LEANDRO SOUZA COSTA

MINERAÇÃO DE PADRÕES FREQÜENTES ORTOGONAIS E SUA APLICAÇÃO EM CLASSIFICAÇÃO ASSOCIATIVA

Belo Horizonte
16 de abril de 2008

LEANDRO SOUZA COSTA
Orientador: Wagner Meira Jr.

MINERAÇÃO DE PADRÕES FREQÜENTES ORTOGONAIS E SUA APLICAÇÃO EM CLASSIFICAÇÃO ASSOCIATIVA

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte
16 de abril de 2008

UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Mineração de Padrões Freqüentes Ortogonais e sua Aplicação em Classificação Associativa

LEANDRO SOUZA COSTA

Dissertação defendida e aprovada pela banca examinadora constituída por:

Ph. D. Wagner Meira Jr. – Orientador
Universidade Federal de Minas Gerais

Ph. D. Marcos André Gonçalves
Universidade Federal de Minas Gerais

Ph. D. Sandra Aparecida de Amo
Universidade Federal de Uberlândia

Belo Horizonte, 16 de abril de 2008

Aos meus pais e irmãos.

Agradecimentos

Aos amigos, pela apoio; à Giselle, pela compreensão; à ATI, pela oportunidade; ao Adriano, pela colaboração; aos professores Wagner Meira Jr. e Mohammed Zaki, pelo conhecimento, orientação e incentivo.

Resumo

Mineração de padrões freqüentes é um dos temas mais explorados da mineração de dados, assumindo um papel essencial em inúmeras tarefas que possuem, como objetivo, encontrar padrões de determinado interesse numa base. Entretanto, grande parte das soluções propostas nesta linha de pesquisa ainda possui problemas não solucionados, sendo muitos deles relacionados com a explosão do número de padrões freqüentes encontrados na base de dados. Isto acontece pelo fato dos padrões freqüentes obedecerem à propriedade da antimonotonia, que diz que, se um padrão é freqüente, todos os seus sub-padrões também o serão. Como conseqüência, o conjunto-solução, por compreender uma grande quantidade de elementos relacionados, acaba por apresentar informações redundantes, provenientes de padrões de baixa significância, que não adicionam, ao resultado, informações úteis o suficiente para justificar a sua importância.

Esta dissertação apresenta uma nova metodologia para obtenção de padrões de interesse numa base de dados que explora o conceito de ortogonalidade - definida como a medida do quanto os elementos de um conjunto contribuem com informações não redundantes para a solução de um problema - e a sua aplicação ao problema da classificação associativa, como forma de aumentar a eficácia de um classificador, diminuindo a redundância e a ambigüidade das regras.

Abstract

Frequent pattern mining is one of the most exploited subjects in data mining, assuming a key role in numerous tasks that have the goal of finding patterns of interest in a given data set. However, most of the solutions proposed in this line of research still have not solved problems, many of them related to the explosion in the number of frequent patterns found in the data set. This happens because frequent patterns conform to the anti-monotony property, which says that if a pattern is frequent, all its sub-patterns are also. This way the solution, by having redundant information from patterns of low significance, does not add to the result information useful enough to justify its importance.

This work presents a new methodology for obtaining patterns of interest in a data set that explores the concept of orthogonality - defined as the measure of how the elements of a set does not contribute with redundant information to the solution of a problem - and its application in associative classification, as a way to increase the effectiveness of a classifier, reducing the redundancy and ambiguity of the rules.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Há muito já não é mais novidade o fato de que vivemos, hoje, na chamada **Era da Informação** (termo cunhado por Daniel Bell, professor emérito da Universidade de Harvard), que se define pelo rápido desenvolvimento da tecnologia de informação, característica marcante da **Sociedade Pós-Industrial**, onde nota-se que, mais importante que possuir a informação, é saber onde e como encontrá-la.

Ao longo dos últimos anos, o desenvolvimento tecnológico fez com que o constante armazenamento de dados se tornasse uma atividade fácil e de baixo custo. A conseqüência deste fato foi a alimentação de enormes bases de dados, o que proporcionou a necessidade de se pensar em novas soluções de gerência, manutenção e, posteriormente, acesso.

Os Sistemas de Gerenciamento de Banco de Dados (**SGBD**) surgiram como uma solução para gerência e manutenção de bases, com o objetivo de retirar da aplicação cliente a responsabilidade de administrar o acesso, a manipulação e a organização dos dados. Entretanto, embora os complexos SGBD's tenham se consolidado como eficientes sistemas na gerência de grandes volumes de dados, da existência de tais bases surgiu a necessidade de métodos avançados de recuperação de informação em amplas coleções, como redução de volume de dados, extração da essência da informação armazenada, descoberta de padrões, dentre outros.

Como resposta à necessidade dos novos métodos de recuperação de informação, surgiu o que chamamos hoje de **Mineração de Dados**.

1.1 Mineração de Dados

Mineração de dados é o resultado de um longo processo de pesquisa e desenvolvimento, considerada uma das mais importantes fronteiras entre bases de dados e sistemas de informação, e um dos mais promissores desenvolvimentos interdisciplinares na tecnologia

da informação (Han e Kamber, 2000). Podemos defini-la como o conjunto de métodos não triviais de análise de dados e extração de informação potencialmente útil, implícita, previamente desconhecida, e suas novas formas de representação e visualização (Tan et al., 2006; Hand et al., 2001; Frawley et al., 1992).

O termo Mineração de Dados é mencionado com freqüência no contexto de **Descoberta de Conhecimento em Bases de Dados** (**KDD - *Knowledge Discovery in Databases***), que é um processo de extração de conhecimentos válidos, novos, potencialmente úteis e compreensíveis para apoiar a tomada de decisão (Fayyad et al., 1996). Para tanto, o KDD faz uso de diversos artifícios, incluindo métodos estatísticos, reconhecimento de padrões, visualização, banco de dados, aprendizado de máquina, inteligência artificial, *data warehouse*, dentre outros (Sassi, 2006).

O processo de descoberta de conhecimento em bases de dados consiste, usualmente, na seguinte seqüência de etapas:

1. Aprendizado: Entendimento do domínio da aplicação, que inclui conhecimento prévio da aplicação e dos seus objetivos;

2. Seleção: Coleta dos dados, constituído pela definição de uma base de dados e pela seleção de um conjunto de dados e atributos a serem considerados;

3. Pré-processamento: Abrange a execução de operações básicas como remoção de ruídos, a definição de procedimentos e estratégias aplicáveis a dados faltosos, decisões relacionadas a tipos de dados, dentre outras tarefas;

4. Transformação: Consiste na definição de atributos indicados para representar os dados, na definição da dimensionalidade dos dados, e em outros métodos de transformação para reduzir o número de variáveis sob consideração;

5. Mineração de Dados: Inclui a escolha da função de mineração de dados, de acordo com o propósito do modelo (podendo ser classificação, agrupamento, sumarização ou até mesmo combinações de duas ou mais funções quaisquer), e do algoritmo a ser aplicado, que depende de decisões tais como qual modelo é mais apropriado considerando-se a natureza dos dados. Por exemplo, modelos para dados categóricos são diferentes de modelos para dados contínuos. A execução da mineração de dados se resume em pesquisar por padrões de interesse numa representação (ou conjunto de representações) definida de acordo com a função de mineração de dados escolhida;

6. Interpretação e Avaliação: Esta etapa pode resultar no retorno a qualquer um dos passos anteriores, caso os resultados não sejam satisfatórios e alguma alteração seja necessária para a sua melhoria;

7. Utilização do Conhecimento: A última etapa consiste na incorporação do conhecimento adquirido promovendo ações ou simplesmente documentando e reportando resultados para as partes de interesse.

De acordo com Fayyad et al. (1996), mineração de dados é um passo do processo KDD que consiste na enumeração de padrões sobre os dados, sujeito a aceitáveis limitações de eficiência computacional. Considerando que padrões enumeráveis sobre uma base finita de dados são potencialmente infinitos, e que tal enumeração envolve alguma forma de pesquisa num espaço amplo, restrições computacionais adicionam severos limites sobre o sub-espaço que pode ser explorado por um algoritmo de mineração de dados. Por este motivo, a constante busca por novas metodologias e algoritmos de enumeração de padrões em grandes bases de dados tem sido foco de inúmeras frentes de pesquisa em mineração de dados.

1.1.1 Padrões Freqüentes

De acordo com Han e Kamber (2000), são chamados padrões freqüentes aqueles que aparecem repetidamente num conjunto de dados. Por exemplo, um par de itens, como "café" e "leite", que aparecem juntos assiduamente em transações de uma base de dados, é considerado um padrão freqüente. Uma seqüência, como comprar uma câmera fotográfica, e logo depois um cartão de memória, se ocorre muitas vezes na base de dados de um estabelecimento comercial, também é considerada um padrão freqüente.

Padrões freqüentes assumem um papel essencial em muitas tarefas de mineração de dados que possuem, como objetivo, encontrar padrões de determinado interesse numa base, tais como regras de associação, correlações, seqüências, episódios, classificadores, agrupamentos e muitas outras das quais a mineração de regras de associação é uma das mais populares (Goethals, 2003). Desde a introdução deste termo, em 1993, por Agrawal et al., a mineração de padrões freqüentes e regras de associação têm recebido grande atenção. Nas últimas décadas, centenas de artigos científicos foram publicados apresentando algoritmos, abordagens e melhorias para resolver problemas de mineração de padrões freqüentes e regras de associação de forma mais eficiente.

Dentre as características mais importantes dos padrões freqüentes, podemos citar o suporte, que representa a medida de popularidade do padrão em relação à base. Um padrão é considerado freqüente se possui suporte maior que um dado limite inferior. Na prática, não estamos interessados apenas no conjunto de padrões freqüentes, mas também nas respectivas medidas de popularidade destes padrões.

1.1.2 Regras de Associação

Em mineração de dados, regras de associação são utilizadas para descobrir elementos que co-ocorrem freqüentemente, sob a relação de causalidade, numa determinada base de dados. A motivação por trás da utilização deste conceito está relacionada com a grande quantidade de aplicações possíveis de se construir com o auxílio de regras de associação, como, por exemplo, questões relacionadas ao comportamento de clientes num determinado estabelecimento comercial.

Considerando uma base de dados de transações em que cada transação represente uma ação de compra de determinado cliente, e cada item da transação represente um produto adquirido, é possível, analisando a freqüência dos itens e a sua co-existência em transações, gerar regras do tipo *se um cliente compra o produto A, existe uma determinada probabilidade de o produto B também ser adquirido na mesma compra.* Chamamos o produto A de termo antecedente, e o produto B de termo conseqüente da regra. Regras de associação surgiram como uma boa opção para responder questões como:

- Encontre todas as regras que possuem "café" como termo conseqüente. Estas regras poderiam auxiliar o usuário a planejar o que deve ser feito no estabelecimento para aumentar a venda daquele produto;
- Encontre todas as regras que possuem "café" como antecedente. Com estas regras seria possível descobrir que produtos serão impactados se o estabelecimento optar por descontinuar a venda daquele;
- Encontre todas as regras que possuem "café" como antecedente e "leite" como conseqüente. Esta requisição poderia ser realizada para todos os produtos do estabelecimento, com a intenção de se encontrar aqueles cujo consumo esteja relacionado;
- Encontre todas as regras relacionando itens localizados nas prateleiras A e B do estabelecimento. Estas regras poderiam auxiliar no planejamento das prateleiras e na organização dos produtos;
- Encontre as k "melhores" regras que possuem "café" como conseqüente. As "melhores" regras podem ser obtidas com o auxílio de métricas associadas a cada regra, como suporte e confiança.

Uma das características mais difundidas das regras de associação é a confiança. A confiança de uma regra representa a probabilidade de que uma transação da base de dados coberta pelo antecedente da regra também seja coberta pelo termo conseqüente.

Em geral, além do conjunto de regras de associação, estamos interessados também no suporte dos termos que as compõem, e na confiança de cada regra.

1.2 Ortogonalidade

Em matemática, ortogonalidade é uma característica que denota perpendicularidade, ou existência de ângulos retos. Formalmente, dois vetores x e y são ortogonais num espaço vetorial V se o produto interno $\langle x, y \rangle$ é zero. Esta situação é descrita por $x \perp y$.

O termo pode ser estendido para o uso geral, denotando a característica de independência, não redundância, não sobreposição, e, até mesmo, a importância de uma entidade qualquer Y dado que X já é conhecida. Neste trabalho, o termo ortogonalidade está mais relacionado com a ausência de redundância entre dois elementos.

Dado um conjunto qualquer, estamos interessados no quanto os elementos deste conjunto contribuem com informações não redundantes para a solução de um problema. Considerando que seja possível medir esta contribuição, e chamá-la de **significância**, podemos definir **ortogonalidade** como a média das significâncias dos elementos do conjunto.

Os termos similaridade e ortogonalidade têm sido explorados em várias áreas da Ciência da Computação, e com diversos objetivos. Um exemplo é o caso do Modelo de Espaço Vetorial, aplicado à Recuperação de Informação, que utiliza modelos vetoriais e métricas de similaridade para se aproximar termos da pesquisa fornecidos por um certo usuário aos documentos de determinada coleção.

Já a ortogonalidade, particularmente, tem sido explorada em áreas como visualização (Cui et al., 2007), reconhecimento facial (Nagao e Sohma, 1998), taxonomia (Smith et al., 2000), dentre outras.

Como exemplo de aplicação do termo em mineração de dados, podemos citar Xin et al. (2006), que utiliza métricas de significância e redundância para extrair, de um grande conjunto de padrões freqüentes, um sub-conjunto (top-k) de padrões com um valor mínimo de redundância. O autor comenta que este estudo abriu uma nova direção na procura por diferentes e significantes top-k respostas para atividades de mineração de dados, que podem resultar em estudos promissores.

1.3 Objetivos

Este trabalho visa explorar o problema de classificação associativa em mineração de dados considerando ortogonalidade entre padrões freqüentes com os seguintes objetivos:

- Minimizar o número de padrões utilizados na geração das regras, extraindo, do

conjunto de padrões freqüentes, um sub-conjunto de padrões ortogonais preservando a mesma qualidade do conjunto original, para que, a partir destes, sejam geradas as regras associativas necessárias para se realizar a classificação;

- Diminuir a redundância das regras geradas, como conseqüência da utilização de ortogonalidade aplicada ao conjunto de padrões freqüentes;
- Diminuir a ambigüidade das regras geradas, como conseqüência da utilização de ortogonalidade aplicada à cobertura de classes e transações pelos padrões;
- Aumentar a efetividade das classificações, como conseqüência da diminuição da redundância e da ambigüidade das regras, de acordo com os itens acima.

1.4 Organização do Documento

Este documento é dividido em 6 capítulos, organizados da seguinte forma:

- O capítulo 2 apresenta uma revisão literária, com alguns dos trabalhos mais relacionados com esta dissertação;
- O capítulo 3 possui uma base de informações relacionadas à classificação associativa, como definição do problema, definição e comparação entre as estratégias *eager* e *lazy*, e uma breve apresentação das métricas mais utilizadas para se comparar regras de associação;
- No capítulo 4 será apresentada a nova estratégia de classificação baseada em ortogonalidade. O conteúdo inclui uma discussão sobre métricas e estratégias de ortogonalidade utilizadas, uma explicação detalhada da utilização de ortogonalidade pelo classificador e das heurística de obtenção de conjuntos ortogonais, e ainda a adaptação da estratégia ORIGAMI (mencionada no capítulo 2) para o problema de classificação;
- No capítulo 5 serão apresentados os experimentos executados e os resultados obtidos durante a realização do trabalho;
- O capítulo 6 possui um breve resumo do que foi apresentado, além de sugestões para futuros trabalhos relacionados ao tema.

Capítulo 2

Trabalhos Relacionados

Este capítulo é dedicado ao levantamento de trabalhos relacionados com a proposta da dissertação. Primeiramente, serão apresentados alguns trabalhos relacionados com a diminuição, ou compactação do conjunto de padrões freqüentes obtidos. Em seguida, trabalhos relacionados com a diminuição da redundância no conjunto-resultado, e, por último, trabalhos relacionados com classificação associativa, por ser este o problema escolhido para se aplicar o conceito de ortogonalidade.

2.1 Diminuição do Conjunto de Padrões Freqüentes

O problema de compactação do conjunto de padrões freqüentes tem sido abordado de várias formas. Algumas metodologias optam por extrair um sub-conjunto (top-k) do conjunto original de padrões freqüentes de forma a maximizar uma determinada função objetivo fornecida pelo usuário, como é o caso encontrado em Mielikäinen e Mannila (2003). Neste trabalho, os autores utilizam, como critério de seleção dos top-k padrões, uma função cujo sub-conjunto obtido produz as melhores estimativas sobre a freqüência dos padrões do conjunto original que não foram selecionados. Em Xin et al. (2006) é apresentado um trabalho semelhante, onde a função objetivo não está ligada à definição do problema. Dessa forma, um critério diferente de seleção dos top-k padrões pode ser definido de acordo com cada aplicação. Outros trabalhos interessados em se obter um sub-conjunto aproximado de padrões freqüentes são encontrados em Afrati et al. (2004), Xin et al. (2005) e Wang e Parthasarathy (2006).

Uma segunda alternativa para ser compactar o conjunto de padrões freqüentes é desenvolver modelos alternativos de representação de tais conjuntos. Neste caso, o resultado obtido pela aplicação não é, necessariamente, constituído de um conjunto de padrões. Em Boulicaut et al. (2003), por exemplo, os autores introduzem uma estrutura chamada *free-sets* de onde é possível obter um valor aproximado para o suporte de

qualquer padrão da base. Os experimentos demonstram que a obtenção de *free-sets* é possível mesmo quando a extração dos padrões freqüentes se torna intratável. Em Zhu et al. (2007) são introduzidos os chamados padrões colossais, e uma nova metodologia de mineração de padrões, que aproxima o conjunto resultado do conjunto de padrões freqüentes colossais. Note que, neste caso, o resultado do algoritmo é constituído, não do conjunto de padrões colossais, mas sim de uma aproximação deste conjunto.

No nosso trabalho, estamos interessados em diminuir o tamanho do conjunto-solução como foi feito nos primeiros trabalhos apresentados, ou seja, obtendo, como resultado, um sub-conjunto (top-k) de padrões a partir do conjunto original de padrões freqüentes.

2.2 Diminuição de Redundância no Conjunto de Padrões Freqüentes

Aqui apresentamos alguns trabalhos relacionados com a obtenção de conjuntos padrões freqüentes com baixa redundância. Em Xin et al. (2006), trabalho que já foi citado na seção anterior, os autores apresentam a aplicação do modelo de obtenção de top-k padrões nos problemas de extração de termos de documentos e *prefetch* de blocos em seqüências de acesso ao disco, utilizando uma função objetivo que relaciona duas métricas propostas - significância e redundância. A idéia é obter, como resultado, um conjunto de padrões com alta significância e baixa redundância, que represente bem todo o conjunto de padrões freqüentes. O artigo apresenta as duas métricas da função objetivo e descreve um algoritmo guloso que resolve o problema com ordem de complexidade de tempo $O(\log k)$.

Em Xin et al. (2005) encontramos uma abordagem diferente para o problema, onde os padrões freqüentes e fechados são agrupados de acordo com uma medida de similaridade baseada em cobertura de transações da base, e, de cada agrupamento, um padrão que possui boa representatividade em relação aos outros elementos do seu conjunto deve ser escolhido e acrescentado à solução. O resultado é um sub-conjunto de padrões freqüentes e fechados com baixa redundância e alta representatividade em relação ao conjunto original.

Em Zaki et al. (2007) encontramos um novo paradigma para obtenção de uma representação resumida do conjunto de padrões freqüentes aplicado ao problema de mineração de grafos. O artigo apresenta a formulação da obtenção do conjunto α-ortogonal e β-representativo como um problema de otimização **NP-Hard**. Um conjunto de padrões é considerado α-ortogonal se a similaridade entre os padrões que o compõem é menor ou igual a um determinado valor α. Dado um conjunto de padrões, um sub-

conjunto deste é β-representativo em relação aos elementos que não fazem parte dele se, para cada um destes elementos, existe um elemento no sub-conjunto cuja similaridade entre os dois é maior ou igual a β. O objetivo é obter sub-conjuntos de padrões não redundantes que possuem um certo nível de representatividade em relação ao conjunto original. Esta abordagem será discutida com mais detalhes na seção 4.4.

Nesta dissertação, estamos interessados em minimizar a redundância do conjunto de padrões freqüentes com o auxílio de uma função objetivo a ser definida de acordo com a aplicação. No nosso caso, a aplicação utilizada é a classificação associativa, e a função objetivo é a medida de ortogonalidade do conjunto, que está relacionada com a métrica de ortogonalidade utilizada.

2.3 Classificação Associativa

Dentre todos os modelos de classificação associativa encontrados na literatura, optamos por utilizar, neste trabalho, um modelo baseado na estratégia *lazy*, introduzida em Veloso et al. (2006b). Este artigo apresenta as vantagens da estratégia *lazy* quando comparado com a estratégia *eager*, bastante explorada em árvores de decisão e classificadores baseados em entropia. Em primeiro lugar, artigo demonstra que classificadores associativos *eager* possuem desempenho melhor que árvores de decisão, e discute como regras de decisão podem ser geradas a partir de um modelo baseado em árvores de decisão. Em seguida, os autores apresentam a estratégia *lazy*, e demonstram que ela produz resultados melhores que a estratégia *eager*. Classificadores *eager* geram regras antes que as instâncias de teste sejam analisadas. A dificuldade de se antecipar todas as direções possíveis da tarefa de classificação faz com que a abordagem produza regras muito generalizadas, o que pode reduzir a eficácia do classificador. A estratégia *lazy*, no entanto, só produz regras aplicáveis à instância de teste, o que faz com que a eficácia do classificador não seja penalizado pela generalização. Os resultados apresentados compravam a superioridade da estratégia *lazy*.

Em Veloso et al. (2006a), os autores utilizam a estratégia *lazy* de classificação associativa direcionado para classificação de documentos. O artigo apresenta um algoritmo de classificação capaz de analisar tanto o conteúdo dos documentos quanto a existências de *links* de saída e de entrada para outros documentos. As inovações introduzidas pelos autores produzem resultados mais efetivos e de maior eficácia que abordagens do estado da arte nas coleções utilizadas. A mesma abordagem ainda obteve resultados satisfatórios em outras aplicações (Veloso e Meira Jr., 2006; Veloso et al., 2007).

No nosso trabalho, aplicamos o conceito de ortogonalidade num modelo de classificação associativa baseada na abordagem *lazy* e direcionado para a classificação de

transações em bases de dados genéricas.

2.4 Considerações

Neste capítulo, fizemos uma revisão da bibliografia relacionada com esta dissertação. Vimos que parte dos trabalhos apresentados possuem, como único objetivo, diminuir o tamanho do conjunto-solução, e outros estão interessados em diminuir a redundância de tais conjuntos. Poucos são os trabalhos que se interessam por extrair, do conjunto de padrões freqüentes, sub-conjuntos com baixa redundância e alta significância ao mesmo tempo. Também encontramos, neste capítulo, trabalhos relacionados com o problema da classificação associativa. No capítulo seguinte, apresentaremos uma descrição detalhada deste problema, além de uma discussão sobre as estratégias **eager** e *lazy*.

Capítulo 3

Classificação Associativa

Neste capítulo apresentaremos fundamentos teóricos relacionadas à Classificação Associativa, como definições de padrões freqüentes e regras de associação, além de uma breve discussão a respeito das estratégias *eager* e *lazy* encontradas na literatura. Também serão apresentadas e discutidas algumas métricas amplamente utilizadas na avaliação de regras de associação.

3.1 Introdução

Classificação já é um problema muito bem explorado na Ciência da Computação. Encontra-se facilmente, na literatura, vários modelos que têm sido propostos ao longo dos anos, incluindo modelos baseados em redes neurais (Lippmann, 1988), modelos estatísticos (James, 1985), árvores de decisão (Breiman et al., 1984; Quinlan, 1993) e algoritmos genéticos (Goldberg, 1989). Dentre todos estes, árvores de decisão é um dos mais apropriados para a mineração de dados, pelo fato desta ser uma abordagem de fácil entendimento (Quinlan, 1993), além da sua construção ser relativamente fácil, quando comparada com outros modelos.

Como uma alternativa para árvores de decisão, a classificação associativa foi proposta (Liu et al., 1998; Dong et al., 1999; Li et al., 2001). Esta abordagem, primeiramente, obtém um conjunto de regras de associação da base de treinamento, e então, constrói um classificador utilizando as regras obtidas. Este processo de classificação produz bons resultados, e ainda melhores acurácias que árvores de decisão (Liu et al., 1998).

De acordo com Veloso et al. (2006b), algoritmos baseados em árvores de decisão executam uma estratégia gulosa (Cormen et al., 2001) de pesquisa por regras selecionando, por meio de uma heurística qualquer, os atributos mais indicados para se representar a classe de uma determinada instância. O algoritmo tem início com um conjunto vazio

de regras de decisão e, gradualmente, adiciona restrições a este conjunto, até que não haja mais evidências para continuar a pesquisa, ou uma discriminação perfeita seja alcançada. O problema das árvores de decisão é que esta estratégia "gulosa" poderia reduzir o espaço de pesquisa, desconsiderando algumas regras de importância para o modelo.

Classificação associativa, por outro lado, executa uma pesquisa global por regras que satisfazem determinadas questões de qualidade, o que faz com que nenhuma regra de importância seja desconsiderada pelo algoritmo.

3.2 Fundamentos Teóricos

Os dados de entrada para um classificador são uma coleção de registros. Cada registro é caracterizado por um par (x, y), onde x é um conjunto de atributos comuns, e y é um atributo especial, designado como classe. De acordo com Tan et al. (2006), classificação é o processo de se descobrir uma função f que realiza o mapeamento de cada conjunto de atributos x para uma das classes y pré-definidas.

Este processo pode ser descrito da seguinte forma: Um primeiro conjunto de dados de entrada, chamado de base de treinamento, é utilizado para se construir um modelo que relaciona os atributos dos registros da base com uma das classes previamente conhecidas. Um segundo conjunto de dados, chamado de base de teste, que possui registros com apenas atributos comuns, é utilizado para se validar o modelo obtido com a base de treinamento.

O processo de validação do modelo consiste em, através deste, designar uma classe para cada instância de teste, e comparar os resultados com o valor real. Espera-se que as classes designadas coincidam com as classes reais de cada instância de teste, o que validaria o modelo obtido com a base de treinamento.

A função de mapeamento de um classificador associativo é representada por um conjunto de regras de associação. Tais regras são geradas a partir de um conjunto de padrões freqüentes extraídos da base de treinamento. Nas seções 3.2.1 e 3.2.2 serão apresentados os fundamentos teóricos que estão por trás dos padrões freqüentes e das regras de associação de um classificador associativo.

3.2.1 Padrões Freqüentes

O problema da mineração de padrões freqüentes pode ser descrito da seguinte forma:

Seja $\mathcal{I}$ um conjunto de itens. Um conjunto $X = \{i_1, \cdots, i_k\} \subseteq \mathcal{I}$ é chamado de *itemset* (ou padrão), ou *k-itemset* se ele contém k itens.

Uma transação sobre $\mathcal{I}$ é um par $T = (tid, I)$ onde tid é o identificador da transação e I é um *itemset*. Dizemos que uma transação $T = (tid, I)$ é coberta por um *itemset* $X \subseteq \mathcal{I}$, se $X \subseteq I$.

Uma base de dados de transações $\mathcal{D}$ sobre $\mathcal{I}$ é um conjunto de transações sobre $\mathcal{I}$. Omitiremos $\mathcal{I}$ sempre que a sua dedução for trivial, de acordo com o contexto.

A cobertura de um *itemset* X em $\mathcal{D}$ consiste no conjunto de identificadores das transações em $\mathcal{D}$ cobertas por X:

$$cobertura(X, \mathcal{D}) := \{tid | (tid, I) \in \mathcal{D}, X \subseteq I\}.$$

A freqüência de um *itemset* X em $\mathcal{D}$ é o número de transações cobertas por X em $\mathcal{D}$:

$$frequencia(X, \mathcal{D}) := |cobertura(X, \mathcal{D})|.$$

Note que $|\mathcal{D}| = frequencia(\{\}, \mathcal{D})$. O suporte de um *itemset* X em $\mathcal{D}$ é a probabilidade de X ocorrer em uma transação $T \in \mathcal{D}$:

$$suporte(X, \mathcal{D}) := P(X) = \frac{frequencia(X, \mathcal{D})}{|\mathcal{D}|}.$$

Omitiremos $\mathcal{D}$ sempre que a sua dedução for trivial, de acordo com o contexto.

Um *itemset* é freqüente se o seu suporte é maior ou igual a um dado valor relativo mínimo σ_{rel}, com $0 \leq \sigma_{rel} \leq 1$. Quando se considera freqüências de *itemsets* ao invés de suportes, é utilizado um valor absoluto σ_{abs}, com $0 \leq \sigma_{abs} \leq |\mathcal{D}|$. Obviamente, $\sigma_{abs} = \lceil \sigma_{rel} \cdot |\mathcal{D}| \rceil$. No restante deste texto, sempre que não especificarmos a natureza de σ, σ_{rel} deve ser considerado.

Definição 1 *Seja $\mathcal{D}$ uma base de dados de transações sobre um conjunto de itens $\mathcal{I}$, e σ um valor mínimo de suporte. A coleção de itemsets freqüentes em $\mathcal{D}$ em relação a σ é dado por:*

$$\mathcal{F}(\mathcal{D}, \sigma) := \{X \subseteq \mathcal{I} | suporte(X, \mathcal{D}) \geq \sigma\}.$$

Problema 1 (Mineração de Padrões Freqüentes) *Dado um conjunto de itens $\mathcal{I}$, uma base de dados de transações $\mathcal{D}$ sobre $\mathcal{I}$, e um suporte mínimo σ, encontre $\mathcal{F}(\mathcal{D}, \sigma)$.*

3.2.2 Regras de Associação

Uma regra de associação é uma implicação da forma $X \Rightarrow Y$, onde X é um conjunto de itens em $\mathcal{I}$, e Y é um único item em $\mathcal{I}$ que não está presente em X. A regra $X \Rightarrow Y$ é satisfeita no conjunto de transações T com confiança $0 \leq c \leq 1$ se, e somente se,

pelo menos $c\%$ das transações em T que satisfazem X também satisfazem Y (Agrawal et al., 1993).

O suporte de uma regra $X \Rightarrow Y$ em $\mathcal{D}$ é o suporte de $X \cup Y$ em $\mathcal{D}$, e a freqüência da regra é a freqüência de $X \cup Y$. Dizemos que uma regra de associação é freqüente se o seu suporte excede um determinado valor mínimo σ.

A confiança de uma regra de associação $X \Rightarrow Y$ em $\mathcal{D}$ é a probabilidade condicional de encontrar Y numa transação, dado que esta contém X:

$$confianca(X \Rightarrow Y, \mathcal{D}) := P(Y|X) = \frac{suporte(X \cup Y, \mathcal{D})}{suporte(X, \mathcal{D})}.$$

Dizemos que a regra é de confiança se $P(Y|X)$ excede um determinado valor mínimo de confiança γ, com $0 \leq \gamma \leq 1$.

