

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Departamento de Computação

Incorporando correlações entre exemplos para classificação
multirrótulo via espaço de classes

Iuri Bonna Mauricio de Abreu



São Carlos – SP

Incorporando correlações entre exemplos para classificação multirrótulo via espaço de classes

Iuri Bonna Mauricio de Abreu

Orientador Prof. Dr. Ricardo Cerri

Monografia final de conclusão de curso apresentada
ao Departamento de Computação – DC-UFSCar,
como requisito parcial para obtenção do título de
Bacharel em Engenharia de Computação.

Área de Concentração: Machine Learning, Inteligên-
cia Artificial, Aprendizado Multirrótulo

UFSCar – São Carlos

Junho de 2021

Abreu, Iuri Bonna Mauricio de
Incorporando correlações entre exemplos para
classificação multirrótulo via espaço de classes / Iuri
Bonna Mauricio de Abreu. - São Carlos - SP, 2021.
48 p.; 29,7 cm.

Orientador Ricardo Cerri.
Monografia (Graduação) - Departamento de Computação -
DC-UFSCar, São Carlos - SP, 2021.

1. Aprendizado de Máquina. 2. Inteligência
Artificial. 3. Aprendizado Multirrótulo. I. Cerri,
Ricardo. II. Departamento de Computação - DC-UFSCar.
III. Título.

AGRADECIMENTOS

Antes de mais nada, gostaria de agradecer à minha família por terem me apoiado durante toda minha trajetória até aqui. Meus pais, irmão, avós, tios, primos e várias outras pessoas muito especiais que nunca mediram esforços para me ver crescer pessoal e profissionalmente, sempre com saúde, amor, educação e liberdade para que eu pudesse buscar meus sonhos — não importando o quão ambiciosos eles fossem. Vocês são, e sempre serão, a minha base!

Ainda, não poderia deixar de agradecer a um grupo de pessoas que foram fundamentais para que eu alcançasse esse momento: meus professores. Com muita paciência, sabedoria e vontade de ensinar me fizeram acreditar na educação como ferramenta de mudar a sociedade.

Nesse sentido, um agradecimento especial a UFSCar, ao Departamento de Computação e a todos os professores, pesquisadores e técnicos administrativos que mantêm uma Universidade pública com educação de qualidade, pesquisa de ponta e um corpo de alunos diverso e de futuro promissor, do qual tive o prazer de fazer parte e nele construir amizades que foram essenciais para a minha experiência de graduação ter sido completa. Gratidão, em especial, ao meu orientador, Prof. Ricardo Cerri, que muito me ensina desde 2015, quando demos início a pesquisa que deu origem a este trabalho.

Na UFSCar eu pude ter vivências fantásticas, e me sinto no dever de ressaltar o Movimento Empresa Júnior (MEJ) como a mais engrandecedora delas. Na CATI Jr., pude evoluir bastante como profissional deixar o meu legado na história da empresa. No MEJ, encontrei um propósito que vivi diariamente e continua ecoando nas minhas ações. E tudo isso foi muito mais fácil por conta das pessoas incríveis que conheci e pude compartilhar ótimos momentos.

É claro que eu também não poderia deixar de mencionar a República Tipo Zero, lugar que foi muito mais do que minha morada pelos últimos 6 anos, onde fui acolhido desde o dia zero e cada vez me senti mais pertencente. Que se sintam abraçados todos os mais de 50 irmãos e a nova mãe que ganhei e levarei para sempre comigo. Obrigado pelos melhores anos da minha vida!

Por fim, agradeço às minhas sinceras e duradouras amizades de Belém, que tenho a alegria de manter contato mesmo com a distância, e a todos aqueles que tenham compartilhado alguns momentos e contribuíram para a minha evolução.

Muito obrigado!

RESUMO

ABREU, I.B.M.. **Incorporando correlações entre exemplos para classificação multirrótulo via espaço de classes**. 2021. 48 f. Monografia (Graduação) – Departamento de Computação – DC-UFSCar, São Carlos – SP.

Classificação multirrótulo é uma tarefa de aprendizado de máquina na qual os exemplos podem ser classificados em duas ou mais classes simultaneamente. Nessa tarefa, existem correlações entre os exemplos que pertencem ao mesmo ou semelhante conjunto de classes. Este estudo propõe a incorporação de correlações entre exemplos por meio de modificação nos conjuntos de dados multirrótulo. Foi utilizado o espaço de classes para criar novos atributos que representam essas correlações. Os conjuntos de dados originais e modificados foram utilizados com diversos métodos de classificação multirrótulo. Experimentos demonstraram que o método obteve resultados de classificação melhores em relação aos seus pares de conjuntos de dados não modificados para a maioria dos algoritmos utilizados. Todos os métodos foram avaliados com base em medidas criadas especialmente para problemas multirrótulo.

Palavras-chave: Aprendizado de Máquina, Inteligência Artificial, Aprendizado Multirrótulo.

ABSTRACT

ABREU, I.B.M.. **Incorporando correlações entre exemplos para classificação multirrotulo via espaço de classes**. 2021. 48 f. Monografia (Graduação) – Departamento de Computação – DC-UFSCar, São Carlos – SP.

Multi-label classification is a machine learning task where instances can be classified into two or more labels simultaneously. In this task, there exist correlations between the instances belonging to same or similar sets of labels. This work proposes the incorporation of instance correlations by modifying the multi-label datasets. The label-space was used to create new features, which represent these correlations. The original and modified datasets were used with different multi-label classification methods. Experiments have shown that the method obtained better classification results in comparison with its original datasets counterparts for most of the algorithms. All methods were evaluated with measures specifically designed for multi-label problems.

Key-words: Machine Learning, Artificial Intelligence, Multi-label learning.

LISTA DE FIGURAS

Figura 1 – Exemplo de um problema de classificação multirrótulo	20
Figura 2 – Estratégias para Classificação Multirrótulo. Adaptado de Carvalho e Freitas (2009)	21
Figura 3 – Inclusão do vetor de correlações (protótipo)	32
Figura 4 – Diagrama Crítico para o teste estatístico <i>post-hoc</i> de Nemenyi - Estratificação <i>Labelset</i>	41
Figura 5 – Diagrama Crítico para o teste estatístico <i>post-hoc</i> de Nemenyi - Estratificação Iterativa	41

LISTA DE TABELAS

Tabela 1 – Características dos conjuntos de dados	33
Tabela 2 – Valores de parâmetros utilizados	34
Tabela 3 – Resultados do <i>Naive Bayes</i> usando a transformação BR	37
Tabela 4 – Resultados do SVM usando a transformação BR	37
Tabela 5 – Resultados do J48 usando a transformação BR	38
Tabela 6 – Resultados do KNN usando a transformação BR	38
Tabela 7 – Resultados do <i>Naive Bayes</i> usando a transformação LP	38
Tabela 8 – Resultados do SVM usando a transformação LP	39
Tabela 9 – Resultados do J48 usando a transformação LP	39
Tabela 10 – Resultados do KNN usando a transformação LP	39
Tabela 11 – Resultados do <i>Back-Propagation Multi-Label Learning</i> (BPMLL)	40
Tabela 12 – Resultados do <i>Multi-Label k-Nearest Neighbours</i> (MLkNN)	40
Tabela 13 – Resultados do <i>Classifier Chains</i>	41

LISTA DE ALGORITMOS

Algoritmo 1 – Algoritmo de Incorporação de Correlações	31
Algoritmo 2 – Classificação de Novos Exemplos	31

LISTA DE ABREVIATURAS E SIGLAS

BPMLL	<i>Back-Propagation Multi-Label Learning</i>
BR	<i>Binary Relevance</i>
CC	<i>Classifier Chains</i>
CR	Cardinalidade de Rótulos
DR	Densidade de Rótulos
IA	Inteligência Artificial
IJCNN	<i>International Joint Conference on Neural Networks</i>
KNN	<i>K-Nearest Neighbors</i>
LP	<i>Label Powerset</i>
MLKNN	<i>Multi-Label K-Nearest Neighbors</i>
NB	<i>Naive Bayes</i>
RakEL	<i>Random k-Labelsets</i>
SVM	<i>Support Vector Machine</i>

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Objetivos	22
1.1.1	<i>Objetivos Gerais</i>	22
1.1.2	<i>Objetivos Específicos</i>	22
1.2	Organização	23
2	REVISÃO BIBLIOGRÁFICA	25
2.1	Abordagem Independente de Algoritmo	25
2.2	Abordagem Dependente de Algoritmo	26
2.3	Incorporação de Correlações	27
3	METODOLOGIA	29
3.1	Método de Correlação Proposto	29
3.2	Conjuntos de Dados	32
3.3	Algoritmos de Classificação	33
3.4	Medidas de Avaliação	34
3.5	Testes Estatísticos	35
4	RESULTADOS E DISCUSSÃO	37
5	CONCLUSÃO E TRABALHOS FUTUROS	43
	REFERÊNCIAS	45

INTRODUÇÃO

A Inteligência Artificial (IA) é um campo da ciência que registrou desenvolvimento exponencial nas últimas décadas. Uma das definições mais aceitas para IA é a de Nils J. Nilsson (2009):

IA é a atividade designada a tornar as máquinas inteligentes, e inteligência é a qualidade que permite uma entidade funcionar apropriadamente e com habilidades de previsão em seu ambiente

Um dos subcampos mais proeminentes da IA é o Aprendizado de Máquina, e é mais um que possui diversas definições. No final da década de 50, quando os primeiros experimentos eram executados em uma ainda nascente área da ciência da computação, Arthur Samuel definiu Aprendizado de Máquina como "o campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados" (SIMON, 2013). Desde então, o reconhecido professor e cientista Tom Mitchell cunhou com certo formalismo a definição mais reproduzida para esse campo de estudo (MITCHELL, 1997):

O estudo do Aprendizado de Máquina está preocupado com a questão de como construir programas de computador que evoluem automaticamente com experiência.

Dizemos que uma máquina aprende em relação a uma tarefa em particular T , métrica de desempenho P , e tipo de experiência E , se o sistema consistentemente melhora o desempenho P na tarefa T , seguindo a experiência E .

Em Aprendizado de Máquina, tarefas de aprendizado supervisionado, em especial tarefas de classificação, são aquelas em que o algoritmo conhece previamente a qual classe (ou rótulo) pertence cada exemplo do conjunto de dados. O algoritmo deve ser treinado para construir um modelo que determine a classe de novos exemplos assertivamente sem depender de novas entradas.

Na literatura, os problemas de classificação convencionais são chamados de problemas monorrótulo. Para a esse tipo de tarefa de aprendizado, a entrada do algoritmo consiste em um conjunto de N exemplos \mathbf{x}_i , com $i = 1, \dots, N$. Cada exemplo, possui, tipicamente um conjunto de A atributos, em que $x_{i,j}$ refere-se ao valor do atributo j , com $j = 1, \dots, A$, para o exemplo \mathbf{x}_i .

