
Combinando conteúdo e contexto para mineração de
imagens em serviços de mídia social

Alceu Ferraz Costa

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Combinando conteúdo e contexto para mineração de imagens em serviços de mídia social

Alceu Ferraz Costa

Orientadora: *Profa. Dra. Agma Juci Machado Traina*

Monografia apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, para o Exame de Qualificação, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional.

USP – São Carlos
Agosto de 2013

Resumo

Serviços de mídia social como Flickr, Twitter e Facebook permitem aos usuários compartilharem conteúdo multimídia, gerando um grande volume de imagens. Essas imagens apresentam dados de contexto, incluindo informações textuais, coordenadas geográficas e características sociais como o grafo formado pelas relações de amigos dos usuários. Por este motivo, serviços de mídia social constituem uma valiosa fonte a partir da qual pode-se minerar imagens combinando conteúdo e contexto. Contudo, a combinação de conteúdo e contexto não é trivial, e apresenta desafios a serem solucionados. Por exemplo, o processo de combinação pode agravar o problema da maldição da dimensionalidade ou não ser capaz de encontrar correlações entre conteúdo e contexto. Este projeto de doutorado propõe tratar dos desafios da combinação de conteúdo e contexto empregando a teoria de mineração de *itemsets* frequentes. A proposta de pesquisa é motivada por trabalhos recentes, que têm empregado a mineração de *itemsets* para encontrar padrões entre características visuais. Esse projeto visa estender essa abordagem para encontrar correlações não somente entre características visuais, mas também entre dados de conteúdo e contexto. Para tanto, propõe-se o desenvolvimento de uma nova abordagem de representação de dados de conteúdo e de contexto por meio de *itemsets*. Os *itemsets* gerados serão utilizados para encontrar padrões e regras de associação que serão aplicadas para minerar imagens de serviços de mídia social.

Abstract

Social media services such as Flickr, Twitter and Facebook allow users to share multimedia content, generating large volume of images. These images present context data, including textual information, geographic coordinates and social features such as the graph of users' friendship relationships. For this reason, social media services constitute a valuable source from which we can mine images combining content and context. However, content and context combination is not trivial, presenting challenges to be solved. For example, content and context combination may aggravate the curse of dimensionality or not being able to detect correlations between content and context. This research project proposes to address the challenges of combining content and context using the theory of frequent itemset mining. The proposed research is motivated by recent studies that have employed itemset mining to find patterns in visual features. This project aims to extend this approach to find correlations not only between visual features, but also between content and context. For this purpose, we propose the development of a new itemset image representation, based on content and context data. The generated itemsets will be used to find patterns and association rules that will be applied to mine social media services images.

Sumário

Lista de Figuras	vi
Lista de Tabelas	vii
1 Introdução	1
1.1 Objetivos	2
1.2 Organização do documento	2
2 Representações de imagens: contexto e conteúdo	3
2.1 Representações de conteúdo	3
2.1.1 Cor	4
2.1.2 Textura	6
2.1.3 Forma	7
2.1.4 Extratores locais	8
2.2 Representações de contexto	10
2.2.1 Dados textuais de contexto	10
2.2.2 Grafo social	12
2.3 Combinando conteúdo e contexto	13
2.3.1 Combinação <i>a priori</i>	14
2.3.2 Combinação <i>a posteriori</i>	15
2.3.3 Expansão e reordenamento de consulta	16
2.4 Considerações finais	16
3 Mineração de imagens	19
3.1 Mineração de <i>itemsets</i> frequentes	20
3.1.1 Regras de associação	21
3.1.2 O Algoritmo <i>Apriori</i>	22
3.1.3 Medidas de interesse para regras de associação	24
3.1.4 Minerando <i>itemsets</i> frequentes em grandes bases de dados	25
3.1.5 Classificadores Associativos	27

3.1.6	Mineração de imagens utilizando <i>itemsets</i> frequentes	28
3.2	Mineração de imagens em serviços de mídia social	31
3.2.1	Deteção de eventos e pontos turísticos	31
3.2.2	Classificação de imagens	31
3.2.3	Anotação automática de imagens	32
3.3	Considerações	33
4	Resultados preliminares	35
4.1	MFS-Map	35
4.1.1	Itens a partir do conteúdo	36
4.1.2	Itens a partir de dados contexto	37
4.2	Geração das regras	37
4.3	Anotação das imagens	38
4.4	Experimentos	39
4.4.1	Extração de características	40
4.4.2	Métodos de anotação	40
4.4.3	Precisão da anotação	41
4.4.4	Tempo de treinamento e anotação	42
4.5	Considerações	43
5	Proposta de pesquisa	45
5.1	Objetivos	45
5.2	Metodologia	46
5.3	Representações por meio de <i>itemsets</i>	46
5.4	Mineração de <i>itemsets</i> frequentes e aplicação na mineração de imagens	47
5.5	Estágio e pesquisa no exterior	47
5.6	Validação e conjuntos de dados	48
5.7	Cronograma	49
	Referências Bibliográficas	51