Definição 2 *Seja $\mathcal{D}$ uma base de dados de transações sobre um conjunto de itens $\mathcal{I}$, σ um valor mínimo para suporte e γ um valor mínimo para confiança, o conjunto de regras de associação freqüentes e de confiança considerando σ e γ é dado por:*

$$\mathcal{R}(\mathcal{D}, \sigma, \gamma) := \{X \Rightarrow Y | X, Y \subseteq \mathcal{I}, X \cap Y = \{\}, X \cup Y \in \mathcal{F}(\mathcal{D}, \sigma), \\ confianca(X \Rightarrow Y, \mathcal{D}) \geq \gamma\}.$$

Problema 2 (Mineração de Regras de Associação) *Dado um conjunto de itens $\mathcal{I}$, uma base de dados de transações $\mathcal{D}$ sobre $\mathcal{I}$, um valor mínimo para suporte σ e um valor mínimo para confiança γ, encontre $\mathcal{R}(\mathcal{D}, \sigma, \gamma)$.*

3.3 Estratégias *eager* e *lazy*

Classificadores que utilizam a estratégia *eager* geram um conjunto de regras a partir de *itemsets* freqüentes em relação à base de treinamento, e as organizam em ordem decrescente, de acordo com métricas apropriadas (alguma delas discutidas na seção 3.4). Então, para cada instância de teste, a primeira regra do conjunto que pode ser aplicada à instância é utilizada para classificá-la.

Intuitivamente, classificadores associativos possuem comportamento melhor que árvores de decisão pelo fato de permitirem que várias regras sejam aplicadas a uma mesma partição da base de treinamento. Enquanto árvores de decisão produzem apenas uma regra possível de se aplicar a uma determinada instância de teste, classificadores associativos geram várias regras aplicáveis, que precisam ser ordenadas para que, posteriormente, a mais indicada seja escolhida para se classificar a instância.

Entretanto, uma estratégia *eager* pode gerar um número muito grande de regras, muitas delas desnecessárias durante a classificação, por não serem aplicáveis a nenhuma instância de teste.

Diferente da estratégia *eager*, um classificador associativo que utiliza a estratégia *lazy* gera regras específicas para cada instância de teste. A estratégia *lazy* obtém uma projeção da base de treinamento somente com instâncias que possuem pelo menos um atributo em comum com a instância de teste. A partir desta projeção e do conjunto de atributos da instância de teste, as regras são induzidas e ordenadas, e a melhor regra do conjunto é utilizada para a classificação. Pelo fato das regras serem induzidas a partir do conjunto de atributos da instância de teste, todas as regras geradas serão aplicáveis.

Em Veloso et al. (2006b), encontramos a demonstração de que estratégias *lazy* de classificadores associativos produzem resultados iguais ou melhores aos classificadores que utilizam a estratégia *eager*.

3.4 Métricas de Regras de Associação

Classificadores associativos, após produzirem um conjunto de regras de associação a partir de documentos que constituem a base de treinamento, organizam as regras em ordem decrescente, de acordo com uma métrica apropriada, para então utilizá-las na classificação de determinada instância de teste. Encontramos, na literatura, diversas métricas para este propósito, desde as mais conhecidas suporte e confiança até outras mais sofisticadas, como coerência, convicção, interesse, correlação, dentre outras.

Abaixo, fornecemos a definição de algumas das métricas mais utilizadas. Utilizamos o termo Z para designar uma regra associativa, que também pode ser descrita como $X \Rightarrow Y$. O termo $\mathcal{D}$ foi utilizado para designar a base de treinamento. Chamamos de $P(W)$ a probabilidade de que uma transação qualquer de $\mathcal{D}$ seja coberta por W:

$$P(W) = \frac{frequencia(W)}{|\mathcal{D}|},$$

onde W pode ser tanto uma regra quanto um de seus elementos (padrão ou classe), e $frequencia(W)$ é o número de transações em $\mathcal{D}$ cobertas por W.

- Suporte: Introduzido por Agrawal et al. (1993), e definido como $P(Z)$, o suporte dá a proporção de transações na base de dados cobertas pela regra. Esta métrica é utilizada como uma medida da significância (ou importância) de um *itemset*. A principal característica do suporte é a anti-monotonicidade, ou seja, se um *itemset* é freqüente, todos os seus subconjuntos também serão. A desvantagem do suporte

está relacionado com os itens raros. Itens que ocorrem com baixa freqüência na base de dados podem ser desprezados, embora talvez sejam importantes para a tarefa de classificação;

- Confiança: Esta métrica, definida como $confianca(X \Rightarrow Y) = P(Y|X)$, e também introduzida por Agrawal et al. (1993), representa a probabilidade de que uma transação coberta pelo antecedente de uma regra também seja coberta pelo termo conseqüente. Confiança está relacionada com a validação da hipótese representada pela regra. Em geral, o suporte é utilizado para se encontrar *itemsets* significativos na base de dados, e a confiança é aplicada como um segundo passo para gerar regras precisas. A desvantagem da confiança é que ela está diretamente relacionada com a freqüência do termo conseqüente. Pela forma que a confiança é calculada, termos conseqüentes com altos suportes tendem a produzir altos valores para confiança, mesmo se não existir nenhuma associação entre os dois termos da regra;

- Convicção: Definida como $conviccao(X \Rightarrow Y) = \frac{P(X) \times P(\neg Y)}{P(X \wedge \neg Y)}$, e introduzida por Brin et al. (1997), convicção foi desenvolvida como uma alternativa para confiança, pelo fato desta não capturar adequadamente a direção das associações. Convicção compara a probabilidade de X aparecer sem Y com a freqüência real do aparecimento de X sem Y;

- Leverage: Definida como $leverage(X \Rightarrow Y) = P(X \wedge Y) - (P(X) \times P(Y))$, e introduzida por Piatetsky-Shapiro (1991), mede a diferença de X e Y aparecendo juntos na base de dados e o que seria esperado se X e Y fossem estatisticamente dependentes;

- Lift: Definida como $lift(X \Rightarrow Y) = \frac{P(X \wedge Y)}{P(X) \times P(Y)}$, e introduzida por Brin et al. (1997), mede quantas vezes X e Y ocorrem juntos a mais que o esperado se eles fossem estatisticamente independentes. Uma das desvantagens do *lift* é ser suscetível a ruídos em pequenas bases de dados. Raros *itemsets* com baixa probabilidade, que ocorrem juntos algumas vezes podem produzir altos valores para *lift*;

- Jaccard: O coeficiente de Jaccard é uma medida estatística utilizada para comparar similaridade e diversidade entre conjuntos, definida pela razão entre a interseção e a união entre dois conjuntos. Esta métrica, obtida pela expressão $jaccard(X \Rightarrow Y) = \frac{P(X \wedge Y)}{P(X)+P(Y)-P(X \wedge Y)}$, é considerada como medida de interesse aplicada a regras de associação em diversos trabalhos encontrados na literatura (Tan et al., 2002; Geng e Hamilton, 2006; Wu et al., 2007);

- Laplace: Definida como $laplace(X \Rightarrow Y) = \frac{frequencia(X \wedge Y)+1}{frequencia(X)+c}$, onde c é o número de classes do domínio, foi introduzida por Clark e Boswell (1991) como uma métrica alternativa para avaliação de regras de associação;

- Kulc: Definida como $kulc(X \Rightarrow Y) = \frac{P(X \wedge Y)}{2}\left(\frac{1}{P(X)} + \frac{1}{P(Y)}\right)$, esta medida, originalmente conhecida como *Kulczynski*, é muito utilizada na área química, e foi proposta como métrica para regras de associação em Wu et al. (2007);

- Cosseno: Esta métrica, bastante utilizada como medida de similaridade para textos, também é amplamente utilizada na avaliação de regras de associação (Tan et al., 2002; Geng e Hamilton, 2006; Wu et al., 2007), e é definida como $cosseno(X \Rightarrow Y) = \frac{P(X \wedge Y)}{\sqrt{P(X) \times P(Y)}}$;

- Sensitividade: Definida como $sensitividade(X \Rightarrow Y) = P(X|Y)$, sensitividade (ou *recall*) é bastante utilizada em sistemas de recuperação de informação. Foi proposta como métrica de avaliação de regras de associação em Geng e Hamilton (2006);

- Especificidade: Definida como $especificidade(X \Rightarrow Y) = P(\neg Y|\neg X)$, e também citada por Geng e Hamilton (2006), esta métrica representa a proporção de verdadeiro-negativos sobre os casos negativos da regra.

Muitas outras métricas tem sido utilizadas em modelos de classificação associativa. Em Tan et al. (2002) encontramos um ótimo estudo comparativo entre várias, com descrição de propriedades chave que podem ser examinadas para se descobrir qual a métrica mais indicada para o domínio de cada aplicação. Em Geng e Hamilton (2006) são apresentados nove critérios a serem considerados para se determinar que métricas devem ser utilizadas de acordo com o interesse e o tipo da aplicação. Em Wu et al. (2007) os autores apresentam um estudo de outro conjunto de métricas sob a perspectiva da propriedade *null-(transaction) invariance* de importância crítica para métricas de interesse.

Neste trabalho, o famoso *framework* suporte-confiança foi utilizado durante a geração das regras associativas. Como medida de qualidade das regras, foram implementadas todas as métricas apresentadas nesta seção, com a intenção de se avaliar, experimentalmente, a qualidade das regras obtidas a partir dos padrões ortogonais quando interpretadas sob a perspectiva de diferentes métricas.

3.5 Considerações

Neste capítulo foram apresentados os fundamentos teóricos relacionadas à Classificação Associativa, e as vantagens da estratégia *lazy* sobre a estratégia *eager*. Em seguida, foram apresentadas algumas métricas bastante utilizadas na avaliação de regras de associação. No capítulo seguinte, será discutido a utilização de ortogonalidade na classificação associativa, que inclui uma discussão sobre métricas e estratégias de ortogonalidade, uma explicação detalhada da utilização de ortogonalidade pelo classificador e das heurística de obtenção de conjuntos ortogonais, e ainda a adaptação da estratégia ORIGAMI para o problema de classificação associativa.

Capítulo 4

Padrões Freqüentes e Ortogonais

Neste capítulo serão apresentadas as razões que justificam a utilização do conceito de ortogonalidade na mineração de padrões freqüentes, e no problema de classificação associativa. Também serão introduzidas as métricas de ortogonalidade aplicáveis a conjuntos de padrões, e o modo como foram utilizadas. Em seguida, será descrito como a ortogonalidade foi empregada num algoritmo de classificação baseado na abordagem *lazy*. Por fim, apresentaremos a estratégia ORIGAMI e a sua adaptação ao problema de classificação associativa.

4.1 Introdução

Como já foi mencionado na seção 1.1.1, padrões freqüentes são largamente utilizados em diversas aplicações na área de mineração de dados, incluindo regras de associação, classificação, agrupamento, indexação, dentre outras. Encontra-se, na literatura, uma grande quantidade de algoritmos de mineração de padrões freqüentes que, para muitas aplicações, produzem resultados satisfatórios (como os apresentados no capítulo 2). Entretanto, grande parte destas soluções ainda possuem problemas não resolvidos.

Uma das razões que contribuem para este fato é que, de acordo com a definição, qualquer sub-conjunto de um padrão freqüente também é freqüente, o que faz com que o tamanho do conjunto-solução do algoritmo cresça de maneira explosiva. A introdução de conceitos como padrões fechados ou padrões maximais ajudou a resolver esta questão, porém, as abordagens existentes minimizam o conjunto-solução apenas sob a perspectiva do suporte, não considerando a semântica dos dados, o que pode fazer com que padrões interessantes ao usuário sejam retirados da solução durante o processo de minimização do resultado.

Outro desafio que ainda persiste na mineração de padrões freqüentes é a eliminação de redundância nos resultados obtidos. Em muitas aplicações encontramos a necessi-

dade de extrair um pequeno conjunto de padrões freqüentes que tenham não só alta significância, mas também baixa redundância. A significância é, em geral, definida pelo contexto da aplicação. De acordo com Xin et al. (2006), alguns estudos recentes têm se concentrado em como extrair top-k padrões com alta significância, e outros em como remover redundância entre padrões, mas poucos têm se dedicado a obter sub-conjuntos de alta significância e baixa redundância ao mesmo tempo.

O objetivo da aplicação de ortogonalidade no problema da mineração de padrões freqüentes é desenvolver uma técnica capaz de extrair um sub-conjunto de padrões com tanto alta significância quanto baixa redundância entre seus elementos.

4.2 Métricas de Ortogonalidade

Para se extrair um sub-conjunto de padrões de alta significância e baixa redundância em relação ao conjunto original de padrões freqüentes, é necessário definir métricas que possibilitem a avaliação dos vários conjuntos-solução possíveis. Na literatura, encontramos alguns exemplos de tais métricas. Em Xin et al. (2006) os autores utilizam, em seus experimentos, o complemento do coeficiente de **Jaccard** (Jain e Dubes, 1988) aplicado à cobertura da base de dados para se obter uma medida de distância entre dois padrões:

$$D(p_1, p_2) = 1 - \frac{|TS(p_1) \cap TS(p_2)|}{|TS(p_1) \cup TS(p_2)|},$$

onde $TS(p)$ é o conjunto de transações cobertas pelo padrão p.

Em Zaki et al. (2007) os autores discutem diferentes métricas de similaridade entre grafos, incluindo, além do coeficiente de **Jaccard**:

$$sim(G_a, G_b) = \frac{|t(G_a) \cap t(G_b)|}{|t(G_a) \cup t(G_b)|},$$

uma métrica de distância indicada para grafos baseada no máximo sub-grafo comum (Bunke e Shearer, 1998):

$$sim_{mc}(G_1, G_2) = \frac{|G_{mc}|}{max(|G_1|, |G_2|)},$$

onde G_{mc} é o maior sub-grafo comum entre G_1 e G_2.

A principal característica destas métricas é que elas foram desenvolvidas para avaliação de pares de padrões, portanto, não são aplicáveis em conjuntos de tamanho maior que 2. No nosso caso, estamos interessados em métricas que definem a ortogonalidade de conjuntos de qualquer tamanho.

Nas próximas sub-seções apresentaremos três métricas de ortogonalidade, sendo as

duas primeiras aplicáveis a qualquer problema relacionado com mineração de padrões freqüentes em bases de transações, e a terceira diretamente relacionada com o problema da classificação associativa.

4.2.1 Estrutura dos Padrões

Considerando a estrutura dos padrões, ou seja, o conjunto de itens pelos quais são definidos, pode-se dizer que os padrões ABC e DEF são ortogonais, mas ABC e CDE não o são, já que o item C está presente nos dois padrões. O mesmo pode ser aplicado a conjuntos maiores, por exemplo, os padrões AB, CD e EF são ortogonais, mas os padrões AB, BC e CD não o são.

Definição 3 *Considerando a estrutura dos padrões, dois padrões são ortogonais se eles não possuem itens em comum.*

Entretanto, a simples informação de que um determinado conjunto de padrões é ortogonal ou não é insuficiente para que um possível conjunto-solução seja avaliado. Para tanto, é preciso desenvolver uma métrica dada por uma função $f : \mathcal{S} \rightarrow [0, 1]$ (onde $\mathcal{S}$ é o conjunto de elementos avaliado) que forneça a medida de ortogonalidade do conjunto.

Uma forma de se obter esta medida, é analisar os itens dos padrões. Definimos que dois ou mais padrões são ortogonais se eles não compartilham itens, ou seja, a presença de um mesmo item em mais de um padrão do conjunto, intuitivamente, deve contribuir de maneira negativa para a medida de ortogonalidade do mesmo. Dessa forma, optou-se por medir a ortogonalidade de um conjunto por meio da qualidade dos itens encontrados nos padrões, onde, neste caso, a qualidade de um item está diretamente relacionada com a sua raridade no conjunto de padrões.

A métrica de ortogonalidade baseada na estrutura dos padrões associa pesos aos itens, de modo que, quanto maior a sua popularidade, menor será o seu peso. O resultado é dado pela média dos pesos de todos os itens que fazem parte dos padrões do conjunto.

Seja $\mathcal{I}$ um conjunto de itens, $\mathcal{D}$ uma base de dados de transações em $\mathcal{I}$, $\mathcal{F}$ o conjunto de padrões freqüentes em $\mathcal{D}$, e $\mathcal{F}^{'}$ um sub-conjunto de $\mathcal{F}$ ($\mathcal{F}^{'} \subseteq \mathcal{F}$). Chamamos de $\mathcal{I}_F^{'} \subseteq \mathcal{I}$ o sub-conjunto de itens que aparecem em, pelo menos, um dos padrões de $\mathcal{F}^{'}$. Para cada item $i \subseteq \mathcal{I}_F^{'}$ é dado um peso:

$$w_i = \frac{|\mathcal{F}^{'}| - |\mathcal{F}_i^{'}|}{|\mathcal{F}^{'}| - 1},$$

onde $\mathcal{F}_i^{'} \subseteq \mathcal{F}^{'}$ é o sub-conjunto de padrões de $\mathcal{F}^{'}$ que contém o item i. É fácil perceber que o numerador da expressão terá o seu valor mínimo (0) quando o item for compartilhado por todos os padrões do conjunto, e terá o seu valor máximo ($|\mathcal{F}^{'}| - 1$) quando o item estiver presente em apenas um padrão. A constante 1 é subtraída de $|\mathcal{F}^{'}|$ no denominador justamente para que o valor máximo obtido com a fração seja 1.

De acordo com esta expressão, se o conjunto possui quatro padrões, e um item i está presente em apenas um deles, o valor do seu peso w_i é equivalente a 1. Se o item estiver presente em 3 padrões do conjunto, o peso w_i será $\frac{1}{3}$. Itens compartilhados por todos os padrões têm peso igual a 0.

Finalmente, a ortogonalidade baseada na estrutura dos padrões do conjunto é dada por:

$$O_e = \frac{\sum_{i \subseteq \mathcal{I}_F^{'}} w_i}{|\mathcal{I}_F^{'}|},$$

onde $0 \leq O_e \leq 1$. Como exemplo, suponha um conjunto de itens $\mathcal{I} = \{A, B, C, D, E\}$ e um conjunto de padrões: $\mathcal{F} = \{AB, ABC\}$. Na tabela 4.1 encontramos o valor do peso de cada item, de acordo com a expressão apresentada. Os itens D e E não possuem pesos, pelo fato de não serem encontrados nos padrões.

Itens	Pesos
A	0
B	0
C	1
D	–
E	–

Tabela 4.1: Pesos dos Itens para Ortogonalidade Baseada na Estrutura dos Padrões (1)

Como podemos ver, o item C é o único que faz parte de apenas um padrão, portanto, é associado o peso 1 para este item. Os outros dois itens (A e B) são encontrados nos dois padrões, e por isso, possuem peso 0. De acordo com a expressão de ortogonalidade, a média destes valores nos dá a medida de ortogonalidade do conjunto: $O_e = 0,33$. Agora, suponha que seja inserido o novo padrão BCD ao conjunto ($\mathcal{F} = \{AB, ABC, BCD\}$). A tabela 4.2 possui os novos pesos para cada item.

Note que o novo padrão possui um novo item (D) que antes não era encontrado no conjunto. A inserção de BCD fez com que o peso de A aumentasse, visto que este item não faz parte do padrão. Já o peso de B continuou o mesmo, e o novo item D entrou no conjunto com peso 1. Assim, a ortogonalidade do novo conjunto é $O_e = 0,5$.

Proposição 1 *De acordo com a definição 3, um conjunto de padrões é totalmente ortogonal por estrutura se e somente se* $O_e = 1$.

Itens	Pesos
A	$0,5$
B	0
C	$0,5$
D	1
E	$-$

Tabela 4.2: Pesos dos Itens para Ortogonalidade Baseada na Estrutura dos Padrões (2)

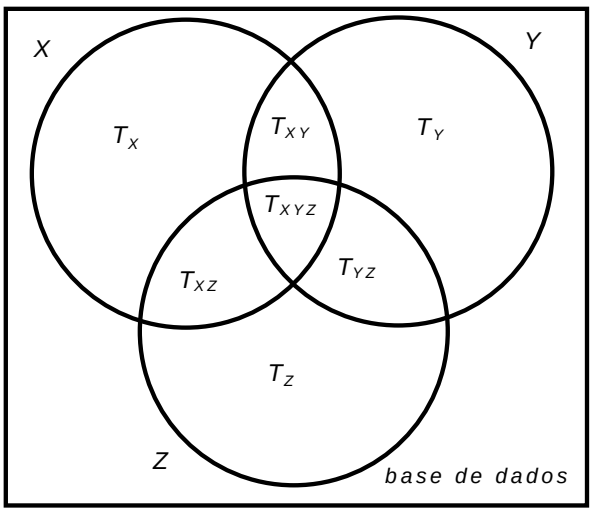

Figura 4.1: Visualização de Cobertura de Transações na Base de Dados

4.2.2 Cobertura de Transações

Considerando o espaço de transações, dizemos que dois padrões são ortogonais se eles cobrem áreas diferentes da base de dados, ou seja, se não existem sobreposições nas coberturas de cada padrão do conjunto.

Definição 4 *Considerando a cobertura de transações, dois padrões são ortogonais se eles não possuem transações em comum.*

Na figura 4.1 temos uma representação de uma base de dados, e os conjuntos de transações cobertas por três padrões (X, Y e Z). As transações que fazem parte das áreas T_X, T_Y e T_Z são cobertas por apenas um dos padrões (X, Y e Z respectivamente). As transações de T_{XY} são cobertas por X e Y, e as transações de T_{XYZ} são cobertas pelos três padrões do exemplo. Se precisamos de padrões que cobrem diferentes áreas da base de dados, então nós estamos interessados em maximizar T_X, T_Y e T_Z, no caso do exemplo.

A métrica de ortogonalidade baseada em cobertura de transações pode ser calculada de maneira análoga à relacionada com a estrutura, sendo que, neste caso, os elementos

que possuem pesos a serem calculados são as transações cobertas pelos padrões, e não os seus itens.

Seja $\mathcal{I}$ um conjunto de itens, $\mathcal{D}$ uma base de dados de transações em $\mathcal{I}$, $\mathcal{F}$ o conjunto de padrões freqüentes em $\mathcal{D}$, e $\mathcal{F}^{'}$ um sub-conjunto de $\mathcal{F}$ ($\mathcal{F}^{'} \subseteq \mathcal{F}$). Chamamos de $\mathcal{D}^{'}_{F} \subseteq \mathcal{D}$ o sub-conjunto transações cobertas por, pelo menos, um dos padrões de $\mathcal{F}^{'}$. Para cada transação $t \subseteq \mathcal{D}^{'}_{F}$ é dado um peso:

$$w_t = \frac{|\mathcal{F}^{'}| - |\mathcal{F}^{'}_{t}|}{|\mathcal{F}^{'}| - 1},$$

onde $\mathcal{F}^{'}_{t}$ é o sub-conjunto de padrões de $\mathcal{F}^{'}$ que cobrem a transação t. As mesmas análises realizadas sobre a métrica baseada na estrutura dos padrões podem ser aplicadas aqui. O numerador da expressão terá o seu valor mínimo (0) quando a transação for coberta por todos os padrões do conjunto, e terá o seu valor máximo ($|\mathcal{F}^{'}| - 1$) quando a transação for coberta por apenas um padrão. Da mesma forma, a constante 1 é subtraída de $|\mathcal{F}^{'}|$ no denominador justamente para que o valor máximo obtido com a fração seja 1.

De acordo com esta expressão, se nós tivermos 4 padrões, e uma transação t coberta por apenas um deles, então o seu peso w_t será 1. Se esta transação for coberta por dois padrões do conjunto, o valor de w_t será $\frac{1}{2}$. As transações cobertas por todos os padrões do conjunto terão peso igual a 0.

A ortogonalidade baseada em cobertura de transações do conjunto é dada por:

$$O_t = \frac{\sum_{t \subseteq \mathcal{D}^{'}_{F}} w_t}{|\mathcal{D}^{'}_{F}|},$$

onde $0 \leq O_t \leq 1$. Suponha uma base de dados $\mathcal{D} = \{A, ABCD, BCDE, CDEF\}$, e um conjunto de padrões $\mathcal{F} = \{AB, BC, CD\}$. Na tabela 4.3 encontramos valor do peso de cada transação, de acordo com a expressão apresentada. A transação A não possui peso, pelo fato de não ser coberta por nenhum dos padrões.

Transações	Pesos
A	–
$ABCD$	0
$BCDE$	0,5
$CDEF$	1

Tabela 4.3: Pesos das Transações para Ortogonalidade Baseada em Cobertura de Transações

A transação $CDEF$ é a única coberta por apenas um padrão do conjunto, portanto, é associado o peso 1 para ela. A transação $BCDE$ é coberta por dois padrões, portanto,

recebe o peso $0,5$, e a transação $ABCD$ recebe o peso 0, já que ela é coberta por todos os padrões do conjunto. A média dos pesos das transações nos dá o valor da ortogonalidade do conjunto: $O_t = 0,5$.

Proposição 2 *De acordo com a definição 4, um conjunto de padrões é totalmente ortogonal por cobertura de transações se e somente se* $O_t = 1$.

4.2.3 Cobertura de Classes

As duas métricas de ortogonalidade apresentadas nas seções 4.2.1 e 4.2.2 estão relacionadas apenas com os padrões e as transações da base, e podem ser utilizadas em qualquer tipo de aplicação. Já a métrica apresentada nesta seção é voltada diretamente para o problema da classificação, pois está relacionada com as classes das transações cobertas pelos padrões do conjunto.

Definição 5 *Dois padrões são ortogonais se são encontrados em transações de classes distintas na base de dados, ou seja, se os conjuntos de transações cobertas por cada um dos padrões não possuem classes em comum.*

Imagine um conjunto de três documentos, d_1, d_2 e d_3. em que d_1 e d_2 pertencem à classe c_1 e d_3 pertence à classe c_2. Agora, imagine um conjunto de três padrões, p_1, p_2 e p_3, onde p_1 é encontrado no documento d_1, p_2 é encontrado no documento d_2 e p_3 aparece no documento d_3. Neste caso, considerando-se a cobertura de classes, poderíamos dizer que p_1 e p_2, apesar de aparecerem em documentos diferentes, não são termos ortogonais, já que os documentos d_1 e d_2 possuem a mesma classe. Em contrapartida, os padrões p_2 e p_3 são ortogonais, visto que são encontrados em documentos de classes diferentes.

A métrica de ortogonalidade por cobertura de classes pode ser calculada de maneira análoga às métricas de similaridade entre padrões e cobertura de transações, sendo que, neste caso, os elementos que possuem pesos a serem calculados são as classes existentes nas transações cobertas pelos padrões.

Antes de apresentar a definição matemática da métrica, é importante falar sobre as características esperadas de um conjunto de padrões ortogonais considerando-se cobertura de classes. Tomemos um conjunto de transações $\mathcal{D}$, um conjunto de padrões $\mathcal{F}$ freqüentes em $\mathcal{D}$, um sub-conjunto de transações $\mathcal{D}_c^{'}$ (onde todas as transações pertencem à classe c) e um sub-conjunto de padrões (independentes, ou seja, ortogonais) $\mathcal{F}^{'}$, freqüentes em $\mathcal{D}_c^{'}$. Por serem padrões ortogonais, espera-se que eles não apareçam juntos em transações de mesma classe, e, ao mesmo tempo, espera-se que todas as transações sejam cobertas pelos padrões freqüentes. Logo, a cobertura esperada de cada padrão para a classe c é de, em média, $\frac{|\mathcal{D}_c^{'}|}{|\mathcal{F}^{'}|}$.

Seja $\mathcal{I}$ um conjunto de itens, $\mathcal{D}$ uma base de dados de transações em $\mathcal{I}$, $\mathcal{F}$ o conjunto de padrões freqüentes em $\mathcal{D}$, e $\mathcal{F}^{'}$ um sub-conjunto de $\mathcal{F}$ ($\mathcal{F}^{'} \subseteq \mathcal{F}$). Chamamos de $\mathcal{D}^{'}_{F} \subseteq \mathcal{D}$ o sub-conjunto transações cobertas por, pelo menos, um dos padrões de $\mathcal{F}^{'}$. Seja $\mathcal{C}_{\mathcal{D}}$ um conjunto de classes associadas às transações de $\mathcal{D}$. Chamamos de $\mathcal{C}^{'}_{D} \subseteq \mathcal{C}$ o sub-conjunto de classes associadas às transações de $\mathcal{D}^{'}_{F}$. Para cada classe $c \subseteq \mathcal{C}^{'}_{D}$ é dado um peso:

$$w_c = \frac{|\mathcal{F}^{'}| - |\mathcal{F}^{'}_{c}|}{|\mathcal{F}^{'}| - 1},$$

onde $\mathcal{F}^{'}_{c}$ é o sub-conjunto de padrões de $\mathcal{F}^{'}$ que cobrem uma quantidade de transações de classe $c \subseteq \mathcal{C}^{'}_{D}$ 10% [1] a mais que a média esperada. Também aqui são válidas as análises realizadas sobre o comportamento desta expressão para as duas métricas apresentadas anteriormente. O numerador da fração terá o seu valor mínimo (0) quando cada padrão aparecer em, pelo menos, uma transação da classe c, e terá o seu valor máximo ($|\mathcal{F}^{'}| - 1$) quando apenas um padrão for encontrado nas transações da classe c. Naturalmente, a constante 1 é subtraída de $|\mathcal{F}^{'}|$ no denominador para que o valor máximo obtido com a fração seja 1.

De acordo com esta expressão, se nós temos 4 padrões, e uma classe c que aparece em transações cobertas por apenas um deles, então o seu peso w_c será 1. Se esta classe aparece em transações cobertas por dois padrões do conjunto de maneira balanceada (por exemplo, 50% das transações da classe aparece em um dos padrões, e os 50% restante no outro), o valor de w_c será $\frac{1}{2}$. As classes que aparecem por igual em transações cobertas por todos os padrões do conjunto terão peso igual a 0.

Por fim, a ortogonalidade baseada em cobertura de classes é dada por:

$$O_c = \frac{\sum_{c \subseteq \mathcal{C}^{'}_{D}} w_c}{|\mathcal{C}^{'}_{D}|},$$

onde $0 \leq O_C \leq 1$. Suponha uma base de dados $\mathcal{D}$ em que as transações e suas respectivas classes são encontradas na tabela 4.4.

Suponha um conjunto de padrões $\mathcal{F} = \{AB, CD, EF\}$. Os pesos de cada classe, de acordo com a expressão apresentada, são encontrados na tabela 4.5.

Como pode ser visto na tabela 4.4, a classe 1 é coberta uma vez pelo padrão AB (transação $ABCD$), e três vezes pelo padrão CD (transações $ABCD$, ACD e BCD), totalizando 4 coberturas. Como o conjunto de padrões possui tamanho 3, considerando que todos os padrões sejam independentes, espera-se que a cobertura de classes seja

[1] O valor 10% é um fator obtido empiricamente que define o ponto a partir do qual a cobertura de uma determinada classe por um determinado padrão é considerada relevante no cálculo da métrica de ortogonalidade. Informações a respeito dos parâmetros considerados e resultados obtidos na obtenção deste fator se encontram no apêndice A.