Nesses problemas, um classificador é treinado em um conjunto de exemplos que estão associados com um única classe c_j de um conjunto de classes disjuntas L , onde $|L| > 1$. Se $|L| = 2$, então o problema é chamado de problema de classificação binária, e se $|L| > 2$, o problema é chamado de problema de classificação multi-classe (TSOUMAKAS; KATAKIS, 2007)

Quando um exemplo pode ser classificado em um conjunto de rótulos simultaneamente, o problema é chamado de classificação multirrótulo. Isto é, um exemplo \mathbf{x}_i pode estar associado a um conjunto de rótulos \mathbf{C}_i . O exemplo \mathbf{x}_i é representado da forma $(\mathbf{A}_i, \mathbf{C}_i)$ onde \mathbf{A} são os atributos do conjunto de dados e $\mathbf{C}_i \in L$ e $L = \{c_j : j = 1 \dots q\}$ é o conjunto dos q rótulos simples que participam dos multirrótulos $\mathbf{C}_i, i = 1 \dots N$. (CHERMAN, 2013)

Mesmo que os problemas de classificação convencionais sejam bem estabelecidos e amplamente cobertos pela literatura, existem uma série de problemas nos quais a simplificação não surte efeito, visto que objetos do mundo real podem ser complicados e estarem associados a diversas categorias simultaneamente e possuírem significados distintos. (ZHANG; ZHOU, 2006)

No passado, a classificação multirrótulo foi fortemente motivada por tarefas de categorização de documentos (GONÇALVES; QUARESMA, 2003; LAUSER; HOTH, 2003; LUO; ZINCIR-HEYWOOD, 2005) e diagnósticos médicos (KARALIC; PIRNAT, 1991). Entretanto, muitos trabalhos podem ser encontrados nas áreas de Bioinformática (CLARE; KING, 2001; ZHANG; ZHOU, 2005; ELISSEFF; WESTON, 2001) e classificação de imagens (BOUTELL *et al.*, 2004; SHEN *et al.*, 2003). Em problemas de categorização de documentos, esses geralmente pertencem a mais de uma classe, por exemplo Ciência da Computação e Biologia. Em problemas de diagnóstico médico, por exemplo, um paciente pode estar sofrendo de diabetes e câncer de próstata ao mesmo tempo. Já imagens, por sua vez, podem conter características de montanha e praia simultaneamente (TSOUMAKAS; KATAKIS, 2007).

A Figura 1 ilustra um problema de classificação multirrótulo. Na figura, um exemplo representa um documento, que pode ser associado a duas ou mais categorias simultaneamente (Esporte, Religião, Ciência e Política). Como pode ser observado, um documento pode abordar assuntos relacionados a esporte e política ao mesmo tempo, ou religião e ciência.

Exemplo	Esporte	Religião	Ciência	Política
1	X			X
2			X	X
3	X			
4		X	X	

Figura 1 – Exemplo de um problema de classificação multirrótulo

A Figura 2 ilustra uma visão geral das diferentes estratégias propostas na literatura para tratar problemas de classificação multirrótulo. De acordo com a figura, essas estratégias pertencem a duas grandes abordagens: abordagem independente de algoritmo e abordagem dependente de algoritmo. A abordagem independente de algoritmo utiliza algoritmos tradicionais de classificação para tratar problemas multirrótulo, transformando o problema multirrótulo original em um conjunto de problemas simples-rótulo. A abordagem dependente de algoritmo cria algoritmos específicos para tratar o problema multirrótulo. Esses algoritmos podem ser baseados em algoritmos de classificação convencionais, como Máquinas de Vetores de Suporte (VAPNIK, 1999) e Árvores de Decisão (QUINLAN, 1993).

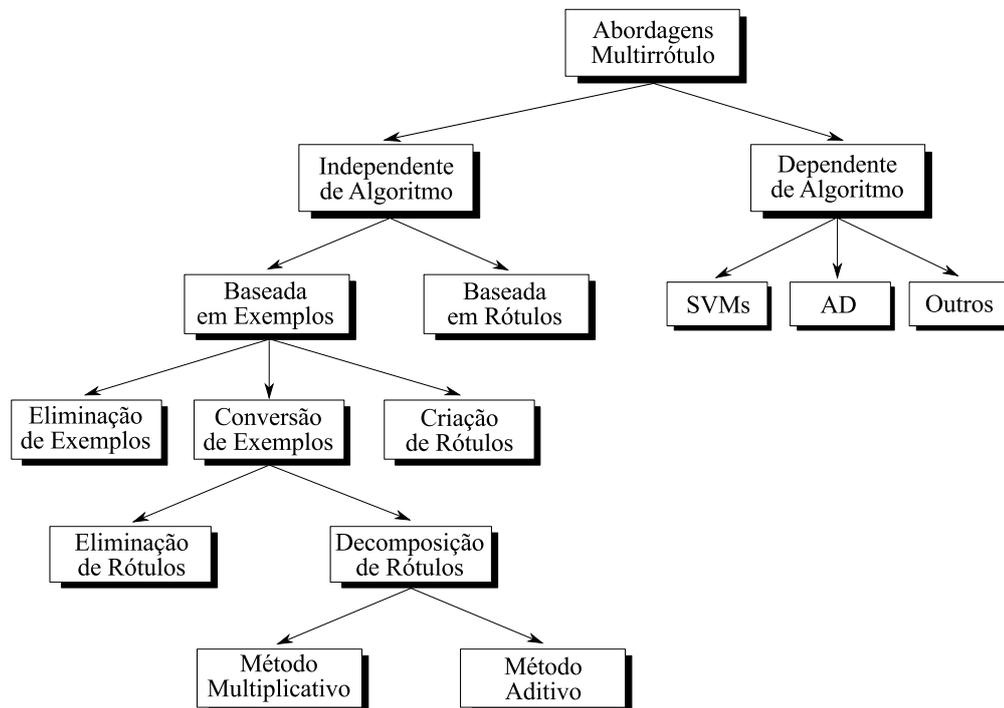


Figura 2 – Estratégias para Classificação Multirrótulo. Adaptado de Carvalho e Freitas (2009)

Em Bioinformática, métodos de Aprendizado de Máquina são frequentemente utilizados na tarefa de predição de funções de proteínas. No entanto, trata-se de uma tarefa complexa, pois uma única proteína pode desempenhar múltiplas funções. Essas funções podem estar relacionadas, por exemplo, à topologia do DNA, à síntese e replicação do DNA, e à recombinação e reparação do DNA. Consequentemente, métodos de Aprendizado de Máquina capazes de lidar com problemas multirrótulo são mais apropriados (TANAKA *et al.*, 2015).

Na tarefa de predição de proteínas (bem como em outras tarefas de classificação multirrótulo), há uma correlação entre as classes do problema. Isso é natural, já que um exemplo é associado a duas ou mais classes (por exemplo, uma proteína associada a várias funções). Assim, espera-se também que haja uma correlação entre os exemplos associados a um mesmo conjunto (ou conjuntos parecidos) de classes, pois é natural que exemplos classificados em um mesmo conjunto de classes compartilhem características semelhantes. No trabalho de Stojanova

et al. (2013), por exemplo, a correlação entre as proteínas foi utilizada durante o aprendizado multirrótulo, melhorando o desempenho de classificação.

Na classificação multirrótulo, há uma relação intrínseca entre os exemplos, representada pelo conjunto de rótulos compartilhado entre eles. Explorar essa dependência entre os rótulos pode levar a melhores desempenhos de classificação (DEMBZYŃSKI *et al.*, 2012). Ademais, várias estratégias exploram essas dependências para representar as relações entre os exemplos. Normalmente, as relações são representadas com a inclusão de novos rótulos binários durante a etapa de treinamento (READ *et al.*, 2009; CHERMAN; METZ; MONARD, 2012; SPOLAÔR *et al.*, 2016).

Este Trabalho de Conclusão de Curso expande um artigo publicado pelo mesmo autor, em conjunto com outros pesquisadores, no qual a incorporação de uma matriz de correlações entre os exemplos mais similares apresentou melhora nos resultados de medidas de avaliação de classificação multirrótulo para diversos algoritmos (ABREU; MANTOVANI; CERRI, 2017).

1.1 Objetivos

Este estudo almeja atingir alguns objetivos relacionados ao estado da arte da classificação multirrótulo e outros intrinsecamente relacionados aos resultados dos métodos propostos. Esses objetivos podem ser divididos em duas categorias:

1.1.1 *Objetivos Gerais*

Dentre as principais expectativas em relação ao estudo, destaca-se a intenção de consolidar a incorporação das correlações entre exemplos como uma etapa válida e valorosa na tarefa de classificação multirrótulo. Conforme será visto na Revisão Bibliográfica, diversos artigos publicados levam em consideração correlações para melhorar os resultados dos algoritmos de classificação. Portanto, os métodos propostos neste estudo visam referendar essa prática e apresentar resultados positivos para diversas medidas de avaliação e algoritmos nos conjuntos de dados avaliados.

Adicionalmente, o estudo é uma extensão do projeto de Iniciação Científica (ABREU; MANTOVANI; CERRI, 2017) do autor, que propôs uma metodologia inicial de incorporação de correlações. Portanto, é esperado que este documento traga um detalhamento maior da metodologia proposta e uma análise assertiva dos resultados a partir da comparação entre métodos de estratificação, algoritmos, características dos conjuntos de dados e relevância estatística dos valores obtidos com uso de medidas de avaliação.

1.1.2 *Objetivos Específicos*

Por outro lado, alguns objetivos estão relacionados diretamente ao método proposto. O primeiro e principal deles é obter resultados melhores nos experimentos realizados com conjuntos de dados modificados pelo método em relação aos seus pares originais. Outros objetivos paralelos

são: investigar se os métodos de estratificação possuem efeito relevante nos resultados de classificação, identificar qual abordagem de algoritmos (dependente ou independente) é mais adequada à metodologia proposta, e compreender como as características dos conjuntos de dados se comportam com os diferentes tipos de algoritmos e transformações.

Ademais, o detalhamento da estratégia de incorporação de correlações entre os exemplos dos conjuntos de dados que será apresentada nas seções seguinte possui objetivos pré-definidos:

- Explorar conceitos e métodos desenvolvidos na bibliografia e alavancá-los como etapas do trabalho;
- Apresentar a metodologia e o seu desenvolvimento para uso geral da comunidade científica;
- Elucidar distinções entre os métodos e potencializar estas para as características de cada conjunto de dados;
- Unir os métodos a algoritmos comumente utilizados na tarefa de classificação multirrótulo e comparar os resultados desses aos encontrados atualmente na literatura;
- Avaliar o desempenho dos métodos associados a algoritmos na classificação de cada conjunto de dados com amparo de medidas de avaliação e testes estatísticos amplamente utilizados na literatura.

