Instituto de Computação
Universidade Estadual de Campinas

Anotação Automática de Imagens Utilizando Regras de Associação

Guilherme Moraes Armigliatto

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Guilherme Moraes Armigliatto e aprovada pela Banca Examinadora.

Campinas, 17 de junho de 2011.

Ricardo Torres
Prof. Dr. Ricardo da Silva Torres
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA POR
Maria Fabiana Bezerra Müller – CRB8/6162 – BIBLIOTECA DO INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA - UNICAMP

Armigliatto, Guilherme Moraes
Ar55a Anotação automática de imagens utilizando regras de associação / Guilherme Moraes Armigliatto. -- Campinas, SP : [s.n.], 2011.

Orientador: Ricardo da Silva Torres.
Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

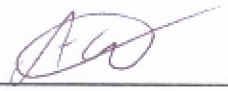
1. Sistemas de recuperação da informação. 2. Processamento de imagens. 3. Mineração de dados (Computação). 4. Programação genética (Computação). I. Torres, Ricardo da Silva, 1977-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em inglês: Automatic image annotation using associative rules
Palavras-chave em inglês:
Information retrieval systems
Image processing
Data mining (Computing)
Genetic programming (Computer)
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Ricardo da Silva Torres [Orientador]
Adriano Alonso Veloso
Claudia Maria Bauzer Medeiros
Data da defesa: 17-06-2011
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Dissertação Defendida e Aprovada em 17 de junho de 2011, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Adriano Alonso Veloso
Instituto de Ciências Exatas / UFMG

Profª. Drª. Claudia Maria Bauzer Medeiros
IC / UNICAMP

Prof. Dr. Ricardo da Silva Torres
IC / UNICAMP

Instituto de Computação
Universidade Estadual de Campinas

Anotação Automática de Imagens Utilizando Regras de Associação

Guilherme Moraes Armigliatto[1]

Junho de 2011

Banca Examinadora:

- Prof. Dr. Ricardo da Silva Torres (Orientador)
- Prof. Dr. Adriano Veloso
  Universidade Federal de Minas Gerais (UFMG)
- Prof[a]. Dr[a]. Claudia Bauzer Medeiros
  Universidade Estadual de Campinas (UNICAMP)
- Prof. Dr. André Santanchè
  Universidade Estadual de Campinas (UNICAMP)
- Prof. Dr. João Paulo Papa
  Universidade Estadual Paulista Júlio de Mesquita Filho (Unesp-Bauru)

[1]Apoio financeiro de: Bolsa do CNPq (processo 133348/2009-1) no período 2009–2011.

Resumo

Com os avanços tecnológicos, grandes coleções de imagens são geradas, manipuladas e armazenadas em bancos de dados. Dado o grande tamanho destes bancos, verifica-se a necessidade de se criar ferramentas para gerenciá-los de forma eficiente e eficaz. Uma das tarefas mais demandadas deste gerenciamento é a recuperação das imagens, e uma forma de fazê-la é baseada no uso de anotações textuais associadas às imagens (por exemplo, palavras-chave e categorias). Entretanto, a anotação manual de grandes coleções de imagens apresenta vários problemas, como o alto consumo de tempo e a não padronização dos termos utilizados.

Desse modo, esta dissertação apresenta quatro novos métodos para anotação automática de imagens, que visam amenizar estes problemas. Estes métodos utilizam as abordagens de descritores de imagens, dicionários visuais, programação genética e regras de associação. Os descritores e os dicionários são utilizados para representar as propriedades visuais das imagens, a programação genética é usada para combinar estas características e as regras de associação são usadas para relacioná-las com anotações. A principal contribuição desta dissertação consiste na análise do comportamento das regras de associação utilizadas para anotação de imagens em um conjunto de experimentos. Resultados experimentais demonstraram que os métodos propostos apresentam desempenho comparável ou superior ao de técnicas tradicionais da literatura.

Abstract

With technological advances, large collections of images are generated, handled and, stored in databases. Given the large size of these collections, there is a need for tools to manage efficiently and effectively these images. One of the most demanding tasks of this management is the retrieval of images from databases, usually based on the use of textual annotations associated with images (for example, keywords and categories). However, manual annotation of large images collections face a lot of problems related to the huge time required to annotate and the lack of standardization of used terms.

This work presents four new methods for automatic image annotation. These methods rely on the use of image descriptors, visual dictionaries, genetic programming, and association rules. The descriptors and dictionaries are used to represent the visual properties of images, genetic programming is used to combine extracted visual features, and association rules are used to associate them with annotations. The main contribution of this work is views on the analyze the behavior of association rules used for annotating images on a set of experiments. Experimental results demonstrated that the proposed methods have performance comparable or superior to traditional techniques of literature.

Agradecimentos

Eu gostaria de agradecer primeiramente a minha família (pai, mãe e irmão) pelo suporte, confiança e apoio. Ao meu orientador Ricardo Torres, pela liderança, paciência e persistência para comigo. Ao Adriano Veloso e Eduardo Valle pela ajuda inicial deste projeto. Gostaria de agradecer à Claudia Bauzer Medeiros por ser minha orientadora inicial. Aos amigos de laboratório (LIS e RECOD) pela colaboração e convivência, em especial o Fábio Faria (Gordinho), Felipe Andrade (Sansão), Otávio Penatti (Otaviano) e o Rodrigo Tripodi (Baiano). Ao time do LIS, que foi campeão do Interlabs IC/2009. Aos amigos e colegas do IC da UFMT e da UNICAMP. Aos amigos de Cuiabá Douglas Leite e Willian Maja, que além do suporte inicial em Campinas, foram companheiros e parceiros para todas as horas. Agradeço ao CNPq, a FAPESP e a CAPES pelo apoio financeiro. Por fim, a todos que contribuíram direta ou indiretamente para a construção desta dissertação de mestrado.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

Com o avanço das tecnologias de aquisição e armazenamento de imagens, juntamente com o uso da internet, surgiram grandes coleções de imagens que vêm sendo usadas em várias áreas do conhecimento. A recuperação da informação de modo eficiente e eficaz é um dos problemas enfrentados no gerenciamento dessa crescente quantidade de dados.

A abordagem mais comum para recuperação de imagens é baseada na associação de descrições textuais às imagens e no uso de técnicas tradicionais de recuperação textual em bancos de dados [61]. Esta operação exige uma etapa anterior voltada à anotação. Outra abordagem é a Recuperação de Imagens baseada em Conteúdo. Neste caso, utilizam-se características das imagens para comparar a similaridade (distância) entre elas, considerando-se propriedades visuais tais como cor, textura e forma. A recuperação por texto tem a vantagem de ser mais simples de ser realizada, além do fato de que geralmente o texto descreve semanticamente a imagem, oferecendo possibilidades mais intuitivas para especificação de consultas.

O processo de se anotar uma imagem consiste em associar metadados textuais, por exemplo palavras-chaves, à imagem. A anotação manual de uma grande base de imagens é custosa e apresenta problemas, como a grande quantidade de tempo para gerar as anotações e a não padronização dos termos utilizados para descrever uma mesma característica. Para tentar resolver alguns desses problemas, foram desenvolvidos métodos de anotação semi-automática de imagens. Esta abordagem tem como objetivo amenizar a intervenção do usuário, a partir, por exemplo, de sugestão de termos a serem usados na anotação. Há também as abordagens de anotação automática, que objetivam diminuir consideravelmente a intervenção do usuário, aumentando assim a eficiência da anotação. Essa última abordagem geralmente exige a utilização de bases previamente anotadas para realização de treinamento, que pode ser feito utilizando-se, por exemplo, técnicas de aprendizagem.

Para representar as propriedades visuais das imagens, nesta dissertação utilizam-se

descritores globais que extraem vetores de características das imagens, levando-se em consideração suas propriedades visuais como um todo, tais como cor e textura. Também, usam-se detectores de pontos de interesse e descritores locais, sendo que os detectores extraem pontos locais de interesse de uma imagem e o descritor local calcula um vetor de característica para cada ponto.

Outra forma de representação utilizada é o dicionário visual, que é uma técnica que utiliza a agregação de características locais de uma imagem para descrevê-la. Por meio do dicionário, pode-se sintetizar o conteúdo visual de imagens em uma representação denominada "saco de palavras visuais" (do inglês *bag of visual words*), em que "palavras" arbitrárias são associadas aos padrões visuais mais marcantes dos dados.

Estas representações visuais podem ser combinadas, pois potencialmente fornecem diferentes informações que se complementam. Para realizar esta combinação, nesta dissertação utiliza-se Programação Genética (PG), que é uma técnica da área de Inteligência Artificial que visa encontrar soluções para problemas baseando-se nos princípios de herança biológica, seleção natural e evolução [6]. Desse modo, PG é empregada na combinação dos valores de similaridade obtidos por meio das representações visuais, objetivando a geração de funções de similaridade mais eficazes.

Assim, de posse das representações visuais das imagens, aplicam-se metodologias que permitem extrair conhecimentos a partir destas massas de dados. Em particular, as chamadas regras de associação (mineração de dados) permitem encontrar os padrões de co-ocorrência mais frequentes nos dados, a partir dos quais se procura inferir correlações, causalidades e outras relações lógicas existentes nos dados. As regras de associação são escritas da forma $X \rightarrow Y$ na qual o X é o antecedente e o Y é o consequente da regra. Desta maneira, quando estas regras são usadas em tarefas de classificação, o antecedente das regras são características e o consequente é uma classe. Nesta dissertação utilizam-se dois classificadores associativos, o *Apriori* [2] e o *Lazy Association Classifier* (LAC) [87], sendo que este último se diferencia por gerar regras de associação sob demanda para cada instância de teste específica.

Levando-se em conta que propriedades visuais semelhantes de diferentes imagens devem receber a mesma anotação, são propostos nesta dissertação quatro novos métodos para anotação automática de imagens, abordando aspectos de descritores de imagens, dicionários visuais, programação genética e regras de associação. Os descritores e os dicionários são utilizados para representar as propriedades visuais das imagens, a programação genética combina as características extraídas, e as regras de associação são usadas para encontrar relações entre estas características e as anotações. A principal contribuição é investigar e analisar a sinergia e o comportamento das regras de associação utilizadas para anotação de imagens.

Dentre as principais contribuições de pesquisa desta dissertação de mestrado, destacam-

se as seguintes:

- Uso de regras de associação para anotação automática de imagens;
- Uso de regras de associação com descritores globais, locais e dicionários visuais em tarefas de anotação automática de imagens;
- Uso de regras de associação com programação genética para anotação automática de imagens;
- Especificação e implementação do método *Lazy Association Classification for Image Annotation* (LIA), que utiliza as características visuais das imagens como atributos do classificador LAC;
- Especificação e implementação do método *Relevance-Based Association Rules for Image Annotation* (RAIA), que utiliza as similaridades entre as imagens como atributos do algoritmo *Apriori*;
- Especificação e implementação do método *Genetic Programming-Based Relevance for Image Annotation Using Association Rules* (GRIA). Este método combina as similaridades (distâncias) entre as imagens utilizando PG, gerando novas funções de similaridade. As novas distâncias geradas, por sua vez, são usadas como atributos de entrada do algoritmo *Apriori*;
- Especificação e implementação do método *Genetic Programming-Based Relevance for Image Annotation Using Lazy Association Classification* (GRIA-LAC). Este método é semelhante ao GRIA, sendo que a principal diferença é a utilização do classificador LAC ao invés do algoritmo *Apriori*.

O restante deste documento é organizado como segue: o próximo capítulo apresenta os conceitos básicos e os trabalhos correlatos. O Capítulo 3 descreve os métodos propostos. O Capítulo 4 exibe a validação experimental destes métodos, bem como a análise dos resultados. Finalmente, o Capítulo 5 apresenta as conclusões e os trabalhos futuros.

Capítulo 2

Conceitos Básicos e Trabalhos Correlatos

Este capítulo apresenta os conceitos fundamentais abordados ao longo desta dissertação, bem como os trabalhos correlatos. A Seção 2.1 apresenta conceitos relacionados à recuperação de imagens. A Seção 2.4 aborda os conceitos de Programação Genética (PG). A Seção 2.2 apresenta os conceitos de anotação de imagens. A Seção 2.3 apresenta o processo de geração de um dicionário visual. A Seção 2.5 apresenta as regras de associação. Por fim, a Seção 2.5.2 apresenta o classificador associativo sob demanda chamado *Lazy Association Classifier* (LAC).

2.1 Recuperação de Imagens

O objetivo da recuperação de imagens é encontrar imagens de um banco de dados que são relevantes para uma dada consulta de um usuário. Dois tipos de recuperação de imagens são discutidos a seguir: recuperação por busca textual e recuperação por conteúdo.

2.1.1 Recuperação por Busca Textual

A busca textual é um dos meios mais rápidos e mais utilizados para recuperação de dados. Este método utiliza descrições textuais (usualmente, palavras-chave) para caracterizar ou descrever as imagens semanticamente. O processo de anotação e recuperação depende da interpretação do conteúdo visual das imagens, e varia de acordo com o conhecimento, o objetivo, a experiência e a percepção de cada usuário. Além deste problema, existe a grande quantidade de tempo que se leva para anotar manualmente uma grande coleção de imagens.

Sistemas tradicionais de recuperação de imagens baseiam-se no cálculo de relevância baseado em informações textuais associadas às imagens da coleção [11,14,38,53,61]. Essas informações podem ser obtidas a partir de diferentes fontes como, por exemplo, anotações prévias associadas à imagem (ver Seção 2.2), textos próximos à imagem em uma página da web, legendas, metadados e até mesmo reconhecimento ótico de caracteres.

Para comparar as descrições textuais, diferentes medidas de similaridades podem ser utilizadas, segundo [13]. Para obter essas similaridades, uma das técnicas mais utilizadas consiste em representar os textos (documentos da coleção e consultas) no Modelo Vetorial (Vector Space Model [5]). Segundo este modelo, documentos são analisados como vetores. Suponha a existência de uma coleção com t termos distintos de índice t_j. Um documento d_i pode ser representado da seguinte maneira: $d_i = (w_{i1}, w_{i2}, \ldots, w_{it})$, onde w_{ij} representa o peso do termo t_j no documento d_i.

O peso para um dado termo t_j do documento d_i pode ser calculado como o respectivo valor $tf \times idf$, onde tf indica a frequência do termo no documento e idf a frequência inversa do termo na coleção. O valor de idf é dado por $log(N/nt)$, sendo N o número total de documentos na coleção e nt o número de documentos em que o termo t_j ocorre. Para um processo de recuperação simples, dado um termo de consulta, a coleção de documentos pode ser ordenada apenas de acordo com o peso do termo de consulta nos documentos. Para consultas com múltiplos termos, uma função de agregação pode ser utilizada para combinar os pesos dos termos e assim gerar uma medida de relevância final.

Considerando o espaço vetorial e o modelo de ponderação tf-idf, algumas medidas de similaridade (retiradas de [13]) são descritas a seguir.

$$Bag - of - words(d_1, d_2) = \frac{|\{d_1\} \bigcap \{d_2\}|}{|d_1|}, \tag{2.1}$$

onde $\{d_i\}$ representa o conjunto de termos que ocorrem no documento d_i. Essa é uma medida simples da porcentagem de palavras em comum dos dois documentos analisados.

$$Cosseno(d_1, d_2) = \frac{\sum_{i=1}^{t} w_{1i} \times w_{2i}}{\sqrt{\sum_{i=1}^{t} w_{1i}^2 \times \sum_{i=1}^{t} w_{2i}^2}} \tag{2.2}$$

onde w_{ij} é o documento como definido anteriormente. Essa fórmula basicamente calcula o cosseno entre os dois vetores que representam os documentos, de forma que quanto mais próximo de 1 seja esse cosseno, mais similares são os documentos.

$$Okapi(d_1, d_2) = \sum_{t \in d1 \bigcap d2} \frac{3 + tf_{d2}}{0,5 + 1,5 \times \frac{tam_{d2}}{tam_{med}} + tf_{d2}} \times log \frac{N - df + 0,5}{df + 0,5} \times tf_{d1} \tag{2.3}$$

onde tf é a freqüência do termo no documento, df é a freqüência do termo do documento na coleção inteira, N é o número de documentos na coleção inteira, tam_{di} é o tamanho do documento i, e tam_{med} é o tamanho médio de todos os documentos da coleção.

2.1.2 Recuperação por Conteúdo

Quando os dados envolvidos em predicados de busca são imagens, uma das maneiras de recuperá-las é por meio da Recuperação de Imagens baseada em Conteúdo (*Content-Based Image Retrieval* — CBIR). Para trabalhar com o conteúdo das imagens são utilizados **descritores de imagens**, que são definidos como um par composto por uma função que extrai características de uma imagem e uma função de distância (ou similaridade) [18]. O vetor de características representa um conjunto de propriedades visuais de uma determinada imagem, como cor, textura e forma, enquanto a função de distância calcula as similaridades entre duas imagens levando-se em conta seus vetores de características. Desse modo, uma consulta baseada em conteúdo normalmente recebe uma imagem de consulta e retorna as imagens mais similares à imagem consultada, ou seja, retorna as imagens com vetores de características mais próximos ao vetor de características da imagem fornecida.

Um dos problemas em CBIR é que a maioria das aplicações que possuem bom desempenho é específica para um contexto e para imagens bem "comportadas", ou seja, imagens já pré-processadas e de um domínio específico. Também, a escolha da imagem de consulta que melhor representa o que o usuário deseja buscar é uma tarefa difícil, uma vez que o usuário precisa traduzir conceitos (ideias) em características de baixo nível.

A seguir, formalizam-se esses conceitos da recuperação de imagens por conteúdo, conforme exposto em [18].

Definição 1 *Uma **imagem** $\hat{I}$ é um par $(D_I, \vec{I})$, onde:*

- *D_I é um conjunto finito de* pixels *(pontos em $\mathbb{Z}^2$, tal que, $D_I \subset \mathbb{Z}^2$), e*
- *$\vec{I} : D_I \to \mathsf{D}'$ é uma função que atribui a cada* pixel *p em D_I um vetor $\vec{I}(p)$ de valores em algum espaço arbitrário D' (por exemplo, $\mathsf{D}' = \mathbb{R}^3$ quando uma cor é atribuída a um* pixel *no sistema RGB).*

Definição 2 *Um **descritor simples** (ou simplesmente, **descritor**) D é definido como um par (ϵ_D, δ_D), onde:*

- *$\epsilon_D : \hat{I} \to \mathbb{R}^n$ é uma função que extrai um* vetor de características *$\vec{v}_{\hat{I}}$ de uma* imagem *$\hat{I}$.*

- $\delta_D : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ *é uma* função de similaridade *(por exemplo, baseada em uma medida de distância) que computa a similaridade entre duas imagens a partir da distância entre seus* vetores de características *correspondentes.*

Definição 3 *Um* ***vetor de características*** $\vec{v}_{\hat{I}}$ *de uma imagem* $\hat{I}$ *é um ponto no espaço* $\mathbb{R}^n$*:* $\vec{v}_{\hat{I}} = (v_1, v_2, ..., v_n)$*, onde* n *é a dimensão do vetor.*

Definição 4 *Um* ***descritor composto*** $\hat{D}$ *é um par* $(\mathcal{D}, \delta_{\mathcal{D}})$ *(veja Figura 2.2 (b)), onde:*

- $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ *é um conjunto de* k *descritores simples pré-definidos.*
- $\delta_{\mathcal{D}}$ *é um função de similaridade que combina os valores de similaridade obtidos de cada descritor* $D_i \in \mathcal{D}$*,* $i = 1, 2, \ldots, k$*.*

A Figura 2.1 ilustra o uso de um descritor simples D para computar a similaridade entre duas imagens $\hat{I}_A$ e $\hat{I}_B$. Primeiro, o algoritmo de extração ϵ_D é usado para computar os vetores de características $\vec{v}_{\hat{I}_A}$ e $\vec{v}_{\hat{I}_B}$ associados às imagens. Depois, a função de similaridade δ_D é utilizada para calcular o valor da similaridade d entre as imagens.

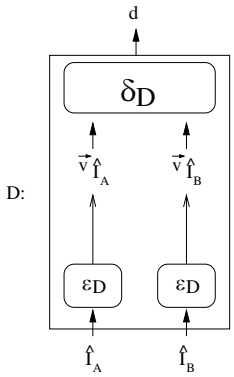


Figura 2.1: O uso de um descritor simples D para computar a similaridade entre duas imagens [18].

Já a Figura 2.2 ilustra um descritor composto $\hat{D}$, na qual são combinadas, utilizando a função $\delta_{\mathcal{D}}$, as distâncias de k descritores simples, retornando ao final uma única distância d.