Transações	Classes
$ABCD$	1
ACD	1
$BCDE$	1
$BDEF$	2
CDE	2
CEF	2

Tabela 4.4: Base de Dados de Transações e Classes

Classes	Pesos
1	1
2	$0,5$

Tabela 4.5: Pesos das Classes para Ortogonalidade Baseada em Cobertura de Classes

homogênea, ou seja, que cada padrão cubra $4/3$ transações de cada classe. De acordo com a expressão apresentada, consideramos $\mathcal{F}_c^{'}$ o conjunto de padrões que cobrem uma quantidade de transações da classe c maior que 90% da média esperada (neste caso, $1,2$). Do nosso conjunto de padrões, o único que satisfaz esta premissa é CD, que ocorre em 3 transações da classe 1. Logo, o peso dessa classe é 1.

Já a classe 2 é coberta duas vezes pelo padrão CD, e duas vezes pelo padrão EF, logo, também possui cobertura esperada igual a $4/3$. Entretanto, os dois padrões possuem cobertura desta classe acima de 90% da média esperada, logo, o peso dessa classe é $0,5$. A média dos pesos das classes nos dá o valor da ortogonalidade do conjunto: $O_c = 0,75$.

4.2.4 Utilização das Métricas

Após a introdução das métricas, é necessário definir como elas serão utilizadas para se obter a medida de ortogonalidade do conjunto. Na literatura, encontramos algumas estratégias interessantes.

Em Zaki et al. (2007) é utilizada uma medida de ortogonalidade aplicável a pares de padrões. Os autores optaram por utilizar, como métrica do conjunto, o valor da ortogonalidade entre os seus dois elementos mais similares.

Em Xin et al. (2006) os autores utilizaram métricas de significância aplicáveis a padrões e métricas de redundância aplicáveis a pares de padrões. O valor da função que compreende as duas métricas é obtido pela soma das significâncias de cada padrão do conjunto menos a média das redundâncias existentes entre todos os pares de padrões.

Neste trabalho, duas soluções foram implementadas. A primeira delas é baseada na medida de ortogonalidade do conjunto, onde aplicamos as métricas apresentadas

na seção 4.2 considerando todo o conjunto-solução. A segunda é baseada na solução proposta por Xin et al. (2006) - as métricas são aplicadas somente a pares de padrões, e a ortogonalidade do conjunto é obtida por meio da média das ortogonalidades entre todos os pares possíveis.

É fácil perceber que as métricas propostas nas seções 4.2.1 e 4.2.2 e 4.2.3 são equivalentes ao complemento do coeficiente de Jaccard quando aplicadas a conjuntos de dois padrões. Tomando a métrica baseada na estrutura dos padrões como exemplo, temos que a ortogonalidade do conjunto é dada pela expressão:

$$O_e = \frac{\sum_{i \subseteq \mathcal{I}'} \left(\frac{|\mathcal{F}'| - |\mathcal{F}'_i|}{|\mathcal{F}'| - 1} \right)}{|\mathcal{I}'|}.$$

Nos casos em que o conjunto de possui necessariamente dois padrões:

$$O_e = \frac{\sum_{i \subseteq \mathcal{I}'} (2 - |\mathcal{F}'_i|)}{|\mathcal{I}'|}.$$

Analisando a expressão acima, temos que $|\mathcal{I}'|$ corresponde ao tamanho do conjunto de itens encontrados nos dois padrões, ou seja,

$$|\mathcal{I}'| = |p_1 \cup p_2|,$$

onde $\mathcal{F}' = \{p1, p2\}$, e a parcela $(2 - |\mathcal{F}'_i|)$ terá valor 1 para itens que estão presentes em apenas um dos padrões, e 0 para itens presentes nos dois padrões, ou seja,

$$\sum_{i \subseteq \mathcal{I}'} (2 - |\mathcal{F}'_i|) = |p_1 \cup p_2| - |p_1 \cap p_2|.$$

Logo, a métrica de ortogonalidade baseada na estrutura dos padrões, quando aplicada par-a-par, é dada pela seguinte expressão:

$$O_e = 1 - \frac{\sum_{p_1, p_2 \subseteq \mathcal{F}', p_1 \neq p_2} \frac{|p_1 \cap p_2|}{|p_1 \cup p_2|}}{\frac{|\mathcal{F}'| \times (|\mathcal{F}'| - 1)}{2}}.$$

A métrica de ortogonalidade baseada em cobertura de transações, quando aplicada par-a-par, é dada pela seguinte expressão:

$$O_t = 1 - \frac{\sum_{p_1, p_2 \subseteq \mathcal{F}', p_1 \neq p_2} \frac{|\mathcal{D}'_{p1} \cap \mathcal{D}'_{p2}|}{|\mathcal{D}'_{p1} \cup \mathcal{D}'_{p2}|}}{\frac{|\mathcal{F}'| \times (|\mathcal{F}'| - 1)}{2}},$$

onde $\mathcal{D}_p^{'}$ é o sub-conjunto de transações cobertas pelo padrão p.

A métrica de ortogonalidade baseada em cobertura de classes, quando aplicada par-a-par, é dada pela seguinte expressão:

$$O_c = 1 - \frac{\sum_{p_1,p_2 \subseteq \mathcal{F}', p_1 \neq p_2} \frac{|\mathcal{C}_{p1}^{'} \cap \mathcal{C}_{p2}^{'}|}{|\mathcal{C}_{p1}^{'} \cup \mathcal{C}_{p2}^{'}|}}{\frac{|\mathcal{F}'| \times (|\mathcal{F}'|-1)}{2}},$$

onde $\mathcal{C}_p^{'}$ é o sub-conjunto de classes cujas transações são cobertas pelo padrão p 90% acima da média esperada.

4.3 Classificação Associativa e Ortogonalidade

Como já foi dito na seção 1.3, a utilização da ortogonalidade no problema da classificação associativa tem como objetivo aumentar a efetividade das classificações, como conseqüência da diminuição da redundância e da ambigüidade das regras. Uma das formas de se fazer isso seria aplicar ortogonalidade no conjunto de regras geradas pelo classificador. Dessa forma, seria possível extrair, de todo o conjunto de regras obtidas, apenas um sub-conjunto representativo de regras com alta significância e baixa redundância, e a partir destas, realizar a classificação das instâncias de teste.

Esta proposta é apresentada em Xin et al. (2006), onde é discutida a utilização de ortogonalidade aplicada a *prefetch* de blocos de acesso a disco. Os autores consideram a similaridade de duas regras igual a 0 quando os termos conseqüentes são diferentes, e igual ao coeficiente de Jaccard aplicado aos termos antecedentes quando os conseqüentes são iguais.

É possível aplicar este modelo no problema da classificação associativa. Nesse caso, seria considerada como métrica de ortogonalidade entre duas regras o valor máximo (1) se as classes para as quais elas apontam são diferentes, e o valor inversamente proporcional ao coeficiente de Jaccard (ou alguma outra métrica de similaridade) aplicado aos termos antecedentes se as classes para as quais elas apontam são iguais. Dessa forma, estaríamos gerando todas as regras possíveis, a partir do conjunto de padrões freqüentes obtidos por meio de um determinado suporte, e extraindo um sub-conjunto que consiste apenas das mais representativas, para então realizar a classificação da instância de teste.

Neste trabalho, porém, optamos por aplicar a ortogonalidade não ao conjunto de regras geradas, mas sim ao conjunto de padrões dos termos antecedentes utilizados para gerar as regras. A idéia é gerar regras a partir de um sub-conjunto dos padrões freqüentes obtidos, que consiste apenas dos padrões ortogonais. A principal diferença desta

abordagem, em relação à anterior, é que as métricas de ortogonalidade são aplicadas apenas aos termos antecedentes das regras, ou seja, as classes não são consideradas.

As três métricas descritas na seção 4.2 foram utilizadas neste trabalho. A métrica baseada na estrutura dos padrões foi utilizada com o objetivo de se encontrar um sub-conjunto de padrões que represente bem todo o conjunto de padrões freqüentes. Por ser uma métrica que considera o conjunto de itens, ela se aplica ao espaço dos padrões, não considerando a sua relação com as transações da base. Neste caso, estamos interessados em diminuir, principalmente, a redundância das regras, ou seja, não gerar regras com um alto nível de similaridade entre os termos antecedentes.

A cobertura de transações foi utilizada com o objetivo de se encontrar padrões ortogonais no espaço de transações. Com esta métrica, espera-se obter um sub-conjunto de padrões que cobrem a base de dados com o mínimo de sobreposições possível, ou seja, que as regras geradas por cada um destes padrões apontem para transações distintas da base. A intenção, neste caso, é diminuir a ambigüidade das regras, além da redundância.

A cobertura de classes é uma métrica definida especialmente para o problema da classificação, já que ela considera as classes das transações da base de treinamento para definir o conjunto de padrões ortogonais. Esta métrica é uma adaptação da cobertura de transações, visto que ela também está centrada na base de dados. A diferença é que, ao contrário da anterior, esta métrica analisa as classes, e não as transações cobertas pelos padrões. A motivação é que não basta extrair padrões que cobrem transações distintas da base de dados se estas transações possuem classes coincidentes. Esta métrica tenta garantir que as regras geradas pelo conjunto-resultado de padrões apontem para classes distintas. A intenção, como na métrica anterior, é diminuir a ambigüidade das regras, além da redundância.

4.3.1 Utilização de Ortogonalidade no LAC

O **LAC** (*Lazy Associative Classifier*) é uma implementação do modelo *lazy* proposto por Veloso et al. (2006b). A estratégia de classificação associativa *lazy* obtém o conjunto de regras de associação relacionadas a cada instância de teste separadamente. Para tanto, ela cria uma projeção da base de treinamento apenas com as transações que possuem itens em comum com a instância de teste. A partir desta projeção, a estratégia obtém um conjunto de padrões freqüentes, de acordo com determinado suporte fornecido pelo usuário, e com estes padrões, gera as regras de associação utilizadas durante a tarefa de classificação.

A ortogonalidade pode ser utilizada de várias maneiras no *lazy*, por exemplo, é possível extrair, *apriori*, o conjunto de itens ortogonais da base de treinamento, con-

siderando cobertura de transações, ou cobertura de classes, e, para cada instância de teste, obter a projeção da base de treinamento considerando apenas os itens que fazem parte do conjunto ortogonal. Esta seria uma boa opção para diminuir o espaço de busca durante a obtenção do conjunto de padrões freqüentes nos casos em que as bases de dados são constituídas de transações com um número muito grande de atributos.

Uma outra maneira seria extrair o conjunto de itens ortogonais, não de toda a base de treinamento, mas sim de cada instância no momento do teste. Entretanto, tanto esta forma quanto a anterior ainda possibilitaria a geração de conjuntos de regras redundantes, já que os itens obtidos, mesmo mantendo a característica de ortogonalidade entre si, poderiam gerar padrões freqüentes similares.

Sendo assim, a forma escolhida neste trabalho foi continuar utilizando todos os itens da instância de teste para gerar a projeção da base. A partir destes, obter o conjunto de padrões freqüentes e então, extrair o sub-conjunto de padrões ortogonais, com os quais são geradas as regras de associação.

4.3.2 Heurística de Obtenção de Conjuntos Ortogonais

O problema de se encontrar o sub-conjunto S de padrões freqüentes com a maior medida de ortogonalidade $O(\mathcal{S})$, dado o conjunto de padrões freqüentes $\mathcal{F}$ ($\mathcal{S} \subseteq \mathcal{F}$) é NP-completo. Como prova, basta considerar o seguinte problema de decisão associado:

Problema 3 (Obtenção de Conjuntos Ortogonais) *Dado um valor qualquer N em que $0 \leq N \leq 1$ e um conjunto de padrões freqüentes $\mathcal{F}$, existe $\mathcal{S} \subseteq \mathcal{F}$ tal que $O(\mathcal{S}) \geq N$?*

Portanto, foi desenvolvida uma heurística gulosa que inicia com um conjunto ortogonal de dois elementos, e, iterativamente, tenta obter um novo conjunto com um elemento a mais, acrescentando padrões candidatos e realizando modificações para que a métrica de ortogonalidade seja maximizada.

Esta abordagem é semelhante à que foi proposta em Zaki et al. (2007). Este artigo apresenta um algoritmo que considera o conjunto de padrões freqüentes como um grafo em que cada vértice representa um padrão, e uma aresta entre dois vértices representa a similaridade entre os padrões. No entanto, só são representadas as arestas que possuem similaridade menor que α (parâmetro do algoritmo). O objetivo da heurística é encontrar um clique de tamanho máximo neste grafo. Para isso, o algoritmo escolhe um vértice aleatoriamente e o adiciona no conjunto-solução. Após este passo, o algoritmo passa a visitar os vizinhos dos vértices que já fazem parte da solução, escolhendo sempre o melhor candidato para adicionar ao conjunto, até que não haja mais vértices para se adicionar.

No nosso caso, não utilizamos o parâmetro α, limite inferior para a métrica de ortogonalidade. Sendo assim, todos os padrões são candidatos ao conjunto-resultado. A obtenção do conjunto ortogonal de padrões é realizada de forma iterativa, onde, no início da execução, o algoritmo inicializa o conjunto-solução com apenas um elemento, e a métrica de ortogonalidade do conjunto com o valor 0 (zero), e então começa o ciclo de iterações em que, a cada etapa:

1. Um novo elemento é incluído ao conjunto;

2. É realizada uma busca por todo o conjunto de padrões que não fazem parte do conjunto-solução. Durante este procedimento, cada padrão verificado é incluído na solução, substituindo, neste conjunto, o elemento que mais se assemelha àquele. Se a métrica de ortogonalidade do conjunto melhorou, o algoritmo mantém a troca. Se não, a troca é desfeita, e o próximo padrão da seqüência é verificado;

3. Ao final do processo, o algoritmo compara a métrica de ortogonalidade obtida com a métrica do conjunto anterior (que possuía um elemento a menos). Se a métrica se manteve, ou melhorou, o algoritmo mantém o novo conjunto como solução, e volta ao início do ciclo. Se não, o algoritmo termina o ciclo, e o conjunto anterior é dado como resultado.

Como medida de semelhança para a escolha do padrão substituto da segunda etapa foi utilizado o coeficiente de Jaccard, aplicado a cada situação de acordo com a métrica utilizada. No caso da métrica baseada na estrutura dos padrões, no numerador da expressão se encontra o tamanho da interseção entre os conjuntos de itens de cada padrão, e no denominador se encontra o tamanho da união dos dois conjuntos. Já para a métrica baseada na cobertura de transações, no numerador da expressão é utilizado o tamanho da interseção entre os conjuntos de transações cobertas por cada padrão, e no denominador é utilizado o tamanho da união dos dois conjuntos. Para a métrica baseada em cobertura de classes, o numerador é formado pelo tamanho da interseção entre os conjuntos das classes cobertas por cada padrão, e no denominador é utilizado o tamanho da união dos conjuntos de classes cobertas por cada padrão.

A ordem como o algoritmo percorre o conjunto de padrões em cada iteração possui uma grande influência no algoritmo, pois um novo padrão verificado só fará parte do conjunto-solução se a ortogonalidade do novo conjunto for maior que a do antigo, logo, em caso de empate, os primeiros padrões terão preferência sobre os últimos. Cinco formas diferentes de ordenação do conjunto de padrões freqüentes foram implementadas:

1. Ordenação lexicográfica crescente;

2. Ordenação lexicográfica decrescente;
3. Ordenação por tamanho crescente;
4. Ordenação por tamanho decrescente;
5. Nenhuma ordenação.

As ordenações lexicográficas foram utilizadas para se diminuir a distância entre dois padrões da seqüência, fazendo com que a modificação do conjunto-solução seja realizada de forma suave. Espera-se que, neste caso, a diferença estrutural entre dois padrões consecutivos do conjunto durante cada iteração seja a menor possível. As ordenações por tamanho foram utilizadas para dar prioridade, ora aos maiores padrões (ordenação decrescente), ora aos menores padrões (ordenação crescente).

O pseudo-código do algoritmo de classificação associativa baseada em ortogonalidade **OLAC** (*Orthogonal Lazy Associative Classifier*) pode ser visto no algoritmo 1.

Algoritmo 1 OLAC

Require: $\mathcal{D}, \sigma$
$\mathcal{F} \leftarrow FindFrequentPatterns(\mathcal{D}, \sigma)$
$Sort(\mathcal{F})$
$\mathcal{O} \leftarrow GetFirstAvailablePattern(\mathcal{F})$
repeat
  $rate \leftarrow GetOrthogonalityRate(\mathcal{O})$
  $\mathcal{O}_{try} \leftarrow \mathcal{O} \cup GetFirstAvailablePattern(\mathcal{F})$
  $rate_{try} = GetOrthogonalityRate(\mathcal{O}_{try})$
  for $P \in \mathcal{F}, P \notin \mathcal{O}_{try}$ **do**
    $S \leftarrow GetMoreSimilar(\mathcal{O}, P)$
    $\mathcal{O}_{try} \leftarrow \mathcal{O}_{try} \cup P \setminus S$
    $rate_{tmp} = GetOrthogonalityRate(\mathcal{O}_{try})$
    if $rate_{tmp} \leq rate_{try}$ **then**
      $\mathcal{O}_{try} \leftarrow \mathcal{O}_{try} \cup S \setminus P$
    else
      $rate_{try} \leftarrow rate_{tmp}$
    end if
  end for
  if $rate_{try} \geq rate$ **then**
    $\mathcal{O} \leftarrow \mathcal{O}_{try}$
  end if
until $rate_{try} < rate$
$\mathcal{R} \leftarrow \mathcal{O}$

A seguir será apresentada a ordem de complexidade do algoritmo (desconsiderando o método *FindFrequentPatterns*, que obtém o conjunto de padrões freqüentes na base.

4.3.2.1 Análise de Complexidade

O método *GetMoreSimilar* (linha 9) recebe um padrão como parâmetro e é responsável por obter, de um conjunto de padrões, o padrão mais similar (ou menos ortogonal) ao fornecido. A ordem de complexidade deste método é $O(\mathcal{F} \times \mathcal{I}^2)$, onde $\mathcal{F}$ é o conjunto de padrões freqüentes, e $\mathcal{I}$ é o conjunto de itens encontrados nos padrões de $\mathcal{F}$.

O método *GetOrthogonalityRate* (linha 11) retorna a ortogonalidade do conjunto-resultado candidato naquela iteração, e possui ordem de complexidade $O(\mathcal{F}^2 \times \mathcal{I}^2)$.

O *loop* da linha 8 itera sobre o conjunto de padrões freqüentes executando os métodos acima para cada padrão encontrado, e possui complexidade $O(\mathcal{F}^2 \times \mathcal{I}^2)$.

O método *GetOrthogonalityRate* (linha 5) possui complexidade $O(\mathcal{F}^2 \times \mathcal{I}^2)$, como já foi demonstrado. Já o método *GetFirstAvailablePattern* possui complexidade $O(\mathcal{F})$, pois é responsável por obter, do conjunto de padrões freqüentes, o primeiro padrão que ainda não faz parte do conjunto-solução.

Na linha 7 temos novamente o método *GetOrthogonalityRate*, e na linha 4, um *loop* com complexidade $O(\mathcal{F}^3 \times \mathcal{I}^2)$, pois ele é responsável por executar todo o seu bloco interno até que o conjunto-solução possua a melhor métrica de ortogonalidade possível.

Logo, conclui-se que o algoritmo **OLAC** possui complexidade de tempo $O(\mathcal{F}^3 \times \mathcal{I}^2)$.

4.4 Estratégia ORIGAMI

O **ORIGAMI** é um algoritmo para mineração de grafos apresentado em Zaki et al. (2007). Neste artigo, os autores introduzem a definição de conjuntos α-ortogonais e β-representativos, e apresentam o novo paradigma de mineração de conjuntos de grafos ortogonais com foco nos padrões, e não nas transações.

4.4.1 Definição de alfa-ortogonalidade

Seja $\mathcal{F}$ o conjunto de todos os sub-grafos freqüentes de uma coleção. Seja $sim : \mathcal{F} \times \mathcal{F} \rightarrow [0, 1]$ uma função binária e simétrica que retorna a *similaridade* entre dois grafos, por exemplo, a similaridade entre dois grafos G_a e G_b baseada no máximo sub-grafo comum (Bunke e Shearer, 1998) dada por:

$$sim(G_a, G_b) = \frac{|G_c|}{max(|G_a|, |G_b|)},$$

onde G_c é o máximo sub-grafo comum entre G_a e G_b.

Dada uma coleção de grafos $\mathcal{G}$, e um limite superior para similaridade $\alpha \in [0, 1]$, dizemos que o sub-conjunto de grafos $\mathcal{R} \subseteq \mathcal{G}$ é **α-ortogonal** em relação a $\mathcal{G}$ se, e

somente se, para quaisquer $G_a, G_b \in \mathcal{R}, sim(G_a, G_b) \leq \alpha$ e para qualquer $G_a \in \mathcal{R}$ e qualquer $G_b \in \mathcal{G} \backslash \mathcal{R}, sim(G_a, G_b) > \alpha$.

Dada uma coleção de grafos $\mathcal{G}$,um conjunto α-ortogonal $\mathcal{R} \subseteq \mathcal{G}$ e um limite inferior para similaridade $\beta \in [0, 1]$, dizemos que $\mathcal{R}$ **representa** um grafo $G \in \mathcal{G}$ se existe algum $G_a \in \mathcal{R}$ tal que $sim(G_a, G) \geq \beta$. Seja $\Upsilon(\mathcal{R}, \mathcal{G}) = \{G \in \mathcal{G} : \exists G_a \in \mathcal{R}, sim(G, G_a) \geq \beta\}$, dizemos que $\mathcal{R}$ é um conjunto β-representativo para $\Upsilon(\mathcal{R}, \mathcal{G})$.

Dada uma coleção de grafos $\mathcal{G}$ e um conjunto α-ortogonal e β-representativo $\mathcal{R}$, chamamos de **resíduo** de $\mathcal{R}$ o conjunto de padrões não representados em $\mathcal{G}$, dado como $\Delta(\mathcal{R}, \mathcal{G}) = \mathcal{G} \backslash \{\mathcal{R} \cup \Upsilon(\mathcal{R}, \mathcal{G})\}$, o *resíduo* de $\mathcal{R}$ é definido como a cardinalidade do seu conjunto resíduo $|\Delta(\mathcal{R}, \mathcal{G})|$. Finalmente, definimos a média de similaridade do resíduo de $\mathcal{R}$ como $ars(\mathcal{R}, \mathcal{G}) = \frac{\sum_{G_b \in \Delta(\mathcal{R}, \mathcal{G})} max_{G_a \in \mathcal{R}}\{sim(G_a, G_b)\}}{|\Delta(\mathcal{R}, \mathcal{G})|}$.

Note que $\alpha < ars(\mathcal{R}, \mathcal{G}) < \beta$, já que, de acordo com a definição, para quaisquer $G_a \in \mathcal{R}$ e $G_b \in \Delta(\mathcal{R}, \mathcal{G}), sim(G_a, G_b) \in (\alpha, \beta)$.

O objetivo é encontrar conjuntos de grafos α-ortogonais e β-representativos em relação ao conjunto de sub-grafos maximais $\mathcal{M}$. Como o conjunto de padrões maximais provê uma síntese de todos os padrões freqüentes, e com uma quantidade bem menor de padrões, parece razoável tentar um conjunto representativo ortogonal em relação àquele. Entretanto, como encontrar todos os sub-grafos maximais em aplicações reais pode se tornar um problema intratável, os autores optaram por utilizar um sub-conjunto do conjunto de sub-grafos maximais $\widehat{\mathcal{M}} \subseteq \mathcal{M}$. Logo, o problema pode ser definido da seguinte forma:

Problema 4 (Mineração de grafos α-ortogonais e β-representativos) *Dado um sub-conjunto $\widehat{\mathcal{M}}$ do conjunto de sub-grafos maximais $\mathcal{M}$ de uma coleção de grafos $\mathcal{G}$, um limite superior para similaridade α e um limite inferior para similaridade β, encontre o melhor sub-conjunto $\mathcal{R}$ que minimize o resíduo $|\Delta(\mathcal{R}, \widehat{\mathcal{M}})|$.*

4.4.2 Estratégia de Ortogonalidade

A utilização dos parâmetros adicionais α e β pelo ORIGAMI enriquece o modelo de conjuntos ortogonais. O parâmetro α permite que a medida da ortogonalidade seja controlada, fazendo com que o conjunto-solução se aproxime ou se distancie do conjunto de padrões freqüentes de acordo com o valor escolhido para α. Já o parâmetro β permite medir a representatividade do conjunto em relação ao restante dos elementos que não fazem parte da solução.

Dependendo dos valores escolhidos para os dois parâmetros, duas variantes do problema são identificadas:

- Caso I ($\beta \leq \alpha$): Pela definição de conjunto α-ortogonal, $G_a \in \mathcal{R}$ e $G_b \in \widehat{\mathcal{M}} \backslash \mathcal{R}$

implica em $sim(G_a, G_b) > \alpha \geq \beta$. Logo, temos que $\Upsilon(\mathcal{R}, \widehat{\mathcal{M}}) = \widehat{\mathcal{M}} \backslash \mathcal{R}$, de onde temos que $\Delta(\mathcal{R}, \widehat{\mathcal{M}}) = 0$. Então, quando $\beta \leq \alpha$, o resíduo de qualquer conjunto α-ortogonal $\mathcal{R}$ é 0, o que implica que qualquer conjunto α-ortogonal é ótimo, considerando o resíduo;

- Caso II ($\beta > \alpha$): Este é o caso geral, onde um conjunto α-ortogonal $\mathcal{R}$ pode não ser um conjunto β-representativo para alguns grafos em $\widehat{\mathcal{M}}$. Em outras palavras, quando $\beta > \alpha$, o resíduo $|\Delta(\mathcal{R}, \widehat{\mathcal{M}})| \geq 0$; logo, a solução ótima é o conjunto de padrões ortogonais que minimiza o resíduo. Um caso especial de $\beta > \alpha$ ocorre quando $\beta = 1$. Neste caso, cada elemento do conjunto α-ortogonal representa somente a si mesmo, e o resíduo é dado por $|\Delta(\mathcal{R}, \widehat{\mathcal{M}})| = |\widehat{\mathcal{M}} \backslash \mathcal{R}|$.

O algoritmo ORIGAMI realiza a mineração de padrões ortogonais em dois passos distintos. O primeiro passo consiste em encontrar, utilizando uma heurística randômica, um sub-conjunto dos padrões maximais da base de dados. O segundo consiste em obter, novamente com o auxílio de uma heurística randômica, um conjunto ortogonal e representativo que minimize o resíduo. O pseudo-código do ORIGAMI pode ser visto no algoritmo 2.

Algoritmo 2 ORIGAMI

Require: $\mathcal{D}, \sigma, \alpha, \beta$
1: $EM \leftarrow EdgeMap(\mathcal{D})$
2: $\mathcal{F}_1 \leftarrow FindFrequentEdges(\mathcal{D}, \sigma)$
3: $\widehat{\mathcal{M}} \leftarrow 0$
4: **while** $\neg StopCondition()$ **do**
5: $\quad M \leftarrow RandomMaximalGraph(\mathcal{D}, \mathcal{F}_1, EM, \sigma)$
6: $\quad \widehat{\mathcal{M}} \leftarrow \widehat{\mathcal{M}} \cup M$
7: **end while**
8: $\mathcal{R} \leftarrow OrthogonalRepresentativeSets(\widehat{\mathcal{M}}, \alpha, \beta)$

4.4.3 Adaptação do Algoritmo

Foi realizada a implementação do algoritmo ORIGAMI adaptado ao problema de classificação associativa utilizando as métricas apresentadas na seção 4.2.

Como heurística de obtenção do conjunto de padrões maximais, o algoritmo inicia a execução com o conjunto-resultado vazio e, a cada iteração, tenta obter o maior padrão freqüente possível adicionando a ele, aleatoriamente, itens que fazem parte da instância de teste, até que não seja mais possível adicionar nenhum novo item, ou a condição de parada local seja atingida. Se, durante a obtenção aleatória dos itens, o item selecionado já ter sido utilizado, ou não gerar um novo padrão freqüente, o

algoritmo decrementa um contador de tentativas. A condição de parada local para a geração de novos padrões maximais é que, durante este processo, o número máximo de escolhas erradas dos itens não pode ser maior que o tamanho da instância de teste.

Ao obter um novo padrão maximal, o algoritmo tenta inseri-lo no conjunto-solução. Esta operação consiste em remover do conjunto todos os sub-padrões do novo candidato, e inserir o candidato caso nenhum dos padrões que ainda existem na solução seja super-padrão dele. A condição de parada para o algoritmo é que, durante todo o processo, o número máximo de padrões candidatos não maximais ou já inseridos no conjunto-solução não pode ser maior que o tamanho da instância de teste.

Como heurística para obtenção do conjunto ortogonal, o algoritmo inicia a execução com o valor de resíduo igual a 0 (zero) e, a cada iteração, tenta obter um conjunto ortogonal adicionando a ele, aleatoriamente, padrões maximais obtidos no primeiro passo do algoritmo, até que não seja mais possível acrescentar novos padrões, ou a condição de parada local seja atingida. Se, durante a obtenção dos padrões, o padrão selecionado já fizer parte do conjunto-resultado, ou não possuir similaridade menor que α para com todos os outros padrões do conjunto-solução, o algoritmo decrementa um contador de tentativas. A condição de parada local para a geração de conjuntos ortogonais é que, durante este processo, o número máximo de escolhas erradas de padrões não pode ser maior que a quantidade de padrões maximais total.

Ao obter um novo conjunto ortogonal, o algoritmo calcula o valor do seu resíduo. Se este valor é menor que o atual, o resíduo é atualizado, e o conjunto-solução passar a ser o conjunto ortogonal recém-encontrado. A condição de parada para o algoritmo é que, durante todo o processo, o número máximo de conjuntos ortogonais candidatos que não melhoram o resultado não pode ser maior que a quantidade de padrões maximais total.

Na seção 4.4.3.1 será apresentado um exemplo para facilitar o entendimento do algoritmo.

4.4.3.1 Exemplo de Execução

Esta seção pretende, com auxílio de um exemplo de execução, facilitar o entendimento da adaptação da estratégia ORIGAMI ao problema de classificação associativa. O foco do exemplo é demonstrar como o algoritmo constrói o conjunto-solução a partir dos padrões maximais gerados na primeira fase da sua execução. O exemplo será apresentado por meio de uma série de iterações, onde, a cada passo, a execução de uma etapa do algoritmo será demonstrada. A seguir, o exemplo de como o conjunto de padrões maximais é obtido:

- O algoritmo inicia com um conjunto-solução vazio, e realiza um ciclo de execuções

onde, a cada iteração, gera o maior padrão freqüente possível selecionando itens que fazem parte da base de dados, até que a condição de parada seja alcançada. Suponha que, na primeira iteração, o algoritmo tenha gerado o padrão **ABC**. Este padrão é adicionado ao conjunto-solução, que, neste momento, passa a ser constituído de apenas 1 elemento;

- Na próxima iteração do algoritmo, um novo padrão freqüente deve ser gerado da mesma forma que na iteração anterior. Suponha que, desta vez, o padrão **ABD** seja gerado. O novo padrão é adicionado ao conjunto. O novo conjunto-solução passa a ser formado pelos padrões **ABC** e **ABD**;

- Na terceira iteração, suponha que o algoritmo tenha gerado o padrão **BC**, e que a condição de parada tenha sido atingida antes que este se tornasse um padrão maximal. Neste caso, o novo padrão não será inserido no conjunto-solução, pois ele é sub-padrão de um dos padrões que já fazem parte do conjunto;

- Na próxima iteração, suponha que o algoritmo tenha gerado o padrão **ABDE**. Agora temos uma situação diferente da anterior, pois no conjunto-solução já existe um sub-padrão deste. Assim, o algoritmo adiciona o novo padrão ao conjunto e remove o seu sub-padrão da solução. Agora, o novo conjunto-solução passa ser formado pelos padrões **ABC** e **ABDE**;

- A execução termina se a condição de parada for alcançada, ou seja, se o número de novos padrões freqüentes gerados que não podem ser adicionados ao conjunto-solução for maior que o número de transações da instância de teste;

- Suponha que a execução deste passo tenha finalizado com um conjunto-solução constituído dos padrões **ABC**, **ABDE**, **ACDE** e **EF**. Este será o conjunto de padrões maximais utilizado na próxima etapa do algoritmo.