Lista de Figuras

2.1	Abordagens de extração de histograma de cores para imagens no espaço RGB.	5
2.2	Em muitas aplicações a textura tem uma forte conexão com a semântica.	6
2.3	Objetos que podem ser reconhecidos somente por sua forma.	7
2.4	Etapas do processo de extração do histograma de palavras.	8
2.5	Dados textuais de contexto nos serviços de mídia social (a) Flickr e (b) Twitter.	10
2.6	Diagrama hierárquico das estratégias de combinação de contexto e conteúdo.	14
3.1	As etapas do processo de mineração de imagens contemplando o conteúdo e contexto. . . .	20
3.2	Descarte de <i>itemsets</i> pelo princípio apriori.	23
4.1	<i>Itemsets</i> para características visuais no método <i>MFS-Map</i>	36
4.2	Entrada e saída do algoritmo de geração de regras de associação.	38
4.3	Exemplo de imagem do conjunto <i>MIR Flickr</i>	39
4.4	Precisão média (AP) obtida para cada anotação individualmente para o conjunto <i>MIR Flickr</i>	42
4.5	Precisão média (AP) contra o tempo de treinamento para cada método de anotação.	43
5.1	Exemplos de imagens obtidas por meio do <i>crawler</i> desenvolvido.	49
5.2	Cronograma de atividades planejadas e desenvolvidas.	49

Lista de Tabelas

2.1	Tamanho do dicionário de palavras visuais adotado por diferentes autores.	9
3.1	Exemplo de dados organizados na forma de transações.	21
3.2	Comparação entre métodos de classificação associativa.	29
3.3	Comparação dos métodos de representação de imagens por <i>itemsets</i> discutidos na seção 3.1.6.	30
4.1	Métodos de extração de características empregados no experimento.	40
4.2	Valores de AP e BEP obtidos por cada método para diferentes configurações.	41

Capítulo 1

Introdução

Serviços de mídia social como Facebook, Google+, Flickr e Youtube permitem aos usuários criar, compartilhar e interagir com conteúdo multimídia [69] gerando um grande volume de imagens. Por exemplo, em outubro de 2012 o Facebook armazenava uma coleção de mais de 219 bilhões de fotos compartilhadas por seus usuários ¹. Essas características fazem dos serviços de mídia social uma valiosa fonte a partir da qual se pode realizar mineração de imagens. A informação minerada pode ser aplicada em várias tarefas, incluindo: anotação automática de imagens [79, 90], detecção de fotografias relacionadas a eventos [9, 90, 80] e recuperação de imagens [20].

A mineração de imagens tem como objetivo descobrir informações úteis de bases de dados de imagens empregando técnicas de diferentes disciplinas, como mineração de dados e visão computacional[46]. A mineração de imagens nos serviços de mídia social pode ser realizada a partir do conteúdo e do contexto das imagens [20, 83]. O conteúdo corresponde às informações visuais da imagem. Métodos de mineração de imagem empregam algoritmos de extração de características para capturar os diferentes aspectos visuais de uma imagem [28]. Por exemplo, histogramas RGB ou HSV são utilizados para capturar informações de cores. O algoritmo de extração GIST [72] captura a estrutura espacial da imagem. Características locais podem ser representadas por meio da abordagem *bag-of-visual-words* utilizando o extrator SIFT [60, 91].

Os dados de contexto de uma imagem variam de acordo com a aplicação e podem ser, por exemplo, descrições textuais, coordenadas geográficas fornecidas por GPS e, no caso de imagens da Web, o texto próximo à imagem. Um importante aspecto dos serviços de mídia social é a riqueza de dados de contexto disponíveis [69]. Os dados de contexto em serviços de mídia social podem incluir comentários de usuários, popularidade (estimada, por exemplo, pelo botão *like* do Facebook), número de compartilhamentos e rede social.

Apesar de sua importância, tanto os dados de conteúdo como os dados de contexto apresentam limitações do ponto de vista de mineração de imagens. Os dados de contexto apresentam ruídos devido à ambiguidade das informações fornecidas pelos usuários [56]. Adicionalmente, dados de contexto podem

¹<http://newsroom.fb.com/download-media/4227>

ser incompletos ou inexistentes para muitas imagens. Já o conteúdo visual frequentemente não corresponde a semântica da imagem. Esse problema é agravado devido ao fato das representações obtidas por métodos de extração somente serem capazes de capturar informações visuais de baixo nível, resultando em um problema conhecido como lacuna semântica (*semantic gap*) [31, 30].

Contudo, apesar de apresentarem limitações, dados de conteúdo e contexto de imagens apresentam informações complementares. Por este motivo, observa-se na literatura de mineração de imagens grande interesse no desenvolvimento de métodos capazes de combinar conteúdo e contexto de imagens [4, 120, 20, 61]. A motivação está na hipótese da correção mútua de erros: ou seja, erros provenientes da análise de dados de conteúdo podem ser corrigidos com base nas informações complementares dos dados de contexto. No entanto, a combinação de conteúdo e contexto apresenta desafios a serem tratados, incluindo a maldição da dimensionalidade (*dimensionality curse*) [53], remoção de ruídos dos dados de contexto [69] e como realizar a detecção de correlações entre dados de conteúdo e contexto [29].