1.2 Organização

O restante do documento está organizado da seguinte forma:

- **Capítulo 2: Revisão Bibliográfica** - Uma revisão sobre estudos consolidados e em estado da arte que constam na literatura do tema. Os principais conceitos de aprendizado multirrótulo também são apresentados neste capítulo.
- **Capítulo 3: Metodologia** - Neste capítulo, todos os componentes do desenvolvimento da solução são apresentados: método proposto, conjuntos de dados, algoritmos, medidas de avaliação e testes estatísticos que foram utilizados são descritos em maiores detalhes.
- **Capítulo 4: Resultados e Discussão** - Resultados obtidos por meio das medidas de avaliação são expostos e discutidos detalhadamente nessa seção.
- **Capítulo 5: Conclusão e Trabalhos Futuros** - Por fim, as considerações finais deste estudo são apresentadas neste capítulo. Uma recapitulação do método proposto é feita, seguida de um resumo da análise de resultados e descrição das limitações do estudo e menções a possíveis trabalhos futuros.

REVISÃO BIBLIOGRÁFICA

A bibliografia do aprendizado multirrótulo é cada vez mais extensa, com artigos qualificados e que discutem temas distintos da área. As principais referências para este estudo estão segmentadas em três focos:

1. **Abordagem Independente de Algoritmo:** trabalhos que propõem métodos que podem ser utilizados com diferentes algoritmos de classificação, inclusive aqueles que não são tradicionalmente utilizados para classificação multirrótulo. Exemplos são as transformações *Binary Relevance* e *Label-Powerset*;
2. **Abordagem Dependente de Algoritmo:** trabalhos que apresentam métodos incorporados a algoritmos adaptados especificamente para a tarefa de classificação multirrótulo, como o *Back-Propagation Multi-Label Learning* (BPMLL);
3. **Incorporação de Correlações:** trabalhos que, independente dos conjuntos de dados, abordagens e algoritmos utilizados, propõem métodos que estipulam a utilização da correlação entre os exemplos e as incorporam na tarefa de treinamento.

Nas próximas seções, as principais referências bibliográficas de cada segmento são apresentadas e suas soluções descritas.

2.1 Abordagem Independente de Algoritmo

Um método muito simples baseado na abordagem independente de algoritmo utiliza L classificadores, sendo L o número de rótulos que estão envolvidos no problema. Cada classificador é então associado a um rótulo e treinado para resolver um problema de classificação binária, na qual é considerado o rótulo ao qual ele está associado contra todos os outros rótulos envolvidos. Esse método é chamado de *Binary-Relevance* (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010).

Um ponto fraco desse método é que ele assume que os rótulos atribuídas a um exemplo são independentes entre si. Isso nem sempre é verdade, e ignorar as possíveis correlações entre os rótulos pode fazer com que o método tenha pouca capacidade de generalização. Seu processo de transformação é reversível, ou seja, é possível recuperar os rótulos do problema original a partir do novo problema criado.

Nos trabalhos de Cherman, Metz e Monard (2012), Read *et al.* (2009) e Dembczyński, Cheng e Hüllermeier (2010) foram propostos métodos baseados na transformação *Binary-Relevance*. A ideia é utilizar os rótulos dos exemplos para complementar os vetores de atributos dos mesmos, de maneira a incorporar no aprendizado as dependências existentes entre os rótulos do problema multirrótulo.

Outro exemplo de método independente de algoritmo é a transformação *Label-Powerset*. Nela, para cada exemplo, todos os rótulos atribuídos aquele exemplo são combinados em um novo e único rótulo. Com essa combinação, são consideradas as correlação entre os rótulos, porém o número de rótulos envolvidos no problema pode aumentar consideravelmente, e alguns rótulos podem terminar com poucos exemplos positivos. Essa técnica de combinação de rótulos foi utilizada nos trabalhos de Tsoumakas e Katakis (2007) e Boutell *et al.* (2004).

Ainda no trabalho de Tsoumakas e Katakis (2007), foi proposto um método chamado *Random k-Labelsets* (RAkEL), baseado no método *Label-Powerset*. Esse método iterativamente constrói uma combinação de m classificadores *Label-Powerset*. Sendo $L = \{\lambda_i\}, i = 1..|L|$ o conjunto de rótulos do problema, um *k-labelset* é dado por um subconjunto $Y \subseteq L$, com $k = |Y|$. O termo L^k representa o conjunto de todos os *k-labelsets* de L . A cada iteração, $1..m$, um *k-labelset* Y_i é selecionado randomicamente de L^k , sem reposição. Um classificador H_i é então treinado para Y_i .

Para a classificação de um novo exemplo, cada classificador H_i toma uma decisão binária para cada rótulo λ_j do *k-labelset* Y_i correspondente. Uma decisão média é calculada para cada rótulo λ_j em L , e a decisão final é positiva para um dado rótulo se a decisão média for maior que um dado limiar t .

O propósito do método RAKEL é levar em consideração as correlações entre os rótulos e, ao mesmo tempo, evitar a desvantagem do método *Label Powerset* (LP), em que alguns rótulos podem terminar com poucos exemplos.

2.2 Abordagem Dependente de Algoritmo

Um trabalho que utiliza árvores de decisão foi proposto por Clare e King (2001). Nesse trabalho, os autores modificaram o algoritmo C4.5 (QUINLAN, 1993) para a classificação de proteínas de acordo com suas funções. O algoritmo C4.5 define os nós da árvore de decisão por meio de uma medida chamada entropia. Os autores modificaram a fórmula dessa medida, originalmente elaborada para problemas monorrótulo, de maneira a permitir seu uso em proble-

mas multirrótulo. Outra modificação feita pelos autores foi a utilização dos nós-folha da árvore para representar conjuntos de rótulos. Quando um nó-folha, alcançado na classificação de um exemplo, contém um conjunto de rótulos, uma regra separada é produzida para cada rótulo.

Em Zhang e Zhou (2005) foi proposto um método para classificação multirrótulo baseado no algoritmo KNN, chamado *Multi-Label K-Nearest Neighbors* (MLKNN). Nesse método, para cada exemplo, os rótulos associados com os K exemplos vizinhos mais próximos são recuperados, e é feita uma contagem dos vizinhos associados a cada rótulo. Então, o princípio *maximum a posteriori* é utilizado para definir o conjunto de rótulos de um novo exemplo.

No trabalho de Zhang e Zhou (2006) foi proposta uma medida de erro para ser utilizada no treinamento de redes neurais com o algoritmo *Backpropagation*. A medida considera os múltiplos rótulos dos exemplos no cálculo do erro de classificação.

Em Schapire e Singer (1999) e Schapire e Singer (2000) foram propostas duas extensões para o algoritmo *Adaboost* (FREUND; SCHAPIRE, 1995), de maneira a permitir seu uso em problemas multirrótulo. Na primeira, é feita uma modificação na maneira de se avaliar o desempenho preditivo do modelo induzido, verificando sua capacidade de predizer um conjunto correto de rótulos para um dado exemplo. Na segunda, uma mudança no algoritmo faz com que ele passe a predizer um *ranking* de rótulos para cada exemplo de entrada.

Em Thabtah *et al.* (2004), foi proposto um algoritmo multirrótulo baseado em regras de associação de classes (rótulos). O algoritmo foi chamado de Classificação Mssociativa Multiclasse Multirrótulo (MMAC, em inglês). Inicialmente é criado um conjunto de regras, e então são removidos todos os exemplos associados a esse conjunto. Os exemplos restantes são então utilizados para criar um novo conjunto de regras. Esse procedimento é realizado até que não restem mais exemplos.

Um algoritmo de classificação baseado em entropia foi utilizado por Zhu *et al.* (2005) para a tarefa de recuperação de informação. Os autores utilizaram o modelo para explorar correlações entre os rótulos de documentos multirrótulo.

Madjarov *et al.* (2012) publicaram um trabalho no qual vários métodos de classificação, baseados tanto na abordagem dependente quanto na independente de algoritmos, foram comparados. Várias medidas de avaliação também foram utilizadas nos experimentos. Os melhores desempenhos foram obtidos por métodos que tentam considerar as dependências entre os rótulos do problema durante as fase de treinamento.

2.3 Incorporação de Correlações

De acordo com Zhang e Zhou (2014), estratégias para levar em conta a dependência de rótulos podem ser divididas em: primeira ordem, segunda ordem e maior ordem. Estratégias de primeira ordem ignoram a existência de outros rótulos, sendo a transformação *Binary Relevance* (BR) (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010) um exemplo clássico. Estratégias de

segunda ordem consideram as relações entre pares de rótulos, enquanto as de maior ordem consideram relações envolvendo mais rótulos.

Com o passar dos anos, melhorias foram propostas para a transformação BR. [Read et al. \(2009\)](#) estenderam o BR criando uma cadeia de classificadores binários. O espaço de atributos de cada classificador é estendido com os rótulos anteriores na cadeia. Uma extensão probabilística ([DEMBCZYŃSKI; CHENG; HÜLLERMEIER, 2010](#)) foi proposta, capaz de estimar a distribuição conjunta dos rótulos, incluindo a noção da minimização de risco e a classificação ótima de Bayes.

Em [Huang e Zhou \(2012\)](#), exemplos foram agrupados e similaridades calculadas dentre cada agrupamento. Essas similaridades foram usadas para incrementar o vetor original de atributos.

[Yu, Pedrycz e Miao \(2014\)](#) usaram conceitos de conjuntos vizinhos irregulares. A ideia era encontrar possíveis rótulos relacionados para um dado exemplo, excluindo todos os não relacionados. [Spolaôr et al. \(2016\)](#) usaram correlações de rótulos emparelhados para construir novos rótulos binários para incrementar o vetor original de atributos.

Outros trabalhos exploram dados específicos de cada rótulo para determinar características próprias e incorporá-las no processo de classificação multirrótulo. [Huang et al. \(2016\)](#) propuseram o método LLSF para as tarefas de classificação e de seleção de atributos. Ainda, uma extensão desse método chamada LLSF-DL é responsável por aprender dependência entre rótulos incorporar correlação de segunda ordem e de ordem maior entre esses. Já no trabalho de [Zhang \(2011\)](#), o algoritmo LIFT identifica características específicas de cada rótulo a partir do agrupamento dos exemplos positivos e negativos.

Conceitos de *Head Label Data* (HLD) e *Tail Label Data* (TLD) são abordados no estudo de [Wei, Tu e Li \(2019\)](#). No trabalho, os autores propõem um método que emprega análises de agrupamento e pondera correlações entre os rótulos para explorar características específicas desses, e gerar aprendizado para incrementar o aprendizado de TLD.

METODOLOGIA

Este Capítulo apresenta os métodos, conjuntos de dados, algoritmos de classificação, medidas de avaliação e testes estatísticos utilizados para desenvolvimento do trabalho, obtenção e análise dos resultados.

3.1 Método de Correlação Proposto

Na Classificação Multirrótulo, há uma intersecção entre os conjuntos de rótulos (ou classes) em que os exemplos são classificados. Em razão disso, existe uma correlação entre esses rótulos e, conseqüentemente, uma correlação entre os exemplos que pertencem ao mesmo conjunto de rótulos. Para incorporar essa correlação na tarefa de classificação, foi adotada uma estratégia baseada nos vizinhos mais próximos de cada exemplo.

