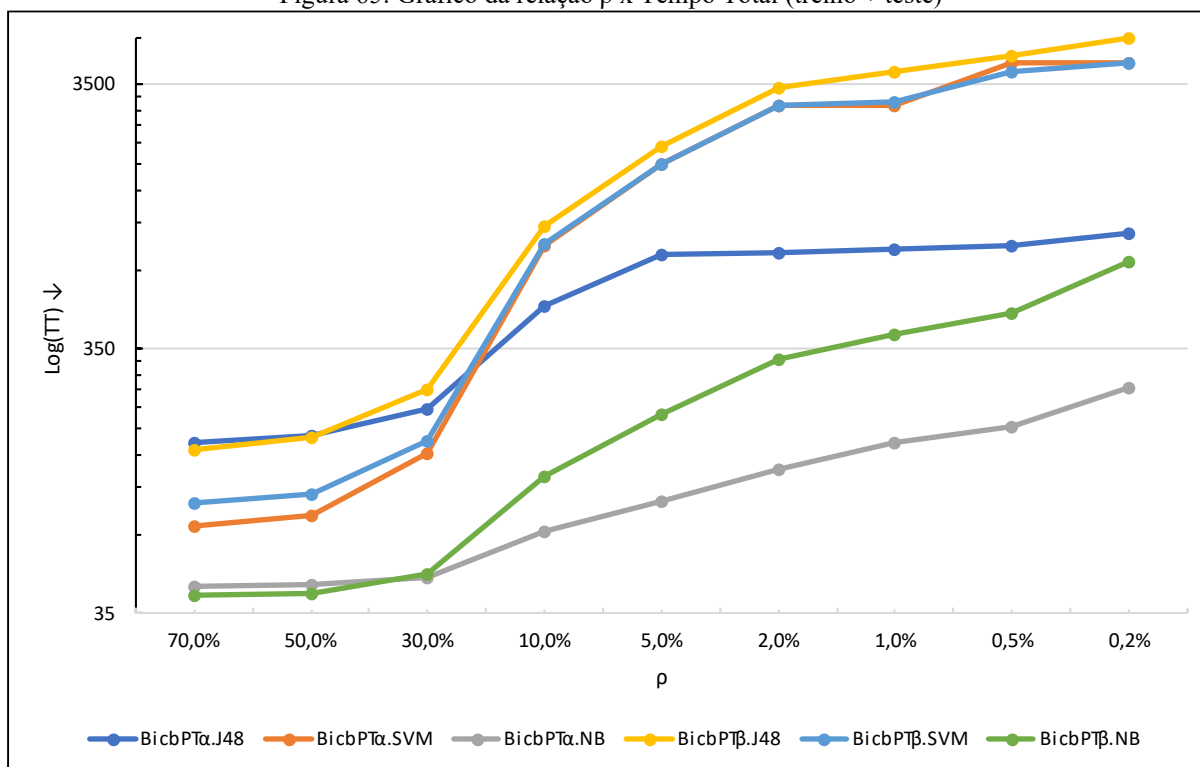


Na Figura 65, ao analisar o tempo total (treino + teste), observam-se valores maiores quando a quantidade de subproblemas tende a ser mais alta, o que já era um comportamento esperado, visto que esta é uma característica comum a maior parte dos algoritmos de classificação multirrotulo por transformação de problema.

Também, é importante destacar que das 12 bases utilizadas, apenas as bases Bibtex e BFRC apresentam tempo total acima de 1500 segundos, o que acaba contribuindo para elevar as curvas dos gráficos.

Aqui, também como ocorre em outras métricas, os algoritmos J48 e SVM apresentam comportamento semelhante e na versão  $\beta$  do BicbPT chegam a produzir quase os mesmos resultados, indicando que a estrutura de árvore de decisão pode ser uma boa escolha, visto que a interpretabilidade do resultado é melhor que em um SVM.