Descritor Global *versus* Descritor Local

Um descritor global é caracterizado por levar em consideração informações de padrão global da imagem, sem focar sua análise em pontos locais. Desta forma, para cada imagem, o descritor global gera um único vetor de características. Como exemplos de descritores

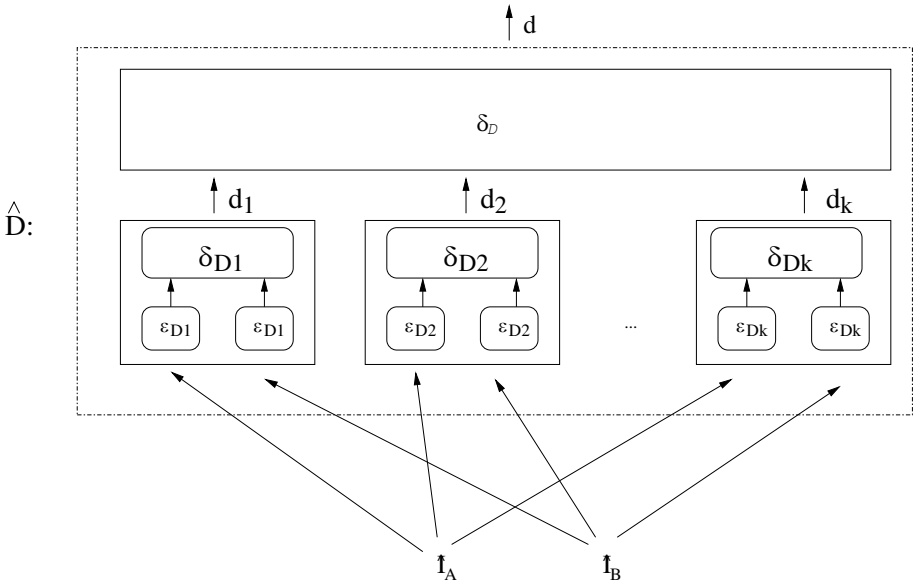


Figura 2.2: O uso de um descritor composto $\hat{D}$ para computar a similaridade entre duas imagens [18].

globais, podemos citar o descritor de cor *Border/Iterior Pixel Classification* (BIC) [75], e o descritor de textura *Local Activity Spectrum* (LAS) [79].

Já o descritor local descreve a imagem a partir de um conjunto de pontos de interesse. Estes pontos são regiões importantes da imagem, como regiões de bordas, e podem ser calculados por meio de detectores, como o Hessian-affine detector [60]. Assim, um detector extrai s pontos de interesse da imagem e o descritor local gera um vetor de dimensão d para cada ponto (gerando uma representação de $s \times d$ dimensões) [20]. A quantidade s de pontos varia para cada imagem, podendo ser até o valor zero. Os descritores locais mais conhecidos são o *Scale-invariant Feature Transform* (SIFT) [57] e o *Speeded up Robust Features* (SURF) [7].

2.2 Anotação de Imagens

Esta seção apresenta o conceito de anotação (2.2.1) e as abordagens de anotação semi-automática (Seção 2.2.2) e automática (Seção 2.2.3).

2.2.1 Visão Geral e Anotação Manual de Imagens

Anotar uma imagem consiste em associar descrições textuais a ela. Um dos objetivos da anotação de imagens é simplificar sua recuperação em uma base de imagens [34], podendo ainda ser usada para explicá-las ou documentá-las. Uma imagem bem descrita

textualmente pode ser recuperada com facilidade usando apenas técnicas de recuperação de texto [53,61]. A principal vantagem da anotação é que as descrições textuais, em geral, refletem o conteúdo semântico de uma imagem.

Os métodos comumente usados para associar informações textuais a uma imagem podem ser divididos em duas abordagens [34]: categorização e metadados. Na categorização, cada imagem é atribuída a uma única categoria dentre várias pré-definidas. Já os metadados usados para descrever a imagem podem ser classificados em:

- **Metadados independentes de conteúdo:** estes metadados estão relacionados com a imagem, mas não a descrevem diretamente. Por exemplo: nome do autor da imagem, a data em que foi obtida, a localização, etc.
- **Metadados referentes ao conteúdo visual:**
  - **Conteúdo direto:** estes metadados estão relacionados com características intermediárias, como cor, textura, forma e movimento.
  - **Descrição de conteúdo:** estes metadados referem-se ao conteúdo semântico. Estão preocupados não apenas com características isoladas da imagem, mas também com relações entre estas características, assim como qualquer informação de alto nível.

Os metadados independentes de conteúdo são difíceis de serem obtidos diretamente da imagem. Pode-se obter, por exemplo, a localização "Rio de Janeiro" de uma imagem reconhecendo algum ponto de referência pré-determinado como "Cristo Redentor". Os metadados de conteúdo direto são os mais fáceis de serem obtidos, para isso normalmente são utilizados descritores de imagens que extraem vetores de características. Já os metadados de descrição de conteúdo são os mais difíceis de obter, podendo ser especificados usando as seguintes abordagens [34]:

Descrição textual livre: combinações de palavras livres e frases que não possuem nenhuma estrutura pré-definida. Esse tipo de anotação é mais fácil de realizar, pois não se necessita de controle. Porém, isto dificulta a recuperação da imagem, visto que existe problemas como a não padronização dos termos, entre outros.

Palavras-chave: a imagem é descrita por meio de palavras-chave. Essas palavras podem ser arbitrárias, escolhidas de forma aleatória, ou restritas, escolhidas de acordo com um vocabulário controlado, definido antecipadamente.

Classificação baseada em ontologias: a anotação de imagens baseada em ontologias consiste, geralmente, em atribuir a uma imagem um conceito pertencente a uma

ontologia. Uma ontologia é uma "conceitualização explícita, formal e compartilhada, de uma área de conhecimento" [84]. Esta abordagem é semelhante à classificação por palavras-chave controladas, mas o fato de que as palavras-chave pertencem a uma rede de conceitos enriquece as anotações, permitindo realizar expansões de consultas [41].

Existem vários problemas associados à obtenção de descrição textual. Primeiro, o processo de anotar manualmente uma grande quantidade de imagens pode ser inviável devido à grande quantidade de tempo e esforço necessários. Além disso, segundo [30], pode ser ineficiente, já que é comum os usuários não realizarem as anotações de forma sistemática, por exemplo, preocupando-se em utilizar termos semelhantes aos usados anteriormente para determinada característica da imagem. Além disso, usuários diferentes têm grande chance de usar anotações distintas para uma mesma característica. Essa falta de sistematização pode prejudicar o desempenho da busca textual tradicional, uma vez que ela baseia-se na igualdade entre termos anotados da imagem e as fornecidas como parâmetros da busca [30].

Vários trabalhos abordam métodos para melhorar a eficácia da anotação manual de imagem, ajudando os usuários a produzirem melhores resultados. Por exemplo, em [34] discute-se a criação de um vocabulário controlado de palavras-chave e fornece-se um conjunto de orientações de anotação. Em [95] são descritos dois métodos de anotação chamados *tagging* e *browsing*, sugerindo-se uma junção das duas técnicas. Em [45] são analisadas as similaridades semânticas entre as palavras-chave para identificar quais palavras são irrelevantes. Em [90] é apresentada uma abordagem probabilística para refinar as anotações de uma imagem, incorporando relações semânticas entre anotações. Finalmente, o trabalho [92] propõe uma abordagem para reduzir o tamanho de ontologias clínicas para uma anotação manual mais eficiente de imagens médicas.

2.2.2 Anotação Semi-Automática de Imagens

Dado o esforço e a grande quantidade de tempo que se leva para anotar manualmente um elevado número de imagens, alguns sistemas utilizam a estratégia da anotação semi-automática. Esse tipo de abordagem realiza uma pré-anotação da imagem, porém trabalha sob controle humano, ou seja, o usuário é quem realmente determina o resultado final da anotação.

Dentre os trabalhos que utilizam essa abordagem, pode-se citar a ferramenta Onto-SAIA [30], que auxilia a busca e anotação de imagens, integrando as abordagens baseadas em conteúdo, ontologia e palavras-chaves. Nesse ambiente, cada imagem está associada a uma anotação, composta por termos, e a um conjunto de vetores de características para os

diferentes descritores de imagens. A anotação é baseada na sugestão de termos, levando em conta os termos definidos pelo usuário e o conteúdo da imagem.

Já o sistema apresentado em [98] utiliza as palavras-chave da imagem, refinando-as com a ajuda do usuário (*feedback*). Este refinamento é tratado por um *framework* de aprendizagem, chamado de *Semi-Automatic Dynamic Auxiliary-Tag-Aided* (SADATA), em que a classificação de uma determinada palavra-chave pode ser melhorada utilizando um subconjunto de outras palavras-chave auxiliares.

O trabalho [22] apresenta um processo de sugestão de palavras-chave que utiliza um descritor para recuperar imagens anotadas de um banco de dados semelhantes a que o usuário quer anotar. Posteriormente, de acordo com a frequência das palavras, utiliza-se mineração de texto para selecionar quais serão as candidatas para serem sugeridas.

Em [76], apresenta-se uma estratégia de anotação semi-automática de fotos baseada em agrupamentos e reconhecimento de faces. O sistema possui um *framework* que identifica automaticamente grupos de fotos em eventos, pessoas e metadados do arquivo, auxiliando desse modo a anotação do usuário.

O estudo [40] propõe uma plataforma online interativa que recomenda *tags* para imagens. O usuário marca um objeto específico em uma imagem, o sistema então retorna imagens que contêm objetos similares. Assim, a anotação pode ser feita utilizando as anotações das imagens retornadas. Se o usuário digitar uma nova *tag*, o sistema propaga esta *tag* para as imagens semelhantes.

Em [23], apresenta-se um sistema de anotação semi-automática que gera anotações baseadas em ontologias para redes sociais, aproveitando-se das anotações fornecidas pelos usuários mais ativos. Para a contextualização de uma imagem, este sistema utiliza vários fatores, como referência geográfica, tempo e a relação entre atores na rede.

O trabalho [12] propõe uma abordagem de sugestões de anotações. Dada uma coleção de imagens pelo usuário, o método seleciona um número pequeno destas imagens e atribui categorias a elas, aprendidas usando 15 categorias da coleção FLICKR [24]. Depois de obter *feedbacks* dos usuários, inferem-se as categorias das imagens restantes a partir da propagação sobre múltiplos grafos esparsos entre as imagens.

O trabalho de Macário [21] anota dados geográficos, dentre eles imagens de satélite, a partir de *workflows* que especificam os precedimentos que devem ser seguidos no processo de anotação. Inicialmente é preciso definir o esquema da anotação (campo) e posteriormente os dados são usados, sendo ligados a anotações.

2.2.3 Anotação Automática de Imagens

O processo de Anotação Automática de Imagens (*Automatic Image Annotation* — AIA) gera anotações com mínima intervenção do usuário, sendo uma abordagem que aumenta

significativamente a eficiência da anotação. A ideia básica da maioria dos trabalhos é que as características visuais que recebem a mesma anotação devem ser coerentes. Ou seja, as imagens ou regiões com características visuais semelhantes podem ser agrupadas e associadas a um determinado conjunto de anotações. Um problema neste tipo de abordagem é que palavras comuns podem ser associadas a regiões diferentes. Como resultado, palavras indesejadas têm chance de serem usadas na anotação. Também existem algumas dificuldades relacionadas ao reconhecimento de fundos muitos texturizados e objetos oclusos [34].

Além de associar características de regiões a textos, também é possível, segundo [102], relacionar textos entre si, melhorando assim o desempenho da anotação da imagem. Por exemplo, se em uma foto reconhecem-se objetos como areia, coqueiro, água e céu, é grande a possibilidade de que a imagem seja uma praia. Este tipo de relação pode ser feita por meio de regras de associação (ver Seção 2.5).

Segundo [47, 63] a maioria dos métodos de AIA são baseados em técnicas de aprendizagem que primeiro aprendem, por meio de treinamento, a relação entre as características visuais e textuais de imagens previamente anotadas, e em seguida utilizam o aprendizado para descrever as imagens ainda não anotadas. Às vezes, quando o número de imagens é pequeno e não é suficiente para um treinamento eficiente, é possível utilizar realimentação de relevância para refinar o resultado da aprendizagem. Realimentação de relevância refere-se a um ciclo interativo em que o usuário seleciona uma parte do resultado de uma busca que ele julga relevante. O sistema então utiliza as indicações de relevância dos resultados para aprimorar as próximas consultas.

O trabalho [54] descreve um método para AIA que utiliza uma base de dados de imagens anotadas, chamado *visual folksonomies*. O método aplica duas técnicas com base em análise de imagens. Em primeiro lugar, usa uma técnica de aprendizagem supervisionada para classificação, que anota as imagens com um vocabulário controlado. Em segundo lugar, utiliza recuperação de imagem por conteúdo para propagar a anotação para imagens visualmente semelhantes.

Em [91], apresenta-se um método que combina características globais, locais e textuais de imagens anotadas, integrando os três tipos de informações para descrever sua semântica por meio de uma estimativa conjunta de probabilidades. As características globais fornecem, utilizando treinamento de imagens, a distribuição global dos temas visuais. Já as características locais ajudam no reconhecimento de objetos.

Já em [100], propõe-se um novo método de AIA baseado em um modelo chamado *Hidden Markov Model — HMM*. O método encontra regiões de maior importância e não leva em conta somente a relevância da relação do conteúdo da imagem com a anotação, mas também a relação que existe entre as palavras-chaves.

O trabalho [83] supõe que usuários geralmente anotam não uma imagem apenas, mas

um grupo de imagens que provavelmente são do mesmo estilo (por exemplo, imagens de esportes). Então, utiliza-se um método baseado em *probabilistic Latent Semantic Analysis* (pLSA) que aprende o conteúdo destas imagens e as casam com uma categoria, dentre mais de 200.000, da base FLICKR.

Em [78], apresenta-se um novo método de aprendizado semi-supervisionado baseado em grafos knn-esparsos para propagação de anotações de imagens. Este método usa a busca aproximada dos k vizinhos mais próximos (do ingês KNN) para garantir a eficiência, enquanto o treinamento eficaz de formação da anotação é feito pelo grafo.

O estudo apresentado em [101] anota automaticamente imagens da web, considerando o texto dos sites em que elas se encontram. Primeiro, um pré-processamento extrai os blocos de texto mais importantes das páginas web. Segundo, agrupam-se as imagens visualmente mais semelhantes, na qual a semântica das imagens do mesmo agrupamento são descritas por uma lista ranqueada dos termos que frequentemente co-ocorrem nos seus blocos de texto.

O trabalho [71] modela a tarefa de anotação como um problema de classificação multi-classe. A relação entre as características de baixo nível das imagens e sua semântica é resolvida utilizando técnicas de aprendizagem supervisionadas Bayesianas. Para cada região da imagem de teste, calcula-se a probabilidade a posteriori para cada conceito a partir do conjunto de treinamento e, em seguida, a probabilidade é modificada de acordo com a relevância das outras regiões da imagem. As probabilidades finais são obtidas combinando-se as probabilidades de cada região, e as palavras-chave são selecionadas de acordo com *ranks*.

2.3 Dicionários Visuais

A técnica de dicionários visuais consiste em representar uma imagem a partir de suas características visuais locais [96]. Os dicionários são usualmente obtidos aplicando-se um algoritmo de agrupamento (*clustering*) em regiões de interesse das imagens, agrupando pequenas quantidades de características semelhantes, chamadas de palavras visuais, calculadas por meio de um descritor local. Os centros dos agrupamentos são as representações das palavras visuais do dicionário [42].

Fazendo uma analogia entre uma base de imagens e um conjunto de documentos de texto, pode-se dizer que os *pixels* das imagens corresponderiam às letras dos documentos, as palavras visuais equivaleriam aos vocábulos dos textos, e, finalmente, o dicionário visual (conjunto de palavras visuais da base) seria análogo ao vocabulário dos documentos. Desconsiderando qualquer estrutura lógica, os *pixels* e as letras possuem pouca informação semântica. Contudo, quando agrupados, formam palavras e agregam maior significado. De maneira similar, a caracterização semântica do documento melhora ao levar em con-

sideração a estrutura lógica entre palavras.

O processo de geração dos dicionários visuais pode ser descrito, de forma simplificada, em três etapas principais [74, 80]:

1. **Detecção de regiões de interesse:** esta etapa é normalmente feita por detectores como o Hessian-affine detector [80] ou o Harris-Laplace [56,85,97]. Alguns trabalhos consideram como regiões de interesse a imagem dividida em células de tamanho fixo (*grids*) [52, 56, 89].

2. **Descrição das regiões detectadas:** esta etapa é realizada por meio de descritores, na qual o mais utilizado é o SIFT [57] ou variações dele.

3. **Agrupamento dos vetores de características:** o espaço dos vetores de características é agrupado e cada grupo representa uma palavra visual. Este agrupamento é feito na maioria dos trabalhos pelo algoritmo *k-means*.

De posse de um dicionário, cada imagem é representada por um histograma de palavras visuais, chamado de saco de palavras visuais. Dessa forma, cada imagem é vista apenas como uma coleção de regiões nas quais a informação espacial da região não é levada em conta, somente a quantidade destas regiões é considerada. Assim, a partir dos dicionários visuais, é possível adaptar técnicas antes utilizadas somente para o processamento de textos, como regras de associação (ver Seção 2.5), e utilizá-las em tarefas relacionadas à imagem. Nesta dissertação, utilizam-se dicionários visuais e regras de associação para anotação automática de imagens.

Como somente a quantidade de palavras visuais é considerada, um dos problemas dessa representação é a perda de informação espacial das mesmas. Para tentar diminuir esta questão, o trabalho [80] organiza as palavras visuais de maneira a manter sua distribuição espacial na imagem e são usadas para formar uma sentença visual. Já em [56], histogramas espaciais são usados para caracterizar a imagem e suas palavras visuais.

Além do alto custo computacional de construção do dicionário [72,80], outro problema é a presença de ruídos de palavras visuais devido ao processo de construção do vocabulário, por exemplo, para agrupar os vetores de características geralmente utilizam-se algoritmos como o k-means, que pode não convergir em um número fixo de iterações, possivelmente gerando maus agrupamentos. O trabalho [80] propõe duas novas técnicas para eliminar palavras inúteis, uma baseada em propriedades geométricas dos vetores de característica, e a outra utiliza uma técnica, originalmente aplicada em análise de documentos textuais, chamada *probabilistic Latent Semantic Analysis* (pLSA) [35].

Após a geração do dicionário visual, uma etapa de treinamento é realizada para associar conceitos semânticos a palavras visuais, geralmente utilizando imagens que já possuem conceitos associados. Uma das maneiras de realizar este treinamento é utilizando técnicas

inicialmente propostas para análise de documentos, como pLSA e a *Latent Dirichlet Allocation* (LDA) [8]. Outra forma é por meio da classificação supervisionada, sendo que as técnicas mais conhecidas são o *Support Vector Machines* (SVM) [32, 46, 50, 85, 89, 94] e o método de vizinhos mais próximos (kNN) [82, 89, 94].

Terminada a associação dos conceitos semânticos às palavras visuais, o dicionário é utilizado em várias tarefas relacionadas às imagens e vídeos, como por exemplo, anotação textual [42], e classificação de cenas e objetos [36, 64, 72, 80, 82, 89, 94, 99]. Os trabalhos [3, 73, 81] propõem a geração automática de uma taxonomia para descrever a estrutura dos objetos.

2.4 Programação Genética

Programação Genética (PG) é uma técnica de Inteligência Artificial que busca achar soluções aproximadas de problemas baseando-se nos principios de herança biológica, seleção natural e evolução. Nesse contexto, cada solução em potencial é chamada de indivíduo em uma população. Sobre essa população aplicam-se transformações genéticas, como *crossover* e mutações, com o intuito de criar indivíduos mais aptos (melhores soluções) em gerações subsequentes. Uma função de adequação (*fitness*) é utilizada para avaliar a qualidade do indivíduo (solução). Sendo assim, esta função é utilizada para definir o grau de evolução do indivíduo. A Tabela 2.1 mostra os principais componentes de um arcabouço de PG.

Em PG, utilizam-se estruturas de dados como árvores (ver Figura 2.3), listas encadeadas ou pilhas [49]. Além disso, o tamanho das estruturas de dados em PG não é fixo, embora seja possível restringir certos limites na implementação. Em virtude do paralelismo intrínseco no mecanismo de busca e poderosa capacidade de exploração global em espaços de dimensões mais elevadas, PG é utilizada para resolver uma ampla gama de problemas de otimização em que normalmente a melhor solução não é conhecida [13, 27].

O Algoritmo 1 (retirado de [26]) mostra os passos da evolução de indivíduos da PG. Dada uma população inicial de indivíduos, para cada uma das N gerações, calcula-se a aptidão de cada indivíduo, selecionam-se os indivíduos para sofrerem operações genéticas (*crossover*, mutação e reprodução) e ao final da N-ésima geração retornam-se os melhores indivíduos, ou seja, as molhores soluções para o problema-alvo.

Em [17], propõe-se o uso de PG como a função $\delta_{\mathcal{D}}$ de um descritor composto $\hat{D}$ (ver Seção 2.1) que combina distâncias de descritores simples. Assim, a função $\delta_{\mathcal{D}}$ será um indivíduo, representado como uma expressão matemática no formato de árvore (como na Figura 2.3), na qual os nós internos são operadores matemáticos e as folhas são distâncias de descritores simples definidas por diferentes descritores. Os índivíduos (funções de combinação de distâncias) são considerados bons se estes enquadram, para uma dada imagem

Tabela 2.1: Componentes essenciais de Programação Genética.