O método descrito nesta seção foi inicialmente objeto de estudo do autor deste trabalho de conclusão de curso, que resultou na publicação de um artigo na *International Joint Conference on Neural Networks (IJCNN)*. Nele, em vez de propor um novo método de classificação ou classes binárias, a incorporação de correlações nos conjuntos de dados se deu pelo uso de um vetor protótipo. Esse protótipo foi composto a partir do cálculo das distâncias Euclidianas entre os exemplos no espaço de atributos original e a média dos vetores de rótulos de cada exemplo no espaço de rótulos. Essa estratégia difere da maior parte dos trabalhos citados na Revisão Bibliográfica, que tentam incorporar correlações ao incluir rótulos específicos como atributos na etapa de treinamento, ou construir conjuntos de classes, transformando o problema multirrótulo original em um problema monorrótulo para incorporar correlações relacionadas a cada conjunto de rótulos. Na proposta aqui apresentada, tenta-se incorporar correlações ao considerar todos os rótulos simultaneamente, a partir da construção do vetor protótipo.

Na proposta, rótulos de um exemplo \mathbf{x}_i são representados por um vetor \mathbf{C}_i , onde a $j^{\text{ésima}}$ posição corresponde ao $j^{\text{ésimo}}$ rótulo. Se \mathbf{x}_i pertence ao rótulo c_j , $C_{i,j}$ recebe 1, e 0 caso contrário. Ainda, os atributos do exemplo são complementados com um vetor protótipo $\bar{\mathbf{v}}$, obtido a partir

da média do vetor de rótulos dos seus vizinhos mais próximos.

A Equação 3.1 exhibe o cálculo de $\bar{\mathbf{v}}$, onde \mathbf{S}_i é o conjunto contendo os k exemplos de treino mais próximos de \mathbf{x}_i , e $\mathbf{S}_{i,j}$ é o conjunto de exemplos em \mathbf{S}_i classificados na classe c_j .

$$\bar{\mathbf{v}}_{i,j} = \frac{|\mathbf{S}_{i,j}|}{|\mathbf{S}_i|} \quad (3.1)$$

Cada posição do vetor protótipo (Equação 3.1) pode ser interpretada como a proporção de exemplos de um conjunto \mathbf{S}_i que é classificada em uma dada classe c_j . Adicionalmente, pode ser interpretada como a probabilidade de um exemplo de \mathbf{S}_i pertencer a uma classe c_j . Dado um novo exemplo \mathbf{x} similar aos exemplos em \mathbf{S}_i (considerando apenas os atributos originais), os novos atributos (protótipo) representam as probabilidades de \mathbf{x} pertencer a qualquer uma das q classes c_j do problema. A intenção é que essa informação possa guiar melhor o algoritmo durante o treinamento. O procedimento de desenvolvimento dessa abordagem possui 8 passos principais, listados abaixo:

1. calcular distâncias Euclidianas, considerando apenas os atributos, entre todos os exemplos na partição de treino;
2. obter, para cada exemplo de treino, \mathbf{x}_i^{treino} , o conjunto \mathbf{K}_i dos seus k vizinhos mais próximos (medidos a partir da distância Euclidiana);
3. para cada conjunto \mathbf{K}_i , calcular o protótipo $\bar{\mathbf{v}}_i$ dos k vetores de classes atribuídos aos exemplos em \mathbf{K}_i ;
4. complementar cada vetor de atributos dos exemplos de treino \mathbf{x}_i^{treino} com o seu protótipo correspondente $\bar{\mathbf{v}}_i$;
5. para cada exemplo de teste \mathbf{x}_i^{teste} , calcular sua distância Euclidiana até todos os exemplos de treino considerando apenas os atributos originais;
6. obter, para cada exemplo de teste, \mathbf{x}_i^{teste} , o conjunto \mathbf{K}_i dos seus k vizinhos mais próximos no treino (medidos a partir da distância Euclidiana);
7. para cada conjunto \mathbf{K}_i , calcular o protótipo $\bar{\mathbf{v}}_i$ dos k vetores de classes associados aos exemplos de treino em \mathbf{K}_i ;
8. complementar cada vetor de atributos dos exemplos de teste \mathbf{x}_i^{teste} com o seu protótipo correspondente $\bar{\mathbf{v}}_i$;

Os Algoritmos 1 e 2 apresentam, respectivamente, o método para incorporar correlações nos dados de treinamento e o procedimento de classificar novos exemplos. Importante ressaltar que os rótulos dos exemplos de teste não são utilizadas para calcular o vetor protótipo. A Figura 3

ilustra o conjunto de dados de treino original e o processo para obter o conjunto de dados de treino com as correlações incorporadas.

Algoritmo 1: Algoritmo de Incorporação de Correlações

```

1 Função: obterCorrelaçõesDeTreino( $X$ )
   Entrada:  $X = [n, (A + q)]$ : conjunto de dados com  $n$  exemplos,  $A$  atributos e  $q$  classes
   Saída:  $X^* = [n, (A + q + q)]$ : conjunto de dados com  $n$  exemplos,  $A$  atributos +
           protótipo com  $q$  valores +  $q$  classes

2 Dada uma matriz  $n \times n$  de distâncias  $D = d_{i,j}$ 
3 para  $i \leftarrow 1$  até  $n$  faça
4   para  $j \leftarrow 1$  até  $n$  faça
5     // Distância Euclidiana considerando os  $A$  atributos em  $\mathbf{x}$ 
6      $d_{i,j} = \text{dist}(\mathbf{x}_{i,1..A}, \mathbf{x}_{j,1..A}) \mid \mathbf{x} \in X \text{ e } i < j$ 
7     // Obtendo o k-NN para cada exemplo
8      $K \leftarrow \mathbf{x}_i$   $k$  exemplos mais próximos
9     // Calculando o vetor protótipo
10     $\bar{\mathbf{v}}_i \leftarrow$  média do vetor de classes de  $K$ 
11    // Atributos originais + protótipos + classes
12     $\mathbf{x}_i^* \leftarrow \mathbf{x}_{i,1..A} + \bar{\mathbf{v}}_i + \mathbf{x}_{i,A+1..A+q}$ 
13  retorna  $\{X^*\}$ 

```

Algoritmo 2: Classificação de Novos Exemplos

```

1 Função: classificação( $\mathbf{X}^{\text{treino}}$ ,  $\mathbf{X}^{\text{teste}}$ )
   Entrada:  $\mathbf{X}^{\text{treino}} = [n, (A + q)]$ : conjunto de dados com  $n$  exemplos,  $A$  atributos e  $q$ 
           classes
            $\mathbf{X}^{\text{teste}} = [m, A]$ : conjunto de dados com  $n$  exemplos,  $A$  atributos
   Saída:  $P = [m, q]$ : matriz de predição com  $m$  linhas e  $q$  colunas

2 Dada uma matriz  $m \times n$  de distâncias  $D = d_{i,j}$ 
3 para  $i \leftarrow 1$  até  $m$  faça
4   para  $j \leftarrow 1$  até  $n$  faça
5     // Distância Euclidiana entre treino e teste considerando
6     apenas os  $A$  atributos
7      $d_{i,j} = \text{dist}(x_{i,1..A}^{\text{teste}}, x_{j,1..A}^{\text{treino}}) \mid x^{\text{treino}} \in \mathbf{X}^{\text{treino}} \text{ e } x^{\text{teste}} \in \mathbf{X}^{\text{teste}}$ 
8     // Obtendo os k-NN dos exemplos de teste
9      $\mathbf{K} \leftarrow x_i^{\text{teste}}$   $k$  exemplos mais próximos
10    // Calculando o vetor protótipo
11     $\bar{\mathbf{v}}_i \leftarrow$  média dos vetores de classes de  $K$ 
12    // Atributos originais + atributos de correlação
13     $\mathbf{x}_i^{\text{teste}*} \leftarrow \mathbf{x}_{i,1..A}^{\text{teste}} + \bar{\mathbf{v}}_i$ 
14   $X^{\text{treino}*} \leftarrow$  obterCorrelaçõesDeTreino( $X^{\text{treino}}$ )
15 modelo  $\leftarrow$  Indução(Algoritmo,  $X^{\text{treino}*}$ )
16  $P \leftarrow$  Prever(modelo,  $X^{\text{teste}*}$ )
17 retorna  $\{P\}$ 

```

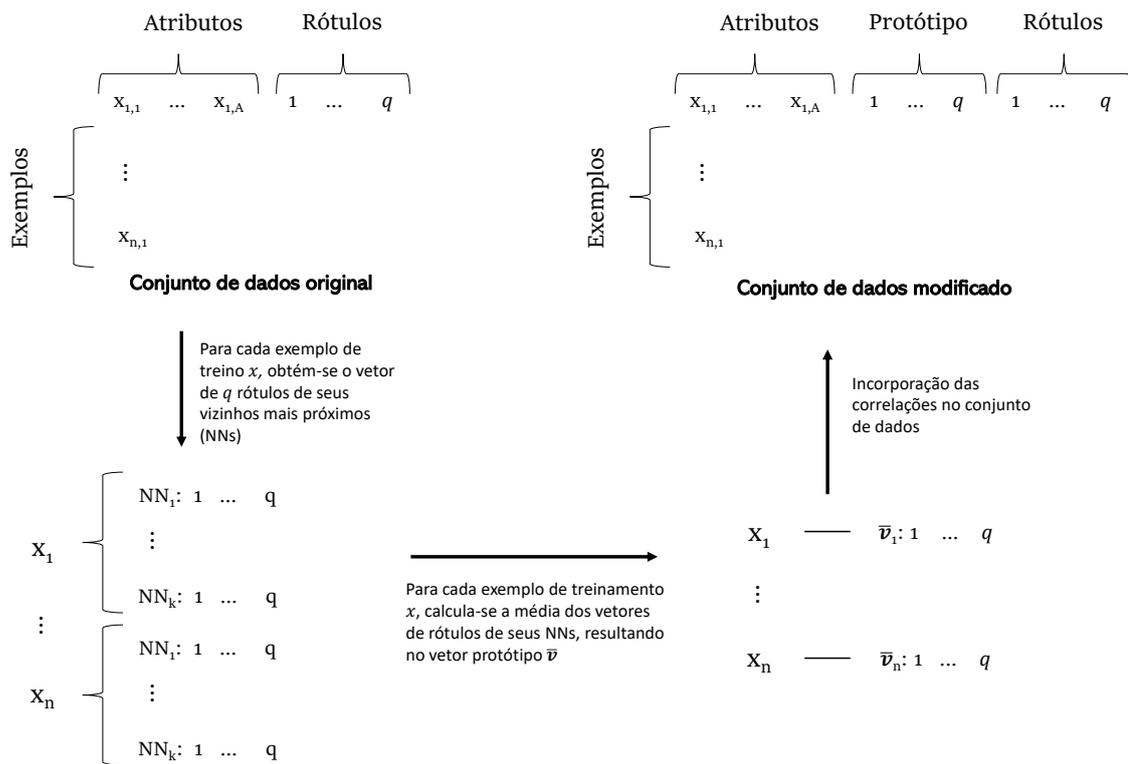


Figura 3 – Inclusão do vetor de correlações (protótipo)

3.2 Conjuntos de Dados

Os conjuntos de dados utilizados neste trabalho estão disponíveis gratuitamente¹. Foram escolhidos sete conjuntos representativos de diferentes domínios: áudio, imagens, música e bioinformática. A Tabela 1 exibe as suas principais características:

- Domínio
- Quantidade de exemplos
- Quantidade de atributos nominais
- Quantidade de atributos numéricos
- Cardinalidade de Rótulos (CR)
- Densidade de Rótulos (DR)
- Quantidade de conjuntos distintos de rótulos

¹ <<http://mulan.sourceforge.net/datasets-mlc.html>>

Cardinalidade de Rótulos (CR) é a quantidade média de rótulos por exemplo, enquanto que a Densidade de Rótulos (DR) é CR dividido pelo número total de rótulos. Nas Equações 3.2 e 3.3, m representa a quantidade de exemplos e q o total de rótulos.