Figura 65: Gráfico da relação  $\rho$  x Tempo Total (treino + teste)



Fonte: Autoria Própria (2022)

Também pode-se verificar que os algoritmos de base escolhidos, SVM e NB possuem comportamento semelhante em ambas as versões do BicbPT, com exceção do C4.5 (J48) que apresentou comportamento distinto entre as versões  $\alpha$  e  $\beta$ , onde nesta última apresentou gráficos similares ao SVM.

Entre os três classificadores base escolhidos, o NB apresentou bom desempenho somente em relação ao tempo total, estando distante em todas as outras métricas, como é possível observar nos gráficos. Este comportamento pode estar ligado ao fato de que o NB considera as variáveis do problema de maneira independentes, o qual é um cenário possível, mas não comum em problemas multirrótulo.

Por fim, como em um problema real o valor ideal de  $\rho$  depende das características do conjunto de dados utilizado e das métricas a serem analisadas, sugere-se que, caso não seja possível realizar testes variados para obter o valor ideal, ele seja definido inicialmente pela Equação 6.1, onde  $T$  corresponde a quantidade de exemplos,  $L$  a quantidade de rótulos,  $X$  a quantidade de atributos do conjunto de dados utilizado e a *Densidade* refere-se a métrica apresentada em 2.2.3.

$$\rho = \begin{cases} 30\%, & T * L * X * Densidade \geq 10M \\ 2\%, & T * L * X * Densidade < 10M \end{cases} \quad ( 6.1 )$$

É importante incluir a densidade no cálculo, pois ao considerar somente  $N * L * X$  para calcular a complexidade, bases com resultados semelhantes, apresentariam comportamentos distintos em relação ao tempo de execução, e isso tornaria a definição de um valor para  $\rho$  mais difícil.

Os valores de 2% e 30% são pontos onde é possível obter um equilíbrio entre métrica e tempo total para a maior parte dos problemas multirrótulo, levando em consideração a complexidade da base e, também é importante lembrar que o valor atribuído para  $\rho$  é um mínimo, não um máximo, e frequentemente os biclusters apresentam uma quantidade maior de exemplos do que o definido no parâmetro.

## 7 CONCLUSÃO E TRABALHOS FUTUROS

Quando se pensa em classificação multirrótulo, podem ser utilizadas as estratégias de transformação de problema, adotada neste trabalho, ou adaptação de algoritmo para a realização desta tarefa. No primeiro caso, os dados multirrotulados sofrem algum processo de transformação de multirrótulo para simples-rótulo, o qual permite com que posteriormente seja possível utilizar algum algoritmo de aprendizagem de máquina simples-rótulo para realização da classificação, enquanto isso, no segundo caso, algum algoritmo simples-rótulo é adaptado para tratar os dados multirrotulados.

Ainda, a transformação de problema pode ser na proporção 1:1, ou seja, os dados multirrotulados são transformados para um único conjunto de dados transformado simples-rótulo, como, por exemplo, no algoritmo LP ou 1:N, onde os dados multirrotulados são transformados para N conjuntos de dados simples-rótulo, como, por exemplo, nos algoritmos BR, CC, ECC e RAKEL.

Na transformação de problema 1:N é possível obter subproblemas multiclasse ou binários, a depender do algoritmo utilizado, sendo que os multirrótulo-binários tendem a possuir uma posição no *ranking* de tempo de execução (treino + teste) melhor que os algoritmos multirrótulo-multiclasse, mas apresentam uma posição no *ranking* de desempenho nas métricas multirrótulo inferior quando comparadas estes últimos.

Também, na classificação multirrótulo por transformação de problema existe uma preocupação da comunidade em como realizar a manutenção das dependências entre os rótulos, ou seja, como levar as relações pré-existentes do problema multirrótulo para o(s) subproblema(s) transformado(s), pois sabe-se que a presença das relações permite uma melhor predição na classificação multirrótulo.