A seguir o exemplo de como o conjunto de padrões ortogonais é obtido:

- O algoritmo inicia com o valor de resíduo igual a 0 (zero) e, a cada iteração, tenta obter um candidato a conjunto-solução selecionando, aleatoriamente, padrões maximais obtidos na etapa anterior. Suponha que, na primeira iteração, o algoritmo tenha selecionado o padrão **ABC**. Neste ponto, o conjunto-solução passa a ser constituído de apenas 1 elemento;

- Na segunda iteração do algoritmo, suponha que o padrão **ABDE** tenha sido selecionado. Para que este padrão seja adicionado ao conjunto-solução, ele deve ter similaridade menor que α com todos os padrões que já fazem parte da solução.

No nosso exemplo, vamos considerar que esta condição seja verdadeira, e que o novo padrão tenha sido adicionado ao conjunto. O novo conjunto-solução é formado pelos padrões **ABC** e **ABDE**;

- Na iteração seguinte, suponha que o padrão **ACDE** tenha sido selecionado, e que este possua similaridade maior que α em relação ao padrão **ABDE**. Neste caso, o novo padrão não é adicionado ao conjunto-solução;

- O algoritmo continua selecionando padrões aleatoriamente e tentando inseri-los no conjunto-solução até que a condição de parada seja alcançada. Neste ponto, o valor do resíduo é calculado. Se o resíduo corrente for menor que o já obtido até o momento, o seu valor é atualizado, e o novo conjunto passa a ser o melhor candidato à solução.

4.5 Considerações

Neste capítulo foram apresentadas as maiores contribuições do trabalho. Em primeiro lugar, foram definidas três métricas de ortogonalidade aplicáveis a conjuntos de padrões, sendo as duas primeiras aplicáveis a qualquer problema relacionado com mineração de padrões freqüentes em bases de transações, e a terceira indicada para o problema da classificação associativa. Em seguida, foram apresentadas a estratégia de utilização de ortogonalidade num algoritmo de classificação baseado na abordagem *lazy* e uma heurística de obtenção de padrões ortogonais. Por último, apresentamos uma descrição detalhada da estratégia ORIGAMI, incluindo conceitos e definições relacionadas, além da adaptação do algoritmo ao problema da classificação associativa. No capítulo seguinte serão apresentados os experimentos e resultados obtidos com as duas abordagens baseadas em ortogonalidade sugeridas, e a sua comparação com a abordagem clássica (não ortogonal) já existente.

Capítulo 5

Avaliação Experimental

Neste capítulo, apresentamos o aplicativo **olac**, descrevemos a sua interface de interação com o usuário, e incluímos um exemplo de execução que demonstra o comportamento da abordagem baseada em ortogonalidade. Em seguida, são apresentados os resultados obtidos com a execução das três abordagens de classificação implementadas no aplicativo em 26 bases de dados do repositório **UCI**.

5.1 O Aplicativo olac

O aplicativo **olac** possui a implementação de três abordagens distintas de um classificador baseado em regras de associação.

A primeira delas, chamada de **LAC** (*Lazy Associative Classifier*), é a abordagem *lazy* na sua versão original (e não-ortogonal), como foi proposta em Veloso et al. (2006b), que simplesmente obtém, para cada instância de teste, um conjunto de padrões freqüentes, e o utiliza para gerar regras associativas. Depois disso, o algoritmo ordena as regras de acordo com uma métrica escolhida pelo usuário, e obtém a classe mais indicada pelas regras que compõem um *ranking* de tamanho também fornecido pelo usuário.

A segunda abordagem, chamada de **OLAC** (*Orthogonal Lazy Associative Classifier*) é a modificação da abordagem *lazy* que considera a ortogonalidade dos padrões durante a tarefa de obtenção de regras. O algoritmo extrai, do conjunto de padrões freqüentes, um sub-conjunto de padrões ortogonais, e utiliza este sub-conjunto para gerar as regras de associação. Depois disso, a classe mais indicada é escolhida considerando uma métrica e o tamanho do *ranking*, da mesma forma que é feito no LAC.

A terceira abordagem, chamada **ORIGAMI** é a implementação da estratégia ORIGAMI (como foi proposta em Zaki et al. (2007)) de acordo com a adaptação discutida na seção 4.4.3.

A interface de interação do usuário com o aplicativo foi realizada por meio de argumentos fornecidos em linha de comando pelo usuário, de acordo com o formato de execução apresentado abaixo:

```
Usage: ./olac [options]
Options:
  -i, --training-file       Set the training file
  -t, --testing-file        Set the testing file
  -s, --support             Set the support
  -c, --confidence          Set the confidence
  -r, --run-mode            Set the run mode [c,o] [CLASSICAL, ORTHOGONAL]
  -p, --pattern-set         Set the pattern set type [f,m,r] [FREQUENT, MAXIMAL,
                              RANDOM MAXIMAL]
  -n, --min-num-rules       Set the minimum number of rules generated
  -l, --max-num-rank-rules  Set the maximum number of rules considered (rank size)
  -m, --min-rule-len        Set the minimum length of the rules
  -x, --max-rule-len        Set the maximum length of the rules
  -o, --orth-mode           Set the orthogonality mode [h,p,o] [HEURISTICAL,
                              POLYNOMIAL, ORIGAMI]
  -e, --orth-metric         Set the orthogonality metric [s,c,l,a] [STRUCTURE,
                              TRANSACTION COVERAGE, CLASS COVERAGE, ALL]
  -w, --orth-method         Set the way metrics are used [s,p,a] [SET, PAIR AVERAGE,
                              ALL]
  -g, --orth-pat-ordering   Set the way patterns are ordered for heuristic
                              [s,r,i,z,n] [SORTED, REVERSE SORTED, SORTED BY SIZE,
                              REVERSE SORTED BY SIZE, NONE]
  -u, --rule-measure        Set the rule measure used [s,c,j,k,o,n,e,p,l,i,v]
                              [SUPPORT, CONFIDENCE, JACCARD, KULC, COSINE,
                              CONVICTION, SENSITIVITY, SPECIFICITY, LAPLACE,
                              LIFT, LEVERAGE]
  -a, --origami-alpha       Set the alpha parameter used by ORIGAMI
  -b, --origami-beta        Set the beta parameter used by ORIGAMI
  -d, --debug               Set the level of debug [-1,0,1,2,3,4] [SILENT, NO DEBUG,
                              LOW LEVEL, MEDIUM LEVL, HIGH LEVEL, MAX LEVEL]
  -v, --verbose             Use verbose mode
  -h, --help                Display this information
```

As opções **-i** e **-t** são utilizadas para fornecer os arquivos de treinamento e teste, respectivamente. O suporte para obtenção de padrões freqüentes, e a confiança para obtenção das regras de associação são fornecidos por **s** e **c**. A opção **-r** é usada para se escolher o modo de execução do algoritmo (**CLASSICAL** para **LAC** e **ORTHOGONAL** para **OLAC** e **ORIGAMI**). A opção **-p** é usada para se escolher o conjunto de padrões a ser obtido. Em geral, padrões freqüentes são utilizados pelas versões LAC e OLAC, e padrões maximais aleatórios são utilizados pela versão ORIGAMI,

no entanto, é possível executar o aplicativo com diferentes combinações de abordagens e conjuntos de padrões, como por exemplo, utilizar o conjunto de todos os padrões maximais no LAC, ou executar o ORIGAMI com padrões freqüentes.

A opção **-n** é utilizada para fornecer o número mínimo de regras a serem geradas. Caso não seja possível gerar um conjunto de regras de tamanho maior que este, de acordo com o parâmetro *confiança*, o aplicativo tenta gerar regras com confianças abaixo do esperado até que este valor mínimo seja alcançado. O parâmetro **-l** é usado para se fornecer o tamanho máximo do *ranking*. As opções **-m** e **-x** são utilizadas para limitar o número mínimo e máximo de itens no lado esquerdo das regras. Por meio da opção **-u** é informada a medida de interesse das regras. O modo de ortogonalidade é dado por **-o**, o usuário deve escolher entre a versão heurística de ortogonalidade (OLAC), a versão polinomial (uma abordagem bastante cara, que tenta todas as combinações para cada conjunto de padrões explorado), e o modo ORIGAMI. A opção **-e** é usada para se escolher a métrica de ortogonalidade utilizada pelo algoritmo em modo ortogonal. As opções **-w** e **-g** são usadas para dar a forma como as métricas serão utilizadas, e a forma como os padrões serão ordenados pelo algoritmo da abordagem heurística. As opções **-a** e **-b** são os parâmetros α e β para a abordagem ORIGAMI.

5.2 Exemplo de Execução

Abaixo será apresentado um exemplo de execução do aplicativo, demonstrando o seu funcionamento e os resultados obtidos pela abordagem **OLAC**:

A base de transações apresentada na tabela 5.1 foi criada sobre o conjunto de itens $\mathcal{I} = \{A, B, C, D, E\}$, e utilizada como base de treinamento para o classificador. Uma única transação ABE de classe $CLASS = 1$ foi utilizada como instância de teste. O aplicativo **olac** foi executado, primeiramente, em modo **LAC** (ou não-ortogonal) utilizando os parâmetros encontrados na tabela 5.2.

Esta execução gerou o conjunto $\mathcal{F} = \{A, B, E, AB, AE, BE, ABE\}$ de padrões freqüentes e classificou corretamente a instância de teste. Depois disso, o aplicativo foi executado em modo **OLAC** (ortogonal) utilizando os mesmos parâmetros da tabela 5.2, e *estrutura* como métrica de ortogonalidade. Esta abordagem obteve o conjunto de padrões ortogonais $\mathcal{O} = \{AE, B\}$ do conjunto original de padrões freqüentes, e classificou corretamente a instância de teste com as regras geradas deste sub-conjunto de padrões. Como podemos observar, o conjunto de padrões ortogonais utilizado pelo OLAC é consideravelmente menor que o conjunto de padrões freqüentes original. E mais, o conjunto de padrões ortogonais não possui redundância entre seus elementos. É esperado que a redundância de um determinado conjunto de padrões ortogonais seja

TID	Class	Itemset
1	$CLASS = 1$	AB
2	$CLASS = 2$	ABC
3	$CLASS = 1$	ABD
4	$CLASS = 1$	$ABDE$
5	$CLASS = 1$	ABE
6	$CLASS = 3$	ADE
7	$CLASS = 2$	BC
8	$CLASS = 2$	BCD
9	$CLASS = 2$	BCE
10	$CLASS = 3$	DE

Tabela 5.1: Base de Dados de Treinamento para Exemplo

Parameter	Value
support	0.1
confidence	0.1
min-num-rules	1
max-num-rank-rules	1000
min-rule-len	1
max-rule-len	1000

Tabela 5.2: Parâmetros para Exemplo de Execução da Abordagem LAC

sempre menor que a redundância do conjunto original de padrões freqüentes.

O aplicativo também foi executado com as outras duas métricas de ortogonalidade (cobertura de transações e cobertura de classes), extraindo, respectivamente, $\mathcal{O} = \{AB, E\}$ e $\mathcal{O} = \{B, E\}$ como conjuntos de padrões ortogonais, e estas execuções também classificaram a instância de teste corretamente.

5.3 Experimentos

Nesta seção serão apresentados os resultados das execuções do aplicativo em 26 bases de dados do repositório **UCI** (*UC Irvine Machine Learning Repository*), amplamente utilizado em pesquisas na área de classificação em mineração de dados.

5.3.1 Metodologia

A metodologia de testes adotada nos experimentos é baseada em trabalhos anteriores relacionados com classificação associativa que realizam experimentos em bases de dados do mesmo repositório (Veloso et al., 2006b; Yin e Han, 2003; Li et al., 2001; Liu et al., 1998; Friedman et al., 1996).

Todas as bases utilizadas durante os testes foram reordenadas aleatoriamente e particionadas em dez sub-conjuntos, de onde foram criadas dez configurações de teste para cada uma delas. Cada configuração de teste consiste de uma parte (um sub-conjunto da base) como arquivo de teste, e nove partes (os nove sub-conjuntos restantes da base) como arquivo de treinamento. Como resultados foram considerados a média das dez execuções diferentes para cada base de dados.

Os parâmetros utilizados nos testes se encontram na tabela 5.3. Todas as combinações possíveis destes parâmetros foram realizadas, com exceção da combinação tamanho máximo de regra 1 e métrica de ortogonalidade s para o OLAC [1], e dos casos em que α é maior ou igual a β para o ORIGAMI. É importante lembrar que alguns dos parâmetros desta tabela não são utilizados por todas as abordagens.

A seguir serão apresentados os resultados obtidos pelos experimentos. Na seção 5.3.2 serão apresentados os melhores resultados obtidos para as três abordagens, onde o conjunto de parâmetros de execução utilizado foi o mais adequado para cada aplicação e base de dados. Já na seção 5.3.3 serão apresentados os melhores resultados obtidos onde o conjunto de parâmetros de execução utilizado em cada abordagem foi o que produziu a melhor média de acurácia para todas as bases.

Parâmetros	**Valores**
support	$\{0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 1\}$
confidence	$\{0.0001, 0.001, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 1\}$
min-num-rules	$\{1\}$
max-num-rank-rules	$\{1, 10, 100, 1000, 10000, 100000, 1000000\}$
min-rule-len	$\{1\}$
max-rule-len	$\{1, 2, 3\}$
rule-measure	$\{s, c, j, k, o, n, e, p, l, i, v\}$
orth-metric	$\{e, c, l, a\}$
orth-method	$\{s, p\}$
orth-pat-ordering	$\{s, r, i, z, n\}$
origami-alpha	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$
origami-beta	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$

Tabela 5.3: Parâmetros Utilizados Durante os Experimentos para Todas as Abordagens

Abaixo, encontramos alguns gráficos comparando execuções das três abordagens implementadas: LAC, OLAC e ORIGAMI.

5.3.2 Melhores Resultados para Cada Base

Na figura 5.1 encontramos um histograma de acurácia obtida com as três abordagens para cada base de dados. A figura 5.2 mostra um histograma com o número médio

[1] Quando a métrica de ortogonalidade baseada na estrutura dos padrões é utilizada em conjunto com o tamanho máximo de regra igual a 1, OLAC terá o mesmo comportamento que LAC, já que a similaridade entre dois padrões freqüentes será sempre zero.

de padrões utilizados para se obter as regras de associação para cada instância de teste, a figura 5.3 mostra o número médio de regras geradas para cada instância pelas três abordagens, e a figura 5.4 mostra o tempo médio de cada classificação. Para estes quatro resultados foram utilizados os melhores conjuntos de parâmetros para cada aplicação e base de dados, por exemplo, para a abordagem OLAC para a base *anneal.ac* foi utilizado suporte igual a 0.0001, mas para a base *horse.ac* foi utilizado suporte igual a 0.3, já que os melhores resultados para cada base foram obtidos com estes valores.

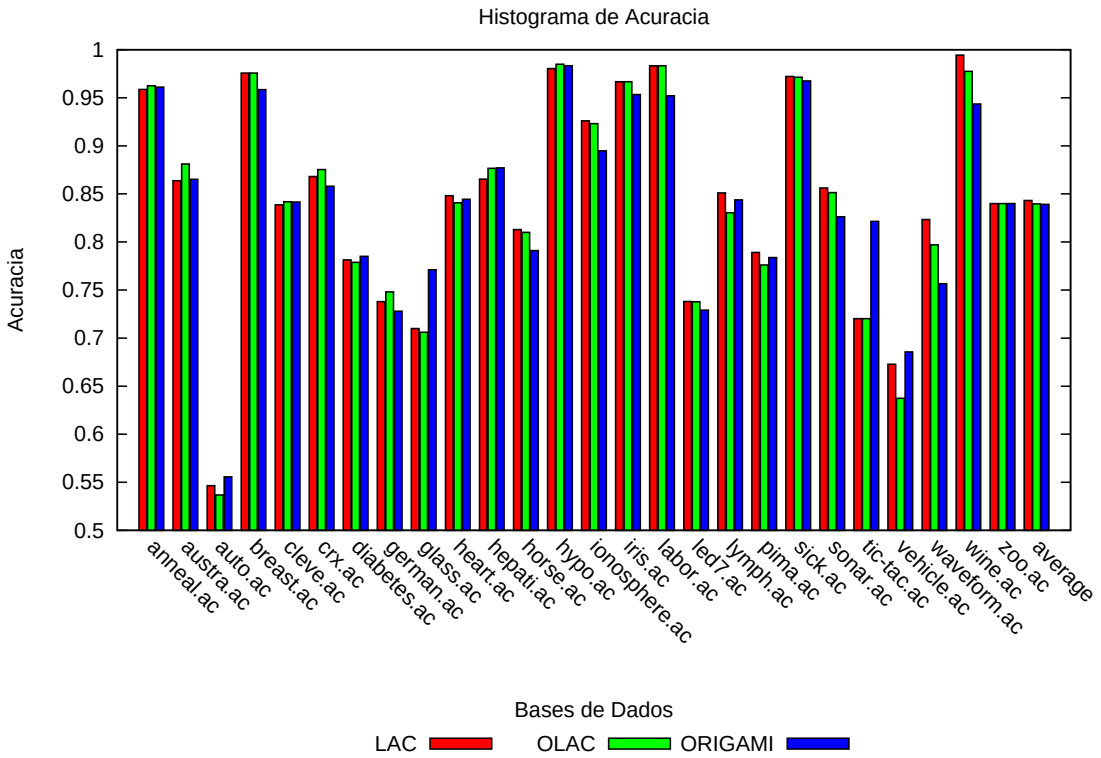

Figura 5.1: Histograma de Acurácia (melhores resultados para cada base)

Como podemos ver na figura 5.1, os valores de acurácia obtidos com as três versões do algoritmo estão muito próximas (de fato, este comportamento será comprovado na seção 5.3.4). As médias obtidas para as abordagens LAC, OLAC e ORIGAMI foram, respectivamente, 0.843, 0.840 e 0.839.

O número de padrões gerados pelas abordagens OLAC e ORIGAMI é sempre menor ou igual ao número de padrões gerados pela abordagem LAC. Na figura 5.2 vemos que, em alguns casos, as três abordagens utilizam uma quantidade de padrões freqüentes bem semelhante para gerar as regras, como na base *crx.ac*. Porém, em casos como os das bases *auto.ac*, *vehicle.ac* ou *waveform.ac*, a diferença chega a duas ou três ordens de grandeza. Em média, foi possível reduzir o número de padrões em uma ordem de

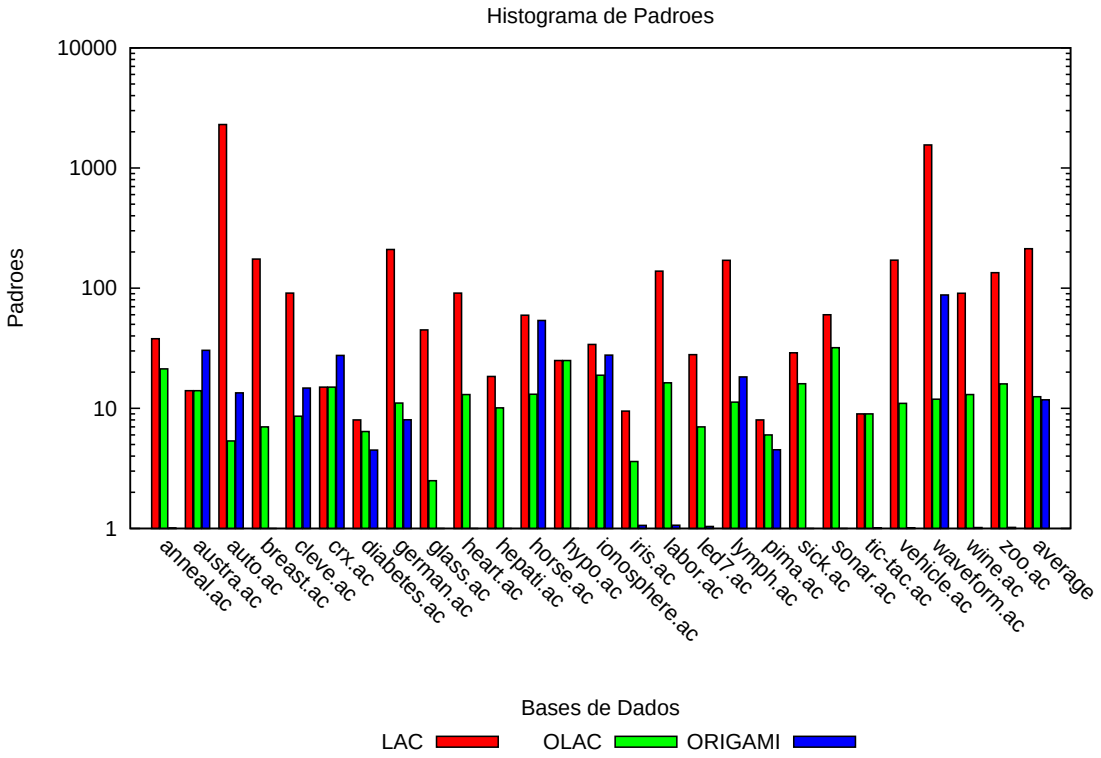

Figura 5.2: Histograma de Padrões (melhores resultados para cada base)

grandeza utilizando a ortogonalidade.

O número de regras geradas pelas abordagens ortogonais também se mostrou sempre menor ou igual ao número de regras geradas pela abordagem não ortogonal, com exceção da base *hepati.ac*, em que o OLAC gerou, em média, 19.78 regras, e o LAC gerou apenas 18.43, e das bases *austra.ac* e *crx.ac*, em que o ORIGAMI gerou mais regras que o LAC.

É possível perceber o motivo pelo qual o OLAC gerou mais regras que o LAC para a base *hepati.ac* analisando as tabelas 5.4 e 5.5. Nessas tabelas, onde temos os parâmetros das melhores execuções para cada base de dados, vemos que a melhor execução do LAC para esta base foi obtida com confiança igual a 0.5, o que reduz o número de regras válidas consideravelmente. Já para o OLAC, a melhor execução para esta base foi obtida com confiança igual a 0.0001.

A mesma observação é feita para o caso do ORIGAMI, analisando a tabela 5.6. Para as bases *austra.ac* e *crx.ac*, a melhor execução do LAC foi obtida com confiança igual a 0.3. Já para o ORIGAMI, a confiança foi de 0.0001 nos dois casos.

De qualquer forma, com as abordagens ortogonais foi possível classificar as instâncias de teste com um número de regras, em média, dez vezes menor que com a abordagem original. Como já era esperado, o LAC possui o menor tempo de execução,

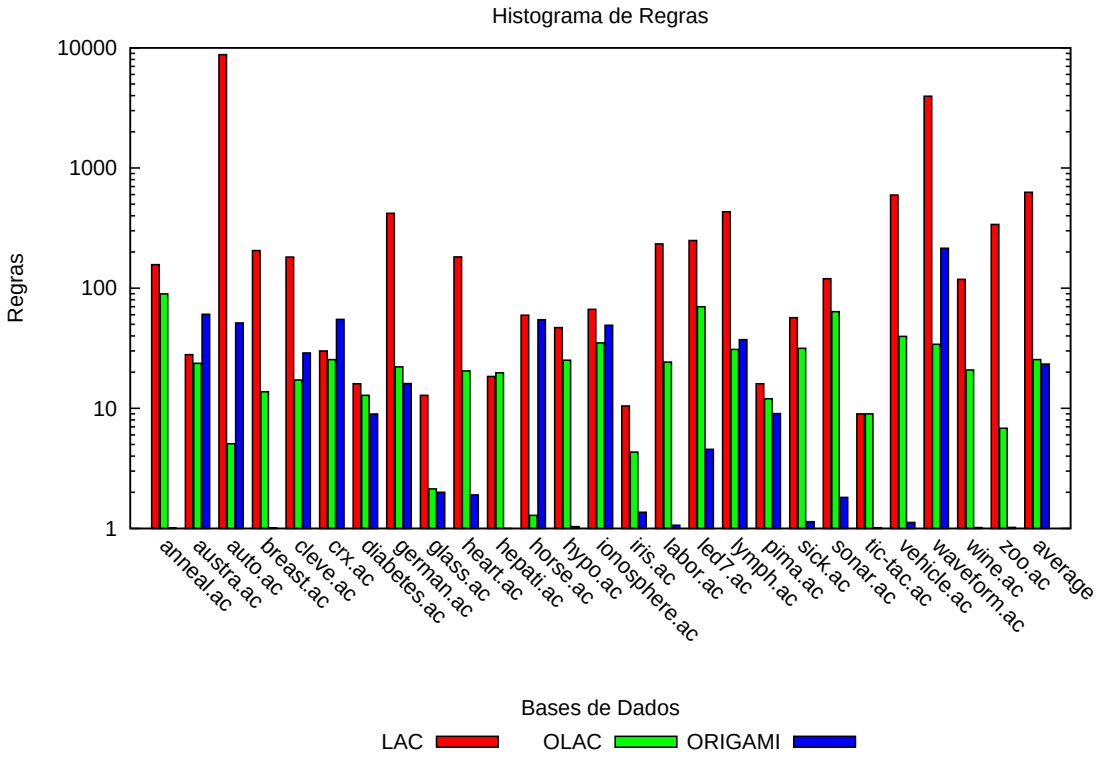

Figura 5.3: Histograma de Regras (melhores resultados para cada base)

visto que o OLAC, além de obter o conjunto de padrões freqüentes, ainda executa uma heurística para extrair o conjunto de padrões ortogonais, e o ORIGAMI possui duas heurísticas que penalizam o seu tempo de execução: a geração de padrões maximais e a heurística de ortogonalidade.

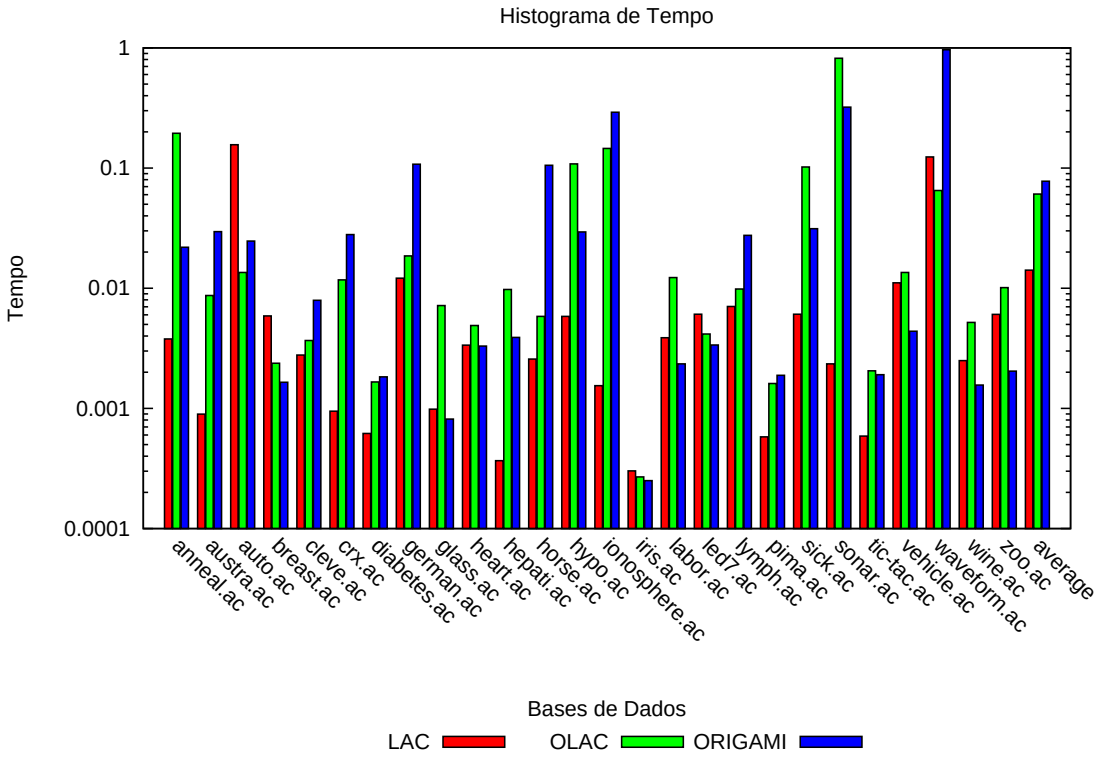

Figura 5.4: Histograma de Tempo (melhores resultados para cada base)

Como já foi dito, na tabela 5.4 se encontram os parâmetros que obtiveram as melhores acurácias para o LAC, e os resultados de cada execução. Analisando a tabela, vemos que a maior parte dos melhores resultado foi obtida utilizando um pequeno valor para suporte (em geral, entre 0.0001 e 0.1) e para confiança (0.01 em média). Isto sugere que, em grande parte das bases, existam transações cujas classes sejam bem representadas por padrões de freqüência muito baixa. A seguir, veremos que os melhores resultados para as abordagens ortogonais também foram obtidos com valores semelhantes a estes.

dataset	**s**	**c**	**n**	**l**	**m**	**x**	**u**	**pat.**	**rul.**	**tim.**	**acc.**
anneal.ac	0.0001	0.01	1	1000	1	1	n	37.99	156.94	0.00	0.96
austra.ac	0.0001	0.01	1	100	1	1	n	14.00	28.00	0.00	0.86
auto.ac	0.0001	0.01	1	10000	1	3	c	2302.02	8770.13	0.16	0.55
breast.ac	0.0001	0.1	1	1000	1	3	c	174.43	205.40	0.01	0.98
cleve.ac	0.0001	0.01	1	100	1	2	i	90.90	181.36	0.00	0.84
crx.ac	0.0001	0.01	1	100	1	1	n	15.00	29.99	0.00	0.87
diabetes.ac	0.0001	0.01	1	100	1	1	n	8.00	16.00	0.00	0.78
german.ac	0.0001	0.01	1	1000	1	2	n	209.91	419.20	0.01	0.74
glass.ac	0.01	0.6	1	100	1	2	c	44.88	12.83	0.00	0.71
heart.ac	0.0001	0.01	1	1000	1	2	n	91.00	182.00	0.00	0.85
hepati.ac	0.1	0.5	1	1	1	1	c	18.44	18.44	0.00	0.87
horse.ac	0.3	0.5	1	100	1	2	c	59.60	59.60	0.00	0.81
hypo.ac	0.0001	0.01	1	100	1	1	n	25.00	46.94	0.01	0.98
ionosphere.ac	0.0001	0.01	1	1000	1	1	n	34.00	66.72	0.00	0.93
iris.ac	0.1	0.3	1	10	1	2	c	9.47	10.47	0.00	0.97
labor.ac	0.0001	0.1	1	1000	1	2	p	138.73	233.42	0.00	0.98
led7.ac	0.0001	0.01	1	1000	1	2	p	28.00	249.25	0.01	0.74
lymph.ac	0.0001	0.01	1	1000	1	2	j	170.25	432.53	0.01	0.85
pima.ac	0.0001	0.01	1	10	1	1	i	8.00	16.00	0.00	0.79
sick.ac	0.0001	0.01	1	100	1	1	n	29.00	56.75	0.01	0.97
sonar.ac	0.0001	0.01	1	1000	1	1	p	60.00	119.97	0.00	0.86
tic-tac.ac	0.1	0.5	1	1	1	1	c	9.00	9.00	0.00	0.72
vehicle.ac	0.0001	0.01	1	1000	1	2	n	170.89	595.59	0.01	0.67
waveform.ac	0.0001	0.0001	1	100000	1	3	c	1555.12	3956.62	0.12	0.82
wine.ac	0.01	0.3	1	1000	1	2	c	90.58	118.38	0.00	0.99
zoo.ac	0.0001	0.1	1	1000	1	2	k	134.59	338.64	0.01	0.84

Tabela 5.4: Melhores Parâmetros e Resultados para cada Base de Dados (LAC)

A tabela 5.5 possui os parâmetros das melhores execuções do OLAC para cada base de dados. Analisando a tabela, vemos que a maior parte dos melhores resultado foi obtida utilizando regras de tamanho máximo igual a 2, e métrica de ortogonalidade baseada na estrutura dos padrões.