Analisar a Cardinalidade e Densidade é importante para entender o comportamento dos algoritmos de classificação. Enquanto CR não considera o número de rótulos, DR considera. A Cardinalidade pode ser utilizada para estimar a quantidade de rótulos alternativos assinalados a um exemplo. Dois conjuntos de dados podem possuir o mesmo CR, mas DR distintos, fazendo com que o mesmo classificador se comporte de maneira diferente. A quantidade de conjuntos de rótulos também é importante, influenciando significativamente métodos que operam em conjuntos de rótulos (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010).

$$CR = \frac{1}{m} \sum_{i=1}^m |Y_i| \quad (3.2)$$

$$DR = \frac{1}{m} \sum_{i=1}^m \frac{|Y_i|}{q} \quad (3.3)$$

Tabela 1 – Características dos conjuntos de dados

Nome	Domínio	# Exemplos	# Nominais	# Numéricos	# Classes	Cardinalidade	Densidade	# Conj.
<i>birds</i>	áudio	645	2	258	19	1.014	0.053	133
<i>cal500</i>	música	502	0	68	174	26.044	0.150	502
<i>emotions</i>	música	593	0	72	6	1.869	0.311	27
<i>flags</i>	imagens	194	9	10	7	3.392	0.485	54
<i>genbase</i>	biologia	662	1186	0	27	1.252	0.046	32
<i>scene</i>	imagens	2407	0	294	6	1.074	0.179	15
<i>yeast</i>	biologia	2417	0	103	14	4.237	0.303	198

3.3 Algoritmos de Classificação

Foram utilizados diversos algoritmos de Aprendizado de Máquina nos experimentos. A intenção era verificar se os métodos propostos poderiam ou não melhorar os resultados de classificação. Ainda, as estratégias foram comparadas com o *Classifier Chains* (CC) (READ *et al.*, 2009), um dos métodos estado-da-arte de incorporação de correlações que utiliza *Binary Relevance*. Essa comparação fornece um *benchmark* relevante de um método da literatura que considera as correlações entre os rótulos no seu processo de aprendizado. Todos os métodos estão implementados no *Mulan* (TSOUMAKAS *et al.*, 2011), um pacote Java para aprendizado multirrótulo. Os classificadores utilizados estão listados abaixo.

- Algoritmos Clássicos monorrótulo: *Naive Bayes* (NB) (LANGLEY; IBA AND; THOMPSON, 1992), *Support Vector Machine* (SVM) (VAPNIK, 1999), árvore de decisão J48 (QUINLAN, 1993) e *K-Nearest Neighbors* (KNN) (AHA; KIBLER; ALBERT, 1991). Todos esses algoritmos foram utilizados com as transformações *Binary Relevance* (BR) e *Label Powerset* (LP);

- *Back-Propagation Multi-Label Learning* (BPMLL) (ZHANG; ZHOU, 2006): um método baseado em redes neurais;
- *Multi-Label k-Nearest Neighbours* (MLkNN) (ZHANG; ZHOU, 2007): um método baseado no KNN;
- *Classifier Chains* (CC) (READ *et al.*, 2009): incorpora correlações utilizando a transformação *Binary Relevance*.

Quanto aos valores de hiperparâmetros, foram utilizados os valores previamente estabelecidos pelo pacote Mulan para os algoritmos BPMLL, ML-kNN e CC. Para os algoritmos de classificação monorrótulo, o Mulan usa os algoritmos fornecidos pelo Weka (HALL *et al.*, 2009). A Tabela 2 exibe os valores de hiperparâmetros utilizados em cada algoritmo.

Tabela 2 – Valores de parâmetros utilizados

Algoritmo	Valores dos Parâmetros
<i>Naive Bayes</i>	Estima as probabilidades para os atributos numéricos usando distribuição normal
SVM	Constante de complexidade = 1.0, Normalização dos dados, Kernel polinomial, Epsilon = $1.0 \times e^{-12}$
J48	Grau de poda = 0.25, Mínimo de exemplos por folha = 2, Não apenas divisões binárias
<i>K-Nearest Neighbours</i>	k = 1, Algoritmo de busca de vizinhos mais próximos = linear, usa distância Euclidiana
BPMLL	Sem camadas escondidas, Normalizar dados, Taxa de aprendizado = 0.05, Regularização do decaimento de peso = 0.00001, Épocas de treinamento = 100
MLkNN	k = 1, Algoritmo de busca de vizinhos mais próximos = linear, usa distância Euclidiana
<i>Classifier Chains</i>	Inicia com uma ordem aleatória para os classificadores binários

3.4 Medidas de Avaliação

A avaliação de classificadores multirrótulo requer medidas diferentes das utilizadas em problemas de classificação monorrótulo. Diferente da classificação monorrótulo, na qual um exemplo é classificado de maneira errada ou correta, na classificação multirrótulo, um exemplo pode ser classificado de maneira parcialmente errada ou parcialmente correta. Esses casos acontecem quando um classificador atribui corretamente a um exemplo pelo menos uma das classes a que ele pertence, mas também não atribui ao exemplo uma ou mais classes as quais ele pertence. Pode acontecer também de o classificador atribuir a um exemplo uma ou mais classes as quais ele não pertence.

Seja Z_i o conjunto de rótulos previstos para um exemplo x_i , Y_i o conjunto de rótulos verdadeiros e S o conjunto de exemplos. Duas medidas amplamente utilizadas são *Precision* e *Recall* (TSOUMAKAS; KATAKIS; VLAHAVAS, 2010), apresentadas nas Equações 3.4 e 3.5.

$$P = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|Y_i \cap Z_i|}{|Z_i|} \quad (3.4)$$

$$R = \frac{1}{|S|} \sum_{i=1}^{|S|} \frac{|Y_i \cap Z_i|}{|Y_i|} \quad (3.5)$$

Para efeitos de simplificação, neste trabalho a medida reportada é a *FMeasure*, que é a média harmônica entre *Precision* e *Recall*, apresentada na Equação 3.6. Na equação Z_i representa o conjunto de classes preditas, e Y_i representa o conjunto de classes verdadeiras.

$$Fmeasure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3.6)$$

A avaliação foi realizada utilizando 10 partições estratificadas via a estratégia de *Cross Validation*. Portanto, os Algoritmos 1 e 2 foram executados 10 vezes. Para dividir os conjuntos de dados em partições de treino e teste, dois métodos de estratificação propostos especificamente para dados multirrótulo (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011) foram utilizados: *Labelset* e Iterativo. O primeiro considera a existência de grupos disjuntos em uma população para produzir amostrar nas quais as proporções desses grupos são mantidas. Por sua vez, o segundo calcula o número de exemplos desejados em cada subconjunto. Então, cada exemplo é assinalada a um subconjunto iterativamente de forma que haja uma distribuição apropriada ao final do processo. Ambos métodos de estratificação foram implementados com uso do pacote `utiml`² da linguagem R (RIVOLLI, 2016).

3.5 Testes Estatísticos

Para avaliar a significância estatística dos resultados a partir da obtenção do *p-value*, foi aplicado o teste de Friedman (DEMŠAR, 2006) considerando a *FMeasure*. Em seguida, o teste *post-hoc* de Nemenyi foi executado para identificar quais pares de comparação possuem diferenças estatisticamente significantes. Os testes foram executados separadamente para cada método de estratificação. A interpretação dos resultados obtidos é apresentada no final do Capítulo 4.

² <<https://cran.r-project.org/web/packages/utiml/index.html>>

RESULTADOS E DISCUSSÃO

As Tabelas 3 a 12 exibem os resultados *FMeasure* no formato *média ± desvio padrão* para os conjuntos originais (Original) e os modificados (Modificado). Os casos em que a metodologia melhorou os resultados estão destacados em negrito.

Tabela 3 – Resultados do *Naive Bayes* usando a transformação BR

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.20 ± 0.10.	0.20 ± 0.10	birds	0.20 ± 0.06	0.20 ± 0.06
cal500	0.33 ± 0.04	0.33 ± 0.04	cal500	0.33 ± 0.03	0.33 ± 0.04
emotions	0.63 ± 0.08	0.64 ± 0.08	emotions	0.63 ± 0.09	0.64 ± 0.08
flags	0.60 ± 0.14	0.56 ± 0.18	flags	0.63 ± 0.17	0.55 ± 0.18
genbase	0.30 ± 0.06	0.96 ± 0.04	genbase	0.30 ± 0.06	0.96 ± 0.04
scene	0.57 ± 0.03	0.58 ± 0.03	scene	0.57 ± 0.07	0.59 ± 0.05
yeast	0.54 ± 0.06	0.53 ± 0.05	yeast	0.54 ± 0.04	0.53 ± 0.06

Tabela 4 – Resultados do SVM usando a transformação BR

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.67 ± 0.08	0.67 ± 0.06	birds	0.66 ± 0.10	0.67 ± 0.12
cal500	0.34 ± 0.03	0.36 ± 0.03	cal500	0.34 ± 0.07	0.36 ± 0.03
emotions	0.60 ± 0.14	0.65 ± 0.11	emotions	0.60 ± 0.11	0.61 ± 0.16
flags	0.70 ± 0.11	0.70 ± 0.16	flags	0.73 ± 0.11	0.71 ± 0.10
genbase	0.99 ± 0.02	0.99 ± 0.02	genbase	0.99 ± 0.02	0.99 ± 0.02
scene	0.62 ± 0.07	0.71 ± 0.08	scene	0.62 ± 0.07	0.70 ± 0.07
yeast	0.61 ± 0.04	0.63 ± 0.03	yeast	0.61 ± 0.03	0.63 ± 0.04

No geral, os resultados dos algoritmos de classificação para os conjuntos de dados Modificados foram melhores ou competitivos comparados com os Originais. Para o algoritmo *Naive Bayes*, especialmente com a transformação LP, os resultados foram melhores na maior parte dos casos (Tabelas 3 e 7). Tal conclusão é especialmente promissora, uma vez que o algoritmo *Naive Bayes* considera que todos atributos são independentes. Esse é um indicativo que a correlação entre exemplos foi incorporada com sucesso, alcançando seu objetivo primário de auxiliar o processo de aprendizado. Se considerarmos a transformação *Binary Relevance*, que