Essa dependência, em muitos algoritmos, como, por exemplo, CC, ECC, RAKEL, não é orgânica, ou seja, ela é pressuposta a partir de aleatoriedade e não utiliza nenhuma medida de correlação, coocorrência ou algum outro método para verificar se ela realmente existe.

Assim, visando apresentar uma forma orgânica para manutenção das dependências entre os rótulos e buscando-se manter a alta escalabilidade da categoria multirrótulo-binário de transformação de problema, neste trabalho foi proposto um novo algoritmo, BicbPT, para classificação multirrótulo baseado em biclusterização, o qual se insere na categoria de algoritmos multirrótulo-binário.

As principais vantagens do BicbPT são a não necessidade de definir, *a priori*, a quantidade de subproblemas que deve ser gerada a partir dos dados multirrotulados, visto que

o algoritmo utiliza a coocorrência entre rótulos para gerar os subproblemas, diferentemente de outros algoritmos que possuem parâmetros fixos ou, onde a quantidade de subproblemas é igual a quantidade de rótulos dos dados multirrotulados.

Também, o BicbPT faz manutenção orgânica dos relacionamentos entre rótulos nos subproblemas, ou seja, as relações não são mantidas com base em aleatoriedade, pela ordem com que os rótulos aparecem nos dados multirrotulados ou outras formas onde não se verifique o relacionamento entre dois ou mais rótulos.

Por fim, além da coocorrência, o algoritmo, em sua versão  $\beta$ , utiliza-se da forma como os rótulos se influenciam para construir os subproblemas. Dessa forma, se é encontrada uma coocorrência ABC, uma influência possível a ser verificada é, por exemplo,  $AB \rightarrow C$ , onde C é o rótulo alvo do subproblema, assim os valores de A e B nos dados irão influenciar se C receberá uma classificação positiva (1) ou negativa (0) no subproblema.

Para avaliar o algoritmo apresentado, foram utilizados 12 conjuntos de dados multirrotulados de diversos domínios e com quantidades de exemplos, atributos e rótulos variados. Também foram utilizadas as métricas multirrótulo *Jaccard Index*, *Hamming Loss*,  $F1_{\text{micro}}$ ,  $F1_{\text{macro}}$  e o tempo total de execução (treino + teste).

Após a avaliação experimental, pode-se concluir que o desenvolvimento do BicbPT permitiu aceitar a hipótese H1, “*A combinação da biclusterização com a transformação de problema multirrótulo-binário permite obter resultados superiores ou similares aos algoritmos que não consideram nenhuma forma de dependência entre os rótulos ou utilizam-se de aleatoriedade*”, apresentada na Seção 1.2.

Embora o ECC, que também é multirrótulo-binário, tenha obtido desempenho melhor em algumas bases, e nos gráficos apresentados pelo teste de Nemenyi com 95% de confiança, tenha demonstrado não existir diferença estatisticamente significativa com o BicbPT, ele possui um tempo de execução mais elevado, em média, 10 vezes mais que o algoritmo proposto.

Na hipótese H2, “*É possível manter a característica de alta escalabilidade da categoria de algoritmos multirrótulo-binários tratando a dependência real entre os rótulos sem degradar o tempo total (treino + teste)*”, a resposta do trabalho também foi positiva, visto que, mesmo identificando relações orgânicas, o que demanda mais tempo do que apenas utilizar a aleatoriedade, o algoritmo conseguiu, estatisticamente, manter a característica de alta escalabilidade da categoria multirrótulo-binário

Por fim, mesmo não sendo formulada hipótese para tal, foi possível observar que a versão  $\alpha$  do BicbPT é estatisticamente similar, em relação ao desempenho nas métricas, aos

algoritmos multirrótulo não-binários e a versão  $\beta$  consegue em várias oportunidades ser superior a estes, inclusive sendo, em média, 4 vezes mais rápido que o algoritmo RAKEL.