O motivo pelo qual as métricas baseadas em cobertura de classes e de transações não produziram resultados melhores pode estar relacionado com a heurística de obtenção de conjuntos ortogonais. Examinando os resultados de tais execuções, percebemos que, em muitas situações, o algoritmo obteve conjuntos pequenos de padrões ortogonais (2 a 4 padrões) com uma medida muito alta de ortogonalidade. Acontece que, de acordo com a nossa heurística, se a adição de um elemento a mais ao conjunto-solução diminui a medida de ortogonalidade do mesmo, este elemento não é adicionado, e o conjunto anterior é tido como resultado.

Um grande número de fatores contribui para que a nossa heurística encontre conjuntos de padrões com alto valor de ortogonalidade para as métricas baseadas em cobertura, a começar pela natureza dos dados. É fácil perceber que padrões pouco freqüentes na base possuem maior probabilidade de fazer parte de conjuntos ortogonais, visto que a sobreposição de cobertura entre estes padrões, tanto de classes quanto de transações, acontecerá em menor freqüência.

Uma forma de evitar que padrões pouco freqüentes sejam inseridos no conjunto-solução é aumentar o valor do suporte utilizado pelo algoritmo, porém, fazendo isto, não será possível classificar corretamente transações cujas classes são pouco freqüentes, ou são bem representadas por padrões pouco freqüentes.

dataset	s	c	n	l	m	x	u	e	w	g	pat.	rul.	tim.	acc.
anneal.ac	0.0001	0.0001	1	100	1	2	n	s	s	s	21.27	89.69	0.19	0.96
austra.ac	0.0001	0.3	1	100	1	2	n	s	s	i	14.00	23.70	0.01	0.88
auto.ac	0.0001	0.4	1	100	1	2	n	l	s	r	5.35	5.08	0.01	0.54
breast.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	7.00	13.71	0.00	0.98
cleve.ac	0.0001	0.0001	1	100	1	2	i	s	s	s	8.60	17.18	0.00	0.84
crx.ac	0.0001	0.3	1	100	1	2	n	s	s	i	15.00	25.43	0.01	0.88
diabetes.ac	0.0001	0.0001	1	100	1	2	n	s	s	s	6.41	12.81	0.00	0.78
german.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	11.06	22.10	0.02	0.75
glass.ac	0.001	0.4	1	10	1	3	n	c	s	z	2.50	2.13	0.01	0.71
heart.ac	0.0001	0.3	1	100	1	2	n	s	s	i	13.00	20.46	0.00	0.84
hepati.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	10.08	19.79	0.01	0.88
horse.ac	0.3	0.7	1	10	1	2	c	s	s	n	13.11	1.29	0.01	0.81
hypo.ac	0.0001	0.3	1	100	1	2	n	s	s	i	25.00	25.14	0.11	0.98
ionosphere.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	18.87	35.05	0.15	0.92
iris.ac	0.01	0.3	1	100	1	2	c	s	s	s	3.61	4.32	0.00	0.97
labor.ac	0.0001	0.3	1	100	1	2	p	s	s	i	16.29	24.23	0.01	0.98
led7.ac	0.0001	0.0001	1	100	1	2	k	s	s	i	7.00	70.00	0.00	0.74
lymph.ac	0.01	0.0001	1	100	1	2	n	s	s	s	11.26	30.96	0.01	0.83
pima.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	6.00	12.00	0.00	0.78
sick.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	16.00	31.58	0.10	0.97
sonar.ac	0.0001	0.0001	1	100	1	2	i	s	s	s	31.98	63.82	0.82	0.85
tic-tac.ac	0.001	0.5	1	1	1	2	l	s	s	n	9.00	9.00	0.00	0.72
vehicle.ac	0.0001	0.0001	1	100	1	2	n	s	s	z	10.99	39.65	0.01	0.64
waveform.ac	0.0001	0.0001	1	100	1	2	i	s	s	z	11.92	34.07	0.07	0.80
wine.ac	0.0001	0.3	1	100	1	2	n	s	s	i	13.00	20.84	0.01	0.98
zoo.ac	0.001	0.5	1	1	1	2	l	s	s	n	15.99	6.82	0.01	0.84

Tabela 5.5: Melhores Parâmetros e Resultados para cada Base de Dados (OLAC)

A tabela 5.6 possui os parâmetros das melhores execuções do ORIGAMI para cada base de dados. É interessante notar que, em boa parte das bases de dados, o algoritmo conseguiu extrair apenas 1 padrão ortogonal. Na maioria destes casos, utilizando valores de α muito pequenos. É fácil perceber que os parâmetros α e β estão diretamente ligados à quantidade de padrões ortogonais extraídos pela abordagem, visto que, quanto menor o tamanho de α, menor será o número de padrões considerados ortogonais no conjunto, e quanto maior o tamanho de β, maior será o número de padrões não representados por pelos candidatos ortogonais.

Novamente, neste caso, percebemos a predominância da métrica baseada na estrutura dos padrões nos melhores resultados obtidos.

dataset	**s**	**c**	**n**	**l**	**u**	**e**	**a**	**b**	**pat.**	**rul.**	**tim.**	**acc.**
anneal.ac	0.0001	0.0001	1	10	c	s	0.7	0.9	1.01	1.01	0.02	0.96
austra.ac	0.1	0.0001	1	100	c	s	0.7	0.8	30.35	60.62	0.03	0.87
auto.ac	0.1	0.0001	1	100	c	s	0.7	0.8	13.47	51.30	0.02	0.56
breast.ac	0.0001	0.0001	1	10	o	s	0.1	0.5	1.00	1.01	0.00	0.96
cleve.ac	0.1	0.0001	1	100	c	s	0.7	0.8	14.76	28.91	0.01	0.84
crx.ac	0.1	0.0001	1	100	c	s	0.7	0.8	27.52	54.99	0.03	0.86
diabetes.ac	0.1	0.0001	1	10	k	s	0.7	0.8	4.49	8.97	0.00	0.79
german.ac	0.1	0.0001	1	10	e	c	0.2	0.8	8.03	16.05	0.11	0.73
glass.ac	0.0001	0.0001	1	10	c	s	0.1	0.5	1.00	2.00	0.00	0.77
heart.ac	0.1	0.0001	1	10	e	s	0.2	0.8	1.00	1.90	0.00	0.84
hepati.ac	0.001	0.001	1	100	s	s	0.1	0.8	1.00	1.00	0.00	0.88
horse.ac	0.1	0.5	1	100	c	s	0.7	0.8	53.77	54.63	0.11	0.79
hypo.ac	0.0001	0.0001	1	10	c	s	0.3	0.5	1.00	1.04	0.03	0.98
ionosphere.ac	0.1	0.0001	1	100	c	s	0.7	0.8	27.71	49.06	0.29	0.89
iris.ac	0.1	0.0001	1	10	n	s	0.2	0.8	1.06	1.37	0.00	0.95
labor.ac	0.0001	0.001	1	1	v	s	0.2	0.6	1.07	1.07	0.00	0.95
led7.ac	0.001	0.0001	1	10	c	l	0.1	0.4	1.04	4.56	0.00	0.73
lymph.ac	0.1	0.001	1	100	n	s	0.6	0.7	18.25	37.23	0.03	0.84
pima.ac	0.1	0.0001	1	10	l	s	0.7	0.9	4.52	9.03	0.00	0.78
sick.ac	0.0001	0.0001	1	10	c	s	0.2	0.7	1.00	1.14	0.03	0.97
sonar.ac	0.1	0.0001	1	10	i	s	0.2	0.9	1.00	1.81	0.32	0.83
tic-tac.ac	0.0001	0.0001	1	1	c	s	0.2	0.7	1.01	1.02	0.00	0.82
vehicle.ac	0.0001	0.0001	1	10	c	s	0.2	0.8	1.01	1.12	0.00	0.69
waveform.ac	0.01	0.0001	1	100	c	s	0.7	0.8	87.71	214.78	0.97	0.76
wine.ac	0.0001	0.0001	1	1	c	s	0.1	0.7	1.02	1.02	0.00	0.94
zoo.ac	0.0001	0.6	1	10	p	s	0.3	0.7	1.02	1.02	0.00	0.84

Tabela 5.6: Melhores Parâmetros e Resultados para cada Base de Dados (ORIGAMI)

5.3.3 Melhor Média dos Resultados para Todas as Bases

Na figura 5.5 se encontra o histograma de acurácia obtido com as abordagens LAC, OLAC e ORIGAMI. A figura 5.6 possui um histograma com a média do número de padrões utilizados por cada classificação, a figura 5.7 possui a média do número de

regras obtidas em cada classificação, e a figura 5.8 possui a média do tempo necessário para a classificação de cada instância de teste. Para estes quatro resultados foi utilizado, para cada aplicação, o conjunto de parâmetros que produziu o melhor valor médio de acurácia para todo o conjunto de bases. Os conjuntos de parâmetros para cada abordagem se encontram na tabela 5.7, sendo que alguns deles não são aplicáveis a todas as abordagens.

	LAC	**OLAC**	**ORIGAMI**
support	0.001	0.0001	0.0001
confidence	0.01	0.0001	0.0001
min-num-rules	1	1	1
max-num-rank-rules	1000	100	10
min-rule-len	1	1	-
max-rule-len	1	2	-
rule-measure	n	n	c
orth-metric	-	s	s
orth-method	-	s	-
orth-pat-ordering	-	s	-
origami-alpha	-	-	0.1
origami-beta	-	-	0.8

Tabela 5.7: Melhores Parâmetros para cada Execução

Como podemos ver, na figura 5.5, as melhores execuções de cada uma das três abordagens obtiveram resultados semelhantes novamente (fato que também será comprovado na seção 5.3.4). As médias obtidas para as abordagens LAC, OLAC e ORIGAMI foram, respectivamente, 0.808, 0.813 e 0.782. É interessante notar que a melhor execução do OLAC foi obtida com tamanho máximo de regra igual a 2 (ou seja, no máximo dois itens do lado esquerdo da regra), enquanto que a melhor execução do LAC teve tamanho máximo de regra igual a 1. Isto faz com que o OLAC encontre uma quantidade de padrões freqüentes muito maior que o LAC, no entanto, o tamanho do *ranking* de regras do OLAC é menor que o do LAC, já que a ortogonalidade diminui o número de padrões considerados durante a geração das regras.

Na tabela 5.6 vemos as quantidades de padrões gerados por cada abordagem. A diferença entre os resultados obtidos com o LAC e o OLAC neste histograma não é tão grande quanto no resultado anterior por influência dos parâmetros de execução. Além da diferença entre o tamanho máximo das regras, que favorece o LAC [2], o suporte utilizado pelo OLAC é 10 vezes menor que o suporte do LAC.

[2]Os padrões freqüentes utilizados pelo classificador são obtidos por meio de uma busca em largura no espaço de padrões, limitada pelo tamanho máximo da regra. Logo, o algoritmo só obtém os padrões de tamanho menor ou igual a este parâmetro

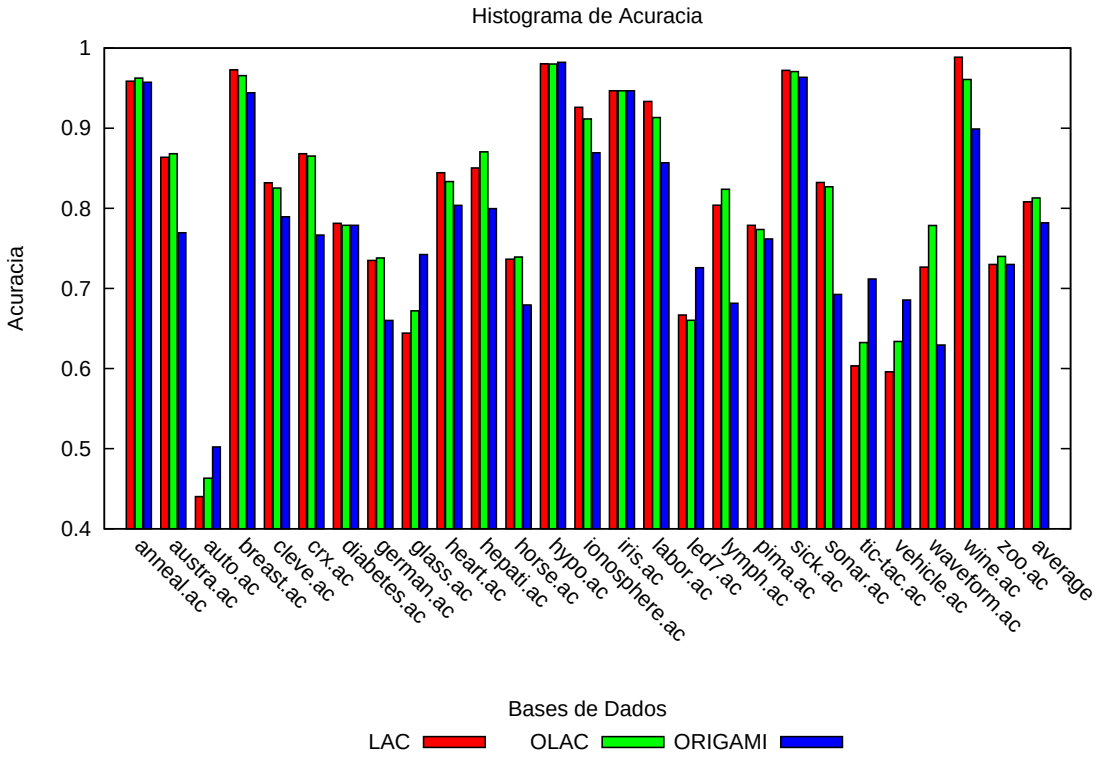

Figura 5.5: Histograma de Acurácia (melhores resultados para cada abordagem)

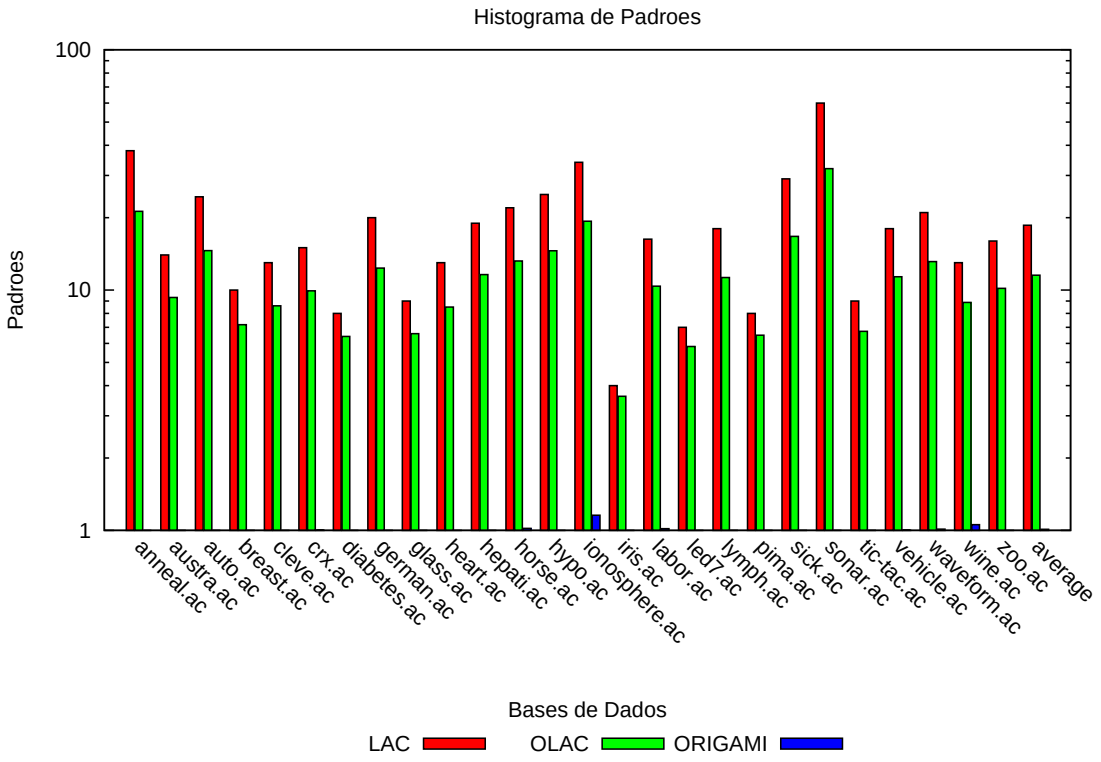

Figura 5.6: Histograma de Padrões (melhores resultados para cada abordagem)

Analisando um exemplo de classificação podemos compreender como a utilização de padrões ortogonais pode auxiliar na classificação das transações. Para isso, usamos, como exemplo, uma configuração de teste da base **austra.ac**. A configuração selecionada possui 621 transações de treinamento e 69 instâncias de teste, das quais o OLAC classificou 61 de maneira correta, e o LAC acertou 60 vezes. A instância da tabela 5.8 foi classificada corretamente pelo OLAC, mas não pelo LAC. Esta instância possui 14 itens, sendo todos eles freqüentes (suporte maior que 0.0001) na projeção obtida, que corresponde às 621 transações da base de treinamento. Como o LAC foi executado com tamanho máximo das regras igual a 1, foram encontrados 14 padrões freqüentes (tabela 5.9), com os quais foram geradas 28 regras de associação com confiança maior ou igual a 0.01 (tabela 5.10).

Classe	**Transação**
CLASS=1	w=attr3=4.2075+ w=attr4=2 w=attr9=0 w=attr10=-0.5 w=attr11=0 w=attr14=-493 w=attr1=1 w=attr6=4 w=attr13=99.5+ w=attr2=38.96+ w=attr8=1 w=attr7=1.02+ w=attr5=7 w=attr12=1

Tabela 5.8: Exemplo de Classificação - Instância de Teste da base *austra.ac*

Padrão	**Suporte**
w=attr3=4.2075+	0.404187
w=attr4=2	0.756844
w=attr9=0	0.576490
w=attr10=-0.5	0.576490
w=attr11=0	0.541063
w=attr14=-493	0.764895
w=attr1=1	0.679549
w=attr6=4	0.586151
w=attr13=99.5+	0.692432
w=attr2=38.96+	0.230274
w=attr8=1	0.516908
w=attr7=1.02+	0.470209
w=attr5=7	0.054750
w=attr12=1	0.083736

Tabela 5.9: Exemplo de Classificação - Padrões Freqüentes Encontrados pelo LAC

Já o OLAC, que foi executado com tamanho máximo de regra igual a 2, obteve um conjunto de 105 padrões freqüentes, dos quais foram extraídos 9 padrões ortogonais (tabela 5.11), e, a partir destes, foram geradas 18 regras com confiança maior ou igual a 0.001 (tabela 5.12).

Examinando as tabelas 5.10 e 5.12, é possível perceber algumas características interessantes que aparecem nas regras obtidas pelo OLAC. Vemos, por exemplo, que

Ranking	Regra	Convicção
1	CLASS=1 ⇐ w=attr8=1	2.540859
2	CLASS=0 ⇐ w=attr12=1	1.898014
3	CLASS=0 ⇐ w=attr9=0	1.742280
4	CLASS=0 ⇐ w=attr10=-0.5	1.742280
5	CLASS=1 ⇐ w=attr7=1.02+	1.548142
6	CLASS=1 ⇐ w=attr2=38.96+	1.409922
7	CLASS=1 ⇐ w=attr3=4.2075+	1.330766
8	CLASS=0 ⇐ w=attr14=-493	1.316782
9	CLASS=0 ⇐ w=attr5=7	1.241009
10	CLASS=0 ⇐ w=attr13=99.5+	1.199627
11	CLASS=1 ⇐ w=attr4=2	1.091481
12	CLASS=0 ⇐ w=attr6=4	1.062888
13	CLASS=0 ⇐ w=attr11=0	1.043752
14	CLASS=0 ⇐ w=attr1=1	1.015590
15	CLASS=1 ⇐ w=attr1=1	0.988178
16	CLASS=1 ⇐ w=attr11=0	0.968364
17	CLASS=1 ⇐ w=attr6=4	0.955920
18	CLASS=0 ⇐ w=attr4=2	0.902901
19	CLASS=1 ⇐ w=attr13=99.5+	0.885196
20	CLASS=1 ⇐ w=attr5=7	0.868540
21	CLASS=1 ⇐ w=attr14=-493	0.842109
22	CLASS=0 ⇐ w=attr3=4.2075+	0.758199
23	CLASS=1 ⇐ w=attr10=-0.5	0.750727
24	CLASS=1 ⇐ w=attr9=0	0.750727
25	CLASS=1 ⇐ w=attr12=1	0.730596
26	CLASS=0 ⇐ w=attr2=38.96+	0.728308
27	CLASS=0 ⇐ w=attr7=1.02+	0.687618
28	CLASS=0 ⇐ w=attr8=1	0.562396

Tabela 5.10: Exemplo de Classificação - Regras de Associação Encontradas pelo LAC

Padrão	Suporte
w=attr3=4.2075+	0.404187
w=attr10=-0.5	0.576490
w=attr11=0	0.541063
w=attr9=0 w=attr14=-493	0.504026
w=attr4=2 w=attr2=38.96+	0.190016
w=attr6=4 w=attr13=99.5+	0.426731
w=attr8=1 w=attr7=1.02+	0.334944
w=attr1=1 w=attr5=7	0.049919
w=attr12=1	0.083736

Tabela 5.11: Exemplo de Classificação - Padrões Freqüentes Encontrados pelo OLAC

Ranking	**Regra**	**Convicção**
1	CLASS=1 ⇐ w=attr8=1 w=attr7=1.02+	3.770817
2	CLASS=0 ⇐ w=attr9=0 w=attr14=-493	2.016103
3	CLASS=0 ⇐ w=attr12=1	1.898014
4	CLASS=0 ⇐ w=attr10=-0.5	1.742280
5	CLASS=1 ⇐ w=attr4=2 w=attr2=38.96+	1.617454
6	CLASS=1 ⇐ w=attr3=4.2075+	1.330766
7	CLASS=0 ⇐ w=attr6=4 w=attr13=99.5+	1.221798
8	CLASS=0 ⇐ w=attr1=1 w=attr5=7	1.131508
9	CLASS=0 ⇐ w=attr11=0	1.043752
10	CLASS=1 ⇐ w=attr11=0	0.968364
11	CLASS=1 ⇐ w=attr1=1 w=attr5=7	0.916942
12	CLASS=1 ⇐ w=attr6=4 w=attr13=99.5+	0.876054
13	CLASS=0 ⇐ w=attr3=4.2075+	0.758199
14	CLASS=1 ⇐ w=attr10=-0.5	0.750727
15	CLASS=1 ⇐ w=attr12=1	0.730596
16	CLASS=1 ⇐ w=attr9=0 w=attr14=-493	0.717980
17	CLASS=0 ⇐ w=attr4=2 w=attr2=38.96+	0.671226
18	CLASS=0 ⇐ w=attr8=1 w=attr7=1.02+	0.514716

Tabela 5.12: Exemplo de Classificação - Regras de Associação Encontradas pelo OLAC

a primeira regra do *ranking*, que possui convicção próxima de 3.77, é uma regra que contribui positivamente para o resultado correto do algoritmo, e foi obtida por meio da combinação de dois padrões que, na abordagem LAC, geraram regras com convicções próximas de 2.54 e 1.55 respectivamente. Nota-se que a combinação dos dois padrões, além de diminuir a redundância das regras, melhorou a sua medida de convicção, fazendo com que o algoritmo se aproximasse do resultado correto.

E adicionalmente, podemos perceber também que a última regra do *ranking* da abordagem OLAC, que possui convicção igual a 0.51 e contribui negativamente para o resultado, e foi obtida por meio da combinação de duas regras da abordagem LAC com convicções iguais a 0.69 e 0.56, respectivamente, ou seja, nesse caso, a medida de convicção diminuiu. Logo, temos aqui um exemplo de como a ortogonalidade pode diminuir a redundância nas informações obtidas, e ainda melhorar a efetividade do algoritmo.

Este fato é notado com freqüência nos casos em que o LAC é executado com o tamanho máximo de regra maior que 1, pois neste caso, o número muito maior de regras será gerado, fazendo com haja redundância tanto das regras que contribuem positivamente para o sucesso da classificação quanto das que contribuem negativamente. Entretanto, não é raro encontrar transações nas bases de dados que são definidas por um número muito pequeno de padrões, e nestes casos, a redundância das regras falsas

exerce maiores influências que a redundância das regras verdadeiras. É justamente nestes casos que a aplicação da ortogonalidade favorece o algoritmo.

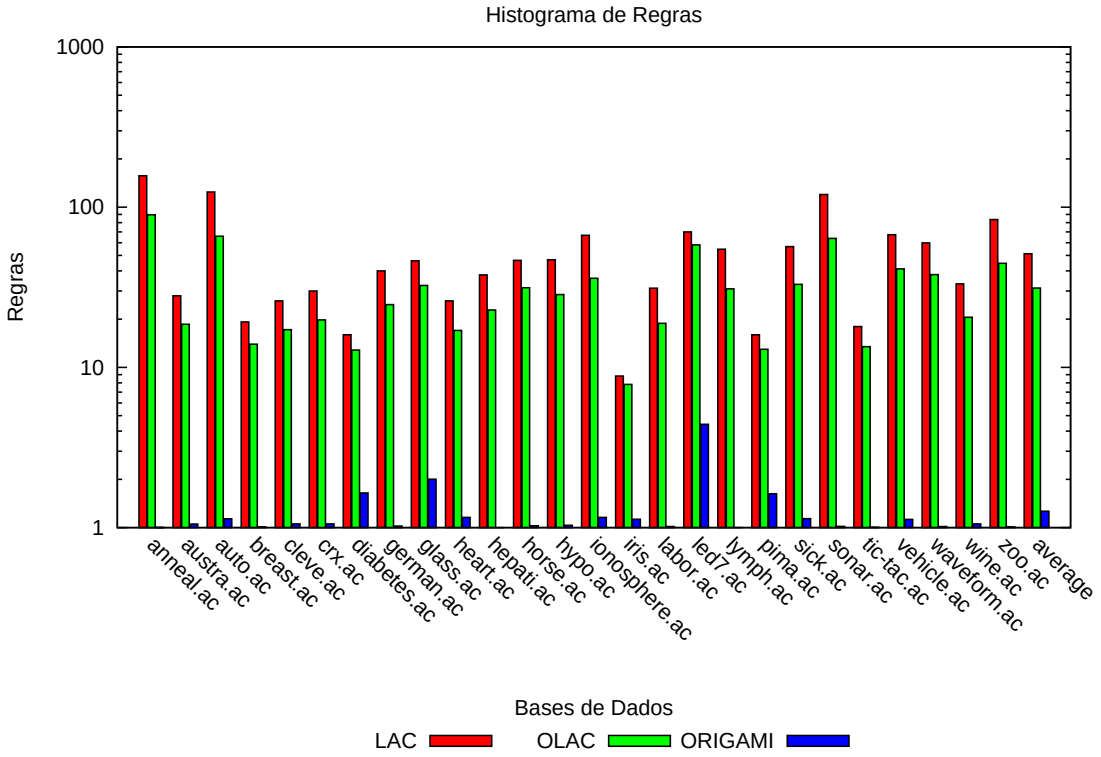

Figura 5.7: Histograma de Regras (melhores resultados para cada abordagem)

Em relação ao ORIGAMI, esta abordagem obteve apenas um padrão freqüente e ortogonal para a maior parte das bases. Isto se deve à combinação dos parâmetros utilizados na sua melhor execução. Um suporte baixo favorece a geração de grandes padrões maximais, que tendem a se aproximar do padrão máximo obtido com todos os itens da instância de teste. Como a métrica de ortogonalidade baseada na estrutura dos padrões foi utilizada, isto torna difícil a obtenção de conjuntos ortogonais. Mesmo utilizando um valor baixo para o parâmetro α, a probabilidade de dois padrões maximais possuírem itens em comum é muito grande. Em grande parte dos experimentos, o tamanho médio dos padrões maximais ficou bem próximo do tamanho da instância de teste.

Novamente, como era de se esperar, o LAC obteve o melhor desempenho das três abordagens.

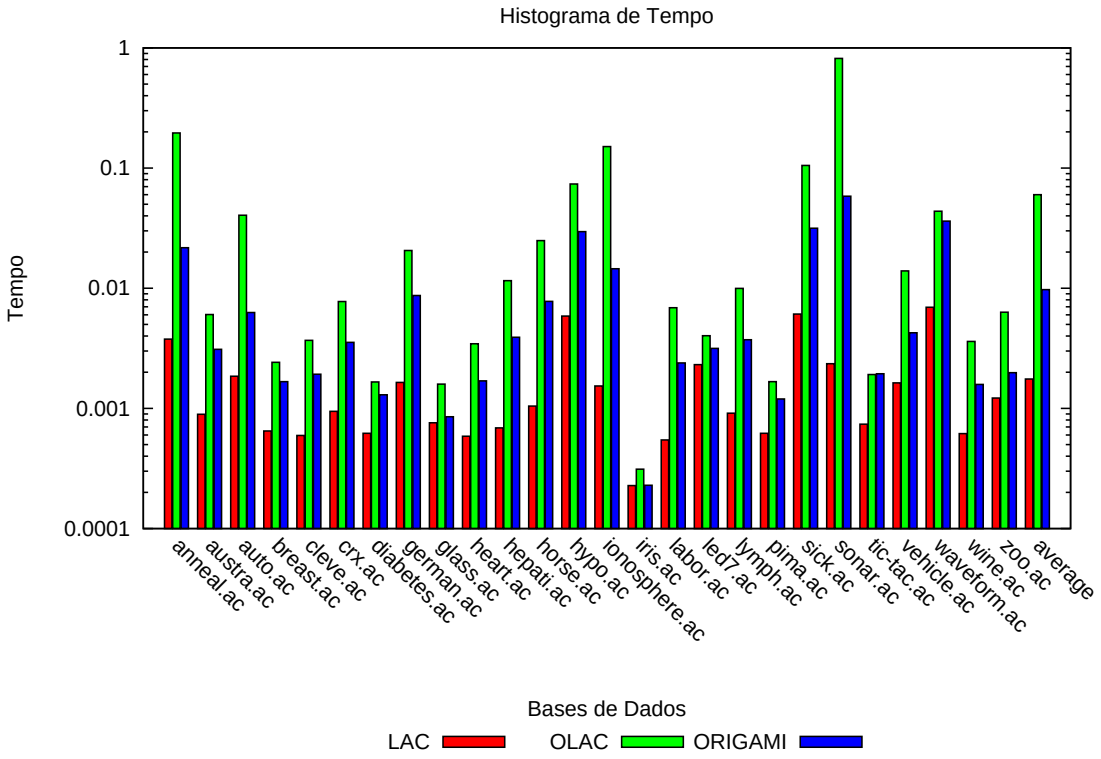

Figura 5.8: Histograma de Tempo (melhores resultados para cada abordagem)

5.3.3.1 Comparação das Métricas de Ortogonalidade (OLAC)

Na figura 5.9 encontramos os valores de acurácia obtidos após a execução da abordagem OLAC com cada uma das métricas de ortogonalidade implementadas, além dos resultados obtidos com a utilização da multiplicação das três métricas como medida de ortogonalidade. Na figura 5.10 encontramos o histograma de padrões, e em 5.11 encontramos o histograma de regras. Na figura 5.12 temos o tempo médio de classificação de uma instância de teste para cada métrica. Todos estes resultados foram obtidos com os parâmetros do OLAC encontrados na tabela 5.7 variando-se a métrica de ortogonalidade.