Tabela 5 – Resultados do J48 usando a transformação BR

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.61 ± 0.05	0.56 ± 0.18	birds	0.61 ± 0.09	0.52 ± 0.27
cal500	0.35 ± 0.06	0.34 ± 0.03	cal500	0.34 ± 0.07	0.35 ± 0.04
emotions	0.55 ± 0.07	0.58 ± 0.16	emotions	0.55 ± 0.08	0.55 ± 0.09
flags	0.68 ± 0.14	0.69 ± 0.14	flags	0.70 ± 0.13	0.71 ± 0.15
genbase	0.99 ± 0.03	0.99 ± 0.03	genbase	0.99 ± 0.03	0.99 ± 0.03
scene	0.57 ± 0.04	0.66 ± 0.09	scene	0.56 ± 0.04	0.67 ± 0.07
yeast	0.56 ± 0.05	0.55 ± 0.03	yeast	0.56 ± 0.06	0.54 ± 0.06

Tabela 6 – Resultados do KNN usando a transformação BR

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.65 ± 0.12	0.66 ± 0.09	birds	0.65 ± 0.10	0.64 ± 0.10
cal500	0.34 ± 0.06	0.34 ± 0.06	cal500	0.34 ± 0.05	0.34 ± 0.04
emotions	0.60 ± 0.11	0.61 ± 0.10	emotions	0.60 ± 0.06	0.60 ± 0.12
flags	0.64 ± 0.19	0.66 ± 0.18	flags	0.65 ± 0.16	0.66 ± 0.17
genbase	0.99 ± 0.02	0.99 ± 0.02	genbase	0.99 ± 0.02	0.99 ± 0.02
scene	0.69 ± 0.10	0.70 ± 0.09	scene	0.70 ± 0.05	0.71 ± 0.05
yeast	0.57 ± 0.07	0.57 ± 0.05	yeast	0.57 ± 0.07	0.58 ± 0.07

Tabela 7 – Resultados do *Naive Bayes* usando a transformação LP

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.23 ± 0.10	0.23 ± 0.10	birds	0.23 ± 0.14	0.23 ± 0.12
cal500	0.33 ± 0.04	0.33 ± 0.03	cal500	0.33 ± 0.06	0.34 ± 0.05
emotions	0.60 ± 0.13	0.61 ± 0.13	emotions	0.60 ± 0.10	0.60 ± 0.12
flags	0.66 ± 0.14	0.67 ± 0.16	flags	0.65 ± 0.14	0.66 ± 0.12
genbase	0.40 ± 0.08	0.96 ± 0.03	genbase	0.40 ± 0.11	0.96 ± 0.04
scene	0.64 ± 0.08	0.67 ± 0.04	scene	0.64 ± 0.08	0.66 ± 0.09
yeast	0.57 ± 0.06	0.58 ± 0.05	yeast	0.57 ± 0.03	0.58 ± 0.04

ignora completamente correlações, o método proposto foi capaz de incorporar as correlações de forma assertiva e melhorar o desempenho dos classificadores em vários casos.

Os resultados com o SVM também foram interessantes, principalmente considerando a transformação BR. A matriz de correlações inserida levou a resultados melhores na maioria das comparações. Em particular, nos conjuntos de dados com maior quantidade de classes (*birds*, *cal500*, *genbase* e *yeast*), os resultados foram sempre melhores ou equivalentes. Entre esses 4 conjuntos de dados, *birds*, *cal500* e *yeast* possuem o maior número de conjuntos de classes, levando ao treinamento de vários classificadores binários do SVM na transformação LP. Por conta disso, quando a estratificação foi realizada pela estratégia *Labelset*, nota-se que os resultados tiveram melhoras na maioria dos casos.

No que diz respeito aos algoritmos J48 e KNN, o desempenho da classificação foi aprimorado na maior parte dos casos. O KNN teve desempenho melhor que o J48 e, salvo algumas exceções, os mesmos conjuntos de dados tiveram melhora de resultados com o método proposto. Curiosamente, os resultados do KNN foram sempre os mesmos tanto para a transformação BR quanto para a transformação LP.

Tabela 8 – Resultados do SVM usando a transformação LP

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.68 ± 0.08	0.69 ± 0.10	birds	0.68 ± 0.09	0.68 ± 0.09
cal500	0.34 ± 0.07	0.34 ± 0.06	cal500	0.34 ± 0.05	0.34 ± 0.04
emotions	0.66 ± 0.10	0.68 ± 0.10	emotions	0.67 ± 0.12	0.66 ± 0.11
flags	0.68 ± 0.05	0.67 ± 0.11	flags	0.67 ± 0.09	0.67 ± 0.10
genbase	0.99 ± 0.02	0.99 ± 0.02	genbase	0.99 ± 0.03	0.99 ± 0.03
scene	0.76 ± 0.05	0.77 ± 0.05	scene	0.75 ± 0.06	0.76 ± 0.07
yeast	0.62 ± 0.06	0.63 ± 0.04	yeast	0.62 ± 0.04	0.63 ± 0.05

Tabela 9 – Resultados do J48 usando a transformação LP

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.61 ± 0.05	0.53 ± 0.26	birds	0.61 ± 0.09	0.58 ± 0.14
cal500	0.33 ± 0.04	0.33 ± 0.05	cal500	0.33 ± 0.05	0.32 ± 0.03
emotions	0.53 ± 0.12	0.57 ± 0.14	emotions	0.55 ± 0.14	0.56 ± 0.12
flags	0.64 ± 0.14	0.66 ± 0.10	flags	0.66 ± 0.15	0.68 ± 0.16
genbase	0.99 ± 0.03	0.99 ± 0.02	genbase	0.99 ± 0.04	0.99 ± 0.03
scene	0.51 ± 0.05	0.67 ± 0.09	scene	0.51 ± 0.06	0.69 ± 0.09
yeast	0.50 ± 0.05	0.51 ± 0.06	yeast	0.51 ± 0.06	0.52 ± 0.05

Tabela 10 – Resultados do KNN usando a transformação LP

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.65 ± 0.12	0.66 ± 0.09	birds	0.65 ± 0.10	0.64 ± 0.10
cal500	0.34 ± 0.06	0.34 ± 0.06	cal500	0.34 ± 0.05	0.34 ± 0.04
emotions	0.60 ± 0.11	0.61 ± 0.10	emotions	0.60 ± 0.08	0.60 ± 0.12
flags	0.64 ± 0.19	0.66 ± 0.18	flags	0.65 ± 0.15	0.66 ± 0.16
genbase	0.99 ± 0.01	0.99 ± 0.02	genbase	0.99 ± 0.02	0.99 ± 0.02
scene	0.69 ± 0.02	0.70 ± 0.09	scene	0.70 ± 0.05	0.71 ± 0.05
yeast	0.57 ± 0.05	0.57 ± 0.05	yeast	0.57 ± 0.07	0.58 ± 0.07

Outro fator de análise interessante é a quantidade de classes dos conjuntos de dados explorados. Na maioria das comparações efetuadas, o desempenho dos algoritmos baseados em métodos de transformação melhorou para conjuntos de dados com poucas classes. Tal fato ocorre porque os exemplos são mais correlacionados quando existem menos classes associadas a um número menor de conjuntos de classes. Isso indica que muitos exemplos são classificados em grupos mais similares de classes e, nesse caso, é mais fácil obter informações acerca da correlação entre os exemplos (vetor protótipo de médias de classes). Esses fatores também podem ser o motivo de todos os métodos terem desempenhos tão bons no conjunto de dados *genbase*, que possui quantidades semelhantes de classes e de conjuntos distintos de classes (27 e 32, respectivamente). Considerando os métodos de estratificação, os resultados demonstram que eles não tiveram grande efeito nos algoritmos de classificação, visto que as medidas obtidas foram bem similares para quase todos os classificadores. No entanto, em uma análise experimental mais minuciosa, uma pequena diferença pode ser notada. A estratégia de estratificação *Labelset* levou a resultados ligeiramente melhores do que a Iterativa para os conjuntos de dados com uma razão *labelsets/exemplos* baixa (≤ 0.1) (SECHIDIS; TSOUMAKAS; VLAHAVAS, 2011). Essa evidência pode ser observada nos resultados do conjunto de dados *emotions*.

Tabela 11 – Resultados do *Back-Propagation Multi-Label Learning* (BPMLL)

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.45 ± 0.07	0.46 ± 0.04	birds	0.44 ± 0.12	0.45 ± 0.06
cal500	0.46 ± 0.04	0.44 ± 0.04	cal500	0.45 ± 0.03	0.45 ± 0.04
emotions	0.62 ± 0.15	0.65 ± 0.12	emotions	0.66 ± 0.10	0.66 ± 0.08
flags	0.68 ± 0.17	0.53 ± 0.71	flags	0.70 ± 0.10	0.55 ± 0.74
genbase	0.05 ± 0.06	0.05 ± 0.12	genbase	0.06 ± 0.06	0.07 ± 0.04
scene	0.50 ± 0.11	0.50 ± 0.22	scene	0.49 ± 0.11	0.49 ± 0.16
yeast	0.63 ± 0.05	0.63 ± 0.04	yeast	0.63 ± 0.06	0.63 ± 0.05

Tabela 12 – Resultados do *Multi-Label k-Nearest Neighbours* (MLkNN)

Conjunto de Dados	Estratificação <i>Labelset</i>		Conjunto de Dados	Estratificação Iterativa	
	Original	Modificado		Original	Modificado
birds	0.56 ± 0.06	0.56 ± 0.10	birds	0.58 ± 0.09	0.56 ± 0.09
cal500	0.33 ± 0.04	0.32 ± 0.03	cal500	0.32 ± 0.06	0.32 ± 0.04
emotions	0.65 ± 0.06	0.63 ± 0.11	emotions	0.63 ± 0.16	0.60 ± 0.15
flags	0.71 ± 0.09	0.72 ± 0.12	flags	0.73 ± 0.11	0.72 ± 0.09
genbase	0.96 ± 0.05	0.96 ± 0.03	genbase	0.96 ± 0.05	0.96 ± 0.04
scene	0.69 ± 0.12	0.70 ± 0.09	scene	0.69 ± 0.06	0.69 ± 0.04
yeast	0.62 ± 0.04	0.63 ± 0.03	yeast	0.62 ± 0.05	0.62 ± 0.04

Adicionalmente, os resultados exibidos neste capítulo comprovam que a estratégia de incorporação de correlação utilizada foi mais efetiva quando usada em conjunto com as transformações BR e LP do que com os algoritmos BPMLL e ML-kNN para a maioria dos conjuntos de dados. Tal fato expõe que pode ser mais fácil usar correlações entre exemplos quando se considera as classes individualmente para induzir os classificadores, em vez de considerá-las em conjunto, como os algoritmos que não fazem transformação do problema fazem.