1.1 Objetivos

Uma importante tarefa de mineração de dados é a mineração de *itemsets* frequentes [42]. Seu objetivo é encontrar padrões em bases de dados organizadas em conjuntos de itens, denominados de *itemsets*. Trabalhos recentes têm explorado a mineração de *itemsets* frequentes para gerar representações de imagens mais próximas da semântica da imagem [34, 84, 71, 112]. Um tópico pouco explorado, no entanto, é a combinação de dados de conteúdo e de contexto baseada na mineração de *itemsets* frequentes.

Este projeto de doutorado tem como objetivo tratar dos desafios da combinação de dados de conteúdo e contexto utilizando a teoria de mineração de *itemsets* frequentes. Para tanto, propõe-se o desenvolvimento de uma nova abordagem de representação de dados de conteúdo e contexto de imagens por meio de *itemsets*.

A abordagem a ser desenvolvida para geração dos *itemsets* de imagem deverá explorar as características particulares dos diferentes dados de contexto e de conteúdo. Assim, os itens que irão compor o *itemset* de imagem poderão ser divididos entre os itens de conteúdo e os itens de contexto. Os itens de conteúdo serão gerados por meio da análise de subespaços de característica gerados por diferentes métodos de extração. Os itens de contexto serão gerados a partir de dados de contexto como comentário, anotações de usuário e grafo social.

Os *itemsets* gerados serão utilizados para minerar *itemsets* frequentes. O objetivo é a remoção de ruídos e a detecção de padrões úteis entre os dados de contexto e conteúdo. Os padrões encontrados serão representados tanto na forma de *itemsets* frequentes como na forma de regras de associação.

1.2 Organização do documento

Este documento está organizado da seguinte forma: nos Capítulos 2 e 3 é apresentada a revisão bibliográfica das áreas de representação e mineração de imagens. O Capítulo 4 descreve os resultados preliminares obtidos na área de representação de imagens por *itemsets*. Por fim, no Capítulo 5 é apresentado o plano de pesquisa para este projeto doutorado.

Capítulo 2

Representações de imagens: contexto e conteúdo

A mineração de imagens pode ser realizada considerando tanto o conteúdo como o contexto das imagens. O conteúdo de uma imagem refere-se às suas características visuais como cores, texturas e formas. Dados de contexto podem ser mais variados e dependem da aplicação. No caso dos serviços de mídia social, dados de contexto incluem informações textuais como título, comentários e *tags* (etiquetas) [47] ou características sociais, como o grafo de amizade dos usuários [35, 81]. Neste capítulo é realizada uma revisão das técnicas existentes para extrair representações do conteúdo e contexto de imagens de serviços de mídia social. A seção 2.1 discute o processo de extração de características visuais para representar o conteúdo das imagens. Na seção 2.2 são apresentadas técnicas para analisar dados de contexto de serviços de mídia social. Por fim, na seção 2.3, é realizada uma revisão de técnicas para combinar representações de conteúdo e de contexto.

2.1 Representações de conteúdo

A mineração de imagens utiliza muitas técnicas da mineração de dados. No entanto, na mineração de dados tradicional, os conjuntos de dados estão organizados na forma de tabelas, nas quais as linhas são exemplos e as colunas são medidas de um determinado atributo. Assim, na mineração de imagens existe o desafio de se obter uma representação adequada do conteúdo visual das imagens. Para este propósito são utilizados extratores de características. As representações são, em geral, organizadas na forma de vetores que possibilitam medir a dissimilaridade entre duas imagens por meio de funções de distância. O processo de extração implica em redução de dimensionalidade, uma vez que a imagem, que pode ser vista como uma matriz de duas dimensões de níveis de cinza, apresenta um maior número de componentes que o vetor de características.

Formalmente, um vetor de características $\vec{v}_I = (v_1, v_2, \dots, v_n)$ de uma imagem I pode ser definido como um ponto no espaço \mathbb{R}^n , onde n é o número de dimensões desse vetor [97]. A imagem em níveis

de cinza I , por sua vez, é um par (P_I, \vec{I}) onde:

- P_I é um conjunto finito de pixels (pontos em \mathbb{N}^2 tal que $P_I \subset \mathbb{N}^2$), e
- $\vec{I}: P_I \rightarrow \mathbb{R}$ é uma função que associa cada pixel p em P_I a um escalar em \mathbb{R} que representa a intensidade do nível de cinza.