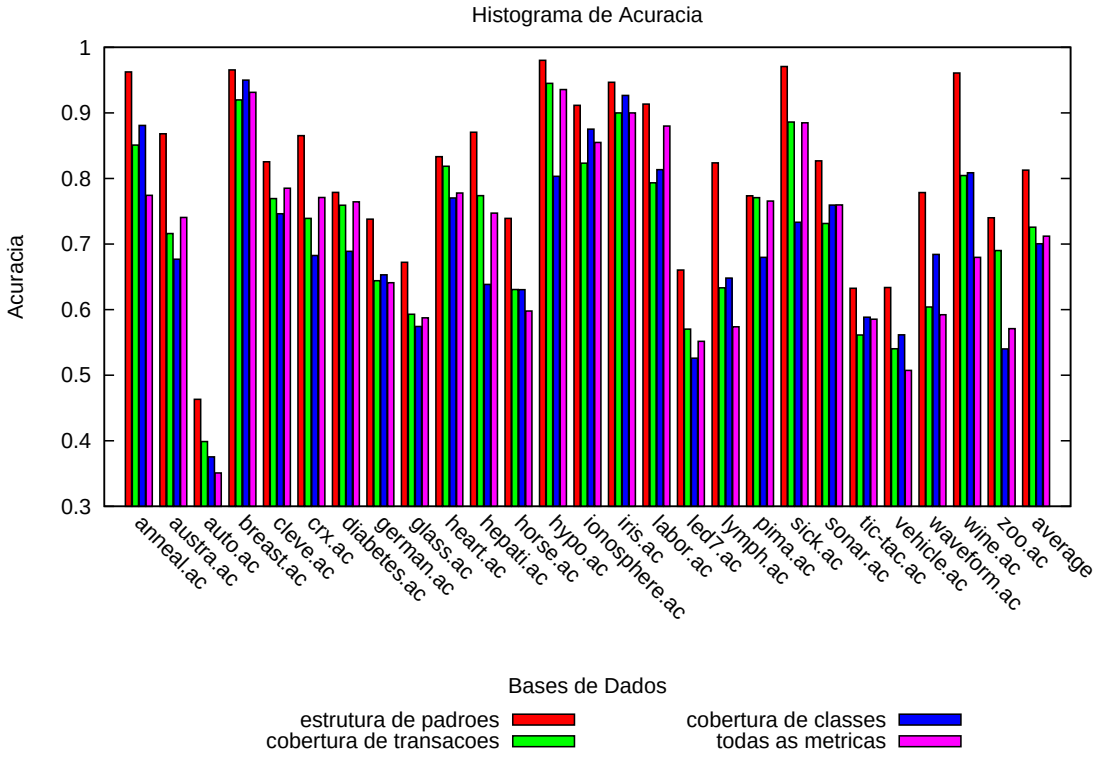

Figura 5.9: Histograma de Acurácia por Métricas de Ortogonalidade

A superioridade da métrica baseada na estrutura dos padrões sobre as outras pode ser notada analisando a figura 5.9. Neste experimento, a métrica baseadas na estrutura dos padrões encontrou conjuntos ortogonais, em geral, maiores que os encontrados pelas métricas baseadas em cobertura. Isto se deve, em parte, ao tamanho máximo das regras, que reduz o tamanho dos padrões freqüentes considerados. Quando comparamos dois padrões diferentes de tamanho unitário, a ortogonalidade será sempre 1 se a métrica baseada em estrutura for considerada. No caso das métricas baseadas em cobertura, este valor deve variar de acordo com a natureza dos dados. Como as

aplicações foram executadas com o limite de tamanho das regras igual a 2, espera-se que uma boa parte do conjunto-solução seja composta de padrões com apenas 1 item.

Analisando os resultados de algumas execuções, percebemos que o problema das métricas baseadas em cobertura não está na qualidade dos padrões extraídos, mas sim nas regras geradas com estes padrões. Em grande parte das instâncias de teste analisadas em que a métrica baseada em estrutura obteve sucesso, e as métricas baseadas em cobertura não obtiveram, o conjunto de padrões ortogonais destas possuía redundância até menor que naquele caso. Entretanto, os padrões selecionados eram indicativos de classes diferentes das verdadeiras. Seria necessário um estudo mais detalhado destes casos, para então chegar a uma conclusão a respeito da utilização de tais métricas.

O tempo de execução do algoritmo para cada métrica varia de acordo com a natureza da base de dados. A métrica baseada em estrutura faz com que o desempenho do algoritmo seja influenciado pelo tamanho do conjunto de itens. Já a métrica baseada em cobertura de transações faz com o que o desempenho seja influenciado pela quantidade de transações da base. A cobertura de classes, por sua vez, faz com que o desempenho seja influenciado pelo tamanho do conjunto de classes encontradas na base de treinamento.

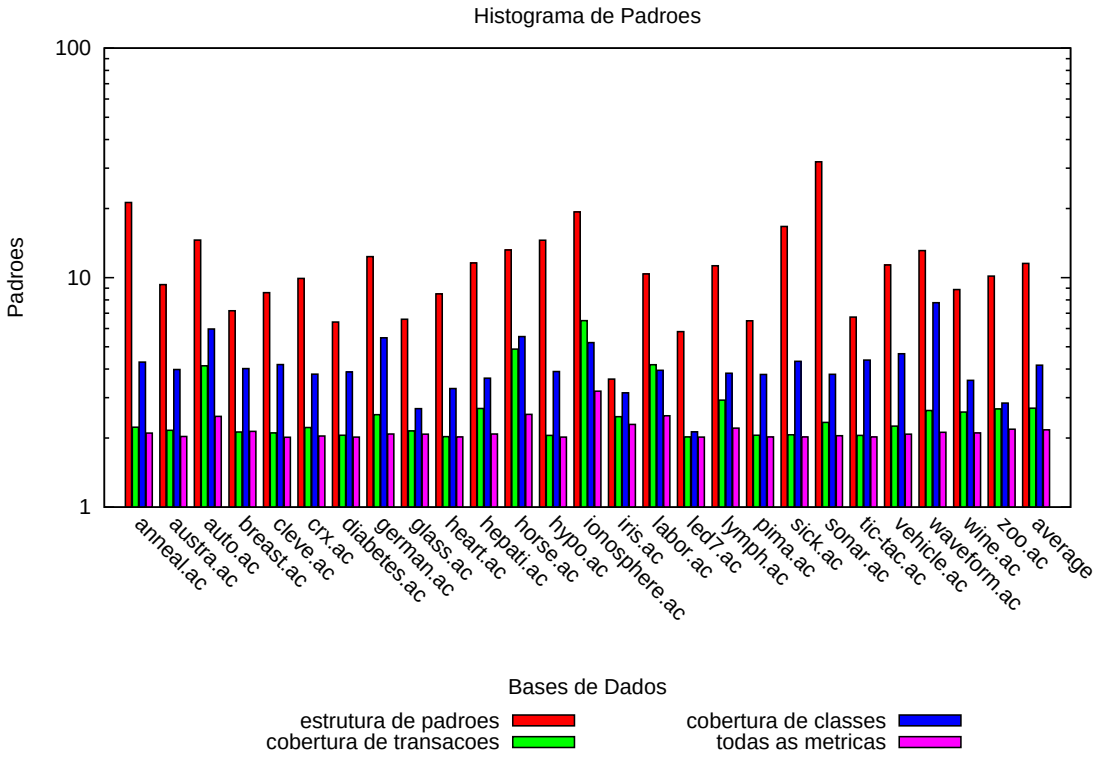

Figura 5.10: Histograma de Padrões por Métricas de Ortogonalidade

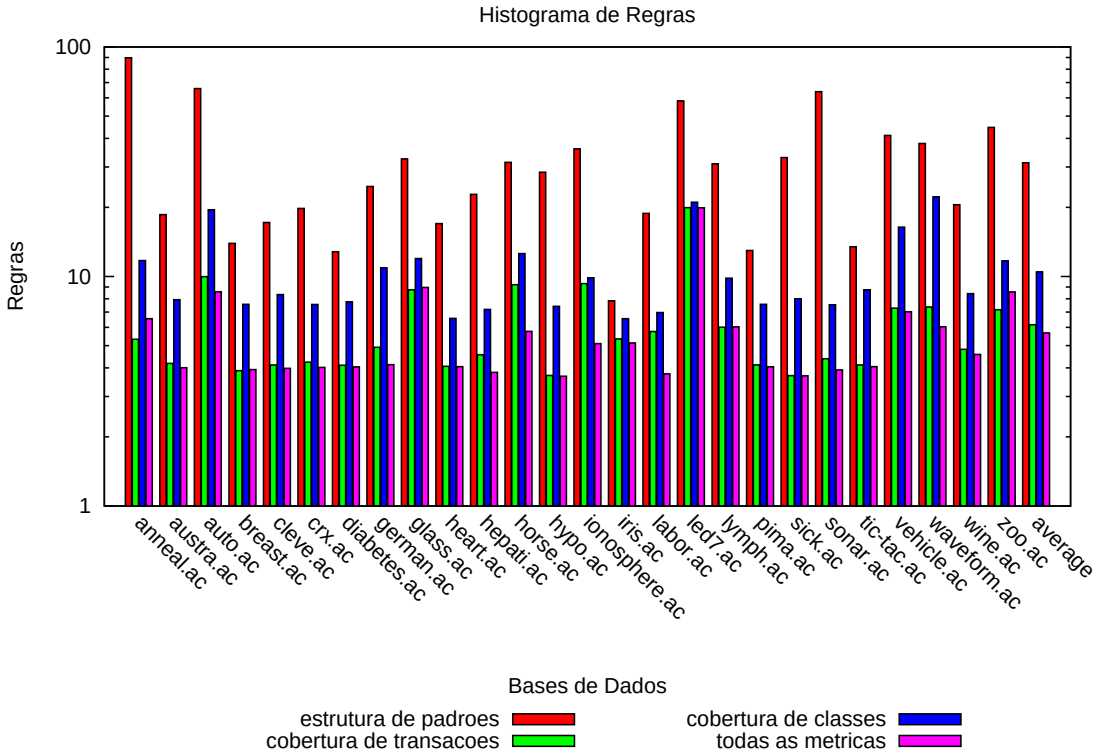

Figura 5.11: Histograma de Regras por Métricas de Ortogonalidade

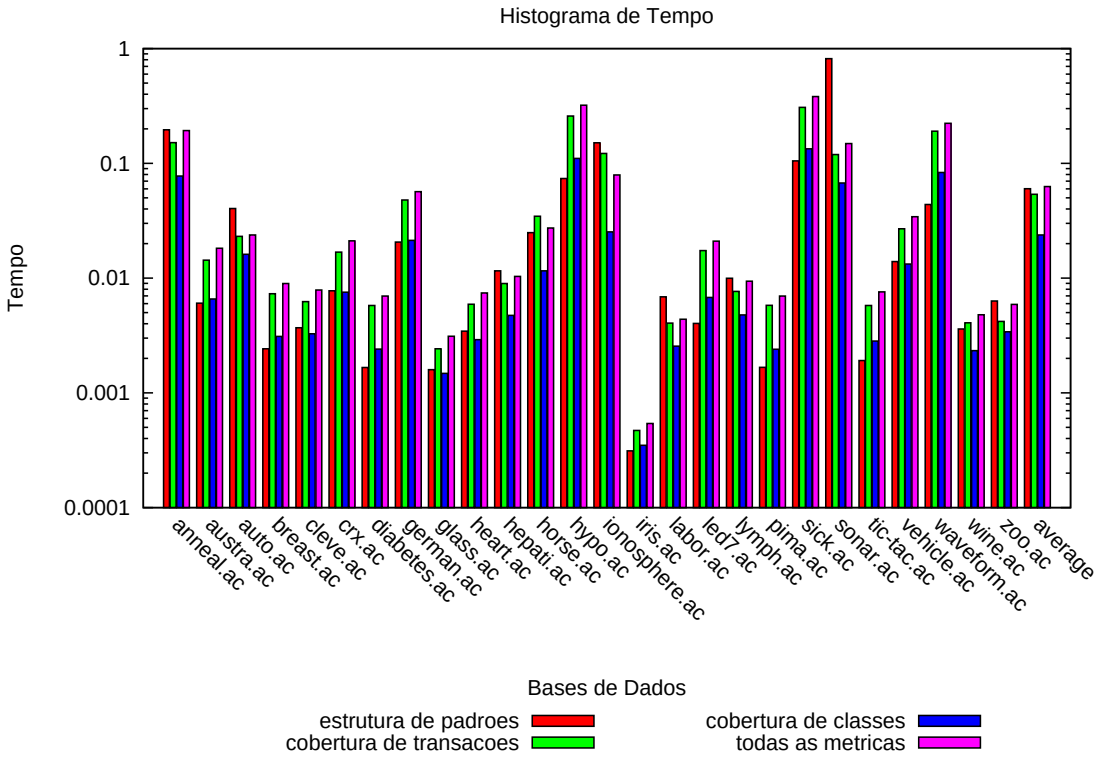

Figura 5.12: Histograma de Tempo por Métricas de Ortogonalidade

5.3.3.2 Comparação das Abordagens Quanto ao Número de Acertos

Na tabela 5.13 temos uma comparação entre as execuções do LAC e do OLAC, onde se encontram, para cada base de dados, a porcentagem de instâncias de teste que as duas abordagens acertaram (segunda coluna), a porcentagem de instâncias que o OLAC acertou e o LAC errou (terceira coluna), a porcentagem de instâncias que o LAC acertou e o OLAC errou (quarta coluna), e a porcentagem de instâncias que as duas abordagens erraram (quinta coluna). Os resultados foram obtidos com as melhores execuções para a média de todas as bases de dados (utilizando os parâmetros da tabela 5.7).

Com estes dados é possível notar que a maioria dos acertos das duas abordagens são coincidentes, ou seja, a abordagem OLAC está realizando, na maior parte das vezes, as mesmas classificações da abordagem LAC. A taxa de erro da estratégia OLAC em relação à quantidade de acertos total da estratégia LAC é de, aproximadamente, 2.3%, e a taxa de erro do LAC em relação ao OLAC é de 3.7%.

Bases de Dados	**OLAC & LAC**	**OLAC & ¬ LAC**	**¬ OLAC & LAC**	**¬ OLAC & ¬ LAC**
anneal.ac	95.11	1.13	0.75	3.01
austra.ac	84.93	1.88	1.45	11.74
auto.ac	39.51	6.83	4.39	49.27
breast.ac	96.28	0.29	1.00	2.43
cleve.ac	81.19	1.32	1.98	15.51
crx.ac	84.93	1.59	1.88	11.59
diabetes.ac	76.82	1.04	1.30	20.83
german.ac	70.40	3.40	3.10	23.10
glass.ac	63.08	4.21	1.40	31.31
heart.ac	82.96	0.37	1.48	15.19
hepati.ac	85.16	1.94	0.00	12.90
horse.ac	70.38	3.53	3.26	22.83
hypo.ac	97.94	0.06	0.09	1.90
ionosphere.ac	90.03	1.14	2.56	6.27
iris.ac	94.00	0.67	0.67	4.67
labor.ac	89.47	1.75	3.51	5.26
led7.ac	61.72	4.31	4.97	29.00
lymph.ac	79.73	2.70	0.68	16.89
pima.ac	76.82	0.52	1.04	21.61
sick.ac	97.04	0.04	0.18	2.75
sonar.ac	80.29	2.40	2.88	14.42
tic-tac.ac	56.47	6.78	3.86	32.88
vehicle.ac	56.86	6.50	2.72	33.92
waveform.ac	71.12	6.74	1.54	20.60
wine.ac	96.07	0.00	2.81	1.12
zoo.ac	73.27	0.99	0.00	25.74
average	78.91	2.39	1.90	16.80

Tabela 5.13: Comparação entre LAC e OLAC (número de acertos)

Na tabela 5.14 temos a comparação entre as execuções do LAC e do ORIGAMI. A taxa de erro do ORIGAMI em relação ao LAC é de 12.7%, e a taxa de erro do LAC em relação ao ORIGAMI é de 10.5%. Analisando os dados, vemos que a solução baseada na abordagem ORIGAMI se distancia um pouco mais do LAC, se compararmos com o resultado da tabela anterior. De fato, os padrões utilizadas pelo ORIGAMI possuem características bem diferentes dos padrões utilizados pelo LAC e pelo OLAC. Enquanto estes utilizam de padrões de tamanho pequeno, o ORIGAMI utiliza apenas padrões maximais para gerar as regras de classificação. Isto faz com que o comportamento do ORIGAMI seja totalmente diferente dos comportamentos do LAC e do OLAC.

Bases de Dados	**ORIGAMI & LAC**	**ORIGAMI & ¬ LAC**	**¬ ORIGAMI & LAC**	**¬ ORIGAMI & ¬ LAC**
anneal.ac	92.61	3.13	3.26	1.00
austra.ac	73.19	3.77	13.19	9.86
auto.ac	20.98	29.27	22.93	26.83
breast.ac	93.71	0.72	3.58	2.00
cleve.ac	72.28	6.60	10.89	10.23
crx.ac	72.61	4.06	14.20	9.13
diabetes.ac	71.35	6.51	6.77	15.36
german.ac	55.30	10.70	18.20	15.80
glass.ac	58.88	15.42	5.61	20.09
heart.ac	75.56	4.81	8.89	10.74
hepati.ac	72.90	7.10	12.26	7.74
horse.ac	55.71	12.23	17.93	14.13
hypo.ac	97.31	0.92	0.73	1.04
ionosphere.ac	84.05	2.85	8.55	4.56
iris.ac	92.67	2.00	2.00	3.33
labor.ac	78.95	7.02	14.04	0.00
led7.ac	61.81	10.78	4.88	22.53
lymph.ac	60.81	7.43	19.59	12.16
pima.ac	69.79	6.38	8.07	15.76
sick.ac	95.50	0.86	1.71	1.93
sonar.ac	60.58	8.65	22.60	8.17
tic-tac.ac	46.97	24.22	13.36	15.45
vehicle.ac	49.41	19.15	10.17	21.28
waveform.ac	49.68	13.26	22.98	14.08
wine.ac	89.33	0.56	9.55	0.56
zoo.ac	65.35	7.92	7.92	18.81
average	69.90	8.32	10.92	10.87

Tabela 5.14: Comparação entre LAC e ORIGAMI (número de acertos)

Na tabela 5.15 temos a comparação entre as abordagens OLAC e ORIGAMI, que comprova o fato de que esta abordagem possui comportamento bem diferente daquela, fazendo com que os resultados das duas execuções se distanciem tanto.

Bases de Dados	**OLAC & ORIGAMI**	**OLAC & ¬ ORIGAMI**	**¬ OLAC & ORIGAMI**	**¬ OLAC & ¬ ORIGAMI**
anneal.ac	93.11	3.13	2.63	1.13
austra.ac	72.90	13.91	4.06	9.13
auto.ac	22.93	23.41	27.32	26.34
breast.ac	93.56	3.00	0.86	2.58
cleve.ac	71.29	11.22	7.59	9.90
crx.ac	72.32	14.20	4.35	9.13
diabetes.ac	70.70	7.16	7.16	14.97
german.ac	54.90	18.90	11.10	15.10
glass.ac	61.68	5.61	12.62	20.09
heart.ac	74.44	8.89	5.93	10.74
hepati.ac	73.55	13.55	6.45	6.45
horse.ac	56.79	17.12	11.14	14.95
hypo.ac	97.38	0.63	0.85	1.14
ionosphere.ac	82.34	8.83	4.56	4.27
iris.ac	93.33	1.33	1.33	4.00
labor.ac	80.70	10.53	5.26	3.51
led7.ac	61.97	4.06	10.62	23.34
lymph.ac	61.49	20.95	6.76	10.81
pima.ac	69.40	7.94	6.77	15.89
sick.ac	95.46	1.61	0.89	2.04
sonar.ac	60.58	22.12	8.65	8.65
tic-tac.ac	49.48	13.78	21.71	15.03
vehicle.ac	52.60	10.76	15.96	20.69
waveform.ac	52.44	25.42	10.50	11.64
wine.ac	87.64	8.43	2.25	1.69
zoo.ac	66.34	7.92	6.93	18.81
average	70.36	10.94	7.86	10.85

Tabela 5.15: Comparação entre OLAC e ORIGAMI (número de acertos)

5.3.4 Avaliação teste-t de Student

Com o objetivo de verificar se os resultados de todas as abordagens são estatisticamente equivalentes, tanto para os experimentos executados com os parâmetros que obtiveram melhores acurácias para cada base quanto para os experimentos executados com os parâmetros que obtiveram as melhores médias de acurácia para todas as bases, a cada conjunto de resultados (comparando as abordagens par-a-par) foi aplicado o teste-t pareado com confiança igual a 99%. Os resultados se encontram nas tabelas 5.16 e

5.17 respectivamente. Cada uma delas possui três comparações distintas: LAC × OLAC, LAC × ORIGAMI e OLAC × ORIGAMI. O valor encontrado em cada coluna, para cada base de dados, é o nome da abordagem que possui os melhores resultados ou um hífem (caso os resultados sejam estatisticamente equivalentes). Analisando estas tabelas nota-se que todos os resultados são equivalentes entre LAC e OLAC, com exceção dos resultados obtidos com a base **waveform.ac** na tabela 5.17.

Bases de Dados	**LAC x OLAC**	**LAC x ORIGAMI**	**OLAC x ORIGAMI**
anneal.ac	-	-	-
austra.ac	-	-	-
auto.ac	-	-	-
breast.ac	-	-	-
cleve.ac	-	-	-
crx.ac	-	-	-
diabetes.ac	-	-	-
german.ac	-	-	-
glass.ac	-	-	-
heart.ac	-	-	-
hepati.ac	-	-	-
horse.ac	-	-	-
hypo.ac	-	-	-
ionosphere.ac	-	-	-
iris.ac	-	-	-
labor.ac	-	-	-
led7.ac	-	-	-
lymph.ac	-	-	-
pima.ac	-	-	-
sick.ac	-	-	-
sonar.ac	-	-	-
tic-tac.ac	-	ORIGAMI	ORIGAMI
vehicle.ac	-	-	-
waveform.ac	-	LAC	OLAC
wine.ac	-	LAC	-
zoo.ac	-	-	-

Tabela 5.16: Resultado do teste-t (melhores resultados para cada base)

5.3.5 Estudo de Casos de Teste

Nesta seção apresentaremos dois casos de teste extraídos do *log* de execução em que as abordagens baseadas em ortogonalidade não obtêm sucesso na classificação das ins-

Bases de Dados	LAC x OLAC	LAC x ORIGAMI	OLAC x ORIGAMI
anneal.ac	-	-	-
austra.ac	-	LAC	OLAC
auto.ac	-	-	-
breast.ac	-	-	-
cleve.ac	-	-	-
crx.ac	-	LAC	OLAC
diabetes.ac	-	-	-
german.ac	-	LAC	OLAC
glass.ac	-	-	-
heart.ac	-	-	-
hepati.ac	-	-	-
horse.ac	-	-	-
hypo.ac	-	-	-
ionosphere.ac	-	-	-
iris.ac	-	-	-
labor.ac	-	-	-
led7.ac	-	ORIGAMI	ORIGAMI
lymph.ac	-	-	OLAC
pima.ac	-	-	-
sick.ac	-	-	-
sonar.ac	-	-	-
tic-tac.ac	-	ORIGAMI	-
vehicle.ac	-	ORIGAMI	-
waveform.ac	OLAC	LAC	OLAC
wine.ac	-	LAC	-
zoo.ac	-	-	-

Tabela 5.17: Resultado do teste-t (melhores resultados para cada abordagem)

tâncias de teste. Foram utilizados os resultados das três abordagens considerando os parâmetros da tabela 5.7.

5.3.5.1 Caso de Teste LAC x OLAC

Serão apresentados, nesta seção, os detalhes do processo de classificação de uma instância aleatória de teste em que o LAC obteve sucesso e o OLAC não. A configuração selecionada possui 668 transações de treinamento e 69 instâncias de teste da base *breast.ac*, das quais o OLAC classificou quatro de maneira errada, e o LAC, apenas duas. Serão apresentados os detalhes da classificação da instância da tabela 5.18. Esta instância possui 10 itens, sendo todos eles freqüentes (suporte maior que 0.0001) na projeção obtida, que corresponde a 630 transações da base de treinamento. Visto que

o LAC foi executado com tamanho máximo das regras igual a 1, foram encontrados 10 padrões freqüentes (tabela 5.19), com os quais foram geradas 20 regras de associação com confiança maior ou igual a 0.01 (tabela 5.20).

Classe	Transação
CLASS=1	w=attr1=ignore w=attr7=-1.5 w=attr8=-2.5 w=attr10=-1.5 w=attr5=-1.5 w=attr9=2.5-9.5 w=attr2=6.5+ w=attr3=4.5+ w=attr4=4.5+ w=attr6=3.5+

Tabela 5.18: Exemplo de Classificação - Instância de Teste da base *breast.ac*

Padrão	Suporte
w=attr1=ignore	1.000000
w=attr7=-1.5	0.576190
w=attr8=-2.5	0.455556
w=attr10=-1.5	0.833333
w=attr5=-1.5	0.585714
w=attr9=2.5-9.5	0.226984
w=attr2=6.5+	0.212698
w=attr3=4.5+	0.247619
w=attr4=4.5+	0.260317
w=attr6=3.5+	0.268254

Tabela 5.19: Exemplo de Classificação - Padrões Freqüentes Encontrados pelo LAC

Já o OLAC, que foi executado com tamanho máximo de regra igual a 2, obteve um conjunto de 55 padrões freqüentes, dos quais foram extraídos 7 padrões ortogonais (tabela 5.21), e, a partir destes, gerou 14 regras com confiança maior ou igual a 0.001 (tabela 5.22).

Analisando as tabelas 5.19 e 5.21, percebe-se que todos os itens da instância de teste são encontrados nos conjuntos de padrões freqüentes das duas execuções. No caso do LAC, já era esperado que cada item correspondesse a um padrão, pelo fato de todos os itens serem freqüentes. Em relação ao OLAC, era esperado um sub-conjunto dos padrões freqüentes com baixa redundância em relação ao conjunto original. Como o OLAC foi executado com tamanho máximo de regra igual a 2, o conjunto de padrões freqüentes encontrados foi bem maior que o mesmo conjunto obtido pela estratégia LAC, porém, nota-se que o conjunto de padrões ortogonais é menor, mesmo cobrindo todos os itens da instância de teste. Além disso, nenhum item foi utilizado mais de uma vez, o que comprova a baixa redundância do conjunto.

Entretanto, apesar da boa qualidade do conjunto de padrões ortogonais encontrado, a classificação não foi realizada com sucesso pelo OLAC. Uma análise simples das tabelas 5.20 e 5.22 é suficiente para entender o motivo deste erro. Na primeira tabela, vemos que as regras que mais contribuem para a classificação correta aparecem no topo

Ranking	**Regra**	**Convicção**
1	CLASS=1 ⇐ w=attr3=4.5+	25.876190
2	CLASS=1 ⇐ w=attr2=6.5+	17.781588
3	CLASS=0 ⇐ w=attr8=-2.5	13.796825
4	CLASS=1 ⇐ w=attr4=4.5+	13.601587
5	CLASS=0 ⇐ w=attr7=-1.5	11.104762
6	CLASS=1 ⇐ w=attr6=3.5+	6.595891
7	CLASS=0 ⇐ w=attr5=-1.5	4.281774
8	CLASS=1 ⇐ w=attr9=2.5-9.5	3.649206
9	CLASS=0 ⇐ w=attr10=-1.5	1.497175
10	CLASS=0 ⇐ w=attr1=ignore	1.000000
11	CLASS=1 ⇐ w=attr1=ignore	1.000000
12	CLASS=1 ⇐ w=attr10=-1.5	0.855856
13	CLASS=1 ⇐ w=attr5=-1.5	0.720084
14	CLASS=1 ⇐ w=attr7=-1.5	0.684226
15	CLASS=1 ⇐ w=attr8=-2.5	0.680079
16	CLASS=0 ⇐ w=attr9=2.5-9.5	0.411287
17	CLASS=0 ⇐ w=attr6=3.5+	0.374144
18	CLASS=0 ⇐ w=attr4=4.5+	0.353765
19	CLASS=0 ⇐ w=attr2=6.5+	0.349551
20	CLASS=0 ⇐ w=attr3=4.5+	0.345363

Tabela 5.20: Exemplo de Classificação - Regras de Associação Encontradas pelo LAC

Padrão	**Suporte**
w=attr1=ignore w=attr7=-1.5	0.576190
w=attr5=-1.5	0.585714
w=attr9=2.5-9.5 w=attr4=4.5+	0.141270
w=attr3=4.5+	0.247619
w=attr2=6.5+	0.212698
w=attr8=-2.5 w=attr10=-1.5	0.436508
w=attr6=3.5+	0.268254

Tabela 5.21: Exemplo de Classificação - Padrões Freqüentes Encontrados pelo OLAC

Ranking	**Regra**	**Convicção**
1	CLASS=0 ⇐ w=attr8=-2.5 w=attr10=-1.5	46.269840
2	CLASS=1 ⇐ w=attr3=4.5+	25.876190
3	CLASS=1 ⇐ w=attr2=6.5+	17.781588
4	CLASS=0 ⇐ w=attr1=ignore w=attr7=-1.5	11.104762
5	CLASS=1 ⇐ w=attr9=2.5-9.5 w=attr4=4.5+	8.435827
6	CLASS=1 ⇐ w=attr6=3.5+	6.595891
7	CLASS=0 ⇐ w=attr5=-1.5	4.281774
8	CLASS=1 ⇐ w=attr5=-1.5	0.720084
9	CLASS=1 ⇐ w=attr1=ignore w=attr7=-1.5	0.684226
10	CLASS=1 ⇐ w=attr8=-2.5 w=attr10=-1.5	0.668353
11	CLASS=0 ⇐ w=attr6=3.5+	0.374144
12	CLASS=0 ⇐ w=attr9=2.5-9.5 w=attr4=4.5+	0.365234
13	CLASS=0 ⇐ w=attr2=6.5+	0.349551
14	CLASS=0 ⇐ w=attr3=4.5+	0.345363

Tabela 5.22: Exemplo de Classificação - Regras de Associação Encontradas pelo OLAC

do *ranking* (posições 1, 2 e 4), com medidas de convicção grandes o suficiente para que a escolha da classe correta seja realizada. A média das convicções das regras que apontam para a classe **CLASS=1** é 7.14, enquanto que, para a classe **CLASS=0**, este valor é igual a 3.35. Já na segunda tabela, nota-se a presença de regras que contribuem para a classificação falsa numa proporção próxima à das regras verdadeiras. A primeira regra do *ranking*, que possui o padrão **w=attr8=-2.5 w=attr10=-1.5** e aponta para a classe **CLASS=0**, é responsável pela maior contribuição para o erro da classificação, com a medida de convicção igual a 46.27. Na primeira tabela, vemos que os padrões **w=attr8=-2.5** e **w=attr10=-1.5** também contribuem para a classe falsa, mas com convicções iguais a 13.80 e 11.10, respectivamente. O que acontece, neste caso, é que existe uma relação entre o padrão **w=attr8=-2.5 w=attr10=-1.5** e a classe **CLASS=0** muito maior que a relação entre os sub-padrões **w=attr8=-2.5** e **w=attr10=-1.5** e a classe **CLASS=0**. Isto fez com que a média das regras que apontam para a classe **CLASS=0** ficasse em 9.01, enquanto que a média da classe **CLASS=1** ficou em 8.68.