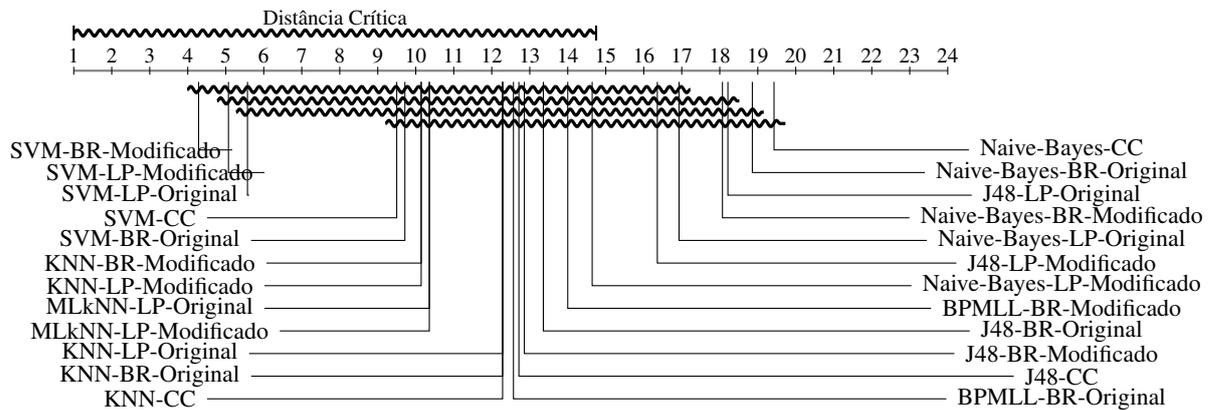
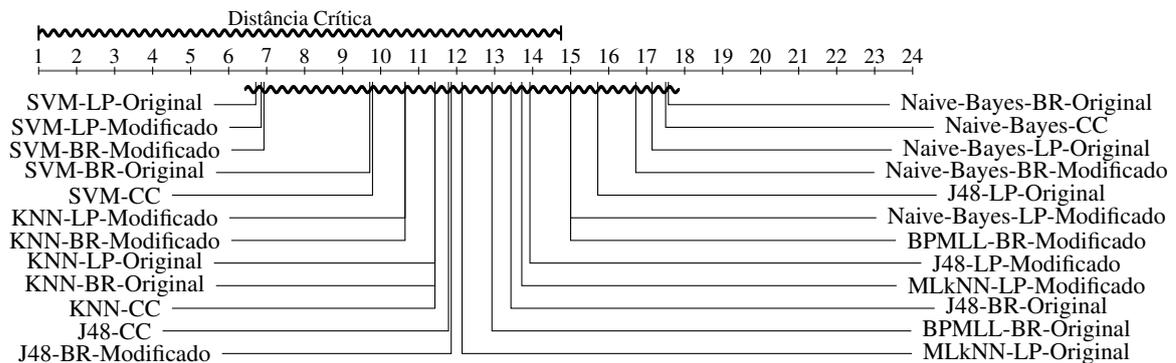
A Tabela 13 mostra os resultados obtidos para o algoritmo *Classifier Chains* (CC), um dos métodos estado-da-arte para incorporar correlações na classificação multirrotulo usando a *Binary Relevance*. Como o CC já incorpora correlações durante o treinamento, foi avaliado apenas com os conjuntos de dados originais e os resultados obtidos serviram de *benchmark* para a solução proposta neste estudo. Ao analisar os resultados do *Classifier Chains* em comparação aos resultados obtidos pelo método proposto com os mesmos algoritmos de classificação e também com uso da transformação BR, nota-se que a estratégia de incorporação dos vetores protótipos resultou em um desempenho ligeiramente superior na maioria dos casos. Diferentemente dos demais algoritmos, nos quais os resultados superiores nos conjuntos de dados modificados foram destacados, a Tabela 13 apresenta destacados em negrito os resultados em que o *Classifier Chains* foi superado pelas medidas obtidas para os conjuntos de dados modificados para cada algoritmo que utilizou a transformação BR.

Para avaliar a significância estatística dos resultados, foi aplicado o teste estatístico de Friedman (DEMÊSAR, 2006) considerando a medida FMeasure. Dois conjuntos de testes foram executados para comparar os métodos de estratificação *Labelset* e Iterativa. Os *p-values* obtidos foram, respectivamente 1.10×10^{-5} e 0.047. Claramente, os resultados do método *Labelset* possuem significância estatística, mas não foram encontradas diferenças estatística-

Tabela 13 – Resultados do *Classifier Chains*

	Estratificação <i>Labelset</i>				Estratificação Iterativa			
	NB	SVM	J48	KNN	NB	SVM	J48	KNN
birds	0.20 ± 0.09	0.68 ± 0.07	0.62 ± 0.10	0.65 ± 0.12	0.20 ± 0.06	0.66 ± 0.14	0.61 ± 0.12	0.65 ± 0.10
cal500	0.29 ± 0.05	0.30 ± 0.05	0.36 ± 0.07	0.34 ± 0.06	0.29 ± 0.05	0.30 ± 0.07	0.36 ± 0.05	0.34 ± 0.05
emotions	0.63 ± 0.08	0.61 ± 0.12	0.57 ± 0.13	0.60 ± 0.11	0.63 ± 0.10	0.61 ± 0.13	0.56 ± 0.14	0.60 ± 0.08
flags	0.64 ± 0.17	0.68 ± 0.15	0.68 ± 0.14	0.64 ± 0.19	0.66 ± 0.15	0.70 ± 0.14	0.69 ± 0.13	0.65 ± 0.19
genbase	0.31 ± 0.07	0.99 ± 0.02	0.99 ± 0.03	0.99 ± 0.01	0.30 ± 0.07	0.99 ± 0.02	0.99 ± 0.03	0.99 ± 0.02
scene	0.57 ± 0.03	0.71 ± 0.07	0.63 ± 0.08	0.69 ± 0.09	0.57 ± 0.06	0.71 ± 0.07	0.61 ± 0.06	0.70 ± 0.05
yeast	0.53 ± 0.05	0.59 ± 0.03	0.53 ± 0.06	0.57 ± 0.05	0.53 ± 0.04	0.59 ± 0.03	0.52 ± 0.05	0.57 ± 0.07

mente significantes quando o método Iterativo foi avaliado. Em seguida, o teste *post-hoc* de Nemenyi foi conduzido para identificar quais comparações entre pares apresentaram diferenças estatisticamente significantes. Os diagramas críticos do teste são apresentados nas Figuras 4 e 5. Os métodos conectados são aqueles que não apresentaram resultados com diferenças estatisticamente significantes.

Figura 4 – Diagrama Crítico para o teste estatístico *post-hoc* de Nemenyi - Estratificação *Labelset*Figura 5 – Diagrama Crítico para o teste estatístico *post-hoc* de Nemenyi - Estratificação Iterativa

Uma consequência interessante dos resultados da Figura 4 é que com a estratificação *Labelset*, todos os métodos obtiveram melhores posições na média quando as correlações foram incorporadas, apenas com exceção dos métodos dependentes de algoritmos. A comparação dos resultados dos conjuntos de dados modificados para algoritmos que utilizaram a transformação

BR com o *Classifier Chains* também demonstra que a metodologia proposta obteve resultados melhores no geral, salvo a exceção da árvore de decisão J48.

Quando a estratificação Iterativa foi utilizada, a incorporação de correlações novamente levou a *rankings* superiores na média para todos os métodos, com exceção daqueles dependentes de algoritmos e do SVM com a transformação *Label-Powerset*. Portanto, isso reforça a conclusão que a incorporação de correlações teve uma influência mais forte nos métodos baseados em transformações (BR e LP), que consideram as classes de forma individual para induzir os classificadores. Quando comparados com o *Classifier Chains*, novamente é possível observar um desempenho geral melhor obtido pela estratégia de incorporação, sendo o classificador J48, mais uma vez, a única exceção.

CONCLUSÃO E TRABALHOS FUTUROS

Neste trabalho, foi proposto um método para incorporar correlações entre exemplos na classificação multirrótulo utilizando o espaço de rótulos. Os conjuntos de dados originais foram transformados antes do treinamento dos classificadores. As correlações entre os exemplos foram feitas considerando os seus vizinhos mais próximos (calculados via distância Euclidiana). Os vetores de classes dos vizinhos mais próximos de cada exemplo tiveram suas médias computadas e transformadas em um novo vetor (protótipo) que foi incorporado ao conjunto de dados original como complemento dos vetores de atributos originais. Todos os conjuntos de dados utilizados, tanto na versão original quanto na modificada, foram submetidos a diversos classificadores multirrótulo e avaliados a partir de medidas especialmente propostas para problemas dessa natureza.

Os resultados demonstraram que o método proposto melhorou o desempenho dos classificadores em vários conjuntos de dados. A incorporação das correlações teve uma influência mais forte nos métodos independentes de algoritmo. Considerando a transformação *Binary Relevance* e o algoritmo *Naive Bayes*, que originalmente ignora completamente as correlações, foi observado que a estratégia teve sucesso em incorporar as correlações, resultando em métricas iguais ou superiores em aproximadamente 60% dos pontos de verificação. Ainda, o método proposto teve resultados iguais ou superiores ao *Classifier Chains*, um dos métodos estado-da-arte, em mais de 70% dos casos comparáveis.

Adicionalmente, foi possível notar que os métodos de estratificação não tiveram grande influência na tarefa de classificação, mas a estratégia *Labelset* teve resultados ligeiramente superiores ao método Iterativo, o que pode ser explicado pela existência de conjuntos de dados com razão baixa entre a quantidade de *labelsets* e de exemplos.

Como trabalhos futuros, existem alguns caminhos que podem ser seguidos para superar as limitações deste estudo. Como forma de expandir as análises já realizadas, o método pode ser testado em mais conjuntos de dados, com algoritmos e medidas de avaliação adicionais.

Ainda, para evoluir o método proposto, as correlações podem ser incorporadas também com outras métricas de distância, como Mahalanobis, e o número de vizinhos mais próximos pode ser variado de acordo com as características de cada conjunto de dados.

Por fim, métodos distintos de incorporação de correlações podem ser desenvolvidos e submetidos a mesma metodologia de avaliação. Algumas possibilidades são: incluir cálculo de similaridade entre classes como extensão aos atributos dos conjuntos de dados e gerar uma matriz de probabilidades Bayesianas de um exemplo pertencer a uma classe dado que ele pertença a outra.