Dessa forma, ao considerar os resultados obtidos com os experimentos e com as hipóteses H1 e H2, pode-se concluir que o BicbPT conseguiu, combinando a transformação de problema e a biclusterização, mostrar que o uso de relações orgânicas nos subproblemas permite melhorar a classificação multirrótulo e, que é possível realiza-la mantendo a escalabilidade do algoritmo.

## 7.1 TRABALHOS FUTUROS

Durante o desenvolvimento e experimentos com o algoritmo proposto foram identificados alguns trabalhos futuros envolvendo o BicbPT e/ou a classificação multirrótulo por transformação de problema.

Como no processo de transformação de problema é necessário realizar uma agregação dos resultados obtidos dos diversos subproblemas, constatou-se a necessidade de desenvolver um trabalho de pesquisa com o intuito de avaliar o impacto das formas de realizar essa agregação, visto que, assim como o classificador base escolhido, o formato de agregação das predições adotado também influencia a saída multirrótulo, e conseqüentemente, as métricas escolhidas.

Em relação as métricas multirrótulo, verificou-se a necessidade de pesquisa e desenvolvimento de uma métrica que avalie, de fato, se as relações entre rótulos, tão importantes no contexto da transformação de problema, são realmente mantidas.

Nenhuma das métricas multirrótulo existentes consideram as relações entre rótulos no seu cálculo, dessa forma, ao considerar o multirrótulo real  $\{A, C, E\}$  para um exemplo  $x$  qualquer, onde o rótulo A só deve ocorrer se o rótulo C também ocorrer, se a saída predita for  $\{A, E\}$ , pelas métricas existentes essa saída está 67% correta, mas se fosse levado em consideração a relação existente entre os rótulos A e C, esta saída estaria somente 33% correta.

Já para o BicbPT avalia-se a possibilidade de utilizar o algoritmo de biclusterização, Bimax, codificado internamente, sem a necessidade de comunicação entre as linguagens de programação Java e R, o que poderia reduzir ainda mais o tempo de execução (treino + teste) do BicbPT.

Por fim, poder-se-ia avaliar a possibilidade de atribuir pesos dinamicamente para as camadas de transformação de problema, independente e biclusterizada, do BicbPT, visto que

em alguns conjuntos de dados multirrotulados a quantidade de rótulos independentes pode ser maior do que os que possuem relacionamentos e vice-versa.

## REFERÊNCIAS

BRUCE, R. A bayesian approach to semi-supervised learning. **Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium - NLPRS2001**, p. 57–64, 2001.

BÜLBÜL, H. I.; ÜNSAL, Ö. Comparison of classification techniques used in machine learning as applied on vocational guidance data. **Proceedings - 10th International Conference on Machine Learning and Applications, ICMLA 2011**, v. 2, p. 298–301, 2011.

CHEN, W. et al. **Document Transformation for Multi-label Feature Selection in Text Categorization**. Seventh IEEE International Conference on Data Mining (ICDM 2007). *Anais...IEEE*, out. 2007.

CHENG, Y.; CHURCH, G. M. Biclustering of Expression Data. **Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology**, v. 8, p. 93–103, 2000.

CURI, Z.; BRITTO JR, A. DE S.; PARAISO, E. C. Multi-label Classification of User Reactions in Online News. **CoRR**, 2018.

CURI, Z.; BRITTO JR., A. DE S.; PARAISO, E. C. Using Correlation for Labelset Selection in Multi-Label Classification of Users Reactions. **Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference**, p. 167–172, 19 maio 2019.

DEMSAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. **The Journal of Machine Learning Research**, v. 7, n. 2006-12– 01, p. 1–30, 2006.

DIETTERICH, T. G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. **Neural Computation**, v. 10, n. 7, 1 out. 1998.

DOSCIATTI, M. M.; FERREIRA, L. P. C.; PARAISO, E. C. Anotando um Corpus de Notícias para a Análise de Sentimento: um Relato de Experiência. **Proceedings of Symposium in Information and Human Language Technology**, p. 121–130, 2015.