Tomando esta definição, uma imagem pode ser representada por uma matriz de N linhas e M colunas na qual $I(x,y)$ denota o valor do pixel p na posição (x,y) . Por fim, um extrator de características ε é definido como se segue:

Definição 2.1. *Um extrator de características ou simplesmente extrator, é uma função $\varepsilon: I \rightarrow \mathbb{R}^n$ que extrai o vetor de características \vec{v}_I da imagem I . O espaço n -dimensional dos vetores de características \vec{v}_I é denotado por \mathcal{F} e denominado espaço de características.*

É importante notar, contudo, que existem extratores para os quais a definição 2.1 não se aplica. Por exemplo, o método de extração histograma métrico [?] retorna vetores de características de dimensionalidade variável. Adicionalmente, muitos métodos de mineração e recuperação de imagens utilizam funções de distância para medir a dissimilaridade entre duas imagens. Uma função de distância $\delta: \mathbb{R}^n \times \mathbb{R}^n$ computa a dissimilaridade entre duas imagens como sendo a distância entre seus respectivos vetores de características.

Métodos de extração de características podem ser categorizados como globais ou locais. Os métodos globais analisam a imagem como um todo e podem ser caracterizados por descritores simples de cor, textura ou forma e são discutidos nas seções 2.1.1, 2.1.2 e 2.1.3. Os métodos locais capturam informações de regiões de interesse da imagem. Na seção 2.1.4 é apresentada a abordagem *Bag-of-Features* para extração de características locais.

2.1.1 Cor

A grande maioria dos extratores baseados em características de cores utilizam histogramas para analisar a frequência de valores dos pixels [55]. Desta maneira, o espaço de cores utilizado para definir os valores dos pixels na imagem tem influência no desempenho dos vetores de características. Por este motivo muitos autores utilizam espaços de cores como o HSV (*hue, saturation and value*), considerado mais próximo da percepção humana[28].

Uma imagem representada no espaço de cores RGB (*red, green e blue*) é composta por três canais: vermelho, verde e azul. Cada um desses canais pode ser visto como uma imagem em níveis de cinza, na qual o nível de cinza descreve a intensidade das cores vermelho, verde e azul. Assim, uma maneira de gerar um vetor de características consiste em concatenar histogramas calculados a partir de cada um dos canais. A Figura 2.1(a) ilustra esse processo.

Uma segunda abordagem para gerar um histograma para imagens coloridas consiste em representar o valor RGB de um pixel como um único valor hexadecimal. Por exemplo, supondo uma imagem na qual cada canal é representado por 8 bits e um dado pixel tem o seguinte valor para a tripla RGB em base decimal:

$$(R = 165, G = 138, B = 25) \tag{2.1}$$

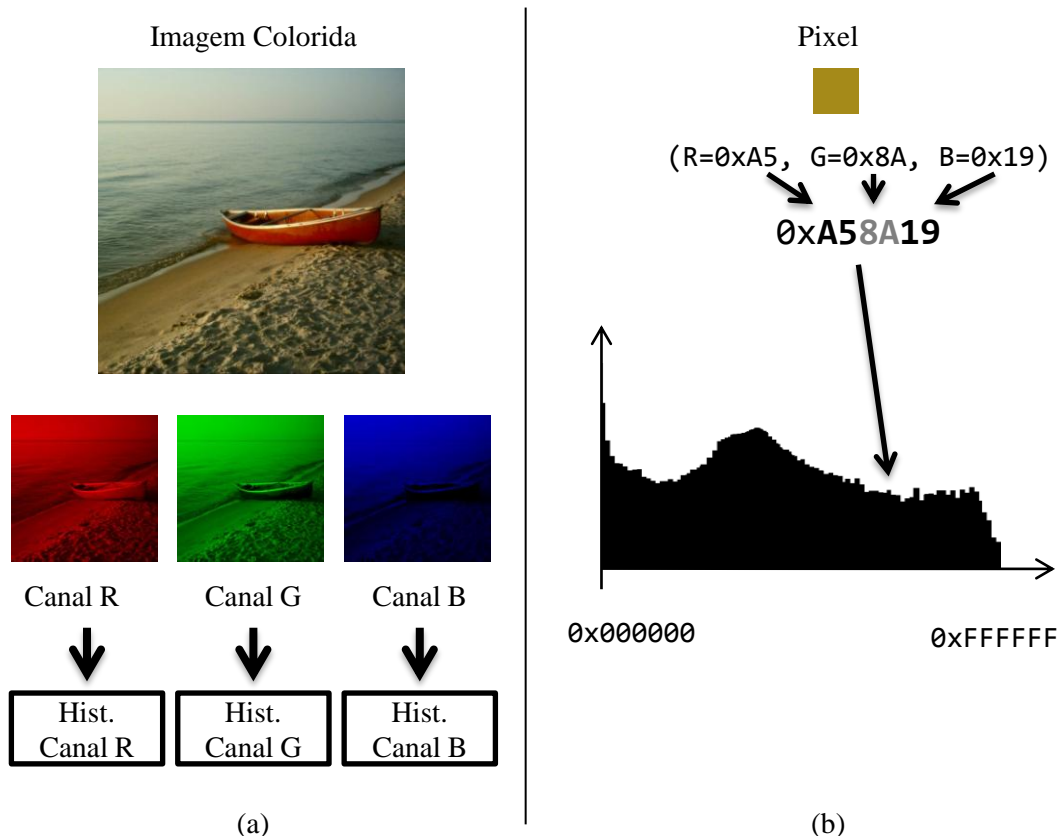


Figura 2.1: Abordagens de extração de histograma de cores para imagens no espaço RGB. (a) Vetor de características é obtido concatenando o histograma de níveis de cinza para cada canal RGB. (b) O vetor de característica corresponde ao histograma que conta frequência de valores hexadecimais utilizados para representar as triplas RGB de cada pixel.