Componentes	Significado
Terminais	Nós folhas na estrutura da árvore.
Funções	Nós não-folhas utilizados para combinar os nós folha. Operações numéricas comuns: +, -, *, /, log, sqrt (raíz).
Função de Adequação	A função que PG busca otimizar.
Reprodução	Um operador genético que copia os indivíduos com os melhores valores de adequação diretamente para a próxima geração, sem passar pela operação de *crossover*.
Crossover	Um operador genético que troca sub-árvores de dois pais para formar dois novos descendentes. A Figura 2.4 ilustra este operador.
Mutação	Um operador genético que troca uma sub-árvore de um determinado indivíduo, cuja raiz é um ponto de mutação escolhido, com uma sub-árvore gerada aleatoriamente. A Figura 2.5 ilustra este operador.

Algoritmo 1 Algoritmo de evolução de indivíduos da Programação Genética.

1: Gere a população inicial de indivíduos
2: Para N gerações faça
3: Calcule a adequação de cada indivíduo
4: Selecione os indivíduos para operações genéticas
5: *crossover*, mutação e reprodução
6: Fim Para
7: Retorna os melhores indivíduos

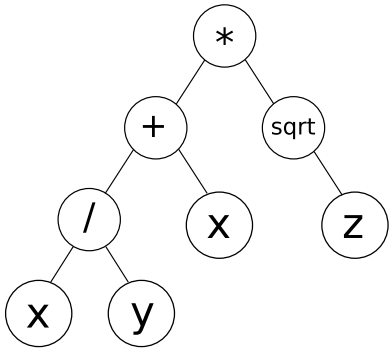


Figura 2.3: Representação de um indivíduo PG. Retirado de [70].

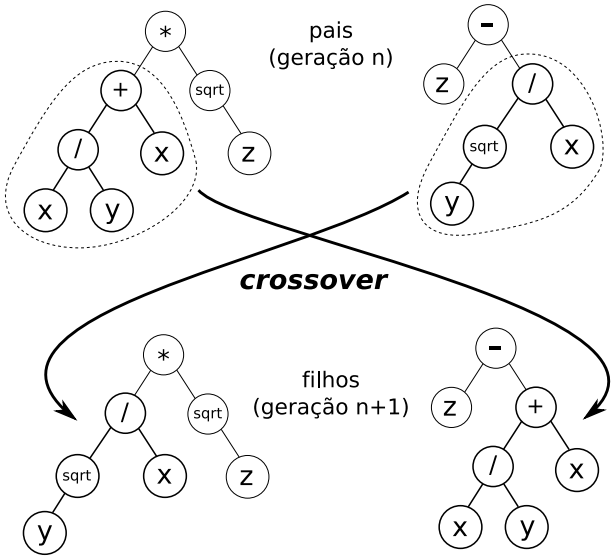


Figura 2.4: Exemplo da operação de *crossover* entre indivíduos. Retirado de [70].

de consulta, imagens relevantes (similares) nas primeiras posições da lista ordenada (*ranked list*) gerada.

Nesta dissertação, utiliza-se PG para combinar distâncias geradas por descritores globais, descritores locais e dicionários visuais (Seção 2.3).

2.5 Regras de Associação

O estudo de regras de associação é uma área de mineração de dados que busca encontrar padrões que descrevam dependências significativas entre conjuntos de eventos um uma base de dados [1], isto é, permite associar itens que ocorrem frequentemente juntos. Seu

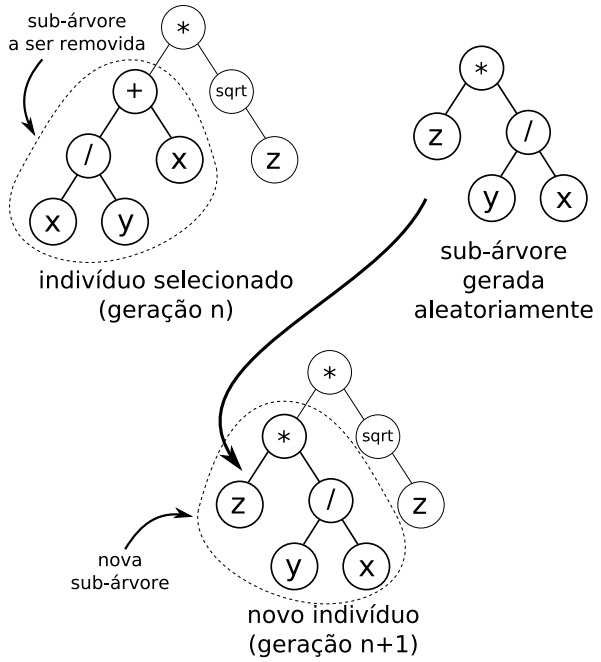


Figura 2.5: Exemplo da operação de mutação entre indivíduos. Retirado de [70].

uso é muito importante para diversas aplicações, como estratégia de negócios, processo de tomada de decisão, determinação de efeitos de alteração climática, entre outras.

Para descrever formalmente a regra de associação, precisamos de algumas definições [1]. Seja $I = \{i_1, ...i_n\}$ um conjunto de m itens, X é um conjunto que pertence a I e é chamado de *itemset* e um *itemset* X com k elementos é chamado de k-itemset. Uma transação T é um conjunto de itens, ou seja , $T \subseteq I$, e é identificada por *tid*. Uma regra de associação é uma implicação da forma $X \rightarrow Y$ (X é o antecedente e Y é o consequente da regra), onde $X \subseteq I$, $Y \subseteq I$ e $X \cap Y = \varnothing$.

As regras de associação possuem medidas de qualidade, sendo que as mais conhecidas são suporte $\sigma(X \rightarrow Y)$ e confiança $\theta(X \rightarrow Y)$. A regra de associação $X \rightarrow Y$ possui suporte s em D, se $s\%$ das transações em D contêm $X \cup Y$. Esta regra possui confiança c no conjunto de transações D, se $c\%$ da transações em D que contêm X também contêm Y.

Sendo assim, dados valores mínimos para para o suporte (σ_{min}) e para a confiança (θ_{min}) especificados pelo usuário, uma regra de associação é denominada *forte* se ela possui suporte e confiança maiores ou iguais aos respectivos valores mínimos definidos. Analogamente, uma regra de associação é *fraca* se ela não atende a esses requisitos.

O problema da mineração de regras de associação (MRA) consiste em encontrar todas

as regras de associação *fortes*. Para obter todas essas regras, os suportes de todos os *itemsets* frequentes da base de dados precisam ser computados. Desse modo, grande parte dos algoritmos para essa tarefa de mineração de dados dividem-se em duas etapas principais [2]:

1. Encontrar $L = \{X \subseteq I \mid X \text{ é frequente}\}$, onde L é o conjunto de todos os *itemsets* frequentes, juntamente com seus respectivos valores de suporte.

2. Para todos os *itemsets* frequentes $X \in L$, calcular a confiança de todas as regras $Y \rightarrow X - Y$, onde $Y \subset X$ e $Y \neq \varnothing$, e eliminar todas aquelas que não satisfazem a confiança mínima.

Determinar a frequência de todos os *itemsets* de uma grande base de dados, na prática, pode ser inviável devido ao elevado custo computacional de busca dos itens. Desse modo, para diminuir a quantidade de itens e consequentemente o processamento, a maioria dos algoritmos criam os chamados conjuntos candidatos, que são subconjuntos de itens da base que provavelmente são frequentes. Por exemplo, alguns algoritmos geram os conjuntos candidatos a serem computados em uma determinada iteração a partir apenas dos *itemsets* considerados frequentes na iteração anterior. A estratégia de escolha dos itens candidatos, da sua quantidade e da frequência com que são criados dependem particularmente da solução de cada algoritmo.

Dentre os algoritmos mais conhecidos para minerar regras de associação, podemos destacar o *Apriori*, o *AprioriTID* e o *AprioriHybrid* apresentados em [2]. O *AprioriTID* possui um desempenho de eficiência superior ao *Apriori* quando os conjuntos de itens candidatos cabem na memória, e inferior caso contrário. O *AprioriHybrid* é uma junção dos dois algoritmos, cuja estratégia é utilizar o *AprioriTID* quando os candidatos cabem na memória, e usar o *Apriori* caso não caibam.

O trabalho [65] exemplifica a tarefa de associação, apresentando uma tabela (Tabela 2.2) que considera dados de compra de produtos em um supermercado. A partir destes dados é possível obter a seguinte regra de associação: $fralda \rightarrow cerveja$. Esta conclusão foi obtida observando que os itens fralda e cerveja aparecem juntos em 60% das transações da base de dados, e de todas as transações que possuem fraldas, 75% também contêm cerveja. Neste caso, 60% e 75% representam, respectivamente, os valores de suporte e de confiança dessa regra.

Regras de Associação para Classificação

Um dos problemas de Aprendizagem de Máquina é a tarefa de classificação. Grande parte dos métodos de classificação realiza um treinamento utilizando conjuntos de exemplos, que são pares de entrada e saída de um determinado problema. Posteriormente, cria-se uma

Tabela 2.2: Exemplo de uma transação de um supermercado [65].

Transação	Itens
1	cerveja, fralda, leite
2	arroz, cerveja, fralda
3	cerveja, fralda
4	fralda, manteiga, pão
5	manteiga, farinha

função de mapeamento entre essas entradas e saídas. Contudo, quando o conjunto de exemplos é pequeno demais para um classificador conseguir mapear corretamente, diz-se que ocorreu um *underfitting*. Analogamente, se o conjunto é tão grande que possa atrapalhá-lo, verifica-se um *overfitting*.

Em [86], formaliza-se o problema de classificação. O conjunto de entradas e saídas é definido como um par $z_i = (x_i, y_i)$, na qual x_i representa um registro de tamanho fixo de atributos da forma $< a_1, ..., a_l >$ e y_i assume um valor de uma classe contida no conjunto $y = (c_1, ..., c_p)$. Existe uma probabilidade de relação entre x e y denominada $P(x|y)$ e a função de mapeamento f é utilizada para tentar aproximá-la. Um conjunto de treinamento (S) são pares de entrada $z = (x_i, y_i)$, onde x_i e y_i são conhecidos. Já o conjunto de teste (T) são pares $z = (x_i, y_i)$ onde $y_i =?$, isto é, y_i é desconhecido. A Tabela 2.3 exemplifica um conjunto S de treinamentos e uma instância de teste no conjunto T.

Tabela 2.3: Conjunto de treinamento e uma instância de teste.

		Entrada (x_i)				Saída (y_i)
		a_1	a_2	a_3	a_4	
S	(x_1, y_1)	1	1	1	2	1
	(x_2, y_2)	2	2	0	1	1
	(x_3, y_3)	3	0	0	1	0
	(x_4, y_4)	1	3	1	1	0
	(x_5, y_5)	3	1	1	2	1
T	(x_6, y_6)	3	2	4	1	?

As regras de associação foram utilizadas para classificação primeiramente em [55], e foram chamadas de Regras de Associação para Classificação (*Classification Association Rules* — CARs). Neste contexto, as CARs caracterizam-se por serem regras de associação do tipo $X \rightarrow c_j$, onde X são características e c_j é uma classe.

Os valores de suporte σ e de confiança θ de uma CAR são definidos segundo as equações 2.4 e 2.5, respectivamente.

$$\sigma(X \rightarrow c_j) = \frac{|(x_i, y_i)| \in S \; tal \; que \; X \subseteq x_i \; e \; c_i = y_i}{|S|} \tag{2.4}$$

$$\theta(X \rightarrow c_j) = \frac{|(x_i, y_i)| \in S \; tal \; que \; X \subseteq x_i \; e \; c_i = y_i}{|(x_i, y_i)| \in S \; tal \; que \; X \subseteq x_i} \tag{2.5}$$

Para gerar regras de classificação, nesta dissertação utilizam-se o algoritmo *apriori* (Seção 2.5.1) e classificador associativo chamado LAC [87], mais especificamente o algoritmo LAC-MR (Algoritmo 2), que está descrito na Seção 2.5.2.

2.5.1 Algoritmo *Apriori*

O algoritmo *apriori* [2] é um dos mais conhecidos para minerar regras de associação devido a sua simplicidade. Este algoritmo inicia-se percorrendo a base de dados para determinar quais *itemsets* de tamanho um são frequentes. Em cada iteração subsequente, geram-se as combinações dos *itemsets* frequentes, criando itens que são possivelmente frequentes. A cada nova iteração, geram-se *itemsets* de tamanho no máximo igual ao tamanho da iteração anterior mais um. Este processo finaliza-se quando não existir novos *itemsets* frequentes em uma determinada iteração.

O *apriori* utiliza duas funções principais segundo [67]: o *Apriori-gen* que encontra os itens candidatos e elimina os itens não frequentes; e o *Genrules* que gera as regras de associação.

Para gerar os itens candidatos em uma iteração, a função *Apriori-gen* utiliza o conjunto de todos os itens frequentes encontrados na iteração anterior e realiza combinações entre eles gerando novos itens que possivelmente são frequentes, descartando os que não possuem suporte mínimo. A intuição por trás desse procedimento é que se um *itemset* X tem suporte mínimo, todos os seus subconjuntos também terão.

Já a função *Genrules* gera as regras de associação utilizando itens frequentes l encontrados pela função *Apriori-gen*. Segundo [67], a geração de regras, para qualquer itemset frequente, implica encontrar todos os *subsets* não vazios de l. Assim, para todo e qualquer *subset* a, produz-se uma regra $a \rightarrow (l - a)$ somente se a razão (suporte (l)/suporte(a)) é ao menos igual a confiança mínima estabelecida pelo usuário. Para gerar regras com múltiplos consequentes, são considerados todos os *subsets*. Por exemplo (retirado de [67]), dado um *itemset* $ABCD$, considera-se o primeiro *subset* ABC, seguido de AB, etc. Se $ABC \rightarrow D$ não atinge uma confiança suficiente (confiança <minconf), não é necessário verificar se $AB \rightarrow CD$.

2.5.2 *Lazy Association Classifier* (LAC)

Classificadores associativos tradicionais realizam uma pesquisa global na base para gerar regras que satisfazem algumas restrições de qualidade, como o suporte e a confiança. No entanto, esta pesquisa global pode gerar um conjunto muito grande de regras, além da possibilidade de nunca extrair regras raras, que são regras em que certas características aparecem poucas vezes no conjunto de treinamento, mas que podem ser importantes [86]. Desse modo, em [87] é proposto um classificador associativo chamado *Lazy Association Classifier* (LAC), que visa superar esses problemas, focando-se somente nas características de ocorrência de uma determinada instância de teste, aumentando a chance de gerar mais regras que serão úteis para sua classificação.

Por isso, um conceito chave para entender o LAC são as projeções. Especificamente, segundo [86], dada uma entrada $x_i \in T$, o conjunto S de treinamento é projetado para S^{x_i} ($S^{x_i} \subseteq S$) de tal forma que só possam ser obtidas regras da forma $X \rightarrow c_j$, na qual $X \subseteq x_i$. As regras de associação obtidas por meio do conjunto S^{x_i} são chamadas de R^{x_i}. Da mesma forma que o conjunto de treinamento, o suporte mínimo também é projetado para $\pi_{min}^{x_i}$ de acordo com o número de exemplos em S^{x_i}, definido na Equação 2.6. A confiança mínima não necessita ser projetada, pois o valor da confiança de uma regra específica será o mesmo, tanto para S quanto para S^{x_i}. A Tabela 2.4 ilustra o conjunto de treinamento da Tabela 2.3 projetado de acordo com a instância de teste (x_6,y_6).

$$\pi_{min}^{x_i} = \lceil \sigma_{min} \times |S^{x_i}| \rceil \tag{2.6}$$

Tabela 2.4: Conjunto de treinamento projetado: S^{x_6}.

		Entrada (x_i)				Saída (y_i)
		a_1	a_2	a_3	a_4	
S^{x_6}	(x_1, y_1)	–	–	–	–	1
	(x_2, y_2)	–	2	–	1	1
	(x_3, y_3)	3	–	–	1	0
	(x_4, y_4)	–	–	–	1	0
	(x_5, y_5)	3	–	–	–	1
T	(x_6,y_6)	3	2	4	1	?

Considerando-se um suporte mínimo (σ_{min}) igual a 0,25 e uma confiança mínima (θ_{min}) igual a 0,6, as regras R^{x_6} geradas para a entrada x_6 por meio do conjunto projetado S^{x_6} (Tabela 2.4) são:

1. $\{a_2$=[2] $\rightarrow$ saída=1$\}$, com θ(1,00)

2. $\{a_2$=[2] $\wedge$ a_4=[1] $\rightarrow$ saída=1$\}$, com θ(1,00)
3. $\{a_1$=[3] $\wedge$ a_4=[1] $\rightarrow$ saída=0$\}$, com θ(1,00)
4. $\{a_4$=[1] $\rightarrow$ saída=0$\}$, com θ(0,66)

Desse modo, o LAC é um classificador sob demanda, pois para cada instância de teste específica é projetado um subconjunto de treinamento, que será utilizado para gerar as CARs. Contudo, mesmo diminuindo consideravelmente o tamanho do conjunto de treinamento, ainda é grande o custo computacional para acessar a base, visto que as entradas de teste podem projetar diferentes subconjuntos de treinamento, e consequentemente gerar diferentes conjuntos de CARs. No entanto, algumas dessas CARs podem ser comuns. Por isso, no LAC implementa-se um mecanismo de memorização (*caching*) que reduz o trabalho replicado e, portanto, o acesso a S. O *cache* em questão é um conjunto de entradas armazenadas na memória principal, e cada entrada tem a forma $< key, data >$, onde $key = \{X, c_j\}$ e $data = \{\sigma(X \rightarrow c_j), \theta(X \rightarrow c_j)\}$.

A seguir, são mostrados os principais algoritmos implementados a partir do LAC de acordo com [86].

LAC-SR (acrônimo derivado de *"lazy association classification using a single rule"*)

Dada uma instância de teste $x_i \in T$, este algoritmo projeta S para S^{x_i}, extrai as regras de associação R^{x_i}, e posteriormente retorna a classe prevista pela regra que possua o maior valor de confiança, de acordo com a Equação 2.7.

$$f_S^{x_i}(x_i) = c_j,\ tal\ que\ \theta(X \rightarrow c_j)\ seja\ argmax(\theta(r))\ \forall\ r \in R^{x_i} \tag{2.7}$$

LAC-MR (acrônimo derivado de *"lazy association classification using multiple rules"*)

Dada uma instância de teste $x_i \in T$, S^{x_i} é projetado e as regras de associação R^{x_i} são extraídas. Em seguida, todas as regras geradas são utilizadas para prever a classe de x_i, na qual cada regra $X \rightarrow c_j$ é interpretada como um voto dado por X à classe c_j, levando-se em conta o seu peso. Este peso é dado por $\theta(X \rightarrow c_j)$, ou seja, quanto maior a confiança, maior o peso. A pontuação (*score*) de cada classe c_j, referente a entrada x_i, é definida pela Equação 2.8 e a probabilidade da classe c_j ser correta referente a entrada x_i é definida segundo a Equação 2.9.

$$s(x_i, c_j) = \frac{\sum_{r \in R_{c_j}^{x_i}} \theta(r)}{|R_{c_j}^{x_i}|} \tag{2.8}$$

$$p(c_j|x_i) = \frac{s(x_i, c_j)}{\sum_{k=1}^{p} s(x_i, c_j)} \tag{2.9}$$

Esta implementação do LAC (Algoritmo 2) possui a melhor relação entre o custo computacional e a eficácia, segundo [86]. Por isso, os experimentos realizados neste trabalho utilizaram este algoritmo.

Algoritmo 2 LAC-MR

Require: Conjunto de treinamento S, entrada $x_i \in T$, σ_{min}
1: $S^{x_i} \Leftarrow S$ projetado de acordo com x_i
2: $R^{x_i} \Leftarrow$ regras $(X \rightarrow c_j)$ extraídas de S^{x_i}, onde $\pi(X \rightarrow c_j) \geq \pi_{min}^{x_i}$
3: Retorna $f_S^{x_i}$ tal que $f_S^{x_i} = c_j$, onde $p(c_j|x_i)$ seja $argmax(p(c_k, x_i)) \ \ \forall\ 1 \leq k \leq p$

LAC-MR-ERM (acrônimo derivado de *"empirical risk minimization"*) e LAC-MR-SRM (acrônimo derivado de *"structural risk minimization"*)

Estes algoritmos baseados no LAC utilizam inequações para encontrar uma função de mapeamento que forneça um erro empírico baixo. Veja o trabalho [86] para maiores informações sobre estes algoritmos e as inequações usadas.

Capítulo 3

Métodos para Anotação Automática de Imagens

Neste capítulo são apresentados 4 métodos para anotação automática de imagens. São eles: o *Lazy Association Classification for Image Annotation* (LIA); o *Relevance-Based Association Rules for Image Annotation* (RAIA); o *Genetic Programming-Based Relevance for Image Annotation Using Association Rules* (GRIA); e o *Genetic Programming-Based Relevance for Image Annotation Using Lazy Association Classification* (GRIA-LAC). Por fim, a Seção 3.5 apresenta uma comparação entre os métodos.

3.1 Método LIA

Esta seção descreve um método para anotação automática de imagens baseado em características de imagens e regras de associação sob demanda. Propõe-se nesta dissertação a utilização do classificador LAC (Seção 2.5.2), formando o método *Lazy Association Classification for Image Annotation* (LIA).