GAMA, P. P. DA; BERNARDINI, F. C.; ZADROZNY, B. Proposta de um novo método para classificação multirrótulo baseado em seleção aleatória e bagging. **Anais do IX Encontro Nacional de Inteligência Artificial (ENIA2012)**, p. 1–12, 2012.

GIBAJA, E.; VENTURA, S. Multi-label learning: A review of the state of the art and ongoing research. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 4, n. 6, p. 411–444, 2014.

GIBAJA, E.; VENTURA, S. A Tutorial on Multilabel Learning. **ACM Computing Surveys**, v. 47, n. 3, p. 1–38, 16 abr. 2015.

GJORGJEVIKJ, D.; MADJAROV, G. **Two Stage Classifier Chain Architecture for efficient pair-wise multi-label learning**. 2011 IEEE International Workshop on Machine Learning for

Signal Processing. **Anais...IEEE**, set. 2011. Disponível em: <http://ieeexplore.ieee.org/document/6064599/>

GLINKA, K.; ZAKRZEWSKA, D. Effective Multi-label Classification Method for Multidimensional Datasets. v. 400, 2016.

GONÇALVES, E. C.; FREITAS, A. A.; PLASTINO, A. A Survey of Genetic Algorithms for Multi-Label Classification. **2018 IEEE Congress on Evolutionary Computation, CEC 2018 - Proceedings**, 2018.

KAISER, S. et al. **biclust.** , 2021.

KASHEF, S.; NEZAMABADI-POUR, H.; NIKPOUR, B. Multilabel feature selection: A comprehensive review and guiding experiments. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 8, n. 2, p. e1240, mar. 2018.

LAI, C.-C.; TSAI, M.-C. An Empirical Performance Comparison of Machine Learning Methods for Spam E-Mail Categorization. **Fourth International Conference on Hybrid Intelligent Systems (HIS'04)**, p. 44–48, 2005.

LOTF, H.; RAMDANI, M. **Multi-Label Classification**. Proceedings of the 13th International Conference on Intelligent Systems: Theories and Applications. **Anais...New York, NY, USA: ACM**, 23 set. 2020.

MADEIRA, S. C.; OLIVEIRA, A. L. Biclustering algorithms for biological data analysis: a survey. **IEEE/ACM Transactions on Computational Biology and Bioinformatics**, v. 1, n. 1, p. 24–45, jan. 2004.

MADJAROV, G. et al. An extensive experimental comparison of methods for multi-label learning. **Pattern Recognition**, v. 45, n. 9, p. 3084–3104, set. 2012.

MADJAROV, G.; GJORGJEVIKJ, D.; DŽEROSKI, S. Two stage architecture for multi-label learning. **Pattern Recognition**, v. 45, n. 3, p. 1019–1034, mar. 2012.

MITCHELL, T. M. **Machine Learning**. [s.l.] McGraw-Hill Higher Education, 1997.

MURALI, T.; KASIF, S. Extracting conserved gene expression motifs from gene expression data. **Pacific Symposium on Biocomputing 2003**, v. 8, p. 77–88, 2003.

PEETERS, R. The maximum edge biclique problem is NP-complete. **Discrete Applied Mathematics**, v. 131, n. 3, p. 651–654, set. 2003.

PONTES, B.; GIRÁLDEZ, R.; AGUILAR-RUIZ, J. S. Biclustering on expression data: A review. **Journal of Biomedical Informatics**, v. 57, p. 163–180, out. 2015.

PRELIĆ, A. et al. A systematic comparison and evaluation of biclustering methods for gene expression data. **Bioinformatics**, v. 22, n. 9, 1 maio 2006.

READ, J. et al. Classifier chains for multi-label classification. **Machine Learning**, v. 85, n. 3, p. 333–359, 30 dez. 2011.



READ, J. et al. MEKA: A Multi-label/Multi-target Extension to Weka. **Journal of Machine Learning Research**, v. 17, n. 21, p. 1–5, 2016.