Um caso semelhante ocorre também com a segunda instância classificada de maneira errada pelo OLAC, onde o padrão **w=attr9=2.5-9.5 w=attr7=?**, por não fazer parte de nenhuma transação de classe **CLASS=0** na projeção, obteve uma medida infinita de convicção, fazendo com que esta classe fosse escolhida como resultado, onde o correto seria escolher a classe **CLASS=1**.

Situações desse tipo contribuíram para que a abordagem OLAC não classificasse corretamente todas as instâncias em que a abordagem LAC obteve sucesso. No entanto, os erros encontrados estão mais relacionados com a natureza dos dados do que com o

conceito de ortogonalidade.

5.3.5.2 Caso de Teste LAC x ORIGAMI

Nesta seção comparamos o comportamento do LAC e do ORIGAMI no processo de classificação de uma instância aleatória de teste em que o LAC obteve sucesso e o ORIGAMI não. A configuração selecionada possui 52 transações de treinamento e 5 instâncias de teste da base *labor.ac*, das quais o ORIGAMI classificou 2 de maneira correta, e o LAC não errou nenhuma classificação. A instância escolhida se encontra na tabela 5.23. Esta instância possui 16 itens, sendo todos eles freqüentes na projeção obtida, que corresponde às 52 transações da base de treinamento. Assim como no exemplo anterior, cada item da instância de teste deu origem a um padrão freqüente (tabela 5.24), com os quais foram obtidas 31 regras de associação com confiança maior ou igual a 0.01 (tabela 5.25).

Classe	Transação
CLASS=0	w=attr1=ignore w=attr3=-3.25 w=attr4=? w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr13=? w=attr14=? w=attr2=2.65+ w=attr7=? w=attr15=yes w=attr10=yes w=attr5=none w=attr16=full w=attr12=generous

Tabela 5.23: Exemplo de Classificação - Instância de Teste da base *labor.ac*

Padrão	Suporte
w=attr1=ignore	0.980769
w=attr3=-3.25	0.211538
w=attr4=?	0.730769
w=attr6=ignore	0.884615
w=attr8=?	0.846154
w=attr9=?	0.461538
w=attr11=ignore	0.923077
w=attr13=?	0.519231
w=attr14=?	0.346154
w=attr2=2.65+	0.711538
w=attr7=?	0.519231
w=attr15=yes	0.461538
w=attr10=yes	0.153846
w=attr5=none	0.365385
w=attr16=full	0.288462
w=attr12=generous	0.250000

Tabela 5.24: Exemplo de Classificação - Padrões Freqüentes Encontrados pelo LAC

Ranking	**Regra**	**Convicção**
1	CLASS=0 ⇐ w=attr7=?	0.925926
2	CLASS=0 ⇐ w=attr12=generous	0.923077
3	CLASS=1 ⇐ w=attr3=-3.25	0.818182
4	CLASS=0 ⇐ w=attr2=2.65+	0.837838
5	CLASS=0 ⇐ w=attr10=yes	0.750000
6	CLASS=0 ⇐ w=attr13=?	0.740741
7	CLASS=0 ⇐ w=attr16=full	0.733333
8	CLASS=0 ⇐ w=attr15=yes	0.708333
9	CLASS=1 ⇐ w=attr5=none	0.421053
10	CLASS=1 ⇐ w=attr9=?	0.416667
11	CLASS=0 ⇐ w=attr8=?	0.681818
12	CLASS=1 ⇐ w=attr4=?	0.394737
13	CLASS=1 ⇐ w=attr11=ignore	0.375000
14	CLASS=0 ⇐ w=attr14=?	0.666667
15	CLASS=1 ⇐ w=attr6=ignore	0.369565
16	CLASS=1 ⇐ w=attr1=ignore	0.352941
17	CLASS=0 ⇐ w=attr1=ignore	0.647059
18	CLASS=1 ⇐ w=attr14=?	0.333333
19	CLASS=1 ⇐ w=attr8=?	0.318182
20	CLASS=0 ⇐ w=attr6=ignore	0.630435
21	CLASS=0 ⇐ w=attr11=ignore	0.625000
22	CLASS=1 ⇐ w=attr15=yes	0.291667
23	CLASS=1 ⇐ w=attr16=full	0.266667
24	CLASS=1 ⇐ w=attr13=?	0.259259
25	CLASS=0 ⇐ w=attr4=?	0.605263
26	CLASS=1 ⇐ w=attr10=yes	0.250000
27	CLASS=0 ⇐ w=attr9=?	0.583333
28	CLASS=0 ⇐ w=attr5=none	0.578947
29	CLASS=1 ⇐ w=attr2=2.65+	0.162162
30	CLASS=1 ⇐ w=attr12=generous	0.076923
31	CLASS=1 ⇐ w=attr7=?	0.074074
32	CLASS=0 ⇐ w=attr3=-3.25	0.181818

Tabela 5.25: Exemplo de Classificação - Regras de Associação Encontradas pelo LAC

O ORIGAMI, por sua vez, obteve um conjunto de 13 padrões maximais, que se encontram na tabela 5.26, de onde foi extraído um único padrão ortogonal, e, posteriormente, a única regra de associação com confiança maior ou igual a 0.0001, que pode ser vista na tabela 5.27. É importante notar que todos os padrões maximais obtidos possuem suporte igual a 0.019231, o que corresponde à fração $\frac{1}{52}$, ou seja, nenhum padrão maximal obtido aparece em mais de uma transação na base de dados. A combinação de parâmetros de execução utilizada foi responsável por fazer com que o algoritmo não selecionasse uma maior quantidade de padrões ortogonais. Como o valor do parâmetro α utilizado foi muito baixo (0.1), e o suporte utilizado favoreceu a obtenção de grandes padrões maximais, a métrica de ortogonalidade por estrutura de padrões não foi adequada. Isso aconteceu porque todos os padrões encontrados possuem similaridade maior que 10%, pelo fato de vários itens aparecerem em mais de um padrão. Os itens **w=attr1=ignore** e **w=attr8=?**, por exemplo, estão presentes em todos os padrões.

Como resultado, a única regra obtida pela abordagem ORIGAMI aponta para a classe **CLASS=1**, fazendo com que o algoritmo classifique a instância de teste de maneira errada.

5.4 Considerações

Neste capítulo, foi apresentado o aplicativo **olac**, sua interface de interação com o usuário, e um exemplo de execução onde foi demonstrado o comportamento da abordagem OLAC. Em seguida, foram apresentados os resultados obtidos com a execução das três abordagens de classificação em 26 bases de dados disponíveis no repositório **UCI**.

Na seção 5.3.2 foram apresentados os melhores resultados obtidos com a execução das abordagens utilizando parâmetros diferentes para cada base de dados. As três abordagens obtiveram resultados muito próximos, em média, sendo que a abordagem LAC obteve uma leve vantagem sobre as outras. Entretanto, podemos perceber que este comportamento não se repetiu em todas as bases.

Analisando os resultados, vemos que o LAC se saiu melhor em 11 bases de dados, o OLAC, por sua vez, se saiu melhor em 6 bases, assim como ORIGAMI. Além disso, as três abordagens obtiveram o mesmo resultado para 1 base, e o LAC e o OLAC ainda empataram em outras 2 bases. A superioridade das abordagens ortogonais em algumas bases indica que, mesmo quando os melhores parâmetros para cada execução são utilizados, ainda é possível obter vantagens com as abordagens baseadas em ortogonalidade.

Na seção 5.3.3 foram apresentados os melhores resultados obtidos com a execução

Padrão	**Suporte**
w=attr1=ignore w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr13=? w=attr14=? w=attr2=2.65+ w=attr7=? w=attr15=yes w=attr16=full w=attr12=generous	0.019231
w=attr1=ignore w=attr3=-3.25 w=attr4=? w=attr6=ignore w=attr8=? w=attr11=ignore w=attr13=? w=attr14=? w=attr7=? w=attr10=yes	0.019231
w=attr1=ignore w=attr3=-3.25 w=attr4=? w=attr8=? w=attr9=? w=attr15=yes w=attr10=yes	0.019231
w=attr1=ignore w=attr3=-3.25 w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr13=? w=attr15=yes w=attr5=none w=attr16=full	0.019231
w=attr1=ignore w=attr3=-3.25 w=attr4=? w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr13=? w=attr14=? w=attr15=yes w=attr16=full	0.019231
w=attr1=ignore w=attr6=ignore w=attr8=? w=attr11=ignore w=attr13=? w=attr14=? w=attr2=2.65+ w=attr7=? w=attr15=yes w=attr5=none w=attr16=full w=attr12=generous	0.019231
w=attr1=ignore w=attr4=? w=attr6=ignore w=attr8=? w=attr11=ignore w=attr13=? w=attr2=2.65+ w=attr7=? w=attr16=full w=attr12=generous	0.019231
w=attr1=ignore w=attr8=? w=attr9=? w=attr13=? w=attr14=? w=attr2=2.65+ w=attr7=? w=attr15=yes w=attr10=yes	0.019231
w=attr1=ignore w=attr4=? w=attr6=ignore w=attr8=? w=attr11=ignore w=attr13=? w=attr14=? w=attr2=2.65+ w=attr7=? w=attr15=yes	0.019231
w=attr1=ignore w=attr3=-3.25 w=attr4=? w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr15=yes w=attr5=none w=attr16=full	0.019231
w=attr1=ignore w=attr4=? w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr2=2.65+ w=attr15=yes w=attr5=none w=attr16=full	0.019231
w=attr1=ignore w=attr4=? w=attr6=ignore w=attr8=? w=attr11=ignore w=attr13=? w=attr2=2.65+ w=attr7=? w=attr15=yes w=attr5=none w=attr16=full	0.019231
w=attr1=ignore w=attr4=? w=attr8=? w=attr9=? w=attr11=ignore w=attr2=2.65+ w=attr7=? w=attr10=yes	0.019231

Tabela 5.26: Exemplo de Classificação - Padrões Freqüentes Encontrados pelo ORIGAMI

Ranking	**Regra**	**Confiança**
1	CLASS=1 ⇐ w=attr1=ignore w=attr3=-3.25 w=attr6=ignore w=attr8=? w=attr9=? w=attr11=ignore w=attr13=? w=attr15=yes w=attr5=none w=attr16=full	1.000000

Tabela 5.27: Exemplo de Classificação - Regras de Associação Encontradas pelo ORIGAMI

das três abordagens utilizando o mesmo conjunto de parâmetros para todas as bases. Este resultado comprova a qualidade dos padrões ortogonais obtidos pelo OLAC, visto que a média das acurácias encontradas por esta abordagem (0.813) foi até um pouco maior que a média encontrada pelo LAC (0.808).

Analisando os resultados, vemos que o LAC novamente se saiu melhor em 11 bases, o OLAC, por sua vez, se saiu melhor em 8 bases, e o ORIGAMI, em 6. As três abordagens ainda empataram em 1 base. Apesar do LAC continuar com os melhores resultados num número maior de bases, o OLAC obteve melhores resultados em média. Isto se deve à qualidade dos resultados obtidos, mesmo nos casos em que os do LAC são melhores.

Na seção 5.3.4 foram apresentados resultados obtidos com o auxílio da metodologia teste-t de Student que comprovam a equivalência estatística entre as abordagens.

Na seção 5.3.5 foram apresentados estudos de caso que comparam a classificação de uma instância pela abordagem ortogonal e não ortogonal para o OLAC e o ORIGAMI. Os estudos mostram um exemplo em que a abordagem ortogonal não consegue classificar corretamente uma instância em que a abordagem não ortogonal obtém sucesso. É possível perceber que a falha na classificação pelas abordagens ortogonais não foi causada pela má qualidade dos conjuntos ortogonais, mas sim pelas características dos padrões encontrados nestes conjuntos, que fizeram com que as métricas de avaliação das regras geradas indicassem a classificação falsa com maior convicção que a verdadeira.

Em geral, os resultados demonstram que as abordagens ortogonais obtiveram resultados de boa qualidade, quando comparados com a abordagem clássica não ortogonal.

Capítulo 6

Conclusão

Neste trabalho foi explorado o conceito de ortogonalidade em mineração de dados com o objetivo de se diminuir a redundância e, conseqüentemente, o tamanho do conjunto de padrões freqüentes de uma base de dados. A estratégia foi aplicada ao problema da classificação associativa, onde a ortogonalidade foi explorada sob três perspectivas: estrutura dos padrões, cobertura de transações e cobertura de classes.

Foi implementado um classificador baseado em regras de associação com três abordagens distintas: a abordagem clássica **LAC** (*Lazy Associative Classifier*), proposta em Veloso et al. (2006b), que simplesmente obtém, para cada instância de teste, um conjunto de padrões freqüentes, e o utiliza para gerar regras associativas, a abordagem ortogonal **OLAC** (*Orthogonal Lazy Associative Classifier*), que extrai, do conjunto de padrões freqüentes, um sub-conjunto de padrões ortogonais, e utiliza este sub-conjunto para gerar as regras de associação, e uma adaptação do **ORIGAMI**, encontrado na literatura, e desenvolvido, originalmente, para obtenção de padrões ortogonais em grafos.

Os experimentos comprovam a qualidade dos padrões obtidos, visto que os algoritmos baseados em ortogonalidade obtiveram resultados semelhantes aos da abordagem clássica. Vimos que, apesar do LAC ter levado uma pequena vantagem na maior parte das bases, os resultados obtidos pelas abordagens ortogonais mantiveram a qualidade, fazendo com que as médias das acurácias para as três abordagens ficassem muito próximas.

Considerando, para cada abordagem, os conjuntos de parâmetros de execução que obtiveram os melhores resultados para cada base de dados, as médias de acurácia obtidas para as abordagens LAC, OLAC e ORIGAMI foram, respectivamente, 0.843, 0.840 e 0.839. Já os valores médios para as quantidades de padrões utilizados na geração das regras foram 213 para o LAC, e 12 para o OLAC e o ORIGAMI. Como conseqüência da diminuição do conjunto de padrões, o número médio de regras geradas

pelas abordagens OLAC e ORIGAMI foram, respectivamente, 25 e 23, enquanto o LAC gerou, em média, 628 regras para cada classificação.

Nas execuções realizadas com o conjunto de parâmetros que obteve a melhor média dos resultados, os resultados obtidos para as abordagens LAC, OLAC e ORIGAMI foram, respectivamente, 0.808, 0.813 e 0.782. Os valores médios para as quantidades de padrões utilizados na geração das regras foram, respectivamente, 19, 12 e 1, e os valores médios para números de regras geradas foram, respectivamente, 51, 31 e 1.

Para as abordagens ortogonais, notamos uma predominância da métrica de ortogonalidade baseada na estrutura dos padrões entre os parâmetros de execução responsáveis pelos melhores resultados para cada base de dados. Vimos que as métricas baseadas em cobertura, tanto de classes quanto de transações, obtiveram conjuntos muito limitados, e compostos de padrões, muitas vezes, pouco freqüentes. Isto fez com que tais métricas, mesmo diminuindo a ambiguidade das regras, não obtivessem resultados tão bons quanto a baseada na estrutura dos padrões.

Vimos também que o ORIGAMI apresentou, em grande parte dos resultados, conjuntos ortogonais com apenas 1 elemento. O motivo disto foram os baixos valores para suporte encontrados entre os parâmetros de execução que obtiveram os melhores resultados para a maioria das bases.

Também foram apresentados alguns casos de teste em que as abordagens ortogonais levaram desvantagem. Nestes casos, foi demonstrado que a falha na classificação não aconteceu por falta de qualidade nos conjuntos ortogonais, mas sim por causa das características dos padrões ortogonais selecionados.

Em resumo, vimos que a aplicação de ortogonalidade no problema da classificação associativa possibilitou a minimização do número de padrões utilizados na geração das regras, e, conseqüentemente, a diminuição do conjunto de regras. Vimos também que, apesar de não obter melhores resultados com a diminuição da ambiguidade, a ortogonalidade possibilitou a diminuição da redundância das regras geradas, mantendo a mesma qualidade dos resultados.

6.1 Trabalhos Futuros

Como trabalhos futuros, podemos citar a utilização de ortogonalidade em outros pontos do algoritmo de classificação. Uma forma alternativa de se abordar o problema seria obter um sub-conjunto de itens ortogonais em relação à base de dados, a partir deste, obter todos os padrões freqüentes que seriam considerados na geração das regras. Outra forma seria gerar todas as regras possíveis a partir do conjunto de padrões freqüentes, e aplicar a ortogonalidade, em conjunto com as demais medidas de interesse,

na extração do sub-conjunto de regras consideradas para a classificação. Também seria interessante trabalhar em novas heurísticas de obtenção de conjuntos ortogonais, com ênfase em desempenho, e investir no desenvolvimento de novos algoritmos de mineração de padrões freqüentes que já considerem ortogonalidade durante a exploração do espaço de busca dos padrões.

A utilização de uma abordagem híbrida OLAC-ORIGAMI também seria uma alternativa para tentar melhorar a acurácia dos resultados. A nova abordagem faria uso dos parâmetros α e β aplicados ao conjunto de padrões freqüentes, e não ao seu sub-conjunto de padrões maximais. Esta solução seria útil para se obter conjuntos de padrões ortogonais de melhor qualidade, já que o usuário poderia controlar a medida de ortogonalidade e de representatividade da solução, e todos os padrões freqüentes seriam candidatos ao conjunto ortogonal.

Apêndice A

Cálculo do Fator da Métrica Cobertura de Classes

A obtenção do fator utilizado no cálculo da métrica de ortogonalidade por cobertura de classes foi realizada com o auxílio da própria ferramenta desenvolvida neste trabalho.

Foram realizadas diversas execuções da ferramenta com um conjunto fixo de parâmetros e vários fatores diferentes para cada uma das vinte e seis bases de dados do repositório **UCI** (*UC Irvine Machine Learing Repository*) mencionadas na seção 5, e o fator que obteve os melhores valores de acurácia em média considerando todas as bases foi selecionado para compor a métrica.

Na tabela A.1 se encontram os parâmetros utilizados no processo, e na tabela A.2 se encontram os resultados obtidos par cada fator avaliado.

Parâmetros	**Valores**
support	0.0001
confidence	0.0001
min-num-rules	1
max-num-rank-rules	100
min-rule-len	1
max-rule-len	2
rule-measure	SUPPORT
orth-metric	CLASS COVERAGE
orth-method	SET
orth-pat-ordering	REVERSE SORTED

Tabela A.1: Parâmetros Utilizados

Fator	**Acurácia**
0.5	0.6185
0.6	0.6262
0.7	0.6302
0.8	0.6361
0.9	0.6467
1.0	0.6485
1.1	0.6490
1.2	0.6439
1.3	0.6372
1.4	0.6320
1.5	0.6331

Tabela A.2: Resultados Obtidos

Apêndice B

Resultados com *confiança* como Medida de Interesse das Regras de Associação

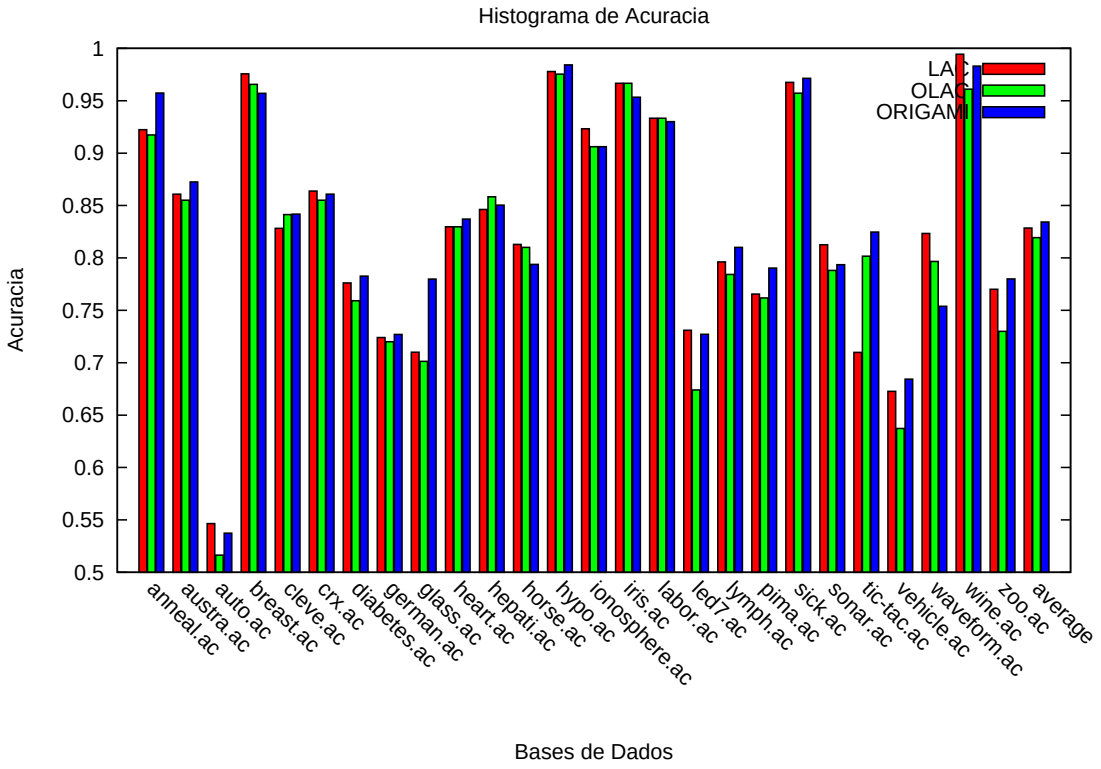

Figura B.1: Histograma de Acurácia (melhores resultados para cada base de dados)

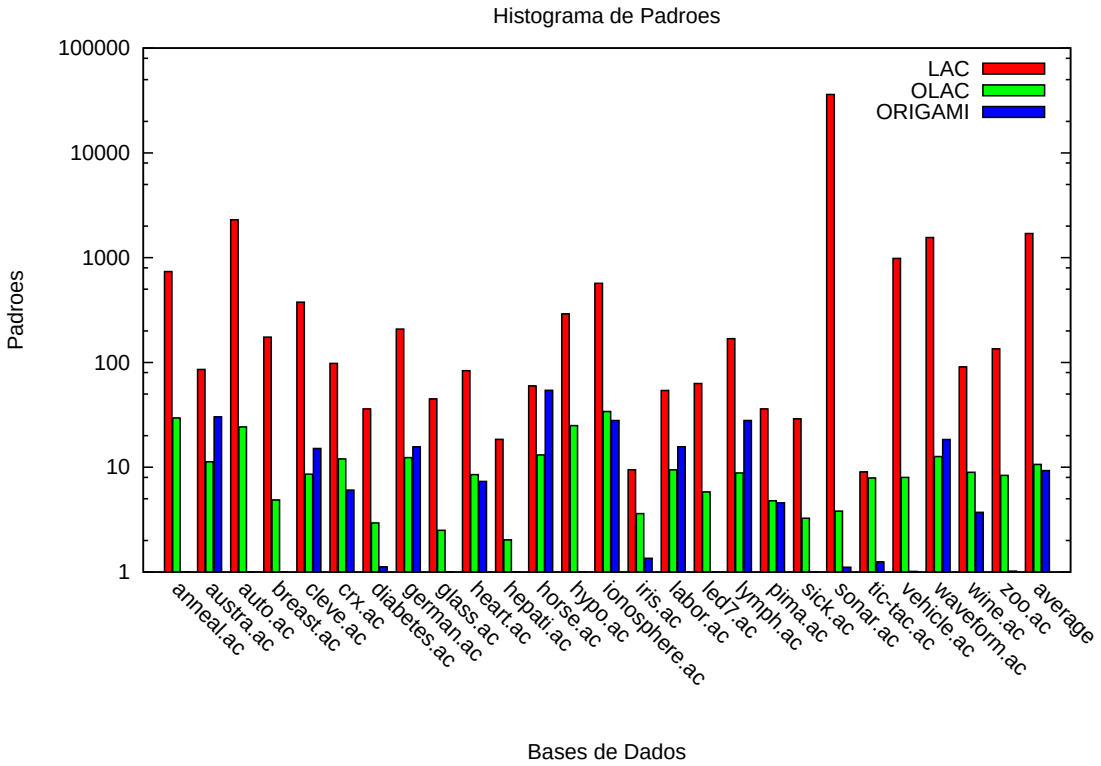

Figura B.2: Histograma de Padrões (melhores resultados para cada base de dados)

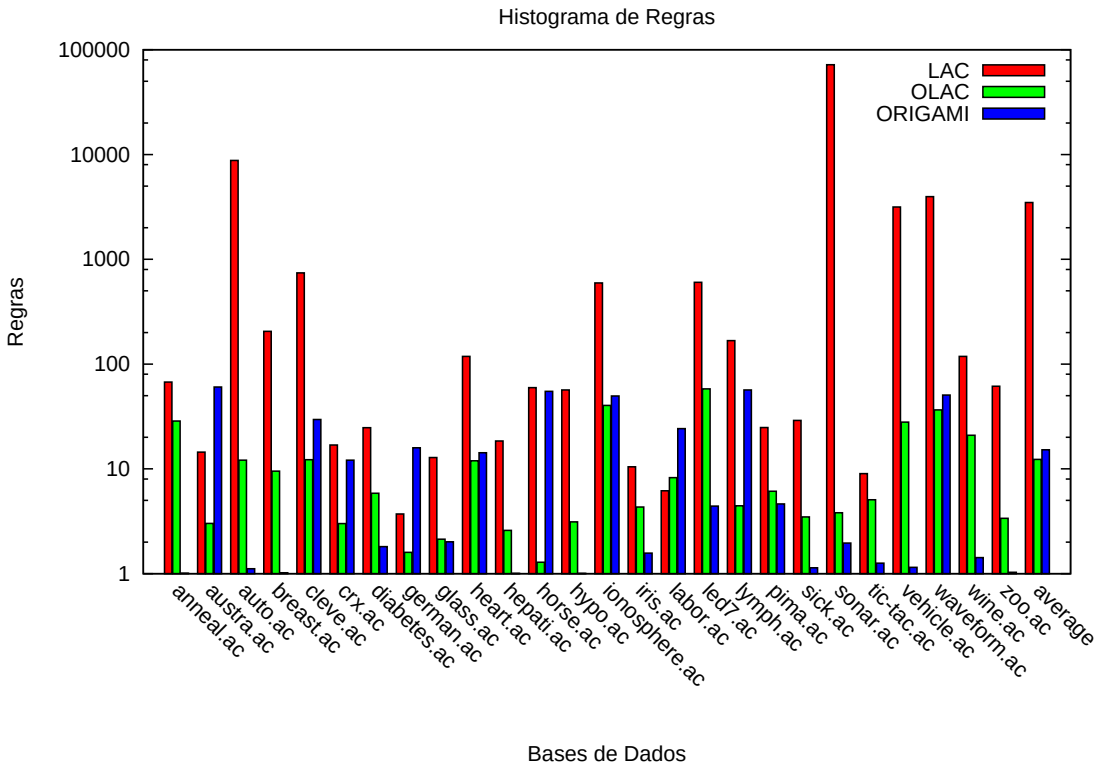

Figura B.3: istograma de Regras (melhores resultados para cada base de dados)

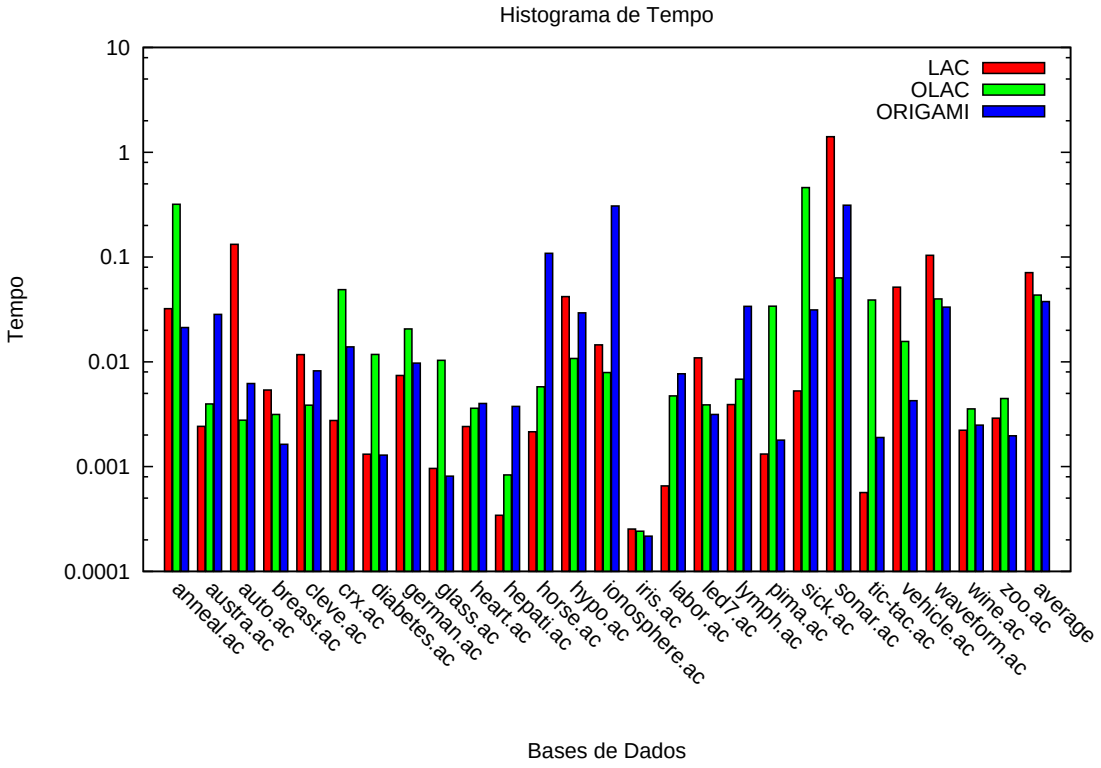

Figura B.4: Histograma de Tempo (melhores resultados para cada base de dados)