O LIA é um método para anotação automática de imagens baseado no Algoritmo 2 do LAC. Neste método, as características das imagens de treinamento são utilizadas para gerar regras de associação. Estas regras são geradas sob demanda, ou seja, para cada imagem de teste, o LAC gera regras que serão utilizadas para sua classificação. A Figura 3.1 ilustra, de modo geral, este processo, na qual o LAC recebe como parâmetros de entrada as características de um conjunto de imagens de treinamento e as características de uma imagem de teste. Ao final, calcula-se a probabilidade da imagem de teste pertencer a cada uma das classes, que no caso são categorias.

Para representar as imagens, neste método utilizam-se dois tipos de características: vetores de características globais e histogramas de palavras visuais. Levando-se em consideração que uma CAR é escrita na forma $X \rightarrow c_j$, onde X é uma característica e c_j é uma

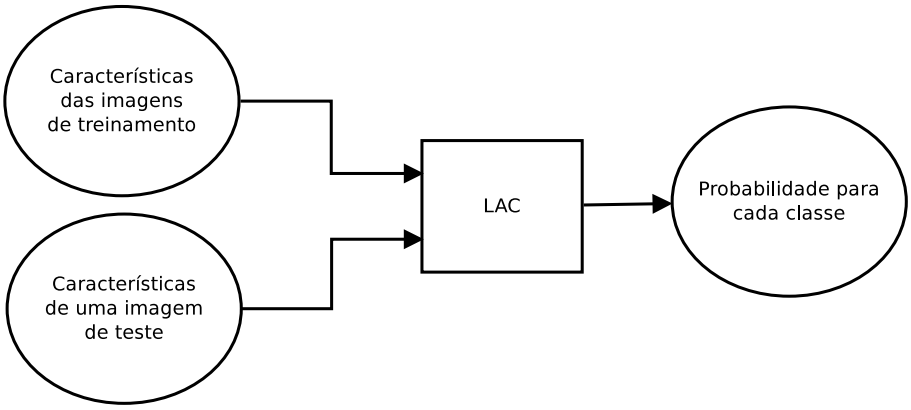


Figura 3.1: Processo de classificação de uma imagem no LAC.

categoria, o antecedente X das regras que utilizam vetores de características globais são os *bins* dos vetores. Já nas que utilizam histogramas de palavras visuais, são os *bins* dos histogramas (palavras visuais). Para serem utilizadas, algumas dessas características podem passar por um pré-processamento, como por exemplo, concatenação de características e discretização (Seção 4.2.4), entre outros.

A Figura 3.2 ilustra o fluxograma do método LIA, na qual as etapas 1 e 2 dizem respeito às extrações de características e pré-processamentos das imagens de treinamento. Paralelamente a estas etapas, são extraídas e pré-processadas as características da imagem de teste (etapas 3 e 4). Em seguida, o conjunto de características de treinamento é projetado (etapa 5) usando as características da imagem de teste, gerando um subconjunto de características de treino que só possuem atributos contidos nas características da imagem de teste. Utilizando este conjunto projetado, geram-se as regras de associação (etapa 6), nas quais características implicam classes. Logo após, aplicam-se as características do teste nas regras geradas (etapa 7), calculando-se uma pontuação para cada uma das classes existentes (etapa 8). Por fim, na etapa 9, retornam-se as probabilidades da imagem de teste ser de cada uma das classes.

O Algoritmo 3 (adaptado do Algoritmo 2) apresenta os passos principais do LIA. Na primeira linha, calculam-se as características das imagens do conjunto de treinamento T e realiza-se um pré-processamento, caso seja necessário. Na segunda linha, calculam-se as características da imagem de teste, e realiza-se o mesmo pré-processamento do treinamento. Na terceira linha, o conjunto de características S do conjunto de treino é projetado para S^x de acordo com as características do teste, para que S^x contenha somente características de x. Na linha 4, o suporte mínimo é projetado de acordo com o tamanho de S^x. Na linha 5, extraem-se as regras de associação utilizando o conjunto S^x, na qual cada regra gerada deve ter um suporte maior ou igual ao suporte mínimo projetado para S^x ($\pi_{min}^{x_i}$) e ter uma confiança maior ou igual à confiança mínima (θ_{min}).

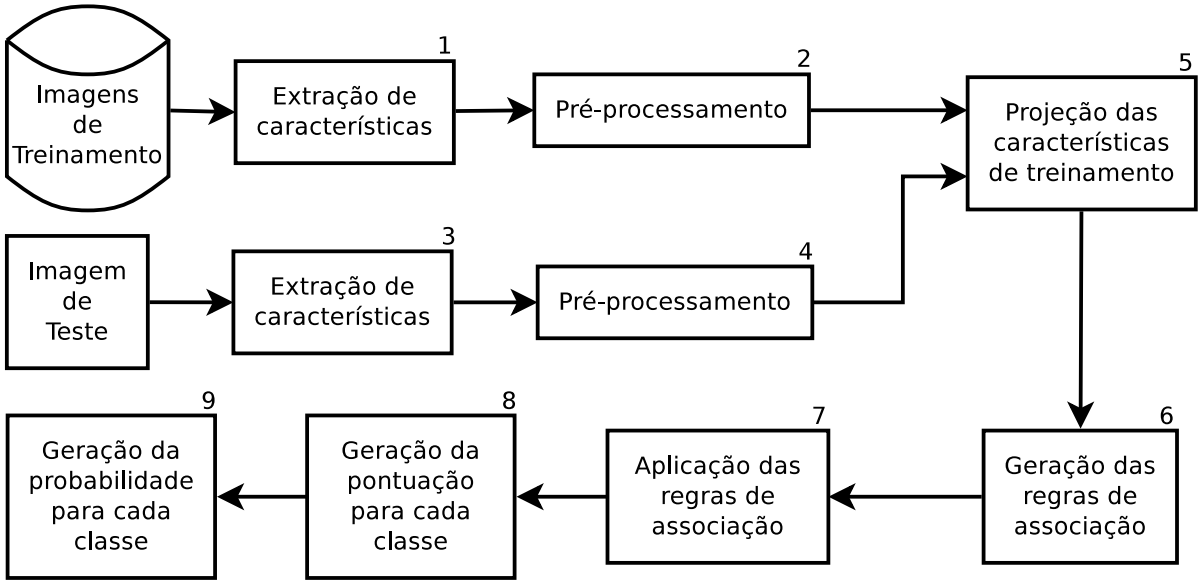


Figura 3.2: Método LIA.

Na linha 6, a pontuação de cada uma das k classes é calculada utilizando-se as regras em R^x e a Equação 2.8. Na linha 7, a probabilidade de todas as k classes é calculada usando-se a pontuação de cada uma delas a partir da Equação 2.9. Na última linha, retornam-se as probabilidades associadas a todas as classes.

Algoritmo 3 Lazy Association Classification for Image Annotation (LIA)

Require: Conjunto de imagens de treinamento T, imagem de teste x, suporte mínimo σ_{min}, confiança mínima θ_{min}

1: $S \Leftarrow$ Conjunto de características pré-processadas das imagens de T
2: $x \Leftarrow$ Características pré-processadas da imagem x
3: $S^x \Leftarrow S$ projetado de acordo com x
4: $\pi_{min}^{x_i} \Leftarrow \lceil \sigma_{min} \times |S^x| \rceil$
5: $R^x \Leftarrow$ Regras extraídas de S^x, na qual $\pi(X \rightarrow c_j) \geq \pi_{min}^{x_i}$ e $\theta(X \rightarrow c_j) \geq \theta_{min}$
6: Utilizando R^x, calcula-se a pontuação $s(x, c_j)\ \forall\ 1 \leq j \leq k$ (Equação 2.8)
7: Utilizando $s(x, c_j)$, calcula-se a probabilidade $p(c_j|x)\ \forall\ 1 \leq j \leq k$ (Equação 2.9)
8: Retorna a probabilidade $p(c_j, x)\ \forall\ 1 \leq j \leq k$

3.2 Método RAIA

Esta seção descreve o método *Relevance-Based Association Rules for Image Annotation* (RAIA). Este método é fundamentado em uma técnica [88] para ranqueamento de docu-

mentos utilizando regras de associação. O trabalho [27] estende este método para ranqueamento de imagens em sistemas de recuperação de imagens por conteúdo. O RAIA, por sua vez, é um método para anotação automática de imagens baseado em similaridade de imagens e regras de associação. Este método é dividido em duas etapas principais, uma de treinamento e outra de teste, descritas a seguir:

Etapa de treinamento

O primeiro passo do RAIA é calcular, para cada imagem do conjunto T de treinamento, d características. Estas características podem ser vetores de características globais , vetores de características locais e/ou histogramas de palavras visuais. Para cada uma das d_u características específicas, gera-se uma matriz D^{d_u} de distâncias. Dessa forma, cada elemento $D^{d_u}_{ij}$ desta matriz é a distância das características d_u da imagem i para as características d_u da imagem j, sendo que a diagonal principal são as distâncias das imagens para elas mesmas. Estas distâncias podem sofrer algum tipo de pré-processamento, como por exemplo, normalização (Seção 4.2.5) e discretização (Seção 4.2.4).

Em seguida, para cada classe c_j de imagens, gera-se um registro G^{c_j} contendo as distâncias das imagens da classe atual para todas as imagens de treinamento (incluindo imagens da própria classe) e as relevâncias $r_i \in \{0, 1\}$. Assim, se uma classe possui $|c_j|$ imagens e o total de imagens de treinamento é $|T|$, este registro terá $|c_j| \times |T|$ transações. Cada transação t possuirá d distâncias (itens) de uma imagem da classe atual para uma outra imagem de treinamento, sendo que a primeira distância refere-se a primeira característica (d_1), a segunda distância equivale a segunda característica (d_2), e assim por diante. Finalmente, marcam-se as transações com relevância 1 (r_1) se estas imagens são da mesma classe, e com relevância 0 (r_0), caso contrário.

A Tabela 3.1 mostra um exemplo do registro G^{c_j}, no qual cada linha é uma transação que contém distâncias, calculadas por descritores ou histrogramas de palavras visuais, de uma imagem de treinamento pertencente a classe c_j para uma imagem de treinamento pertencente a base de imagens. As transações marcadas com relevância 1 são distâncias entre imagens da mesma classe, ou seja, da classe c_j, enquanto as marcadas com relevância 0 são de classes diferentes.

Posteriormente, aplica-se o algoritmo *Apriori* (Seção 2.5) em cada um destes registros, gerando regras de associação para cada classe c_j (R^{c_j}). Estas regras são escritas da forma $X \rightarrow r_i$, na qual X são distâncias e r_i é relevância 0 ou 1. Levando-se em consideração que as distâncias entre imagens da mesma classe são possivelmente pequenas, então espera-se que as regras geradas sejam distâncias pequenas implicando relevância 1 e distâncias grandes implicando relevância 0.

O Algoritmo 4 apresenta os passos de treinamento do RAIA. Na linha 1, as d características do conjunto de treinamento T são armazenadas em S. Na linha 2, S^D recebe as

Tabela 3.1: Exemplo do registro G^{c_j} do RAIA.

Transação/Descritor	BIC	LAS	GCH	Relevância
1	222	0,34	553	1
2	187	0,88	678	1
3	122	0,22	234	0
4	344	0,55	690	0
5	28	0,11	455	0

matrizes D^{d_u} para cada característica d_u. Na linha 3, S_p^D recebe as distâncias das matrizes em S^D com algum pré-processamento, como por exemplo, normalização e discretização. Da linha 4 até a linha 17, executam-se os passos para cada classe c_j existente na base. Da linha 5 até a linha 15, executam-se os passos para cada imagem m pertencente a classe atual (c_j). Da linha 6 à 14, executam-se os passos para cada imagem n do conjunto de treinamento T. Na linha 7, a transação t recebe as distância da imagem m para a imagem n que estão em S_p^D (matrizes pré-processadas). Na linha 9, a transação t é marcada com relevância 1, caso a imagem n pertença a classe c_j. Na linha 11, a transação t é marcada com relevância 0, caso a imagem n não pertença a classe c_j. Na linha 13, insere-se a transação t no registro de transações da classe c_j (G^{c_j}). Por fim, na linha 16 aplica-se o algoritmo *Apriori* no registro G^{c_j}, gerando regras de associação da classe c_j. A Figura 3.3 ilustra o fluxograma do Algoritmo 4.

Etapa de teste

Para uma imagem de teste x específica, calculam-se d características e as distâncias destas características para as respectivas características de todas as imagens de treinamento, realizando o mesmo pré-processamento do treino. Posteriormente, utilizam-se, para cada classe c_j, as distâncias da imagem de teste para cada imagem pertencente a c_j, gerando $|c_j|$ transações (1 imagem de teste $\times$ $|c_j|$ imagens), onde não se sabem suas relevâncias. Assim, tem-se um representação em forma de matriz M^{c_j} de tamanho $|c_j| \times d$, na qual as linhas são transações e as colunas são distâncias. Cada elemento $M_{ij}^{c_j}$ desta matriz é a distância da característica d_j da imagem de teste para a característica d_j da imagem i da classe c_j.

Posteriormente, aplica-se nesta matriz de transações uma técnica de agrupamento denominada *polling*, na qual usa-se uma função que gera um único valor para cada coluna. Por exemplo, pode-se calcular a média dos valores de uma coluna ou ainda utilizar o seu valor máximo. No RAIA, usa-se o menor valor de distância por coluna, pois quanto menor o valor da distância, mais próximo a imagem de teste está de uma imagem de treinamento.

Algoritmo 4 Etapa de treinamento do método RAIA

Require: Conjunto de imagens de treinamento T, suporte mínimo σ_{min}, confiança mínima θ_{min}

1: $S \Leftarrow$ Conjunto das d características de T
2: $S^D \Leftarrow$ Para cada características d_u, calcula-se a matriz D^{d_u} de distância
3: $S_p^D \Leftarrow$ Pré-processamento das distâncias $\in S^D$
4: Para cada classe c_j faça
5: Para cada imagem $m \in c_j$ faça
6: Para cada imagem $n \in T$ faça
7: $t \Leftarrow d$ distâncias $\in S_p^D$ de m para n
8: Se $n \in c_j$ então
9: marca t com relevância 1
10: Senão
11: marca t com relevância 0
12: Fim Se
13: insere t em G^{c_j}
14: Fim Para
15: Fim Para
16: $R^{c_j} \Leftarrow$ Aplique o algoritmo *apriori* para obter regras de associação da classe c_j, geradas a partir de G^{c_j}, na qual $\theta(X \rightarrow c_j) \geq \theta_{min}$ e $\sigma(X \rightarrow c_j) \geq \sigma_{min}$
17: Fim Para

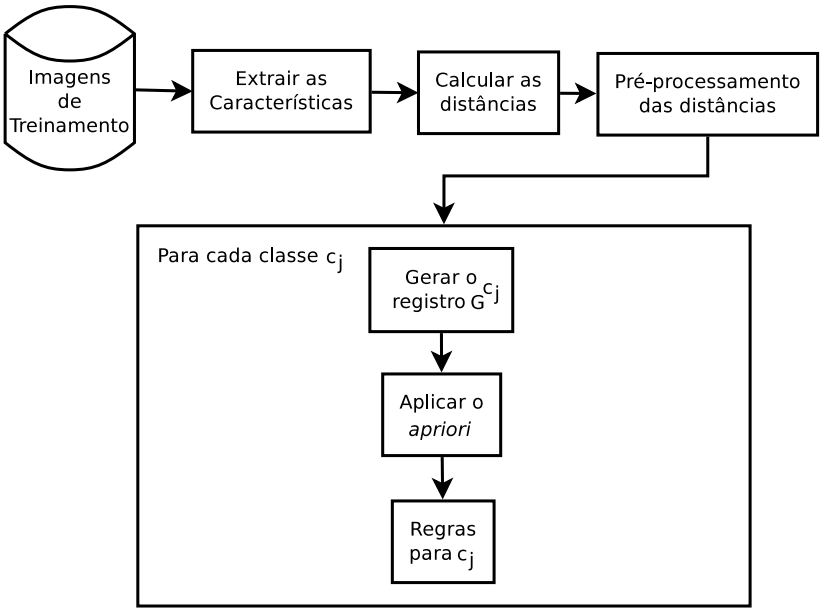


Figura 3.3: Fluxograma de treinamento do RAIA.

Ao final, $M_{ij}^{c_j}$ é transformado em uma única transação ($I_x^{c_j}$) que representa as menores distâncias da imagem x para as imagens em c_j.

A Tabela 3.2 ilustra a matriz $M_{ij}^{c_j}$ na qual cada linha é uma transação que contém distâncias da imagem de teste para uma imagem de treinamento da classe c_j. A transação $I_x^{c_j}$ recebe as menores distância de $M_{ij}^{c_j}$.

Tabela 3.2: Exemplo da matriz $M_{ij}^{c_j}$ e da transação $I_x^{c_j}$ do RAIA.

	Transação/Descritor	BIC	LAS	GCH	Relevância
$M_{ij}^{c_j}$	1	211	0,23	854	?
	2	342	0,55	642	?
	3	233	0,32	306	?
	4	544	0,75	679	?
	5	433	0,05	683	?
$I_x^{c_j}$	MIN	211	0,05	306	?

A seguir, aplicam-se as regras de associação da classe atual R^{c_j}, geradas no treinamento, na transação $I_x^{c_j}$. Desse modo, se a imagem de teste pertencer à classe atual, as distâncias deste teste para as imagens de treinamento desta classe possivelmente serão pequenas, consequentemente muitas regras aplicadas implicarão relevância 1. Caso não pertençam à mesma classe, possivelmente as distâncias serão grandes, implicando muitas

relevâncias 0.

Para cada classe c_j e cada relevância r_i, calcula-se uma estimativa de relevância $s_{(x,c_j,r_i)}$, combinando as regras da classe c_j que implicam r_i e que foram aplicadas em $I_x^{c_j}$. Esta combinação é feita levando-se em conta as confianças das regras aplicadas, dando-se um peso maior para regras com maior confiança. Desse modo, a estimativa $s_{(x,c_j,r_i)}$ é calculada a partir da Equação 3.1, na qual somam-se as confianças das regras em R^{c_j} que foram aplicadas em $I_x^{c_j}$ e que implicam relevância r_i ($R^{c_j}_{(I_x^{c_j},r_i)}$) dividida pelo número total destas regras aplicadas ($|R^{c_j}_{(I_x^{c_j},r_i)}|$).

$$s(x,c_j,r_i) = \frac{\sum_{X \to r_i \in R^{c_j}_{(I_x^{c_j},r_i)}} \theta(X \to r_i)}{|R^{c_j}_{(I_x^{c_j},r_i)}|} \tag{3.1}$$

Então, a relevância da imagem x para a classe c_j ($relevancia(x,c_j)$) é estimada por uma combinação linear da normalização dos *scores* associados a cada relevância r_i ($s(x,c_j,r_i)$), segundo a Equação 3.2.

$$relevancia(x,c_j) = \sum_{i \in \{0,1\}} \left(r_i \times \frac{s(x,c_j,r_i)}{\sum_{z \in \{0,1\}} s(x,c_j,r_z)} \right) \tag{3.2}$$

Por fim, normalizam-se as relevâncias, por meio da Equação 3.3, que divide a relevância da classe c_j pela soma total de relevâncias das k classes, obtendo-se a probabilidade $p_{(c_j|x)}$ da imagem x ser da classe c_j.

$$p_{(c_j|x)} = \frac{relevancia(x,c_j)}{\sum_{i=1}^{k} relevancia(x,c_i)} \tag{3.3}$$

O Algoritmo 5 apresenta os passos de teste do RAIA. Na linha 1, S^x recebe as d características da imagem de teste. Na linha 2, calcula-se a distância da imagem de teste para cada imagem de treinamento, ou seja, cada característica d_u do teste possuirá uma distância para a respectiva característica de cada imagem de treinamento. Na linha 3, realiza-se o mesmo pré-processamento que foi feito no treinamento. Da linha 4 até a linha 16, executam-se os passos para cada classe c_j. Da linha 5 até a linha 8, gera-se uma transação t para cada imagem m pertencente à classe c_j, contendo as distâncias da imagem de teste para m, e insere-se t na matriz M^{c_j}. Da linha 9 até a linha 11, aplica-se a técnica de agrupamento na matriz M^{c_j}, na qual a transação $I_x^{c_j}[c]$ receberá na primeira posição o menor valor da primeira coluna da matriz, na segunda posição receberá o menor valor da segunda coluna, e assim por diante. Da linha 12 à linha 14, utilizando a transação $I_x^{c_j}[c]$ e das regras R^{c_j} geradas no treinamento, calcula-se para cada relevância, uma estimativa de relevância a partir da Equação 3.1. Na linha 15 calcula-se, por meio da Equação 3.2, uma estimativa de relevância da imagem de teste ser da classe c_j. Na linha 17, normalizam-se

estas relevâncias, obtendo-se uma probabilidade da imagem de teste ser de cada uma das k classes. Finalmente, na linha 18 retornam-se estas probabilidades. A Figura 3.4 ilustra o fluxograma de teste do RAIA.