convertendo a tripla RGB para base hexadecimal obtém-se:

$$(R = 0xA5, G = 0x8A, B = 0x19) \quad (2.2)$$

Concatenando os valores na base hexadecimal dos canais RGB obtém-se um único valor ($0xA58A19$) para representar o pixel. Desta maneira, o histograma de cores pode ser computado a partir da frequência dos valores hexadecimais concatenados como ilustrado na Figura 2.1(b). Para ambas as abordagens, a dimensionalidade do vetor de característica resultante pode ser muito alta, prejudicando seu desempenho. Uma solução para o problema consiste em aplicar requantização para reduzir o número de níveis de intensidade para cada canal de cor

Extratores de cor baseados em histogramas são eficientes pois o processo de extração consiste em uma única varredura da imagem, resultando em custo computacional linear. Essa eficiência, contudo, é acompanhada por uma importante limitação: histogramas de cores não são capazes de representar a disposição espacial de objetos na imagem. Por este motivo, extratores baseados em cores são utilizados em conjunto com outros extratores mais elaborados [104, 118].

2.1.2 Textura

Em muitas aplicações a textura tem uma forte conexão com a semântica. Por exemplo, em imagens médicas a textura pode ser utilizada para detectar a ocorrência de anormalidades indicativas de doenças pulmonares [100, 92, 24]. Na Figura 2.2(a) é mostrada a diferença entre as texturas de tecidos saudáveis e não saudáveis em uma tomografia computadorizada pulmonar (retirada de [25]). Em imagens de satélite (Figura 2.2(b)) a textura pode ser empregada para diferenciar diferentes terrenos tais como floresta, construções e mar [50].

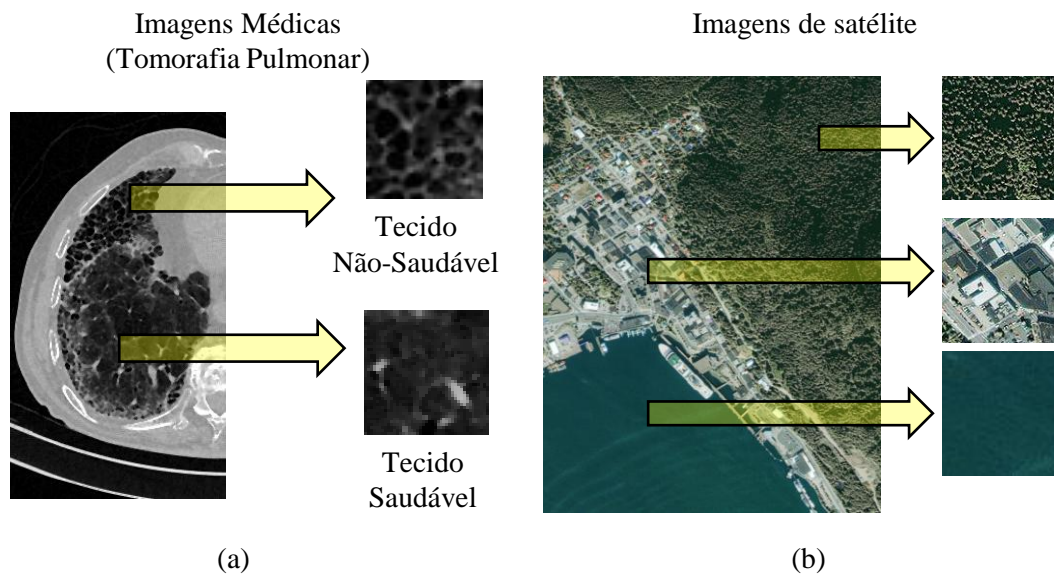


Figura 2.2: Em muitas aplicações a textura tem uma forte conexão com a semântica. (a) A textura pode ser utilizada para diferenciar tecidos saudáveis e não saudáveis em imagens de tomografias pulmonares [25]. (b) Em imagens de satélite características de textura podem ser empregadas para diferenciar diferentes terrenos.

Texturas presentes em uma imagem podem ser interpretadas como a ocorrência de padrões espaciais com propriedades de homogeneidade [40]. Assim, uma estratégia para extração de características de textura consiste em caracterizar esses padrões espaciais por meio de medidas estatísticas, tais como média, desvio padrão e entropia dos níveis de cinza [18].

Outra abordagem de caracterização de texturas consiste em utilizar matrizes de coocorrência, conhecidas como *GLCMs* (*gray level co-occurrence matrices*). As *GLCMs* contam a ocorrência de pares de níveis de cinza em diferentes deslocamentos e ângulos. Estatísticas como contraste, energia e entropia são calculadas a partir das *GLCMs* para obter vetores de características [44].