REFERÊNCIAS

- ABREU, I. B. M. de; MANTOVANI, R. G.; CERRI, R. Incorporating instance correlations in multi-label classification via label-space. In: **2017 International Joint Conference on Neural Networks (IJCNN)**. [S.l.: s.n.], 2017. p. 581–588. Citado na página [22](#).
- AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. **Mach Learn**, v. 6, n. 1, p. 37–66, 1991. Citado na página [33](#).
- BOUTELL, M. R.; LUO, J.; SHEN, X.; BROWN, C. M. Learning multi-label scene classification. **Pattern Recognition**, v. 37, n. 9, p. 1757–1771, 2004. Citado 2 vezes nas páginas [20](#) e [26](#).
- CARVALHO, A.; FREITAS, A. A tutorial on multi-label classification techniques. In: _____. [S.l.]: Springer, 2009. (Studies in Computational Intelligence 205, v. 5), p. 177–195. ISBN 9783642015359. Citado 2 vezes nas páginas [9](#) e [21](#).
- CHERMAN, E. A. **Aprendizado de máquina multirrótulo: explorando a dependência de rótulos e o aprendizado ativo**. Tese (Doutorado) — University of São Paulo, 2013. Citado na página [20](#).
- CHERMAN, E. A.; METZ, J.; MONARD, M. C. Incorporating label dependency into the binary relevance framework for multi-label classification. **Expert Syst Appl**, Pergamon Press, Inc., Tarrytown, NY, USA, v. 39, n. 2, p. 1647–1655, 2012. ISSN 0957-4174. Citado 2 vezes nas páginas [22](#) e [26](#).
- CLARE, A.; KING, R. D. Knowledge discovery in multi-label phenotype data. In: **5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2001)**. [S.l.]: Springer, 2001. (LNAI, v. 2168), p. 42–53. Citado 2 vezes nas páginas [20](#) e [26](#).
- DEMBCZYŃSKI, K.; CHENG, W.; HÜLLERMEIER, E. Bayes optimal multilabel classification via probabilistic classifier chains. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2010. p. 279–286. Citado 2 vezes nas páginas [26](#) e [28](#).
- DEMBCZYŃSKI, K.; WAEGEMAN, W.; CHENG, W.; HÜLLERMEIER, E. On label dependence and loss minimization in multi-label classification. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 88, n. 1-2, p. 5–45, 2012. ISSN 0885-6125. Citado na página [22](#).
- DEMŠAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. **Journal of Machine Learning Research**, v. 7, p. 1–30, 2006. Citado 2 vezes nas páginas [35](#) e [40](#).
- ELISSEFF, A.; WESTON, J. Kernel Methods for Multi-labelled Classification and Categorical Regression Problems. In: **Advances in Neural Information Processing Systems**. [S.l.]: MIT Press, 2001. p. 681–687. Citado na página [20](#).
- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. In: **European Conference on Computational Learning Theory**. [S.l.: s.n.], 1995. p. 23–37. Citado na página [27](#).

GONÇALVES, T.; QUARESMA, P. A preliminary approach to the multilabel classification problem of portuguese juridical documents. In: **EPIA**. [S.l.: s.n.], 2003. p. 435–444. Citado na página 20.

HALL, M.; FRANK, E.; HOLMES, G.; PFAHRINGER, B.; REUTEMANN, P.; WITTEN, I. H. The WEKA data mining software: an update. **SIGKDD Explor. Newsl.**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, 2009. ISSN 1931-0145. Citado na página 34.

HUANG, J.; LI, G.; HUANG, Q.; WU, X. Learning label-specific features and class-dependent labels for multi-label classification. **IEEE Transactions on Knowledge and Data Engineering**, v. 28, n. 12, p. 3309–3323, 2016. Citado na página 28.

HUANG, S.-J.; ZHOU, Z.-H. Multi-label learning by exploiting label correlations locally. In: **AAAI Conference on Artificial Intelligence**. [S.l.: s.n.], 2012. p. 949–955. Citado na página 28.

KARALIC, A.; PIRNAT, V. Significance level based multiple tree classification. In: **Informatica**. [S.l.: s.n.], 1991. v. 15, p. 12. Citado na página 20.

LANGLEY, P.; IBA AND, W.; THOMPSON, K. An analysis of Bayesian classifiers. In: **National Conference on Artificial intelligence**. [S.l.: s.n.], 1992. p. 223–228. ISBN 0-262-51063-4. Citado na página 33.

LAUSER, B.; HOTHO, A. Automatic multi-label subject indexing in a multilingual environment. In: **Proc. of the 7th European Conference in Research and Advanced Technology for Digital Libraries, ECDL 2003**. [S.l.]: Springer, 2003. v. 2769, p. 140–151. Citado na página 20.

LUO, X.; ZINCIR-HEYWOOD, N. A. Evaluation of two systems on multi-class multi-label document classification. In: **International Symposium on Methodologies for Intelligent Systems**. [S.l.: s.n.], 2005. p. 161–169. Citado na página 20.

MADJAROV, G.; KOCEV, D.; GJORGJEVIKJ, D.; DZEROSKI, S. An extensive experimental comparison of methods for multi-label learning. **Pattern Recognition**, v. 45, n. 9, p. 3084–3104, 2012. Citado na página 27.

MITCHELL, T. **Machine Learning**. [S.l.]: McGraw Hill, 1997. Citado na página 19.

NILSSON, N. J. **The Quest for Artificial Intelligence**. 1st. ed. USA: Cambridge University Press, 2009. ISBN 0521122937. Citado na página 19.

QUINLAN, J. R. **C4.5: programs for machine learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Citado 3 vezes nas páginas 21, 26 e 33.

READ, J.; PFAHRINGER, B.; HOLMES, G.; FRANK, E. Classifier chains for multi-label classification. In: **European Conference on Machine Learning and Knowledge Discovery**. [S.l.: s.n.], 2009. p. 254–269. ISBN 978-3-642-04173-0. Citado 5 vezes nas páginas 22, 26, 28, 33 e 34.

RIVOLLI, A. **utiml: Utilities for Multi-Label Learning**. [S.l.], 2016. R package version 0.1.0. Disponível em: <<http://CRAN.R-project.org/package=utiml>>. Citado na página 35.

SCHAPIRE, R. E.; SINGER, Y. Improved Boosting Algorithms Using Confidence-rated Predictions. In: **Machine Learning**. Hingham, MA, USA: Kluwer Academic Publishers, 1999. v. 37, p. 297–336. ISSN 0885-6125. Citado na página [27](#).

_____. BoosTexter: a boosting-based system for text categorization. In: **Machine Learning**. Hingham, MA, USA: Kluwer Academic Publishers, 2000. v. 39, p. 135–168. Citado na página [27](#).

SECHIDIS, K.; TSOUMAKAS, G.; VLAHAVAS, I. On the stratification of multi-label data. In: **European Conference on Machine Learning and Knowledge Discovery**. [S.l.: s.n.], 2011. p. 145–158. ISBN 978-3-642-23808-6. Citado 2 vezes nas páginas [35](#) e [39](#).

SHEN, X.; BOUTELL, M.; LUO, J.; BROWN, C. Multilabel machine learning and its application to semantic scene classification. In: **Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series**. [S.l.: s.n.], 2003. v. 5307, p. 188–199. Citado na página [20](#).

SIMON, P. **Too Big to Ignore: The Business Case for Big Data**. 1st. ed. [S.l.]: Wiley Publishing, 2013. ISBN 1118638174. Citado na página [19](#).

SPOLAÔR, N.; MONARD, M. C.; TSOUMAKAS, G.; LEE, H. D. A systematic review of multi-label feature selection and a new method based on label construction. **Neurocomputing**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 180, n. C, p. 3–15, 2016. ISSN 0925-2312. Citado 2 vezes nas páginas [22](#) e [28](#).

STOJANOVA, D.; CECI, M.; MALERBA, D.; DZEROSKI, S. Using ppi network autocorrelation in hierarchical multi-label classification trees for gene function prediction. **BMC Bioinformatics**, v. 14, n. 1, p. 285, 2013. ISSN 1471-2105. Citado na página [22](#).

TANAKA, E. A.; NOZAWA, S. R.; MACEDO, A. A.; BARANAUSKAS, J. A. A multi-label approach using binary relevance and decision trees applied to functional genomics. **Journal of Biomedical Informatics**, p. 85–95, 2015. Citado na página [21](#).

THABTAH, F. A.; COWLING, P.; PENG, Y.; RASTOGI, R.; MORIK, K.; BRAMER, M.; WU, X. Mmac: A new multi-class, multi-label associative classification approach. In: **Fourth IEEE International Conference on Data Mining**. [S.l.: s.n.], 2004. p. 217–224. Citado na página [27](#).

TSOUMAKAS, G.; KATAKIS, I. Multi Label Classification: An Overview. **International Journal of Data Warehousing and Mining**, v. 3, n. 3, p. 1–13, 2007. Citado 2 vezes nas páginas [20](#) e [26](#).

TSOUMAKAS, G.; KATAKIS, I.; VLAHAVAS, I. P. Mining Multi-label Data. In: **Data Mining and Knowledge Discovery Handbook**. [S.l.]: Springer, 2010. p. 667–685. Citado 4 vezes nas páginas [25](#), [27](#), [33](#) e [34](#).

TSOUMAKAS, G.; SPYROMITROS-XIOUFIS, E.; VILCEK, J.; VLAHAVAS, I. Mulan: A java library for multi-label learning. **J Mach Learn Res**, v. 12, p. 2411–2414, 2011. Citado na página [33](#).

VAPNIK, V. N. **The Nature of Statistical Learning Theory**. [S.l.]: Springer-Verlag New York, Inc., 1999. Hardcover. ISBN 0387987800. Citado 2 vezes nas páginas [21](#) e [33](#).

WEI, T.; TU, W.-W.; LI, Y.-F. Learning for tail label data - a label-specific feature approach. **IJCAI**, p. 3842–3848, 2019. Citado na página [28](#).

YU, Y.; PEDRYCZ, W.; MIAO, D. Multi-label classification by exploiting label correlations. **Expert Syst. Appl.**, v. 41, n. 6, p. 2989–3004, 2014. ISSN 0957-4174. Citado na página 28.

ZHANG, M.-L. Lift: Multi-label learning with label-specific features. In: . [S.l.: s.n.], 2011. v. 37, p. 1609–1614. Citado na página 28.

ZHANG, M.-L.; ZHOU, Z.-H. A k-Nearest Neighbor Based Algorithm for Multi-label Classification. In: **IEEE International Conference on Granular Computing**. [S.l.: s.n.], 2005. v. 2, p. 718–721 Vol. 2. Citado 2 vezes nas páginas 20 e 27.

_____. Multilabel neural networks with applications to functional genomics and text categorization. **IEEE T Knowl Data En**, v. 18, p. 1338–1351, 2006. ISSN 1041-4347. Citado 3 vezes nas páginas 20, 27 e 34.

_____. ML-KNN: A lazy learning approach to multi-label learning. **Pattern Recogn**, New York, NY, USA, v. 40, p. 2038–2048, 2007. ISSN 0031-3203. Citado na página 34.

ZHANG, M. L.; ZHOU, Z. H. A review on multi-label learning algorithms. **IEEE T Knowl Data En**, v. 26, n. 8, p. 1819–1837, 2014. Citado na página 27.

ZHU, S.; JI, X.; XU, W.; GONG, Y. Multi-labelled classification using maximum entropy method. In: **International conference on research and development in information retrieval**. New York, NY, USA: ACM, 2005. p. 274–281. Citado na página 27.