Algoritmo 5 Etapa de teste do método RAIA

Require: Conjunto S das características das imagens de treinamento,
imagem de teste x, conjunto de regras R^{c_j}

1: $S^x \Leftarrow$ Conjunto das d características da imagem de teste x
2: $S^{(D,x)} \Leftarrow$ Para cada características d_u, calcula-se a distância de S^x para S
3: $S_p^{(D,x)} \Leftarrow$ Pré-processamento das distâncias $\in S^{(D,x)}$
4: Para cada classe c_j faça
5: Para cada imagem m em c_j faça
6: $t \Leftarrow d$ distâncias $\in S_p^{(D,x)}$ de x para m
7: insere t em M^{c_j}
8: Fim Para
9: Para cada coluna c de M^{c_j} faça
10: $I_x^{c_j}[c] \Leftarrow min(c)$
11: Fim Para
12: Para cada $r_i \in \{0,1\}$ faça
13: Utilizando R^{c_j} e $I_x^{c_j}$, calcula-se $s(x, c_j, r_i)$ (Equação 3.1)
14: Fim Para
15: Utilizando $s(x, c_j, r_0)$ e $s(x, c_j, r_1)$, calcula-se a $relevancia(x, c_j)$ (Equação 3.2)
16: Fim Para
17: Utilizando $relevancia(x, c_j)$, calcula-se $p(c_j|x)\ \forall\ 1 \leq j \leq k$ (Equação 3.3)
18: Retorna a probabilidade $p(c_j|x)\ \forall\ 1 \leq j \leq k$

3.3 Método GRIA

Esta seção descreve o método *GP-Based Relevance for Image Annotation Using Association Rules* (GRIA). Este método é semelhante ao RAIA, porém, utiliza PG para combinar as similaridades das imagens.

Os primeiros passos do treinamento do GRIA são iguais ao treinamento RAIA, até o momento de criação das matrizes de distância (linha 2 do Algoritmo 4). Em seguida, aplica-se a normalização gaussiana (Seção 4.2.5) em cada uma das matrizes, para enquadrar suas respectivas distâncias no intervalo [0,1]. Posteriormente, executam-se os passos do Algoritmo 1 de PG, onde inicialmente gera-se uma população aleatória de indivíduos, e ao final retornam-se os indivíduos que combinam melhor as distâncias destas matrizes normalizadas. A seguir, aplicam-se os k melhores indivíduos nestas matizes, criando-se k

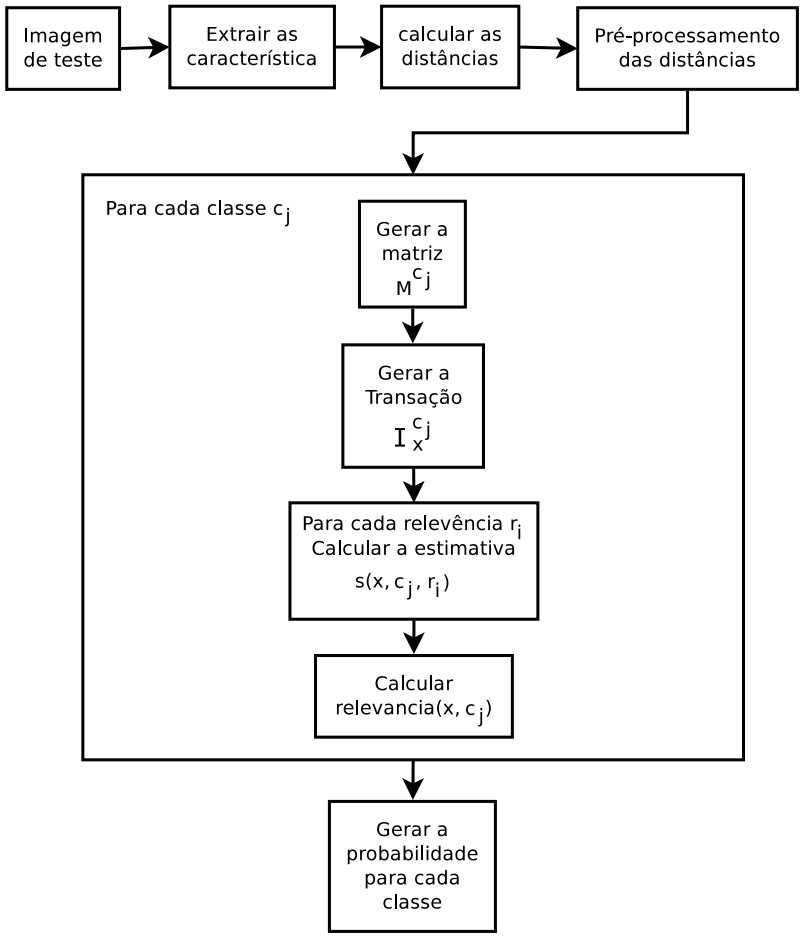


Figura 3.4: Fluxograma de teste do RAIA.

novas matrizes de distâncias. O restante dos passos do treinamento do GRIA são iguais ao RAIA (da linha 3 à linha 17 do Algoritmo 4).

A Tabela 3.3 ilustra um exemplo do registro G^{c_j} do GRIA, no qual cada linha é uma transação que contém distâncias, calculadas a partir de indivíduos GP (como por exemplo, o indivíduo da Figura 2.3), de uma imagem pertencente a classe c_j para uma imagem pertencente a base.

O Algoritmo 6 mostra os passos de treinamento do GRIA. As linhas 1 e 2 deste algoritmo são as mesmas das linha 1 e linha 2 do Algoritmo 4. Na linha 3, cada matriz de distância é normalizada no intervalo [0,1], utilizando-se a normalização gaussiana (Seção 4.2.5). Na linha 4, geram-se os melhores indivíduos usando o Algoritmo 1. Na linha 5, aplicam-se estes indivíduos nas matrizes normalizadas, gerando uma matriz por indivíduo. A linha 6 à linha 20 são os mesmo comandos da linha 3 à linha 17 do Algoritmo 4. A Figura 3.5 ilustra o fluxograma de treinamento do GRIA.

Tabela 3.3: Exemplo do registro G^{c_j} do GRIA.

Transação/Indivíduo	d_{GP1}	d_{GP2}	d_{GP3}	Relevância
1	1,48	2,70	0,58	1
2	2,36	1,36	0,50	1
3	1,62	1,77	0,63	0
4	1,45	2,63	0,59	0
5	0,32	1,82	0,32	0

Algoritmo 6 Etapa de treinamento do método GRIA

Require: Conjunto de imagens de treinamento T, suporte mínimo σ_{min}, confiança mínima θ_{min}

1: $S \Leftarrow$ Conjunto das d características de T
2: $S^D \Leftarrow$ Para cada características d_u, calcula-se a matriz D^{d_u} de distância
3: $S^{Dn} \Leftarrow$ Cada matriz $D^{d_u} \in S^D$ normalizada pela gaussiana (Seção 4.2.5)
4: Calculam-se os melhores indivíduos por meio de S^{Dn} e do Algoritmo 1
5: $S^D \Leftarrow$ Novas matrizes de distância, geradas por meio dos indivíduos aplicados em S^{Dn}
6: $S_p^D \Leftarrow$ Pré-processamento das distâncias $\in S^D$
7: Para cada classe c_j faça
8: Para cada imagem $m \in c_j$ faça
9: Para cada imagem $n \in T$ faça
10: $t \Leftarrow d$ distâncias $\in S_p^D$ de m para n
11: Se $n \in c_j$ então
12: marca t com relevância 1
13: Senão
14: marca t com relevância 0
15: Fim Se
16: insere t em G^{c_j}
17: Fim Para
18: Fim Para
19: $R^{c_j} \Leftarrow$ Regras de associação da classe c_j, geradas a partir de G^{c_j}, na qual $\theta(X \rightarrow c_j) \geq \theta_{min}$ e $\sigma(X \rightarrow c_j) \geq \sigma_{min}$
20: Fim Para

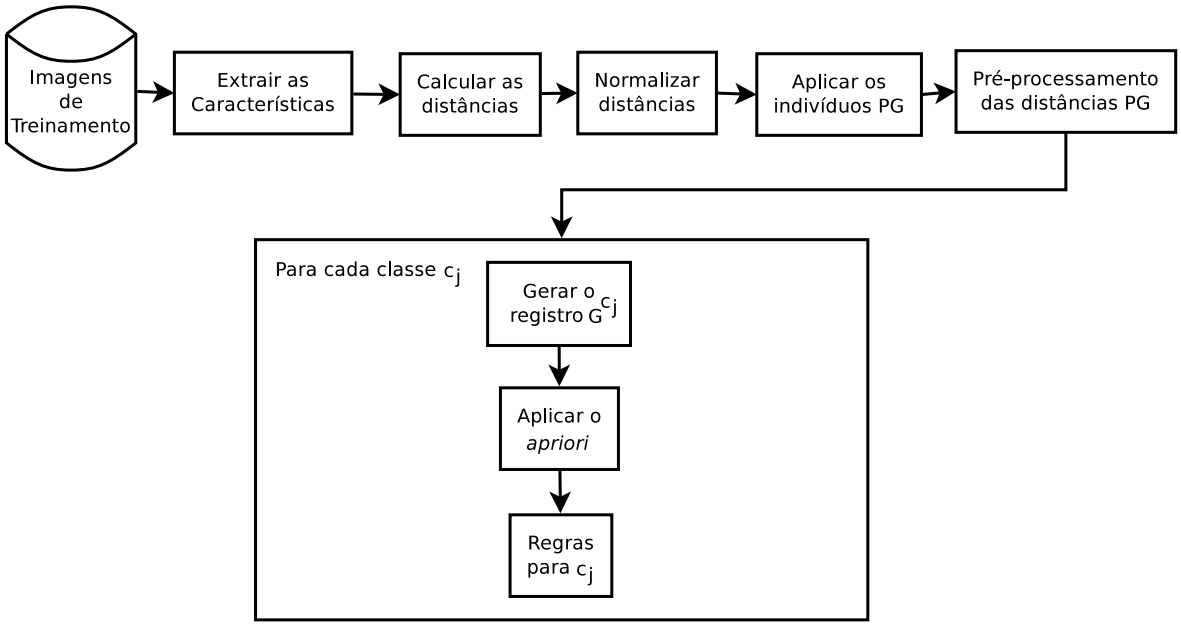


Figura 3.5: Fluxograma de treinamento do GRIA.

A etapa de teste do GRIA também é semelhante ao RAIA, como pode ser visto no Algoritmo 7. Após o cálculo das distâncias da imagem de teste para as imagens de treinamento (linha 1 e 2 do Algortimo 7), na linha 3 estas distâncias são normalizadas pela gaussiana (Seção 4.2.5). Em seguida, na linha 4 aplicam-se os melhores indivíduos nestas distâncias, gerando uma distância da imagem de teste para uma imagem de treinamento por indivíduo. O restante dos passos de teste do GRIA (linhas 5 à 20 do Algoritmo 7) são os mesmos do RAIA (linhas 3 à 18 do Algoritmo 5). A Figura 3.6 ilustra o teste do GRIA.

A Tabela 3.4 ilustra a matriz $M_{ij}^{c_j}$ na qual cada linha é uma transação que contém distâncias, calculadas por indivíduos GP, da imagem de teste para uma imagem de treinamento da classe c_j. A transação $I_x^{c_j}$ recebe as menores distância de $M_{ij}^{c_j}$.

3.4 Método GRIA-LAC

Esta seção descreve um método para anotação automática de imagens chamado de *GP-Based Relevance for Image Annotation Using Lazy Association Classification* (GRIA-LAC). O GRIA-LAC é uma versão do GRIA, sendo que a principal diferença é utilizar o classificador LAC (Seção 2.5.2).

A fase de treinamento de GRIA-LAC é igual ao GRIA até a etapa de criação dos registros de cada classe (G^{c_j}). O registro G^{c_j} do GRIA-LAC é igual ao registro G^{c_j} do

Tabela 3.4: Exemplo da matriz $M_{ij}^{c_j}$ e da transação $I_x^{c_j}$ do GRIA.

	Transação/Indivíduo	d_{GP1}	d_{GP2}	d_{GP3}	Relevância
$M_{ij}^{c_j}$	1	2,65	4,00	0,54	?
	2	4,39	2,27	0,43	?
	3	5,50	2,80	0,99	?
	4	4,44	3,20	0,48	?
	5	3,05	1,16	0,59	?
$I_x^{c_j}$	MIN	2,65	1,16	0,43	?

Algoritmo 7 Etapa de teste do método GRIA

Require: Conjunto S das características das imagens de treinamento, imagem de teste x, conjunto de regras R^{c_j}, melhores indivíduos

1: $S^x \Leftarrow$ Conjunto das d características da imagem de teste x
2: $S^{(D,x)} \Leftarrow$ Para cada característica d_u, calcula-se a distância de S^x para S
3: $S^{(Dn,x)} \Leftarrow$ Normalização das Distâncias $\in S^{(D,x)}$
4: $S^{(D,x)} \Leftarrow$ Distâncias geradas pelos indivíduos aplicados em $S^{(Dn,x)}$
5: $S_p^{(D,x)} \Leftarrow$ Pré-processamento das distâncias $\in S^{(D,x)}$
6: Para cada classe c_j faça
7: Para cada imagem m em c_j faça
8: $t \Leftarrow d$ distâncias $\in S_p^{(D,x)}$ de x para m
9: insere t em M^{c_j}
10: Fim Para
11: Para cada coluna c de M^{c_j} faça
12: $I_x^{c_j}[c] \Leftarrow min(c)$
13: Fim Para
14: Para cada $r_i \in \{0,1\}$ faça
15: Utilizando R^{c_j} e $I_x^{c_j}$, calcula-se $s(x, c_j, r_i)$ (Equação 3.1)
16: Fim Para
17: Utilizando $s(x, c_j, r_0)$ e $s(x, c_j, r_1)$, calcula-se a $relevancia(x, c_j)$ (Equação 3.2)
18: Fim Para
19: Utilizando $relevancia(x, c_j)$, calcula-se $p(c_j|x)\ \forall\ 1 \leq j \leq k$ (Equação 3.3)
20: Retorna a probabilidade $p(c_j|x)\ \forall\ 1 \leq j \leq k$

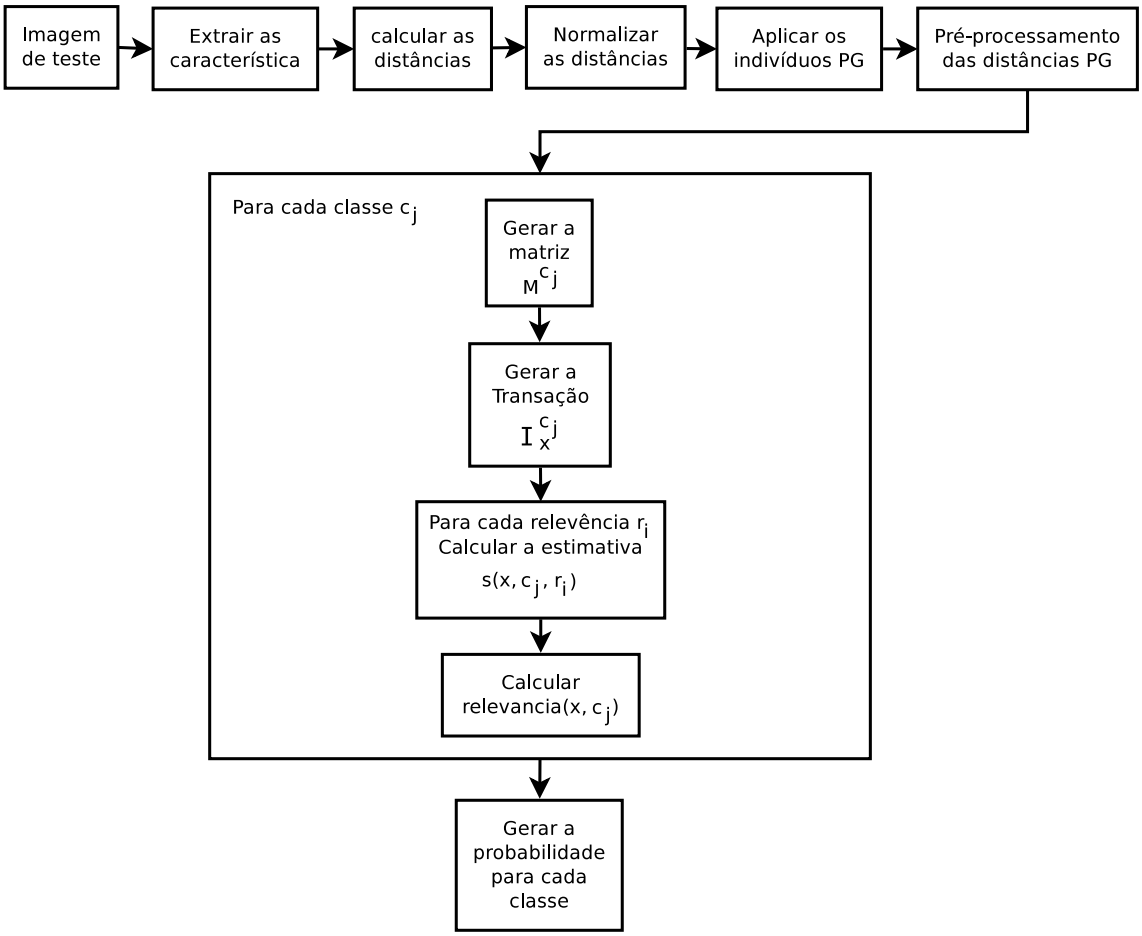


Figura 3.6: Fluxograma de teste do GRIA.

GRIA (ilustrado na Tabela 3.3). Porém, no GRIA-LAC não é aplicado o algoritmo *Apriori* nestes registros. O Algoritmo 8 mostra os passos do treinamento. A Figura 3.7 ilustra o fluxograma de treinamento do GRIA-LAC.

A etapa do teste do GRIA-LAC é igual as do GRIA até o momento da criação da matriz M^{c_j} (ilustrado na Tabela 3.4) para cada classe, que possui transações com distâncias GP da imagem de teste para as imagens de treinamento da classe c_j. Em seguida, utilizam-se como entrada do LAC, para cada transação t em M^{c_j}, o registro G^{c_j} e a transação t. Assim, o LAC criará regras de associação sob demanda para cada transação de M^{c_j}, e retornará a probabilidade de cada transação ser de relevância 0 e relevância 1. O GRIA-LAC associa o maior valor das relevâncias 1 como sendo a relevância da imagem x para a classe c_j ($relevancia(x, c_j)$). Finalmente, utiliza-se a Equação 3.3 para retornar uma probabilidade de x ser de cada uma das c_j classes. O Algoritmo 9 demonstra os passos da etapa de teste. A Figura 3.8 ilustra o fluxograma de teste do GRIA-LAC.

Algoritmo 8 Etapa de treinamento do método GRIA-LAC

Require: Conjunto de imagens de treinamento T, suporte mínimo σ_{min}, confiança mínima θ_{min}

1: $S \Leftarrow$ Conjunto das d características de T
2: $S^D \Leftarrow$ Para cada características d_u, calcula-se a matriz D^{d_u} de distância
3: $S^{Dn} \Leftarrow$ Cada matriz $D^{d_u} \in S^D$ normalizada pela gaussiana (Seção 4.2.5)
4: Calculam-se os melhores indivíduos por meio de S^{Dn} e do Algoritmo 1
5: $S^D \Leftarrow$ Novas matrizes de distância, geradas por meio dos indivíduos aplicados em S^{Dn}
6: $S_p^D \Leftarrow$ Pré-processamento das distâncias $\in S^D$
7: Para cada classe c_j faça
8: Para cada imagem $m \in c_j$ faça
9: Para cada imagem $n \in T$ faça
10: $t \Leftarrow d$ distâncias $\in S_p^D$ de m para n
11: Se $n \in c_j$ então
12: marca t com relevância 1
13: Senão
14: marca t com relevância 0
15: Fim Se
16: insere t em G^{c_j}
17: Fim Para
18: Fim Para
19: Fim Para

Tabela 3.5: Exemplo da matriz $M_{ij}^{c_j}$ e da porcentagem de relevância 0 e 1 do GRIA-LAC.