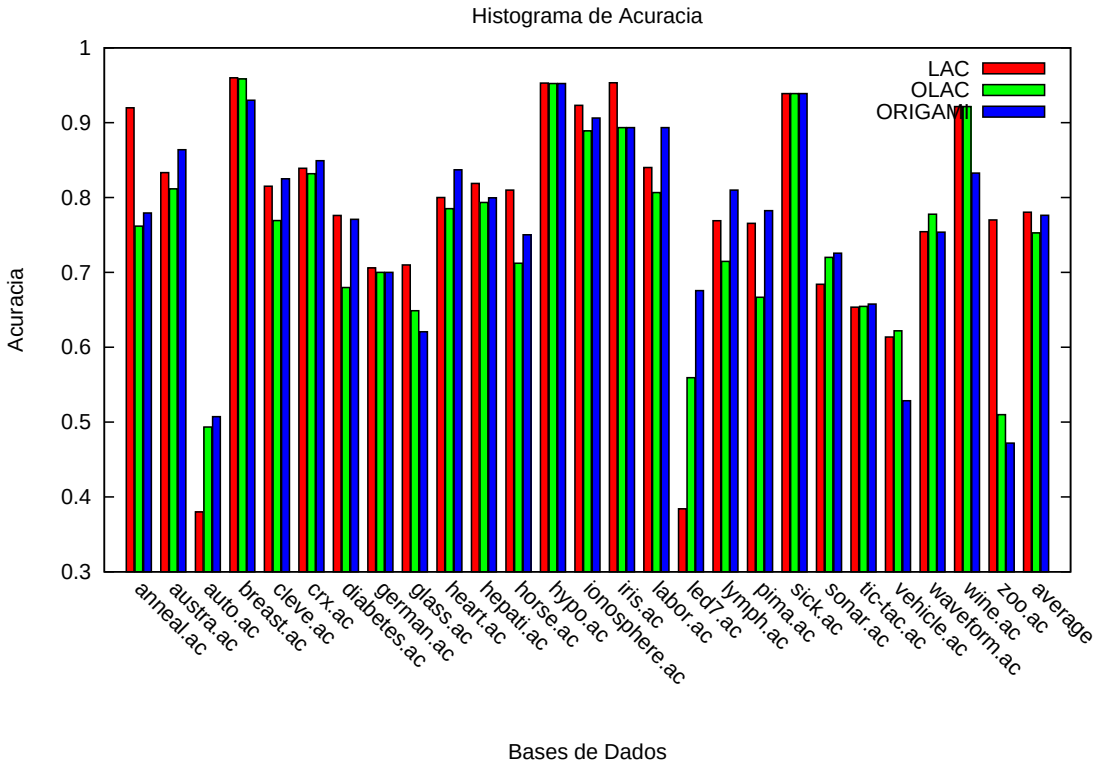

Figura B.5: Histograma de Acurácia (média dos resultados para todas as bases de dados)

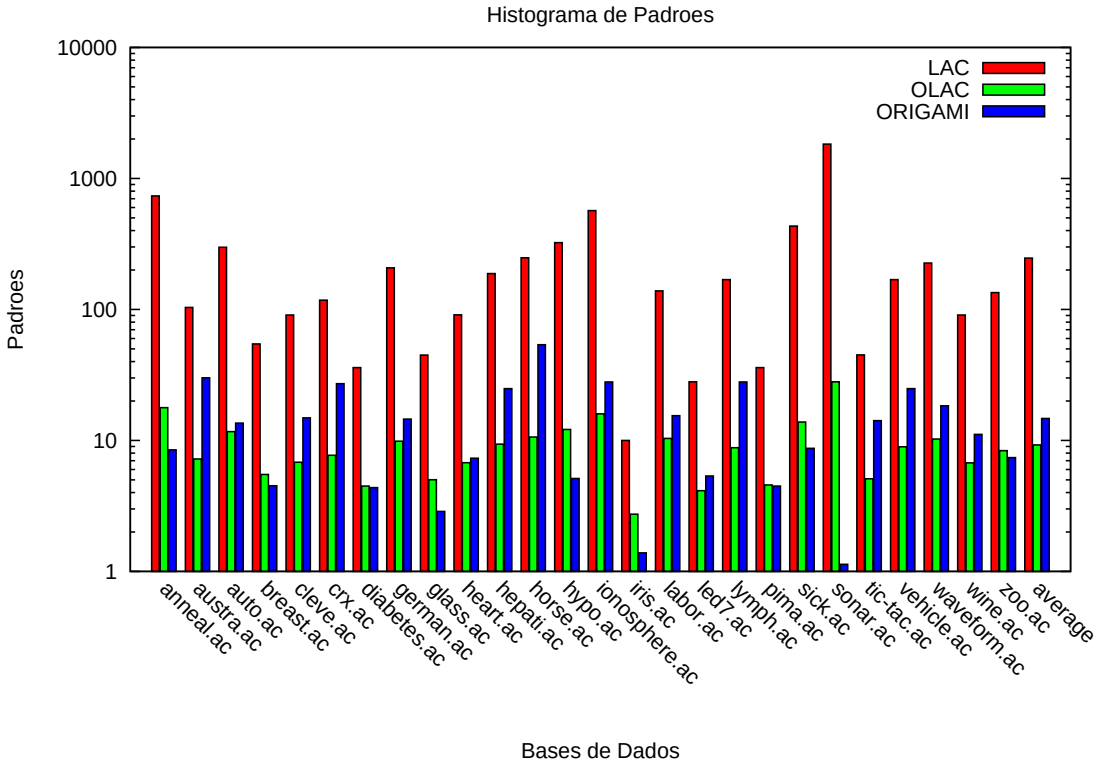

Figura B.6: Histograma de Padrões (média dos resultados para todas as bases de dados)

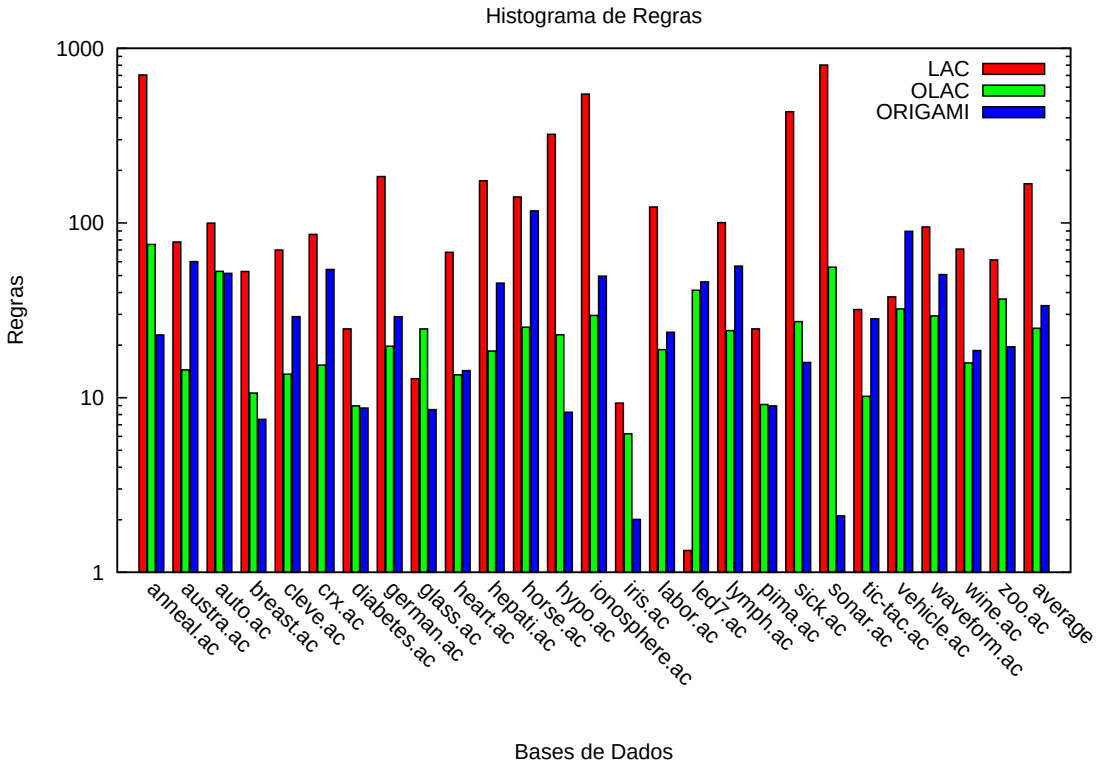

Figura B.7: Histograma de Regras (média dos resultados para todas as bases de dados)

dataset	**s**	**c**	**n**	**l**	**m**	**x**	**pat.**	**rul.**	**tim.**	**acc.**
anneal.ac	0.01	0.9	1	100	1	2	736.32	67.32	0.03	0.92
austra.ac	0.1	0.8	1	100	1	2	85.82	14.46	0.00	0.86
auto.ac	0.0001	0.01	1	1000000	1	3	2302.02	8770.13	0.13	0.55
breast.ac	0.0001	0.1	1	1000	1	3	174.43	205.40	0.01	0.98
cleve.ac	0.0001	0.0001	1	1000	1	3	375.57	742.42	0.01	0.83
crx.ac	0.1	0.8	1	1000	1	2	97.78	16.92	0.00	0.86
diabetes.ac	0.01	0.6	1	1000	1	2	35.99	24.77	0.00	0.78
german.ac	0.01	0.9	1	10	1	2	207.91	3.71	0.01	0.72
glass.ac	0.01	0.6	1	100	1	2	44.88	12.83	0.00	0.71
heart.ac	0.2	0.3	1	1000	1	2	83.33	118.57	0.00	0.83
hepati.ac	0.1	0.5	1	1	1	1	18.44	18.44	0.00	0.85
horse.ac	0.3	0.5	1	100	1	2	59.60	59.60	0.00	0.81
hypo.ac	0.1	0.99	1	10	1	2	290.84	56.59	0.04	0.98
ionosphere.ac	0.01	0.4	1	1000	1	2	568.61	594.37	0.01	0.92
iris.ac	0.1	0.3	1	10	1	2	9.47	10.47	0.00	0.97
labor.ac	0.3	0.9	1	10	1	2	54.03	6.18	0.00	0.93
led7.ac	0.0001	0.0001	1	1000	1	3	63.00	603.96	0.01	0.73
lymph.ac	0.01	0.5	1	1000	1	2	168.91	167.61	0.00	0.80
pima.ac	0.01	0.6	1	1000	1	2	35.99	24.77	0.00	0.77
sick.ac	0.0001	0.4	1	1	1	1	29.00	29.00	0.01	0.97
sonar.ac	0.0001	0.0001	1	100000	1	3	36045.44	71839.85	1.41	0.81
tic-tac.ac	0.1	0.5	1	1	1	1	9.00	9.00	0.00	0.71
vehicle.ac	0.0001	0.0001	1	10000	1	3	983.01	3159.35	0.05	0.67
waveform.ac	0.0001	0.0001	1	10000	1	3	1555.12	3956.62	0.10	0.82
wine.ac	0.01	0.3	1	1000	1	2	90.58	118.38	0.00	0.99
zoo.ac	0.01	0.6	1	1000	1	2	134.59	61.43	0.00	0.77

Tabela B.1: Melhores Parâmetros e Resultados para cada Base de Dados (LAC)

dataset	**s**	**c**	**n**	**l**	**m**	**x**	**e**	**w**	**g**	**pat.**	**rul.**	**tim.**	**acc.**
anneal.ac	0.0001	0.6	1	100	1	2	s	s	r	29.52	28.61	0.32	0.92
austra.ac	0.3	0.7	1	10	1	2	s	s	n	11.28	3.01	0.00	0.86
auto.ac	0.01	0.4	1	10	1	1	s	s	n	24.25	12.12	0.00	0.52
breast.ac	0.0001	0.0001	1	10	1	2	s	s	r	4.88	9.51	0.00	0.97
cleve.ac	0.01	0.3	1	1000	1	2	s	s	s	8.59	12.25	0.00	0.84
crx.ac	0.3	0.7	1	10	1	3	s	p	n	12.00	3.00	0.05	0.86
diabetes.ac	0.0001	0.0001	1	10	1	2	c	s	n	2.93	5.86	0.01	0.76
german.ac	0.0001	0.8	1	1000	1	2	s	s	s	12.35	1.60	0.02	0.72
glass.ac	0.0001	0.4	1	10	1	3	c	s	z	2.50	2.13	0.01	0.70
heart.ac	0.01	0.3	1	1000	1	2	s	s	s	8.50	11.95	0.00	0.83
hepati.ac	0.0001	0.3	1	100	1	1	c	s	s	2.03	2.59	0.00	0.86
horse.ac	0.3	0.7	1	10	1	2	s	s	n	13.11	1.29	0.01	0.81
hypo.ac	0.0001	0.99	1	10	1	1	s	p	z	25.00	3.13	0.01	0.98
ionosphere.ac	0.001	0.3	1	100	1	1	s	s	s	34.00	40.34	0.01	0.91
iris.ac	0.01	0.3	1	1000	1	2	s	s	s	3.61	4.32	0.00	0.97
labor.ac	0.1	0.6	1	10	1	2	s	s	z	9.44	8.25	0.00	0.93
led7.ac	0.0001	0.0001	1	100	1	2	s	s	s	5.82	58.09	0.00	0.67
lymph.ac	0.01	0.6	1	10	1	2	s	s	s	8.79	4.44	0.01	0.78
pima.ac	0.0001	0.4	1	100	1	5	l	s	r	4.80	6.11	0.03	0.76
sick.ac	0.001	0.2	1	100	1	2	c	s	s	3.26	3.48	0.46	0.96
sonar.ac	0.01	0.5	1	100	1	2	l	s	s	3.82	3.82	0.06	0.79
tic-tac.ac	0.01	0.7	1	10	1	5	l	s	z	7.89	5.07	0.04	0.80
vehicle.ac	0.0001	0.0001	1	100	1	2	s	s	z	8.00	27.95	0.02	0.64
waveform.ac	0.0001	0.0001	1	100	1	2	s	s	r	12.64	36.60	0.04	0.80
wine.ac	0.0001	0.0001	1	100	1	2	s	s	z	8.96	20.94	0.00	0.96
zoo.ac	0.01	0.6	1	10	1	2	s	s	s	8.36	3.38	0.00	0.73

Tabela B.2: Melhores Parâmetros e Resultados para cada Base de Dados (OLAC)

dataset	**s**	**c**	**n**	**l**	**e**	**a**	**b**	**pat.**	**rul.**	**tim.**	**acc.**
anneal.ac	0.0001	0.0001	1	1000	s	0.7	0.9	1.01	1.01	0.02	0.96
austra.ac	0.1	0.0001	1	1000	s	0.7	0.8	30.24	60.39	0.03	0.87
auto.ac	0.0001	0.0001	1	1	s	0.1	0.4	1.00	1.11	0.01	0.54
breast.ac	0.0001	0.0001	1	1	s	0.1	0.7	1.00	1.02	0.00	0.96
cleve.ac	0.1	0.0001	1	1000	s	0.7	0.8	15.11	29.60	0.01	0.84
crx.ac	0.1	0.0001	1	1000	s	0.3	0.8	6.06	12.11	0.01	0.86
diabetes.ac	0.0001	0.0001	1	1000	c	0.1	0.5	1.12	1.81	0.00	0.78
german.ac	0.0001	0.0001	1	100	c	0.5	0.9	15.64	15.91	0.01	0.73
glass.ac	0.0001	0.0001	1	1000	s	0.2	0.3	1.00	2.02	0.00	0.78
heart.ac	0.1	0.0001	1	1000	s	0.7	0.8	7.31	14.27	0.00	0.84
hepati.ac	0.0001	0.0001	1	10	s	0.2	0.5	1.01	1.01	0.00	0.85
horse.ac	0.1	0.5	1	1000	s	0.7	0.8	54.24	55.03	0.11	0.79
hypo.ac	0.0001	0.1	1	1	s	0.2	0.9	1.00	1.01	0.03	0.98
ionosphere.ac	0.1	0.0001	1	1000	s	0.7	0.8	27.97	49.60	0.31	0.91
iris.ac	0.1	0.1	1	1000	s	0.7	0.8	1.35	1.57	0.00	0.95
labor.ac	0.1	0.0001	1	1000	s	0.7	0.8	15.70	24.23	0.01	0.93
led7.ac	0.0001	0.0001	1	100	s	0.2	0.4	1.00	4.41	0.00	0.73
lymph.ac	0.1	0.0001	1	1000	s	0.7	0.8	27.96	56.69	0.03	0.81
pima.ac	0.1	0.5	1	1000	s	0.7	0.8	4.55	4.63	0.00	0.79
sick.ac	0.0001	0.0001	1	100	s	0.5	0.8	1.00	1.14	0.03	0.97
sonar.ac	0.1	0.1	1	1000	s	0.7	0.8	1.11	1.96	0.31	0.79
tic-tac.ac	0.0001	0.1	1	1	s	0.4	0.7	1.25	1.26	0.00	0.82
vehicle.ac	0.0001	0.0001	1	1000	s	0.2	0.7	1.02	1.15	0.00	0.68
waveform.ac	0.1	0.0001	1	1000	s	0.7	0.8	18.38	50.65	0.03	0.75
wine.ac	0.1	1	1	1	l	0.2	0.9	3.71	1.43	0.00	0.98
zoo.ac	0.0001	0.0001	1	1	s	0.3	0.8	1.02	1.03	0.00	0.78

Tabela B.3: Melhores Parâmetros e Resultados para cada Base de Dados (ORIGAMI)

	LAC	**OLAC**	**ORIGAMI**
support	0.01	0.01	0.1
confidence	0.6	0.001	0.0001
min-num-rules	1	1	1
max-num-rank-rules	1000	100	1000
min-rule-len	1	1	-
max-rule-len	2	2	-
metric	-	s	s
method	-	s	-
pat-ordering	-	s	-
alpha	-	-	0.7
beta	-	-	0.8

Tabela B.4: Melhores Parâmetros para cada Execução

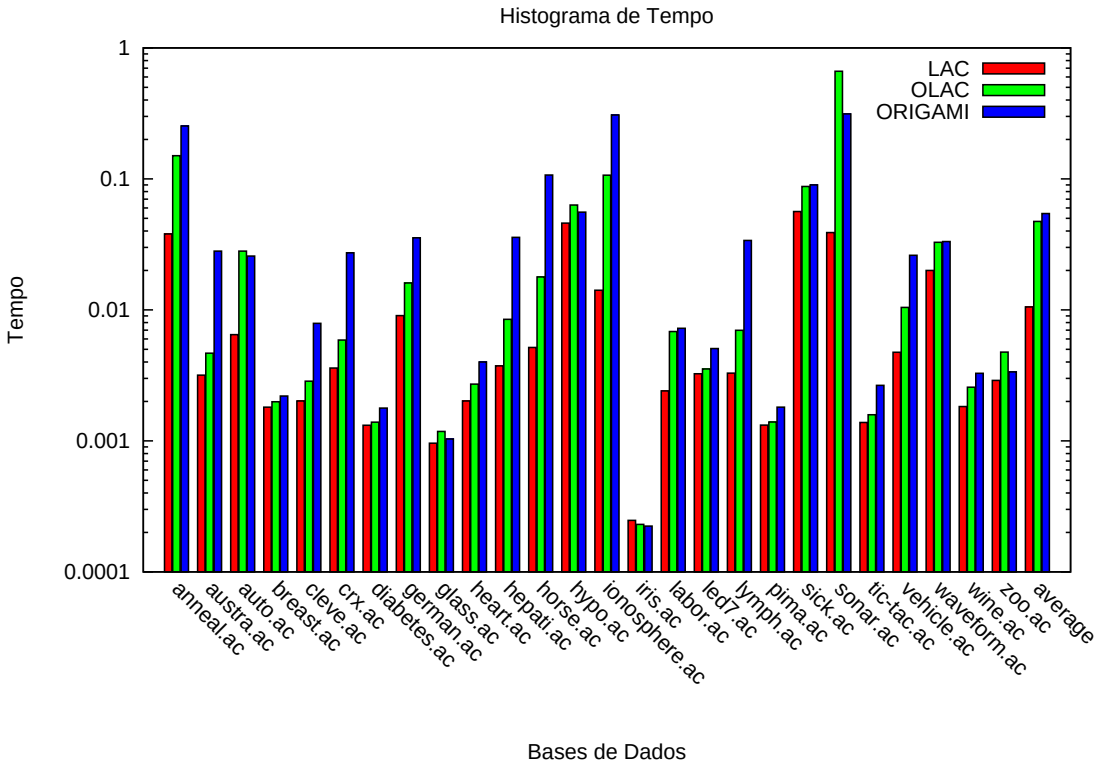

Figura B.8: Histograma de Tempo (média dos resultados para todas as bases de dados)

Data Sets	**OLAC & LAC**	**OLAC & ¬ LAC**	**¬ OLAC & LAC**	**¬ OLAC & ¬ LAC**
anneal.ac	704	28	32	34
austra.ac	573	17	21	79
auto.ac	77	29	35	64
breast.ac	674	1	8	16
cleve.ac	242	13	9	39
crx.ac	579	11	17	83
diabetes.ac	557	26	39	146
german.ac	631	89	93	187
glass.ac	138	12	14	50
heart.ac	218	6	6	40
hepati.ac	120	13	11	11
horse.ac	298	0	1	69
hypo.ac	3083	2	10	68
ionosphere.ac	314	4	10	23
iris.ac	145	0	0	5
labor.ac	51	2	2	2
led7.ac	2095	62	244	799
lymph.ac	106	10	12	20
pima.ac	544	41	44	139
sick.ac	2631	49	78	42
sonar.ac	143	21	26	18
tic-tac.ac	601	167	79	111
vehicle.ac	513	26	56	251
waveform.ac	3920	63	197	820
wine.ac	171	0	6	1
zoo.ac	67	7	11	16
average	738.27	26.88	40.81	120.50

Tabela B.5: Comparação entre LAC e OLAC (número de acertos)

Data Sets	**ORIGAMI & LAC**	**ORIGAMI & ¬ LAC**	**¬ ORIGAMI & LAC**	**¬ ORIGAMI & ¬ LAC**
anneal.ac	713	51	23	11
austra.ac	567	35	27	61
auto.ac	64	46	48	47
breast.ac	662	7	20	10
cleve.ac	245	10	6	42
crx.ac	562	32	34	62
diabetes.ac	562	39	34	133
german.ac	574	153	150	123
glass.ac	141	26	11	36
heart.ac	213	13	11	33
hepati.ac	115	17	16	7
horse.ac	289	3	10	66
hypo.ac	3060	53	33	17
ionosphere.ac	310	8	14	19
iris.ac	140	3	5	2
labor.ac	50	3	3	1
led7.ac	2221	106	118	755
lymph.ac	108	12	10	18
pima.ac	554	53	34	127
sick.ac	2675	45	34	46
sonar.ac	149	16	20	23
tic-tac.ac	595	195	85	83
vehicle.ac	455	124	114	153
waveform.ac	3682	87	435	796
wine.ac	174	1	3	0
zoo.ac	66	13	12	10
average	728.69	44.27	50.38	103.12

Tabela B.6: Comparação entre LAC e ORIGAMI (número de acertos)

Data Sets	**OLAC & ORIGAMI**	**OLAC & ¬ ORIGAMI**	**¬ OLAC & ORIGAMI**	**¬ OLAC & ¬ ORIGAMI**
anneal.ac	706	26	58	8
austra.ac	550	40	52	48
auto.ac	61	45	49	50
breast.ac	657	18	12	12
cleve.ac	238	17	17	31
crx.ac	547	43	47	53
diabetes.ac	548	35	53	132
german.ac	594	126	133	147
glass.ac	144	6	23	41
heart.ac	211	13	15	31
hepati.ac	120	13	12	10
horse.ac	289	9	3	67
hypo.ac	3053	32	60	18
ionosphere.ac	309	9	9	24
iris.ac	140	5	3	2
labor.ac	49	4	4	0
led7.ac	2039	118	288	755
lymph.ac	104	12	16	16
pima.ac	546	39	61	122
sick.ac	2662	18	58	62
sonar.ac	139	25	26	18
tic-tac.ac	661	107	129	61
vehicle.ac	431	108	148	159
waveform.ac	3637	346	132	885
wine.ac	168	3	7	0
zoo.ac	65	9	14	13
average	718.00	47.15	54.96	106.35

Tabela B.7: Comparação entre OLAC e ORIGAMI (número de acertos)

Referências Bibliográficas

Afrati, F.; Gionis, A. e Mannila, H. (2004). Approximating a collection of frequent sets. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 12–19, New York, NY, USA. ACM.

Agrawal, R.; Imieliński, T. e Swami, A. (1993). Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pp. 207–216, New York, NY, USA. ACM.

Boulicaut, J.; Bykowski, A. e Rigotti, C. (2003). Free-Sets: A condensed representation of boolean data for the approximation of frequency queries. *Data Min. Knowl. Discov.*, 7(1):5–22.

Breiman, L.; Friedman, J.; Olshen, R. e Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Brin, S.; Motwani, R.; Ullman, J. D. e Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. In Peckham, J., editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*, pp. 255–264. ACM Press.

Bunke, H. e Shearer, K. (1998). A graph distance metric based on the maximal common subgraph. *Pattern Recogn. Lett.*, 19(3-4):255–259.

Clark, P. e Boswell, R. (1991). Rule induction with cn2: Some recent improvements. In *EWSL '91: Proceedings of the European Working Session on Machine Learning*, pp. 151–163, London, UK. Springer-Verlag.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. e Stein, C. (2001). *Introduction to Algorithms*. The MIT Press, 2nd edição.

Cui, Y.; Fern, X. Z. e Dy, J. G. (2007). Non-redundant multi-view clustering via orthogonalization. In *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)*, Omaha, NE.

Dong, G.; Zhang, X.; Wong, L. e Li, J. (1999). CAEP: Classification by aggregating emerging patterns. In *Discovery Science*, pp. 30–42.

Fayyad, U.; Piatetsky-Shapiro, G. e Smyth, P. (1996). The kdd process for extracting useful knowledge from volumes of data. *Commun. ACM*, 39(11):27–34.

Frawley, W. J.; Piatetsky-Shapiro, G. e Matheus, C. J. (1992). Knowledge discovery in databases: an overview. *AI Mag.*, 13(3):57–70.

Friedman, J. H.; Kohavi, R. e Yun, Y. (1996). Lazy decision trees. In Shrobe, H. e Senator, T., editores, *Proceedings of the Thirteenth National Conference on Artificial Intelligence and the Eighth Innovative Applications of Artificial Intelligence Conference*, pp. 717–724, Menlo Park, California. AAAI Press.

Geng, L. e Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. *ACM Comput. Surv.*, 38(3):9.

Goethals, B. (2003). Survey on frequent pattern mining.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional.

Han, J. e Kamber, M. (2000). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Hand, D. J.; Smyth, P. e Mannila, H. (2001). *Principles of data mining*. MIT Press, Cambridge, MA, USA.

Jain, A. K. e Dubes, R. C. (1988). *Algorithms for clustering data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

James, M. (1985). *Classification algorithms*. Wiley-Interscience, New York, NY, USA.

Li, W.; Han, J. e Pei, J. (2001). CMAR: Accurate and efficient classification based on multiple class-association rules. In *ICDM*, pp. 369–376.

Lippmann, R. P. (1988). An introduction to computing with neural nets. *SIGARCH Comput. Archit. News*, 16(1):7–25.

Liu, B.; Hsu, W. e Ma, Y. (1998). Integrating classification and association rule mining. In *Knowledge Discovery and Data Mining*, pp. 80–86.

Mielikäinen, T. e Mannila, H. (2003). The pattern ordering problem. In Lavrac, N.; Gamberger, D.; Blockeel, H. e Todorovski, L., editores, *PKDD*, volume 2838 of *Lecture Notes in Computer Science*, pp. 327–338. Springer.

Nagao, K. e Sohma, M. (1998). Recognizing faces by weakly orthogonalizing against perturbations. In *ECCV '98: Proceedings of the 5th European Conference on Computer Vision-Volume II*, pp. 613–627, London, UK. Springer-Verlag.

Piatetsky-Shapiro, G. (1991). Discovery, analysis, and presentation of strong rules. In *Knowledge Discovery in Databases*, pp. 229–248. AAAI/MIT Press.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sassi, R. J. (2006). *Uma arquitetura híbrida para descoberta de conhecimento em bases de dados: teoria dos rough sets e redes neurais artificiais mapas auto-organizáveis*. PhD thesis, Escola Politécnica da Universidade de São Paulo.

Smith, J. M.; Stotts, D. e Kum, S. (2000). An orthogonal taxonomy for hyperlink anchor generation in video streams using ovaltine. In *HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia*, pp. 11–18, New York, NY, USA. ACM.

Tan, P.; Kumar, V. e Srivastava, J. (2002). Selecting the right interestingness measure for association patterns. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 32–41, New York, NY, USA. ACM.

Tan, P.; Steinbach, M. e Kumar, V. (2006). *Introduction to Data Mining*. Addison-Wesley.

Veloso, A. e Meira Jr., W. (2006). Lazy associative classification for content-based spam detection. In *LA-WEB '06: Proceedings of the Fourth Latin American Web Congress*, pp. 154–161, Washington, DC, USA. IEEE Computer Society.

Veloso, A.; Meira Jr., W.; Cristo, M.; Gonçalves, M. e Zaki, M. J. (2006a). Multi-evidence, multi-criteria, lazy associative document classification. In *CIKM ' 06: Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 218–227, New York, NY, USA. ACM.

Veloso, A.; Meira Jr., W.; Gonçalves, M. e Zaki, M. J. (2007). Multi-label lazy associative classification. In *European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pp. 605–612.

Veloso, A.; Meira Jr., W. e Zaki, M. J. (2006b). Lazy associative classification. *ICDM*, 0:645–654.

Wang, C. e Parthasarathy, S. (2006). Summarizing itemset patterns using probabilistic models. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 730–735, New York, NY, USA. ACM.

Wu, T.; Chen, Y. e Han, J. (2007). Association mining in large databases: A re-examination of its measures. In *PKDD*, pp. 621–628.

Xin, D.; Cheng, H.; Yan, X. e Han, J. (2006). Extracting redundancy-aware top-k patterns. In Eliassi-Rad, T.; Ungar, L. H.; Craven, M. e Gunopulos, D., editores, *KDD*, pp. 444–453. ACM.

Xin, D.; Han, J.; Yan, X. e Cheng, H. (2005). Mining compressed frequent-pattern sets. In *VLDB '05: Proceedings of the 31st international conference on Very large data bases*, pp. 709–720. VLDB Endowment.

Yin, X. e Han, J. (2003). Cpar: Classification based on predictive association rules. In Barbará, D. e Kamath, C., editores, *SDM*. SIAM.

Zaki, M. J.; Hasan, M.; Chaoji, V.; Salem, S. e Besson, J. (2007). ORIGAMI: Mining representative orthogonal graph patterns. In Press, I. C. S., editor, *Proc. IEEE Int. Conf. on Data Mining ICDM'07*, pp. 153–163.

Zhu, F.; Yan, X.; Han, J.; Yu, P. S. e Cheng, H. (2007). Mining colossal frequent patterns by core pattern fusion. In *ICDE*, pp. 706–715. IEEE.