Métodos baseados em bancos de filtros também são utilizados para extração de características de textura [102, 17], sendo os filtros de Gabor um exemplo popular [63, 109]. Os kernels dos filtros de Gabor são calculados por meio de uma gaussiana modulada por uma função seno complexa. Como parâmetros para gerar o kernel dos filtros de Gabor são utilizados a orientação θ e escala σ . Diferentes valores de θ e σ permitem analisar texturas em diferentes escalas e orientações. O vetor de característica é obtido computando a energia da resposta da operação de convolução entre o filtro e a imagem.

No contexto de visão computacional, a dimensão fractal pode ser utilizada para estimar e quantificar

a complexidade da forma e textura de objetos [7, 82]. Em [25, 24] é proposto um método de extração de características denominado *FFS (Fast Fractal Stack)*. O *FFS* calcula a dimensão fractal de um conjunto de imagens binárias obtidas a partir de uma imagem em níveis de cinza. Para obter o conjunto de imagens binárias é utilizada a operação de decomposição em pilha de imagens binárias, proposta em [18].

2.1.3 Forma

Características de forma representam o contorno e interior de objetos presentes nas imagens [113]. A forma é uma característica visual importante para descrever imagens. Por exemplo, na Figura 2.3 os objetos podem ser reconhecidos somente por seu contorno ainda que não existam informações sobre cor ou textura [26].

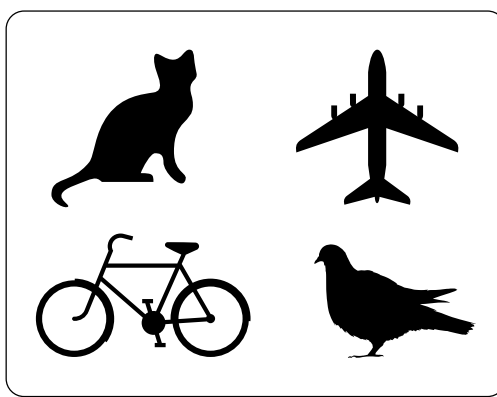


Figura 2.3: Objetos que podem ser reconhecidos somente por sua forma (fonte: *Wikimedia Commons*, <http://commons.wikimedia.org>).

O processo de extração de características de forma é usualmente precedido pela etapa de segmentação da imagem. A segmentação tem como objetivo identificar regiões da imagem que compartilhem características tais como textura e cor. Trata-se de uma etapa fundamental na qual os objetos a serem caracterizados são evidenciados [116]. Exemplos de abordagens de segmentação incluem os algoritmos baseados em limiarização [73], detecção de bordas [65, 15, 52], transformada *Watershed* [10], processos estocásticos [22] e transformada imagem floresta [32].

Uma das abordagens mais simples para caracterizar a forma de um objeto consiste em utilizar medidas relacionadas a aspectos métricos de sua forma. O total de pixels internos a um objeto segmentado pode ser utilizado para calcular sua área. Uma vez que esse valor é dado em número de pixels, pode ser necessário em determinadas aplicações realizar uma conversão da medida para as unidades originais da cena, como, por exemplo, em m^2 ou cm^2 . Outra medida que pode ser extraída do objeto segmentado é seu centróide, que consiste na média das coordenadas de todos os pixels internos ao objeto. O contorno de um objeto também pode ser utilizado para obtenção de descritores básicos de forma como o perímetro, alongamento, curvatura e eixos principais [26].

Um aspecto importante da forma de um objeto é sua irregularidade e complexidade. Extratores baseados na transformada de Fourier podem ser empregados para caracterizar a irregularidade do contorno de objetos [89, 96]. Outra abordagem utilizada para caracterizar a complexidade de formas presentes nas imagens consiste em utilizar a geometria Fractal [98, 82, 14, 6].

2.1.4 Extratores locais

Extratores de características podem ser classificados como globais ou locais [28]. Extratores globais descrevem as características gerais de uma imagem. Já extratores locais resumem características de pontos de interesse detectados nas imagens. Extratores globais tendem a ser mais eficientes que extratores locais, pois não necessitam detectar pontos de interesse ou sumarizar características. No entanto, extratores globais podem falhar na identificação de aspectos visuais que somente aparecem localmente nas imagens. Por este motivo, observa-se um interesse na representação de imagens por meio de extratores locais [99].