					Porcentagem de relevância	
	Transação/Indivíduo	d_{GP1}	d_{GP2}	d_{GP3}	0	1
$M_{ij}^{c_j}$	1	2,65	4,00	0,54	30,0%	70,0%
	2	4,39	2,27	0,43	23,0%	77,0%
	3	5,50	2,80	0,99	50,0%	50,0%
	4	4,44	3,20	0,48	10,0%	90,0%
	5	3,05	1,16	0,59	44,0%	56,0%
Maior probabilidade						90,0%

Algoritmo 9 Etapa de teste do método GRIA-LAC

Require: Conjunto S das características das imagens de treinamento, imagem de teste x, conjunto de registros G^{c_j}, melhores indivíduos

1: $S^x \Leftarrow$ Conjunto das d características da imagem de teste x
2: $S^{(D,x)} \Leftarrow$ Para cada características d_u, calcula-se a distância de S^x para S
3: $S^{(Dn,x)} \Leftarrow$ Normalização das Distâncias $\in S^{(D,x)}$
4: $S^{(D,x)} \Leftarrow$ Distâncias geradas pelos indivíduos aplicados em $S^{(Dn,x)}$
5: $S_p^{(D,x)} \Leftarrow$ Pré-processamento das distâncias $\in S^{(D,x)}$
6: Para cada classe c_j faça
7: Para cada imagem m em c_j faça
8: $t \Leftarrow d$ distâncias $\in S_p^{(D,x)}$ de x para m
9: insere t em M
10: Fim Para
11: Maior_probabilidade $\Leftarrow 0$
12: Para cada transação t $\in M^{c_j}$ faça
13: P1 $\Leftarrow$ Probabilidade da transação t ser de relevância 1, usando o Algoritmo 2 do LAC e o registro G^{c_j}
14: Maior_probabilidade = max(Maior,P1)
15: Fim Para
16: $relevancia(x, c_j) \Leftarrow$ Maior_probabilidade
17: Fim Para
18: Utilizando $relevancia(x, c_j)$, calcula-se $p(c_j|x)\ \forall\ 1 \leq j \leq k$ (Equação 3.3)
19: Retorna a probabilidade $p(c_j|x)\ \forall\ 1 \leq j \leq k$

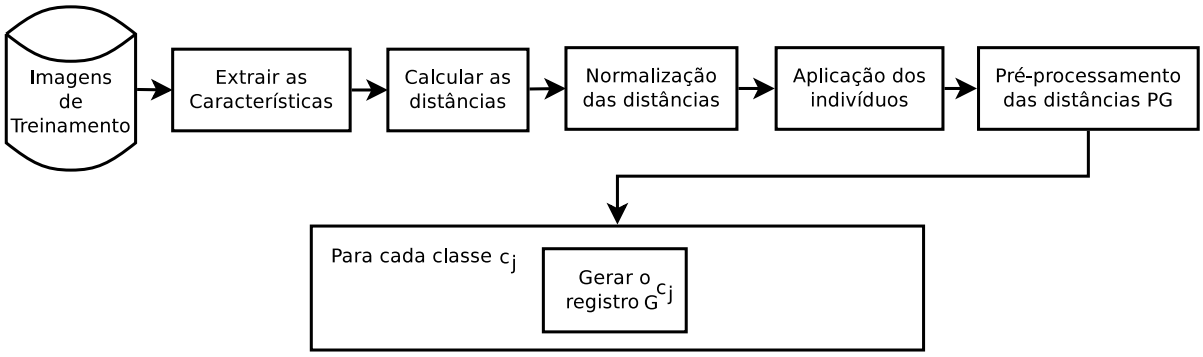


Figura 3.7: Fluxograma de treinamento do GRIA-LAC.

3.5 Comparação dos Métodos

Os quatro métodos apresentados (LIA, RAIA, GRIA e GRIA-LAC) possuem a característica comum de utilizar regras de associação para classificação como técnica de aprendizagem para anotação de imagens. Os métodos LIA e GRIA-LAC utilizam o classificador associativo LAC para gerar as regras de associação, enquanto o RAIA e o GRIA utilizam o algoritmo *apriori*. O método LIA gera regras da forma $X \rightarrow c_j$, onde X são características (valores dos vetores de características ou dos histogramas de palavras visuais) e c_j é uma classe (categoria). Já os métodos RAIA, GRIA e GRIA-LAC geram regras para cada classe, que são escritas da forma $X \rightarrow r_i$, na qual X são similaridades (distâncias) entre duas imagens e r_i é relevância 0 ou 1 para a classe. As funções de similaridade do GRIA e do GRIA-LAC são obtidas por meio da PG. A Tabela 3.6 apresenta a comparação dos métodos.

Tabela 3.6: Comparação dos métodos propostos.

	Regras de associação			Usa PG
	Algoritmo	Antecedente	Consequente	
LIA	LAC	características	classe	-
RAIA	*apriori*	distâncias	relevância	não
GRIA	*apriori*	distâncias	relevância	sim
GRIA-LAC	LAC	distâncias	relevância	sim

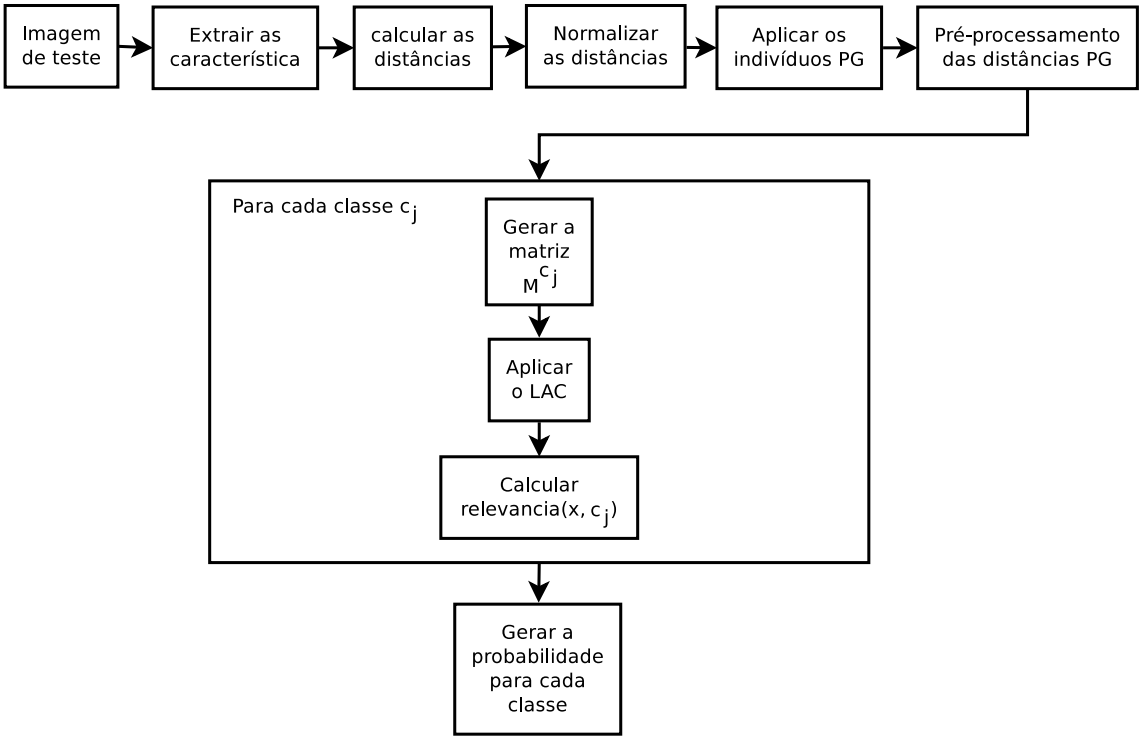


Figura 3.8: Fluxograma de teste do GRIA-LAC.

Capítulo 4

Validação

Este capítulo descreve os experimentos realizados utilizando os métodos para anotação automática de imagens apresentados no Capítulo 3.

4.1 Projeto Experimental

Esta seção apresenta as coleções de imagens utilizadas, bem como as métricas de avaliação e a metodologia empregada.

4.1.1 Bases de Imagens

Os experimentos foram realizados utilizando-se duas bases de imagens previamente anotadas em categorias. A primeira coleção é um subconjunto da base Caltech-256 [33] e a segunda é um subconjunto da *FreeFoto.com*.

Caltech-256

A Caltech-256 é uma base que contém 30.607 imagens coloridas, distribuídas em 256 categorias ("avião", "carro", "face", "tênis", "pessoa", "cão", entre outras). O número de imagens por categoria varia entre 80 e 827, sendo que a média de imagens por categoria é 119.

Para a execução dos experimentos foi escolhido um subconjunto da Caltech-256 contendo 4.991 imagens de 25 categorias, chamado daqui em diante de Caltech-25. A escolha das categorias se deu levando-se em conta a quantidade de imagens contidas em cada uma delas, sendo escolhidas as oito categorias de menor tamanho, oito de tamanho médio e as nove maiores. A Figura 4.1 ilustra algumas imagens da Caltech-25 e suas respectivas categorias.

Imagem					
Categoria	Calculadora	Galáxia	Piano	Cavalo	Escorpião

Figura 4.1: Exemplo de imagens da base Caltech-25 e suas respectivas categorias.

FreeFoto Nature

A *FreeFoto.com* é uma base categorizada que contém 131.863 imagens coloridas organizadas em 3.623 categorias. Nesta dissertação, utilizou-se um subconjunto desta base chamado de FreeFoto Nature, que possui imagens de natureza. Este subconjunto é formado por 3.462 imagens em 9 categorias. O número de imagens por categoria varia de 70 a 854 imagens. A Figura 4.2 ilustra algumas imagens da FreeFoto Nature e suas respectivas categorias.

Imagem					
Categoria	Árvore	Nuvem	Jardim	Cachoeira	Pôr do sol

Figura 4.2: Exemplo de imagens da base FreeFoto Nature e suas respectivas categorias.

4.1.2 Medidas de Avaliação

Para avaliar a eficácia das anotações, algumas métricas foram utilizadas, como precisão e o coeficiente *Kappa* [15], descritas a seguir.

Precisão

A precisão ou taxa de acerto é a razão entre as imagens de teste classificadas corretamente e todas as imagens de teste utilizadas. Esta métrica pode ser calculada por meio da matriz de confusão C, definida como:

$$C = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1c} \\ c_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \\ c_{c1} & & & c_{cc} \end{bmatrix}$$

onde C é uma matriz quadrada do tamanho do número de classes, sendo que cada elemento c_{ij} representa o número de elementos da classe i classificados como sendo da classe j e c_{ii} representa o número de elementos da classe i classificadas corretamente. A precisão é calculada somando-se os elementos da diagonal principal e dividindo pela soma total dos elementos da matriz.

Coeficiente Kappa

O coeficiente *Kappa* é uma medida estatística que determina o grau de concordância, além do que seria esperado tão somente pelo acaso, entre dois métodos de classificação. Uma possível interpretação para esse coeficiente, segundo [51], é que ele expressa a "quantidade ou proporção"de erros que o método de classificação aplicado evita cometer se comparado com um resultado puramente aleatório. No contexto da classificação supervisionada, onde sabem-se as classes do conjunto de teste (*ground truth*), mede-se o grau de concordância entre as classes previstas pelo classificador e o *ground truth*.

A expressão de cálculo do *Kappa* a partir da matriz de confusão é definida segundo a Equação 4.1 [16], onde r é o número de linhas da matriz (número de classes), x_{i+} é a soma dos elementos da linha i, x_{+i} é a soma dos elementos da coluna i e N é a soma total dos elementos.

$$\kappa = \frac{N \sum_{i=1}^{r} c_{ii} - \sum_{i=1}^{r} x_{i+} x_{+i}}{N^2 - \sum_{i=1}^{r} x_{i+} x_{+i}} \tag{4.1}$$

A medida *kappa* tem como valor máximo um, que representa a concordância total entre os classificadores, já os valores próximos de zero representam a concordância próxima ao acaso, enquanto valores negativos indicam uma concordância pior que o acaso. A Tabela 4.1 ilustra uma das possíveis interpretações destes valores, de acordo com [16].

Tabela 4.1: Coeficiente *Kappa* e a interpretação do desempenho da classificação.

Coeficiente *Kappa*	Desempenho da Classificação
$\kappa \leq 0$	Péssimo
$0 < \kappa \leq 0.2$	Ruim
$0.2 < \kappa \leq 0.4$	Razoável
$0.4 < \kappa \leq 0.6$	Bom
$0.6 < \kappa \leq 0.8$	Muito Bom
$0.8 < \kappa \leq 1.0$	Excelente

4.1.3 Metodologia

A Caltech25 foi utilizada para estudo e ajustes de parâmetros dos métodos propostos no Capítulo 3. As imagens desta base foram redimensionadas para metade do seu tamanho original. Posteriormente, um conjunto aleatório de um sexto destas imagens foi utilizado para geração dos dicionários visuais e depois foi descartado. O restante das imagens foi usado como treinamento e teste, aplicando-se a abordagem de *"2-fold cross-validation"* [66].

Já para a FreeFoto Nature, manteve-se o tamanho original das imagens. Um conjunto aleatório de um sexto das imagens foi utilizado para construção dos dicionários visuais e posteriormente foi descartado. O restante das imagens foram usadas como treino e teste, utilizando a abordagem de *"5-fold cross-validation"*.

4.2 Descrição dos Experimentos

Nas próximas seções são apresentados as descrições dos experimentos, discutindo os parâmetros utilizados e os pré-processamentos.

4.2.1 Funções de Distâncias

Os métodos RAIA, GRIA e GRIA-LAC (Capítulo 3) utilizam como atributos a similaridade entre as imagens. Estas similaridades são calculadas por meio de funções de distâncias, que calculam a distância entre características, por exemplo, vetores de característica globais, vetores de características locais ou ainda histogramas de palavras visuais. A seguir, descrevem-se as funções utilizadas.

Manhattan (L1)

A função de distância Manhattan é conhecida como distância de quarteirão, na qual a distância entre um ponto $P = (p_1, p_2, .., p_n)$ e um ponto $Q = (q_1, q_2, .., q_n)$ é a soma das diferenças absolutas de suas coordenadas, de acordo com a Equação 4.2.

$$d(P, Q) = \sum_{i=1}^{n} |p_i - q_i| \tag{4.2}$$

Euclidiana (L2)

A função de distância Euclidiana é baseada no teorema de Pitágoras, onde a distância entre um ponto P e Q é definida segunda a Equação 4.3.

$$d(P, Q) = \sqrt{\sum_{i=1}^{n} (|p_i - q_i|)^2} \tag{4.3}$$

Distância dLog

A distância dLog foi proposta em [75] e possui como vantagem o fato de amenizar o efeito que vetores com valores muito altos causam no valor final da distância. O cálculo desta distância é definida segunda a Equação 4.4.

$$dLog(P, Q) = \sum_{i=1}^{n} |f(p_i) - f(q_i)| \tag{4.4}$$

$$na\ qual\ f(x) = \begin{cases} 0, & \text{se } x = 0 \\ 1, & \text{se } 0 < x \leq 1 \\ \lceil log_2 x \rceil + 1, & \text{caso contrário} \end{cases}$$

Earth Mover's Distance (EMD)

Earth Mover's Distance [68] é uma distância que mede o custo mínimo a ser pago para transformar uma distribuição em outra. Intuitivamente, uma distribuição pode ser vista como uma massa de terra espalhada no espaço, enquanto a outra pode ser vista como um conjunto de buracos neste mesmo espaço [4]. Assim, a EMD mede o mínimo de esforço necessário para preencher estes buracos com terra.

Utilizando-se EMD, pode-se calcular uma única distância entre x vetores de características de dimensões d de uma imagem e y vetores de mesma dimensão de uma outra imagem. Dessa maneira, usa-se esta distância para calcular a similaridade entre duas imagens usando seus respectivos vetores de características locais extraídos.

4.2.2 Descrição do Conteúdo Visual

Esta seção apresenta os métodos utilizados para descrever o conteúdo visual das imagens. As técnicas usadas foram: descritores globais; detectores de pontos de interesse e descritores locais; e dicionários visuais.

Descritores Globais

No método LIA, utilizam-se os vetores de característica dos descritores globais diretamente com as regras de associação. Já nos métodos que utilizam similaridade entre imagens (RAIA, GRIA e GRIA-LAC), usam-se as distâncias entre os vetores de características como atributos.

A seguir, descrevem-se os descritores globais utilizados. A Tabela 4.2 ilustra estes descritores, informando seus respectivos tipos de evidência, as funções de distância utilizadas e a quantidade de *bins* (valores) gerados para o vetor de característica.

Border/Interior Pixel Classification (BIC): o BIC [75] é um descritor global baseado em cor, em que cada imagem é quantizada em 64 intervalos de cores. Cada *pixel* da imagem é classificado como sendo de interior ou de borda, levando-se em consideração seus vizinhos-4 (acima, à direita, abaixo e à esquerda) e suas respectivas cores. Desta forma, se o *pixel* estiver no mesmo intervalo de cor dos seus vizinhos-4 ele é de interior, caso contrário, é de borda. Ao final, constroem-se e concatenam-se dois histogramas de cor, um para os *pixels* de interior e outro para os *pixels* de borda, formando um histograma de 128 dimensões.

Global Color Histogram (GCH): o GCH [77] gera um histograma baseando-se na cor, em que segundo [62], seu algoritmo quantiza o espaço de cor em uma quantidade uniforme de *bins*, percorrendo a imagem calculando a quantidade de *pixels* que pertence a cada *bin*. Os experimentos realizados foram quantizados em 64 *bins* (valores).

Joint Auto-correlogram (JAC): o JAC [93] é um descritor que utiliza a cor como evidência principal. Porém, segundo [62], também utiliza as propriedades: magnitude do gradiente, *rank* e *texturedness*. O vetor de características gerado contém todas as combinações possíveis entre estas quatro propriedades juntamente com funções de distâncias diferentes. Sendo assim, gera-se um vetor com QC $\times$ QG $\times$ QR $\times$ QT $\times$ QD valores, onde QC é a quantidade de cores no espaço de cor quantizados (64), QG, QR e QT é a quantidade de valores possíveis (cinco para cada) no espaço quantizado da magnitude do gradiente, *rank* e *texturedness*, respectivamente, e QD é a quantidade de distâncias utilizadas (quatro). Desse modo, ao final tem-se um vetor de tamanho $64 \times 5 \times 5 \times 5 \times 4$, ou seja, 32000 valores.

Local Activity Spectrum (LAS): o LAS [79] é baseado em textura, e segundo [62], ele captura a atividade espacial de textura, de cada *pixel*, nas direções vertical, horizontal, diagonal e anti-diagonal. Cada uma destas atividades é quantizada separadamente, formando no total um vetor de 256 valores.

Quantized Compound Change Histogram (QCCH): o QCCH [37] é um descritor de textura. Este utiliza uma vizinhança-4 de cada *pixel* para calcular a média da taxa de variação em quadro direções, gerando ao final um vetor de 40 valores.

Tabela 4.2: Descritores globais utilizados nos experimentos.

Descritor Global	Tipo de Evidência	Função de Distância	Quantidade de *Bins*
BIC	Cor	dLog	128
GCH	Cor	L1	64
JAC	Cor	L1	32000
LAS	Textura	L1	256
QCCH	Textura	L1	40

Detectores e Descritores Locais

Utilizou-se o detector de pontos de interesse Harris-Laplace juntamente com o descritor local SIFT. Também usou-se o detector Fast-Hessian em conjunto com o descritor local SURF. As extrações do detector Harris-Laplace e do descritor SIFT foram realizadas utilizando uma implementação disponibilizada por Krystian Mikolajczyk[1]. Já o detector Fast-Hessian e o descritor SURF foram obtidos no site do autor[2].

Nos métodos que usam similaridade, aplica-se a função EMD para calcular a distância entre um conjunto de vetores de características locais de uma imagem para outro conjunto de vetores de uma outra imagem.

A seguir, descrevem-se os detectores Harris-Laplace detector e Fast-Hessian, e os descritores locais SIFT e SURF.

Harris-Laplace: O Harris-Laplace é um detector de pontos de interesse que resolve problemas como alterações significativas na localização do ponto, variações de escala, da forma de vizinhança do ponto, e até mesmo da iluminação.

[1]http://www.robots.ox.ac.uk/∼vgg/research/affine/index.html (data do último acesso: 16/07/2011)
[2]http://www.vision.ee.ethz.ch/∼surf/ (data do último acesso: 16/07/2011)

Fast-Hessian: O Fast-Hessian é uma implementação rápida do Hessian-Affine detector. O Hessian-affine detector é um detector de pontos de interesse e faz parte do subconjunto de classes de detectores chamado de *affine-invariant*. Este método resolve problemas como alterações de localização, variação de escala e diferentes formas de vizinhança de um ponto. Esta técnica é baseada em três etapas principais [59]. Primeiro, calcula-se a matriz de um ponto (chamada de segunda matriz), que se usa para normalizar a região deste ponto, tornando-o invariante a inclinação e alongamento. Segundo, calcula-se a dimensão da estrutura local utilizando-se extremos locais da derivada normalizada sobre a escala. Por fim, usa-se um detector *affine* adaptado que determina a localização dos pontos de interesse.

SIFT: O SIFT é um descritor local robusto, sendo invariante à escala, rotação, iluminação, ruídos, distorções, e até mesmo a alterações de ponto de vista em 3D. Inicialmente, após a detecção dos pontos de interesse da imagem (utilizando por exemplo o Hessian-affine), o SIFT seleciona alguns destes pontos com base em medidas de estabilidade, sendo que pontos de baixo contraste e de borda são eliminados. Atribui-se uma escala e orientação para cada ponto, e a partir disso, cria-se um histograma de orientações levando-se em conta a região em volta do ponto. Finalmente, utiliza-se o histograma de orientações para criar um vetor de característica do ponto.

SURF: O SURF é um descritor local invariante a rotação e escala. Foi baseado, em parte, no descritor SIFT. O primeiro passo do SURF é fixar uma orientação baseando-se nas informações de uma região circular em volta do ponto de interesse. Em seguida, constrói-se um retângulo alinhado com a orientação e extrai-se o vetor de característica deste retângulo.