REZENDE, S. O. **Sistemas Inteligentes: Fundamentos e Aplicações**. Barueri, SP: Manole, 2005.

RODRIGUEZ-BAENA, D. S.; PEREZ-PULIDO, A. J.; AGUILAR–RUIZ, J. S. A biclustering algorithm for extracting bit-patterns from binary datasets. **Bioinformatics**, v. 27, n. 19, p. 2738–2745, 1 out. 2011.

SANTOS, A. DE M. **Investigando a combinação de técnicas de aprendizado semissupervisionado e classificação hierárquica multirrótulo**. [s.l.] Universidade Federal do Rio Grande do Norte, Natal, 2012.

SARKAR, D.; BALI, R.; SHARMA, T. **Practical Machine Learning with Python**. Berkeley, CA: Apress, 2018.

SOONSIRIPANICHKUL, B.; MURATA, T. Domination dependency analysis of sales marketing based on multi-label classification using label ordering and cycle chain classification. **Proceedings - 2016 5th IIAI International Congress on Advanced Applied Informatics, IIAI-AAI 2016**, p. 1048–1053, 2016.

SUN, L.; KUDO, M. Multi-label classification by polytree-augmented classifier chains with label-dependent features. **Pattern Analysis and Applications**, v. 22, n. 3, p. 1029–1049, 15 ago. 2019.

TANAY, A.; SHARAN, R.; SHAMIR, R. Discovering statistically significant biclusters in gene expression data. **Bioinformatics**, v. 18, n. Suppl 1, p. S136–S144, 1 jul. 2002.

TSOUMAKAS, G. et al. Mulan: A Java Library for Multi-Label Learning. **Journal of Machine Learning Research**, v. 12, p. 2411–2414, 2011.

TSOUMAKAS, G.; KATAKIS, I. Multi-Label Classification: An Overview. **International Journal of Data Warehousing and Mining**, v. 3, n. 3, p. 1–13, jul. 2007.

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. Mining Multi-label Data. In: **Data Mining and Knowledge Discovery Handbook**. Boston, MA: Springer US, 2009. p. 667–685.

TSOUMAKAS, G.; VLAHAVAS, I. Random k-Labelsets: An Ensemble Method for Multilabel Classification. **Machine Learning: ECML 2007**, p. 406–417, 2007.

UBIK, S.; ŽEJDL, P. Evaluating application-layer classification using a machine learning technique over different high speed networks. **Proceedings - 5th International Conference on Systems and Networks Communications, ICSNC 2010**, p. 387–391, 2010.

URBANEK, S. **rJava**. , 2021.

WEVER, M. et al. LiBRe: Label-Wise Selection of Base Learners in Binary Relevance for Multi-label Classification. p. 561–573, 2020.

WITTEN, I. H. et al. **Data Mining: Practical Machine Learning Tools and Techniques**. 4<sup>o</sup> ed. Cambridge, Estados Unidos: Elsevier, 2016.

XU, R.; WUNSCH II, D. Survey of Clustering Algorithms. v. 16, n. 3, p. 645–678, 2005.

ZHAN, Y. et al. Chinese text categorization study based on feature weight learning. **Proceedings of the 2009 International Conference on Machine Learning and Cybernetics**, v. 3, n. July, p. 1723–1726, 2009.

ZHANG, J.-J.; FANG, M.; LI, X. Multi-label learning with discriminative features for each label. **Neurocomputing**, v. 154, p. 305–316, abr. 2015.

ZHANG, M. L.; ZHOU, Z. H. A review on multi-label learning algorithms. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 8, p. 1819–1837, 2014.

ZHANG, M.-L. et al. Binary relevance for multi-label learning: an overview. **Frontiers of Computer Science**, v. 12, n. 2, p. 191–202, 23 abr. 2018.

ZHU, X.; GOLDBERG, A. B. Introduction to Semi-Supervised Learning. **Synthesis Lectures on Artificial Intelligence and Machine Learning**, v. 3, n. 1, p. 1–130, jan. 2009.