Um dos avanços recentes mais significativos na área de representação de imagens foi a introdução da abordagem *Bag-of-Features* (BoF) [91] para sumarização de características locais. A abordagem *Bag-of-Features* é inspirada na representação *Bag-of-Words*, amplamente utilizada na área de recuperação de informações textuais. As representações *Bag-of-Words* representam um documento textual por meio de um histograma com a frequência de ocorrência das palavras. De maneira análoga, as representações *Bag-of-Features* utilizam histogramas de palavras visuais representar imagens. A Figura 2.4 mostra as etapas do processo de extração do histograma de palavras visuais de acordo com a abordagem *Bag-of-Features*:

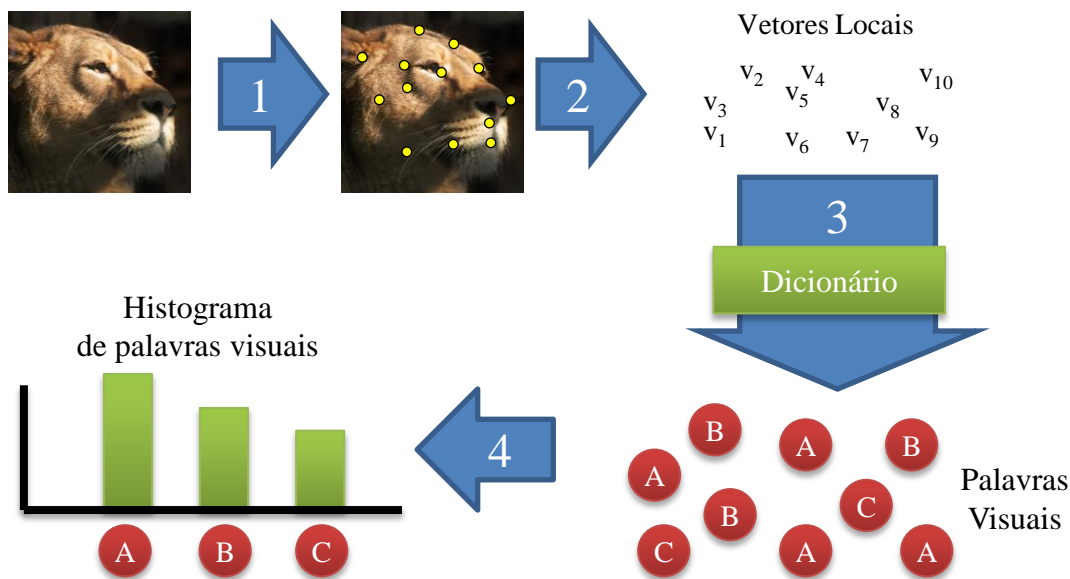


Figura 2.4: Etapas do processo de extração do histograma de palavras visuais na abordagem *Bag-of-Features*. (1) Detecção de regiões de interesse. (2) Extração de vetores de características locais. (3) Conversão dos vetores para palavras visuais. (4) Geração do histograma de palavras visuais.

1. Detecção de regiões de interesse na imagem;
2. Extração de vetores de características locais das regiões detectadas;
3. Conversão dos vetores de características locais para palavras visuais;
4. Geração de um histograma de ocorrência das palavras visuais.

Existem diferentes abordagens para a detecção de regiões de interesse [99]. Em [64] é proposta a utilização de uma grade regular para dividir a imagem em regiões a partir das quais são extraídos os

vetores locais. Devido a sua simplicidade, a utilização de grades regulares tem um baixo custo computacional. Uma outra abordagem consiste em encontrar pontos salientes que apresentem características visuais relevantes tais como bordas e cantos. Exemplos de detectores de regiões de interesse baseados em medidas de saliência incluem os filtros de diferença de gaussianas (*DoG - difference of gaussians*) [59], laplaciano de gaussiana (*LoG - laplacian of gaussian*) [57] e o detector de Harris [68].

A etapa 3 do diagrama da Figura 2.4 utiliza um dicionário visual para converter os vetores locais para palavras chaves visuais. Ou seja, para um dado ponto no espaço de características dos vetores locais, o dicionário visual retorna uma palavra chave. O processo que gera o dicionário visual pode ser visto como uma quantização do espaço de características dos vetores locais. Uma maneira de realizar essa quantização consiste em aplicar um algoritmo de agrupamento como o *k-means* nos vetores locais extraídos de diversas imagens. O objetivo do agrupamento é encontrar centróides que serão utilizados para definir regiões do espaço de características. Desta maneira, a palavra visual atribuída a um vetor de característica local será correspondente ao centróide mais próximo.

Em [87] os autores argumentam que é possível tornar a geração do dicionário visual significativamente mais eficiente substituindo o processo de agrupamento por uma simples amostragem aleatória. Ou seja, ao invés de empregar agrupamento de dados para encontrar centróides, os autores propõem que vetores locais sejam selecionados para representar regiões do espaço de características. Outro fator que pode influenciar o desempenho da abordagem *Bag-of-Features* é o tamanho do dicionário. O número de diferentes palavras visuais determinará o tamanho do histograma de frequência de palavras visuais. Diferentes autores adotam tamanhos variados de dicionários na extração de características utilizando a abordagem *Bag-of-Features*, conforme mostrado na Tabela 2.1.

Tabela 2.1: Tamanho do dicionário de palavras visuais adotado por diferentes autores para extração de características utilizando a abordagem *Bag-of-Features*.