Dicionários Visuais

Foram criados dois dicionários visuais para cada base de imagens. Na primeira etapa de criação do dicionário, usam-se os detectores Harris-Laplace e Fast-Hessian. Na segunda etapa, calculam-se os vetores de característica de cada ponto obtido pelo Harris-Laplace por meio do descritor SIFT, e cada ponto do Fast-Hessian por meio do SURF. Na terceira etapa, 500 vetores aleatórios de cada descritor foram escolhidos para serem os centros dos agrupamentos, que representam as palavras visuais. Assim, um ponto extraído de uma imagem é de uma determinada palavra visual se ele estiver mais perto do ponto (usando distância L2) que representa esta palavra. Ao final, tem-se dois dicionários para cada base: o Harris-Laplace—SIFT—K500 e o Fast-Hessian—SURF—k500.

No método LIA, utilizam-se os histogramas de palavras visuais diretamente com as regras de associação. Já nos métodos que usam similaridade, aplica-se a função L1 para calcular a distância entre histogramas.

4.2.3 Programação Genética

A Tabela 4.3 ilustra os parâmetros PG utilizados nos experimentos da base Caltech25 e FreeFoto. Estes parâmetros foram escolhidos de acordo com [29].

Tabela 4.3: Parâmetros PG utilizados nos experimentos.

Parâmetros	Valores
População	30
Gerações	10
Reprodução	0,05
Crossover	0,8
Mutação	0,2
Funções	+, -, *, /, raiz
Altura máxima da árvore	6
Função de Adequação	FFP4 [25]

4.2.4 Discretização

Discretização é uma técnica que transforma dados contínuos em discretos. Nesta dissertação, transformam-se atributos numéricos em nominais. Este método é frequentemente utilizado como pré-processamento de regras de associação, pois os atributos numéricos possuem uma quantidade infinita de valores, podendo causar alta dimensionalidade nas regras [86]. Dessa forma, os atributos numéricos são agrupados em intervalos nominais. Espera-se que cada intervalo contenha números que estejam relacionados entre si, ou seja, que possuam a mesma informação.

Nos experimentos, aplica-se o discretizador somente no conjunto de treinamento, criando intervalos para cada atributo. As imagens de teste não são utilizadas no processo de geração dos intervalos, para não contaminar o treinamento. Posteriormente, o conjunto de treinamento e de teste são projetados para os valores nominais, de acordo com estes intervalos gerados.

Existem vários algoritmos para discretização dos dados, podendo ser divididos em supervisionados [28, 43, 48], nos quais usam-se os atributos e as classes dos exemplos de entrada; semi-supervisionados [9], nos quais as classes de apenas alguns exemplos de entrada são conhecidas; e não supervisionados [44, 58], que não utilizam as classes dos exemplos.

Nesta dissertação, utiliza-se um discretizador não supervisionado denominado *binning*,

implementado na ferramenta *Waikato Environment for Knowledge Analysis* [3] (WEKA). Neste método, cada atributo (a_1, a_2, ..., a_n) é discretizado em um número n de intervalos, escolhido pelo usuário. Para cada atributo a_i específico, o menor valor deste atributo ($min(a_i)$) é o início do seu primeiro intervalo, e o maior valor ($max(a_i)$) é o final do seu último intervalo. O tamanho t de cada um destes intervalos é obtido por meio da Equação 4.5, que calcula a diferença entre o maior valor e o menor valor deste atributo, dividido pelo número n de intervalos. Por fim, os valores do menor e do maior atributo são substituídos por menos infinito e mais infinito, respectivamente, para abranger todos os valores numéricos possíveis, incluindo assim valores que não foram utilizados na geração dos intervalos, mas que possam estar no conjunto de teste.

$$t = \frac{max(a_i) - min(a_i)}{n} \tag{4.5}$$

A Tabela 4.4 ilustra um conjunto de treinamento com três atributos numéricos (a_1, a_2 e a_3). Já a Tabela 4.5 mostra dois intervalos nominais gerados pelo discretizador para cada um destes atributos. Finalmente, a Tabela 4.6 exibe o conjunto de treinamento da Tabela 4.4 projetado para os valores nominais, de acordo com os intervalos da Tabela 4.5 gerados pelo discretizador.

Tabela 4.4: Atributos numéricos.

Imagem	Atributos Numéricos		
	a_1	a_2	a_3
01	84	0.546755	358
02	112	0.523954	363
03	100	0.461385	323
04	86	0.372389	326
05	123	0.441345	367

Tabela 4.5: Intervalos gerados pelo discretizador.

Atributo	Intervalos	
a_1	[-inf — 103.5]	]103.5 — +inf]
a_2	[-inf — 0.459572]	]0.459572 — +inf]
a_3	[-inf — 345]	]345 — +inf]

[3] http://www.cs.waikato.ac.nz/ml/weka/ (data do último acesso: 16/07/2011)

Tabela 4.6: Atributos numéricos discretizados.

Imagem	Atributos discretizados.		
	a_1	a_2	a_3
01	[-inf — 103.5]	]0.459572 — +inf]	]345 — +inf]
02	]103.5 — +inf]	]0.459572 — +inf]	]345 — +inf]
03	[-inf — 103.5]	]0.459572 — +inf]	[-inf — 345]
04	[-inf — 103.5]	[-inf — 0.459572]	[-inf — 345]
05	]103.5 — +inf]	[-inf — 0.459572]	]345 — +inf]

4.2.5 Normalização Gaussiana

Nesta dissertação, utiliza-se a Normalização Gaussiana [69] para enquadrar valores de similaridade (distâncias) no intervalo [0,1]. Para isto, calcula-se a média M e o desvio padrão σ de uma matriz contendo distâncias, de uma certa característica, das imagens de treinamento. Em seguida, aplica-se a Equação 4.6 em cada distância d, que podem ser distâncias da própria matriz ou distâncias da imagem de teste para as imagens de treinamento.

$$Gaussiana = \frac{\frac{d-M}{3*\sigma} + 1}{2} \tag{4.6}$$

Esta normalização possui a propriedade de que 99,7% do valores são enquadrados no intervalo [0,1]. Para os valores que possivelmente deram menor que 0, truncam-se seus resultados para 0. Analogamente, para os valores maiores que 1, truncam-se para 1.

4.2.6 Técnicas de Aprendizagem

Nesta dissertação, comparam-se os resultados dos métodos propostos no Capítulo 3 com duas técnicas de aprendizagem, o *k-Nearest Neighbor* e o *Support Vector Machine*, descritos a seguir.

k-Nearest Neighbor (kNN)

O kNN é uma técnica simples de classificação de objetos [31]. Esta técnica classifica um objeto de teste baseando-se nos k exemplos de treinamento (vizinhos) mais próximos no espaço de características. Sendo assim, a tarefa de classificação da imagem de teste é decidida pela votação majoritária das classes dos vizinhos mais próximos.

***Support Vector Machine* (SVM)**

O SVM é uma técnica de aprendizagem de máquina introduzida por [10]. Esta técnica projeta as características dos objetos de treinamento em um plano com a mesma dimensão destas características. Em seguida, cria-se um hiperplano que separa linearmente as características, de tal forma que maximizam-se as margens entre as classes. Intuitivamente, a margem pode ser interpretada como uma medida de separação entre duas classes.

4.3 Parametrização

Esta seção apresenta os experimentos que foram realizados para acerto dos parâmetros que são comuns entre os métodos do Capítulo 3. Estes experimentos foram feitos com o LIA na Caltech25, usando vetores de características globais do descritor BIC. Por padrão, nestes experimentos utilizam-se os parâmetros: confiança igual a 0,0000001; suporte que utilize no mínimo um item (no caso deste experimento, uma imagem), ou seja, um valor muito baixo; tamanho da regra igual a 4; e um número máximo de imagens por classe. Para cada experimento específico, variou-se apenas um destes parâmetros.

4.3.1 Valores de Confiança

Este experimento tem como objetivo avaliar quanto o valor da confiança interfere na eficácia e eficiência da anotação. Como pode ser visto na Tabela 4.7, quanto menor a confiança maior é a eficácia. Isto ocorre porque quanto menor a confiança, mais regras serão utilizadas na classificação, aumentando a quantidade de informações. Porém, a eficiência diminui, como pode ser visto nesta mesma tabela. Sendo assim, utilizou-se o menor valor de confiança nos métodos, pois foi o valor que mais melhorou a eficácia.

Tabela 4.7: Diferentes valores do parâmetro confiança.

Valores de Confiança	0,5	0,3	0,1	0,0001	0,0000001
Porcentagem de acerto	0,13	0,19	0,23	0,33	0,34
Tempo em minutos	41	43	45	46	47

4.3.2 Valores de Suporte

Igualmente a confiança, o suporte obteve uma melhor eficácia quando usou-se o menor valor (Tabela 4.8). Então, utilizou-se nos métodos o valor de suporte que use no mínimo 1 item.

Tabela 4.8: Valores do parâmetro suporte.

Valores de Suporte	100	50	5	1
Porcentagem de acerto	0,17	0,20	0,32	0,34
Tempo em minutos	38	39	42	47

4.3.3 Tamanho das Regras de Associação

As regras de associação podem ter diferentes tamanhos, por exemplo, a regra $fralda \Rightarrow cerveja$ possui tamanho 2, já a regra $carro, pessoa, casa \Rightarrow cidade$ possui tamanho 4. Então, o objetivo deste experimento foi avaliar se o tamanho máximo permitido para gerar as regras influencia na eficiência e eficácia da anotação. Segundo a Tabela 4.9, verificou-se que quanto maior o tamanho das regras utilizadas, maior a eficácia. Porém, a eficiência diminui bastante. Então, para o LIA, que utiliza grandes quantidade de atributos, utilizaram-se regras de tamanho máximo igual a 20. Para os outros métodos, que utilizam uma quantidade menor de atributos, utilizou-se o tamanho máximo possível.

Tabela 4.9: Diferentes valores para os tamanhos das regras.

Tamanho das Regras	2	3	4	20	50
Porcentagem de acerto	0,29	0,29	0,34	0,38	0,38
Tempo em minutos	1	3	47	134	134

4.3.4 Número de Imagens no Treinamento

Neste experimento, variou-se o número de imagens em cada classe. O valor 30 é quando o *fold* da Caltech25 utilizado está balanceado, isto é, quando todas a classes de treinamento possuem o mesmo número de imagens. Passando deste valor, somente as maiores classes aumentarão o número de imagens na classe. Assim, como pode ser visto na Tabela 4.10, aumentando-se o número de imagens por classe, aumenta-se a eficácia. Por isso, optou-se por usar todas as imagens no treinamento, mesmo se a base não estivesse balanceada.

4.4 Resultados

Esta seção descreve os resultados obtidos para as bases Caltech25 e FreeFoto. Na primeira coleção, realiza-se uma comparação entre os métodos propostos nesta dissertação. Já na

Tabela 4.10: Diferentes quantidades de imagens por classe no treinamento.

Imagens	1	15	30	100
Porcentagem de Acerto	0,06	0,12	0,14	0,19
Tempo em minutos	38	40	41	42

segunda, faz-se uma comparação entre estes métodos e os métodos de aprendizagem KNN e SVM.

4.4.1 Base Caltech25

Os experimentos da base Caltech25 foram realizados utilizando-se descritores globais e dicionários visuais. Os descritores usados foram BIC, LAS e GCH. Já os dicionários foram o Harris-Laplace—SIFT—K500 e o Fast-Hessian—SURF—k500. No LIA, concatenaram-se os vetores de características globais com os histogramas de palavras visuais, e como pré-processamento, realizou-se uma discretização em 10 intervalos (Seção 4.2.4). Nos restantes dos métodos, utilizaram-se as funções de distâncias da Tabela 4.2 para os vetores de características globais e a função L1 para os dicionários, aplicando-se como pré-processamento uma normalização gaussiana (Seção 4.2.5) e uma discretização em 10 intervalos. Nos métodos que utilizam GP, foram utilizados os 5 melhores indivíduos para cada *fold*. As distâncias geradas pela PG são normalizadas pela gaussiana e discretizadas.

Nesta base, efetuou-se uma comparação dos métodos propostos no Capítulo 3. A Tabela 4.11 ilustra a porcentagem de acerto para cada método, na qual nota-se que o RAIA possui a melhor porcentagem para cada *fold*. Devido à capacidade da PG combinar diversas evidências para descrever as imagens, esperava-se um desempenho superior do GRIA. Contudo, os resultados mostraram o oposto. Uma das possíveis razões para este resultado é um *overfitting* dos dados. Um *overfitting* ocorre quando o treinamento não consegue generalizar uma solução encontrada para o problema do conjunto de treinamento para o de teste. Então, uma das maneiras para resolver isto é separar o treinamento da PG do treinamento das regras de associação, ou seja, separar um *fold* de treinamento para a PG e um *fold* de validação para as regras de associação. Outra possível razão seria a má qualidade dos indivíduos gerados pela PG. Isto, por sua vez, pode ser resolvido testando-se várias combinações de parâmetros da PG, como por exemplo, o número da população e a função de adequação. Nota-se, também, que o LIA gerou o pior resultado. Uma explicação para isto seria que as funções de distância influenciam melhor na interpretação das características, pois, estas últimas foram criadas e otimizadas com o intuito de serem usadas em conjunto com uma função de distância.

Tabela 4.11: Porcentagem de acerto dos métodos para a base Caltech25.

Caltech25	LIA	RAIA	GRIA	GRIA-LAC
Fold 1	0,38	0,45	0,42	0,40
Fold 2	0,31	0,39	0,25	0,37
Média	0,35	0,42	0,33	0,39

A Tabela 4.12 ilustra a média das porcentagens de acerto por categoria desta base. Segundo [33], as categorias *145.motorbikes-101*, *251.airplanes-101* e *253.faces-easy-101* são consideradas categorias fáceis. Sendo assim, verifica-se que os métodos possuem um viés para elas, ou seja, acertam estas categorias com mais facilidade que as outras. Porém, verifica-se que o viés delas para o método GRIA-LAC é muito grande, sendo que a média de porcentagem das outras categorias é muito menor.

A seguir, exemplificam-se algumas regras de associação geradas no *fold* 1 para a categoria *011.billiards* no método RAIA, e seus respectivos valores de confiança. Verificam-se nestas regras que distâncias pequenas implicaram relevância 1, e distâncias maiores implicaram relevância 0.

1. BIC= [0,4—0,5[$\wedge$ LAS= [0,406431—0,491227[$\wedge$ siftBOF= [0,54242—0,607789[$\wedge$ surfBOF= [0,664389—0,731511] $\rightarrow$ Relevância= 0, com θ= 1,00

2. BIC= [0,7—0,8[$\wedge$ GCH=]0,637016—0,716643[$\wedge$ LAS= [0,745613—0,830409] $\wedge$ siftBOF= [0,607789—0,673157] $\wedge$ surfBOF= [0,597267—0,664389] $\rightarrow$ Relevância= 0, com θ= 1,00

3. GCH= [-inf—0,079627] $\wedge$ LAS= [0,406431—0.491227[$\wedge$ siftBOF= [-inf—0.411683[$\wedge$ surfBOF= [-inf—0.3959] $\rightarrow$ Relevância= 1, com θ= 0,18

Também, exemplificam-se alguns indivíduos de PG gerados para o *fold* 1. Notam-se que os indivíduos combinam as similaridades obtidas a partir dos descritores globais com as similaridades obtidas pelos dicionários visuais.

1. (((((surfBOF * siftBOF) + siftBOF) + (bic + siftBOF)) + (bic + siftBOF)) + (surfBOF * (siftBOF + ((bic + las) + (surfBOF * las))))) * las

2. ((bic + siftBOF) + (siftBOF * ((surfBOF / (bic + siftBOF)) * siftBOF))) * las

3. ((bic + las) + (((surfBOF / siftBOF) * ((((0.57 * gch)) * gch) * gch)) * (((0.89 * gch)) * gch))) * las

Tabela 4.12: Média das porcentagens de acerto por categoria da Caltech25.

Categoria	LIA	RAIA	GRIA	GRIA-LAC
011.billiards	0,22	0,43	0,63	0,02
027.calculator	0,10	0,21	0,27	0,00
034.centipede	0,20	0,25	0,35	0,00
082.galaxy	0,39	0,49	0,45	0,03
086.golden-gate-bridge	0,06	0,28	0,21	0,00
096.hammock	0,15	0,30	0,25	0,00
099.harpsichord	0,05	0,11	0,14	0,00
105.horse	0,08	0,20	0,21	0,00
126.ladder	0,06	0,10	0,08	0,01
145.motorbikes-101	**0,68**	**0,70**	**0,55**	**0,92**
179.scorpion-101	0,03	0,23	0,10	0,00
183.sextant	0,03	0,15	0,03	0,00
186.skunk	0,00	0,10	0,04	0,01
197.speed-boat	0,07	0,11	0,16	0,00
200,stained-glass	0,11	0,17	0,14	0,00
201.starfish-101	0,06	0,21	0,09	0,01
204.sunflower-101	0,44	0,47	0,19	0,05
213.teddy-bear	0,02	0,07	0,18	0,01
218.tennis-racket	0,02	0,16	0,05	0,00
223.top-hat	0,00	0,08	0,06	0,00
232.t-shirt	0,16	0,22	0,11	0,03
233.tuning-fork	0,11	0,06	0,08	0,00
249.yo-yo	0,05	0,03	0,00	0,01
251.airplanes-101	**0,73**	**0,74**	**0,54**	**0,96**
253.faces-easy-101	**0,55**	**0,73**	**0,47**	**0,90**

4.4.2 Base FreeFoto Nature

Os experimentos da base Freefoto foram realizados utilizando-se descritores globais, descritores locais e dicionários visuais. Os descritores globais usados foram BIC, LAS, GCH, QCCH e JAC. Os descritores locais utilizados foram o SIFT juntamente com o detector Harris-Laplace e o SURF com o Fast-Hessian. Já os dicionários foram o Harris-Laplace—SIFT—K500 e o Fast-Hessian—SURF—k500. No LIA, concatenaram-se os vetores de características globais com os histogramas de palavras visuais, e como pré-processamento, realizou-se uma discretização com 10 intervalos. No RAIA, utilizaram-se as funções de distâncias da Tabela 4.2 para os vetores de características globais e a função L1 para os dicionários, aplicando-se também como pré-processamento uma normalização gaussiana e uma discretização. Nos métodos que utilizam GP, utilizou-se a distância EMD para os descritores locais, e foram utilizados os 5 melhores indivíduos para cada *fold*.

Para a FreeFoto, compararam-se os métodos com o KNN e o SVM. Como pode ser visto na Tabela 4.13, dentre os métodos com melhores resultados, destaca-se o RAIA, com média da porcentagem de acerto igual a 0,9133. Segundo o intervalo de confiança calculado com *alpha* igual a 0,05, estatisticamente existe um empate entre o RAIA e o SVM, ou seja, não existe diferença significativa entre eles. Porém, nota-se que o mesmo não acontece entre o RAIA e o KNN, sendo que os intervalos de confiança dos mesmos não se sobrepõem.

Tabela 4.13: Porcentagens de acerto para a base FreeFoto.

FreeFoto	LIA	RAIA	GRIA	GRIA-LAC	KNN	SVM
Fold 1	0,5529	0,9116	0,9012	0,8371	0,8891	0,9151
Fold 2	0,4974	0,9133	**0,9255**	0,8111	0,8769	**0,9203**
Fold 3	0,5130	0,9116	0,9029	0,7972	0,8666	0,9012
Fold 4	0,5563	0,9012	0,8908	0,8284	0,8925	**0,9203**
Fold 5	0,5615	**0,9289**	0,9151	0,8475	0,8891	0,9116
Média	0,5362	**0,9133**	0,9071	0,8243	0,8828	**0,9137**
Desvio Padrão	0,0290	0,0100	0,0134	0,0202	0,0109	0,0079
Intervalo de Confiança	0,0254	0,0087	0,0117	0,0177	0,0095	0,0069
Média + Intervalo	0,5617	0,9221	0,9189	0,8419	0,8924	0,9206
Média – Intervalo	0,5108	0,9046	0,8954	0,8066	0,8733	0,9068

A Figura 4.3 ilustra algumas quantidades de indivíduos usados no método RAIA e a acurácia obtida para a porcentagem de acerto e para o índice *Kappa*. Como pode ser visto, a acurácia melhorou pouco entre quatro e cinco indivíduos.

A Tabela 4.14 realiza uma comparação dos índices *Kappa* (Seção 4.1.2) obtidos para a

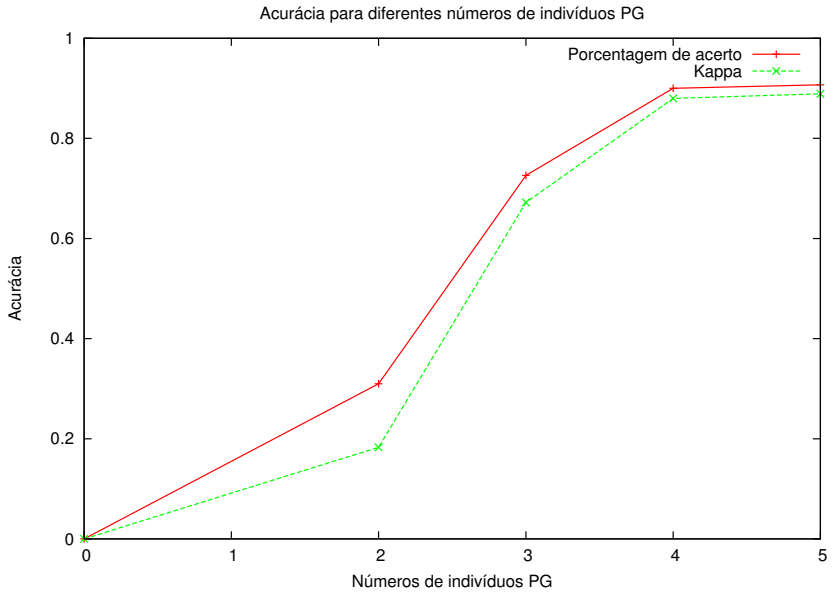


Figura 4.3: Acurácias para o método RAIA usando diferentes quantidades de indivíduos PG.

FreeFoto, sendo que a interpretação para este índice foi feita de acordo com a Tabela 4.1. Verificou-se, então, que os métodos RAIA, GRIA, KNN e SVM tiveram um desempenho excelente para esta coleção de imagens.

Tabela 4.14: Índice *Kappa* para a base FreeFoto.

FreeFoto	LIA	RAIA	GRIA	GRIA-LAC	KNN	SVM
Fold 1	0,46	0,89	0,88	0,80	0,87	**0,90**
Fold 2	0,39	**0,90**	**0,91**	0,77	0,85	**0,90**
Fold 3	0,41	0,89	0,88	0,75	0,84	0,88
Fold 4	0,46	0,88	0,87	0,79	0,87	**0,90**
Fold 5	0,47	**0,92**	**0,90**	0,82	0,87	0,89
Média do Kappa	0,44	**0,90**	0,89	0,79	0,86	**0,90**
Interpretação	Bom	Excelente	Excelente	Muito Bom	Excelente	Excelente

A vantagem de utilizar os métodos em relação ao KNN e o SVM é que as regras de associação geradas são passíveis de interpretação. Por exemplo, pode-se concluir que uma característica está gerando regras mais confiáveis para uma determinada anotação. Outra vantagem seria a interpretação dos indivíduos PG, analisando por exemplo, qual característica se sobressai (mais importante dado sua peso) em um indivíduo.

Capítulo 5

Conclusões e Trabalhos Futuros

5.1 Conclusões

Atualmente, grandes coleções de imagens são utilizadas em várias áreas do conhecimento. Um dos problemas enfrentados no gerenciamento dessa grande quantidade de dados é a recuperação das imagens no banco de dados. A forma mais comum de se recuperar uma imagem é associar anotações (descrições textuais) a ela (por exemplo, palavras-chave e categorias), e utilizar técnicas tradicionais de recuperação de texto. Este trabalho propôs quatro novos métodos para anotação automática de imagens, cujo objetivo foi avaliar a usabilidade das regras de associação para anotação de imagens.

O primeiro método, chamado de LIA, utiliza as representações do conteúdo visual das imagens (por exemplo, vetores de características e histogramas de palavras visuais) como atributos para o LAC, gerando assim regras de associação sob demanda, onde características implicam anotações. O segundo método, denominado RAIA, utiliza as distâncias entre as representações visuais como atributos. O terceiro, chamado de GRIA, utiliza PG para gerar novas funções de similaridade que, por sua vez, são utilizadas como atributos. O último método, chamado de GRIA-LAC, é uma adaptação do GRIA que utiliza o algoritmo de classificação LAC para gerar regras de associação sob demanda.

Estes métodos foram validados por meio de experimentos em duas coleções de imagens, nas quais utilizaram-se como medidas de avaliação o *Kappa* e a precisão de acerto. Estes experimentos mostraram a viabilidade do uso das regras de associação para anotação de imagens, sendo que os resultados indicam um desempenho comparável ou superior ao de técnicas tradicionais da literatura, como o SVM e o KNN.

5.2 Trabalhos Futuros

Existem vários tópicos de extensão e aprimoramento dos métodos de anotação automática, sendo que os principais são discutidos a seguir:

- Utilizar métodos discretização supervisionada [28,43,48] como forma de pré-processamento das regras de associação.
- Testar vários parâmetros do dicionário (como detectores, descritores e números diferentes de agrupamentos), e ainda outros métodos de agrupamento.
- Testar vários parâmetros da Programação Genética, como por exemplo, diferentes valores para população e geração. O trabalho [19] apresenta estratégias de investigação dos parâmetros da PG.
- Separar o treinamento das regras de associação do treinamento da PG. Isto é, utilizar um conjunto de treino para as regras e um conjunto de validação para PG, evitando assim um possível *overfitting*.
- Utilizar coleções de imagem anotadas com palavras-chave, como a coleção MIR FLICKR [39]. Esta base possui possui 25.000 imagens coloridas anotadas com 38 palavras-chave controladas, onde cada imagem pode ser anotada com zero ou várias palavras. Além das 38 palavras-chave, a base também é anotada com *tags*, que são palavras ou frases livres (não controladas). O número médio de *tags* por imagem é 8,94, sendo que 1386 delas são comuns, isto é, aparecem em no mínimo 20 imagens. As *tags* podem, por exemplo, serem utilizadas como atributos das imagens.
- Combinar o resultados dos métodos.
- Combinar os métodos com outros métodos de aprendizagem, como KNN e SVM.
- Aplicar os métodos na tarefa de recuperação de imagens por conteúdo.
- Utilizar os métodos para classificação de documentos textuais, usando por exemplo, medidas de similaridade (distâncias) entre os documentos.
- Adaptar os métodos que utilizam distâncias para serem usados com outros métodos de aprendizagem (por exemplo, o SVM), ao invés das regras de associação.
- Usar regras de associação em tarefas de agrupamento.

Referências Bibliográficas

[1] R. Agrawal, T. Imieliński, and A. Swami. 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*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *VLDB '94: Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994.

[3] N. Ahuja and S. Todorovic. Learning the taxonomy and models of categories present in arbitrary images. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[4] J. Almeida. Recuperação de imagens por cor utilizando análise de distribuição discreta de características. Master's thesis, Instituto de Computação, Unicamp, Campinas, Brazil, Aug. 2007.

[5] R. A. Baeza-Yates, R. Baeza-Yates, and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1999.

[6] W. Banzhaf, P. Nordin, R. Keller, and F. Francone. *Genetic Programming - An Introduction*. Morgan Kaufmann Publishers, Inc, San Francisco, CA, 1998.

[7] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. In *In European Conference on Computer Vision*, pages 404–417, 2006.

[8] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[9] A. Bondu, M. Boulle, V. Lemaire, S. Loiseau, and B. Duval. A non-parametric semi-supervised discretization method. *Data Mining, IEEE International Conference on*, 0:53–62, 2008.

[10] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

[11] D. Cai, X. He, Z. Li, W. Ma, and J. Hierarchical clustering of www image search results using visual, textual and link information. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 952–959, New York, NY, USA, 2004.

[12] J. Cai, Z.-J. Zha, Q. Tian, and Z. Wang. Semi-automatic flickr group suggestion. In *Proceedings of the 17th international conference on Advances in multimedia modeling - Volume Part II*, MMM'11, pages 77–87, Berlin, Heidelberg, 2011. Springer-Verlag.

[13] R. T. Calumby. Recuperação multimodal de imagens com realimentação de relevância baseada em programação genética. Master's thesis, Universidade Estadual de Campinas, 2010.

[14] T. A. S. Coelho, P. P. Calado, L. V. Souza, B. Ribeiro-Neto, and R. Muntz. Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):408–417, 2004.

[15] J. Cohen. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, 20(1):37–46, April 1960.

[16] R. G. Congalton. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, 37(1):35 – 46, 1991.

[17] R. da S. Torres, A. X. Falcão, M. A. Gonçalves, J. P. Papa, B. Zhang, W. Fan, and E. A. Fox. A genetic programming framework for content-based image retrieval. *Pattern Recognition*, 42(2):283–292, 2009.

[18] R. da S. Torres and A. X. Falcão. Content-Based Image Retrieval: Theory and Applications. *Revista de Informática Teórica e Aplicada*, 13(2):161–185, 2006.

[19] M. G. de Carvalho, A. H. F. Laender, M. A. Gonçalves, and T. C. Porto. The impact of parameter setup on a genetic programming approach to record deduplication. In *Proceedings of the 23rd Brazilian symposium on Databases*, SBBD '08, pages 91–105, 2008.

[20] T. J. de Carvalho. Aplicação de técnicas de visão computacional e aprendizado de máquina para a detecção de exsudatos duros em imagens de fundo de olho. Master's thesis, Universidade Estadual de Campinas, 2010.

[21] C. G. do Nascimento Macario. *Anotação Semântica de Dados Geoespaciais*. PhD thesis, Universidade Estadual de Campinas, 2009.

[22] A. Dorado and E. Izquierdo. Semi-automatic image annotation using frequent keyword mining. In *IV 2003. Proceedings. Seventh International Conference on Information Visualization.*, pages 532–535, 2003.

[23] N. Elahi, R. Karlsen, and W. Younas. Image annotation by leveraging the social context. In *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication*, ICUIMC '11, pages 40:1–40:8, New York, NY, USA, 2011. ACM.

[24] J. Fan, D. A. Keim, Y. Gao, H. Luo, and Z. Li. JustClick: Personalized Image Recommendation via Exploratory Search From Large-Scale Flickr Images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(2):273–288, Dec. 2008.

[25] W. Fan, E. A. Fox, P. Pathak, H. Wu, and F. E. Al. The effects of fitness functions on genetic programming-based ranking discovery for web search. *Journal of the American Society for Information Science and Technology*, 55:2004, 2004.

[26] F. A. Faria. Uso de técnicas de aprendizagem para classificação e recuperação de imagens. Master's thesis, Universidade Estadual de Campinas, 2010.

[27] F. A. Faria, A. Veloso, H. M. Almeida, E. Valle, R. d. S. Torres, M. A. Gonçalves, and W. Meira, Jr. Learning to rank for content-based image retrieval. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pages 285–294, New York, NY, USA, 2010. ACM.

[28] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[29] C. D. Ferreira. Recuperação de imagens com realimentação de relevância baseada em programação genética. Master's thesis, Universidade Estadual de Campinas, 2010.

[30] R. B. Freitas and R. da S. Torres. OntoSAIA: Um ambiente Baseado em Ontologias para Recuperação e Anotação Semi-Automática de Imagens. *Primeiro Workshop de Bibliotecas Digitais, Simpósio Brasileiro de Banco de Dados*, pages 60–79, October 2005.

[31] J. Friedman, T. Hastie, and R. Tibshirani. *The Elements of Statistical Learning*. Springer Series in Statistics, 1 ediction, New York, NY, USA, 2001.

[32] K. Grauman and T. Darrell. The pyramid match kernel: discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465 Vol. 2, Oct. 2005.

[33] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[34] A. Hanbury. A survey of methods for image annotation. *Journal of Visual Languages and Computing*, 19(5):617–627, 2008.

[35] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1/2):177–196, 2001.

[36] E. Hörster, R. Lienhart, and M. Slaney. Continuous visual vocabulary models for plsa-based scene recognition. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 319–328, New York, NY, USA, 2008. ACM.

[37] C. Huang and Q. Liu. An orientation independent texture descriptor for image retrieval. In *International Conference on Computational Science*, pages 772–776, 2007.

[38] P. Huang, J. Bu, C. Chen, and G. Qiu. *Advances in Information Retrieval*, volume 4956 of *Lecture Notes in Computer Science*, chapter Improving Web Image Retrieval Using Image Annotations and Inference Network, pages 617–621. Springer Berlin / Heidelberg, 2008.

[39] M. J. Huiskes and M. S. Lew. The mir flickr retrieval evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.

[40] I. Ivanov, P. Vajda, L. Goldmann, J.-S. Lee, and T. Ebrahimi. Object-based tag propagation for semi-automatic annotation of images. In *Proceedings of the international conference on Multimedia information retrieval*, MIR '10, pages 497–506, New York, NY, USA, 2010. ACM.

[41] P. Janecek and P. Pu. Searching with semantics: An interactive visualization technique for exploring an annotated image collection. In *On The Move to Meaningful Internet Systems 2003: OTM 2003Workshops*, volume 2889 of *Lecture Notes in Computer Science*, pages 185–196. Springer Berlin / Heidelberg.

[42] S. Ji, L. Yuan, Y.-X. Li, Z.-H. Zhou, S. Kumar, and J. Ye. Drosophila gene expression pattern annotation using sparse features and term-term interactions. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 407–416, 2009.

[43] F. Jiang, Z. Zhao, and Y. Ge. A supervised and multivariate discretization algorithm for rough sets. In *Proceedings of the 5th international conference on Rough set and knowledge technology*, RSKT'10, pages 596–603, Berlin, Heidelberg, 2010. Springer-Verlag.

[44] S. Jiang and W. Yu. A local density approach for unsupervised feature discretization. In *Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, ADMA '09, pages 512–519, Berlin, Heidelberg, 2009. Springer-Verlag.

[45] Y. Jin, L. Khan, L. Wang, and M. Awad. Image annotations by combining multiple evidence & wordnet. In *MULTIMEDIA '05: Proceedings of the 13th annual ACM international conference on Multimedia*, pages 706–715, 2005.

[46] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 604–610 Vol. 1, Oct. 2005.

[47] F. Kang, R. Jin, and J. Y. Chai. Regularizing translation models for better automatic image annotation. In *CIKM '04: Proceedings of the 13th ACM international conference on Information and knowledge management*, pages 350–359, 2004.

[48] I. Kononenko. The minimum description length based decision tree pruning. In H.-Y. Lee and H. Motoda, editors, *PRICAI'98: Topics in Artificial Intelligence*, volume 1531 of *Lecture Notes in Computer Science*, pages 228–237. Springer Berlin / Heidelberg, 1998. 10.1007/BFb0095272.

[49] W. B. Langdon. *Data Structures and Genetic Programming: Genetic Programming + Data Structures = Automatic Programming!* Kluwer, Boston, 1998.

[50] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178, 2006.

[51] A. L. M. Levada. *Combinação de modelos de campos aleatórios markovianos para classificação contextual de imagens multiespectrais*. PhD thesis, Instituto de Física de São Carlos, 2009.

[52] L. Li and L. Fei-Fei. What, where and who? classifying events by scene and object recognition. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[53] H. Lieberman, E. Rosenzweig, and P. Singh. Aria: An agent for annotating and retrieving images. *IEEE Computer*, 34(7):57–62, 2001.

[54] S. Lindstaedt, R. Mörzinger, R. Sorschag, V. Pammer, and G. Thallinger. Automatic image annotation using visual content and folksonomies. *Multimedia Tools and Applications*, 42(1):97–113, 2009.

[55] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *KDD: International conference on Knowledge discovery and data mining*, pages 80–86, 1998.

[56] D. Liu, G. Hua, P. Viola, and T. Chen. Integrated feature selection and higher-order spatial feature extraction for object categorization. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[57] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[58] M.-C. Ludl and G. Widmer. Relative unsupervised discretization for association rule mining. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '00, pages 148–158, London, UK, 2000. Springer-Verlag.

[59] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference Computer Vision*, pages 128–142. Springer Verlag, 2002.

[60] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1-2):43–72, 2005.

[61] V. E. Ogle and M. Stonebraker. Chabot: Retrieval from relational database of images. IEEE *Computer*, 28(9):40–48, Sep 1995.

[62] O. A. B. Penatti. Estudo comparativo de descritores para recuperação de imagens por conteúdo na web. Master's thesis, Universidade Estadual de Campinas, 2009.

[63] X. Qi and Y. Han. Incorporating multiple svms for automatic image annotation. *Pattern Recognition*, 40(2):728–741, 2007.

[64] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct. 2007.

[65] M. X. Ribeiro. Mineração de dados em múltiplas tabelas fato de um data warehouse. Master's thesis, Universidade Federal de São Carlos, UFSCAR, 2004.

[66] A. Rocha and S. Goldenstein. Randomização progressiva para esteganálise. Master's thesis, Campinas, SP, Brazil, 2006.

[67] W. Romão, C. A. P. Niederauer, A. Martins, A. Tcholakian, R. C. S. Pacheco, and R. M. Barcia. Extração de regras de associação c&t: O algoritmo apriori. In *XIX Encontro Nacional em Engenharia de Produção*, Rio de Janeiro, 1999.

[68] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:2000, 2000.

[69] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, 1998.

[70] J. A. Santos. Reconhecimento semi-automaático e vetorização de regiões em imagens de sensoriamento remoto. Master's thesis, Campinas, SP, Brazil, 2009.

[71] F. Shi, J. Wang, and Z. Wang. Region-based supervised annotation for semantic image retrieval. *AEU - International Journal of Electronics and Communications*, In Press, Corrected Proof:–, 2011.

[72] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 370–377 Vol. 1, Oct. 2005.

[73] J. Sivic, B. C. Russell, A. Zisserman, W. T. Freeman, and A. A. Efros. Unsupervised discovery of visual object class hierarchies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[74] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. *Computer Vision, IEEE International Conference on*, 2:1470–1477 vol.2, April 2003.

[75] R. Stehling, M. Nascimento, and A. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *CIKM '02: Proceedings of*

ACM International Conference on Information and Knowledge Management, pages 102–109, 2002.

[76] B. Suh and B. B. Bederson. Semi-automatic photo annotation strategies using event based clustering and clothing based person recognition. *Interacting with Computers*, 19(4):524–544, 2007.

[77] M. Swain and D. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[78] J. Tang, R. Hong, S. Yan, T.-S. Chua, G.-J. Qi, and R. Jain. Image annotation by knn-sparse graph-based label propagation over noisily tagged web images. *ACM Transactions on Intelligent Systems and Technology*, 2:14:1–14:15, February 2011.

[79] B. Tao and B. Dickinson. Texture recognition and image retrieval using gradient indexing. *Journal of Visual Communication and Image Representation*, 11(3):327–342, 2000.

[80] P. Tirilly, V. Claveau, and P. Gros. Language modeling for bag-of-visual words image categorization. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 249–258, 2008.

[81] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[82] B. Tomasik, P. Thiha, and D. Turnbull. Tagging products using image classification. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 792–793, 2009.

[83] A. Ulges, M. Worring, and T. Breuel. Learning visual contexts for image annotation from flickr groups. *Multimedia, IEEE Transactions on*, 13(2):330 –341, april 2011.

[84] M. Uschold and M. Grüninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.

[85] K. van de Sande, T. Gevers, and C. Snoek. Evaluation of color descriptors for object and scene recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[86] A. Veloso. *Demand-Driven Associative Classification*. PhD thesis, Federal University of Minas Gerais, 2009.

[87] A. Veloso, W. Meira Jr., and M. J. Zaki. Lazy associative classification. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 645–654, Washington, DC, USA, 2006.

[88] A. A. Veloso, H. M. Almeida, M. A. Gonçalves, and W. Meira Jr. Learning to rank at query-time using association rules. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, Special Interest Group on Information Retrieval '08, pages 267–274, New York, NY, USA, 2008. ACM.

[89] M. Wang, X. Zhou, and T. Chua. Automatic image annotation via local multi-label classification. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 17–26, New York, NY, USA, 2008. ACM.

[90] Y. Wang and S. Gong. Refining image annotation using contextual relations between words. In *CIVR '07: Proceedings of the 6th ACM international conference onImage and video retrieval*, pages 425–432, 2007.

[91] Y. Wang, T. Mei, S. Gong, and X.-S. Hua. Combining global, regional and contextual features for automatic image annotation. *Pattern Recognition*, 42(2):259–266, 2009.

[92] P. Wennerberg, K. Schulz, and P. Buitelaar. Ontology modularization to improve semantic medical image annotation. *Journal of Biomedical Informatics*, 44(1):155 – 162, 2011. Ontologies for Clinical and Translational Research.

[93] A. Williams and P. Yoon. Content-based image retrieval using joint correlograms. *Multimedia Tools and Applications*, 34(2):239–248, 2007.

[94] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1800–1807 Vol. 2, Oct. 2005.

[95] R. Yan, A. Natsev, and M. Campbell. An efficient manual image annotation approach based on tagging and browsing. In *MS '07: Workshop on multimedia information retrieval on The many faces of multimedia semantics*, pages 13–20, 2007.

[96] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206, 2007.

[97] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[98] S. Zhang, B. Li, and X. Xue. Semi-automatic dynamic auxiliary-tag-aided image annotation. *Pattern Recognition*, 43:470–477, February 2010.

[99] W. Zhang, A. Surve, X. Fern, and T. Dietterich. Learning non-redundant codebooks for classifying complex objects. In *ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1241–1248, 2009.

[100] Y. Zhao, Y. Zhao, and Z. Zhu. Tsvm-hmm: Transductive svm based hidden markov model for automatic image annotation. *Expert Systems with Applications*, 36(6):9813–9818, 2009.

[101] N. Zhou, Y. Shen, J. Peng, X. Feng, and J. Fan. Leveraging auxiliary text terms for automatic image annotation. In *Proceedings of the 20th international conference companion on World wide web*, WWW '11, pages 175–176, New York, NY, USA, 2011. ACM.

[102] X. Zhou, M. Wang, Q. Zhang, J. Zhang, and B. Shi. Automatic image annotation by an iterative approach: incorporating keyword correlations and region matching. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 25–32, 2007.