Referência	Tam. do Dicionário
[108]	10.000
[75]	500
[104]	100 e 1.000
[112]	250
[107]	100
[83]	128
[19]	128
[117]	200 e 400

A abordagem *Bag-of-Features* tradicionalmente não considera a disposição espacial dos pontos de interesse detectados na imagem. No entanto, existem trabalhos recentes que incorporam informação espacial à abordagem *Bag-of-Features* para melhorar o desempenho das representações de imagem. Por exemplo, em [5] as coordenadas das regiões de interesse são incorporadas ao vetor de características. Em [76] o histograma de palavras visuais é substituído por um histograma de *n-grams*. *N-grams* são conjuntos de *n* palavras visuais que aparecem próximas na imagem. *N-grams* visuais têm a vantagem de representar a coocorrência de palavras visuais.

2.2 Representações de contexto

Serviços de mídia social são caracterizados por possuírem grande variedade e riqueza de dados de contexto que não estão presentes em outras aplicações como, por exemplo, em documentos da Web [69]. Esses dados de contexto adicionais podem ser explorados, por exemplo, para o desenvolvimento de novas abordagens de mineração de imagens. Nesta seção serão discutidas duas modalidades de dados de contexto que pretende-se explorar neste projeto de pesquisa. Na seção 2.2.1 são apresentadas técnicas para a análise dos dados de contexto textuais e na seção 2.2.2 é discutido o uso do grafo social como fonte de dados de contexto.

2.2.1 Dados textuais de contexto

Imagens em serviços de mídia social frequentemente são acompanhadas de informações textuais. A Figura 2.5 mostra imagens provenientes dos *websites* Flickr e Twitter. No Flickr (Figura 2.5(a)) as fotografias são acompanhadas de um título, descrição, comentários e *tags* (etiquetas). No Twitter (Figura 2.5(b)), usuários compartilham *tweets*: pequenos textos de no máximo 140 caracteres. Muitos dos *tweets* contém *URLs* de imagens, sobretudo de fotografias capturadas por meio de dispositivos móveis.

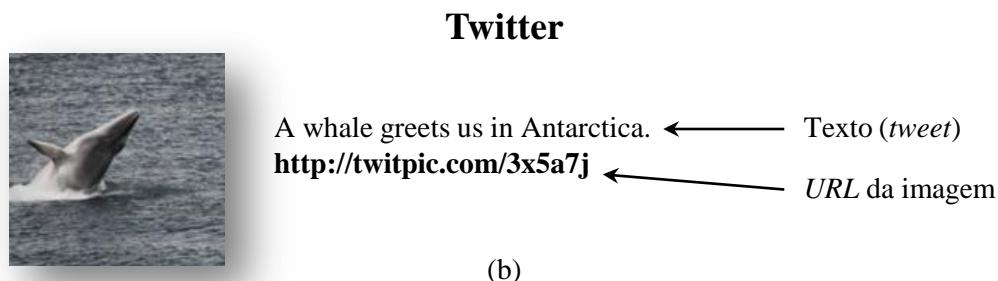
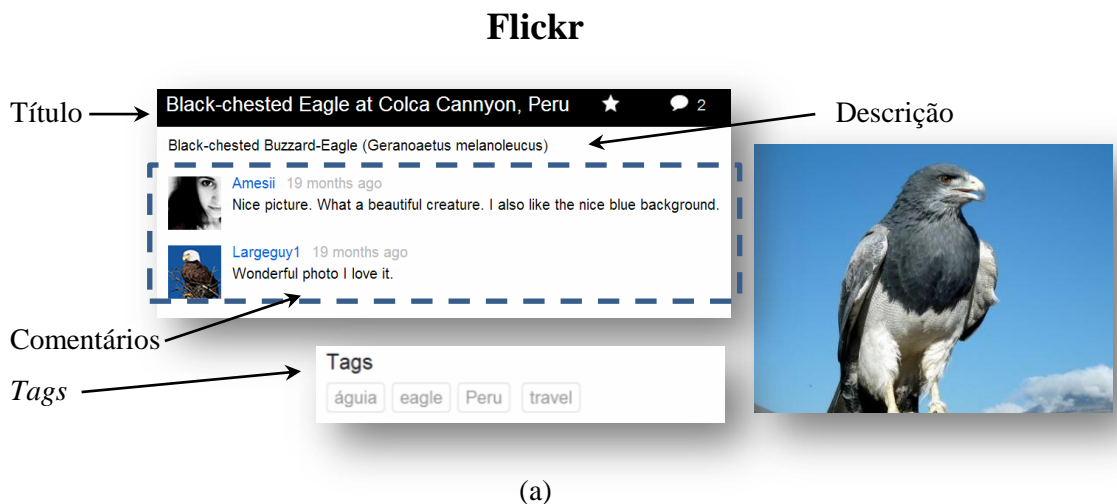


Figura 2.5: Dados textuais de contexto nos serviços de mídia social (a) Flickr e (b) Twitter.

Um modo de representar dados textuais de contexto é pela abordagem *Bag-of-Words* [47]. A abordagem *Bag-of-Words* consiste em representar um objeto textual por meio de um histograma de frequência

de termos (palavras). O histograma pode ser modelado como um vetor de características no qual cada componente corresponde a uma palavra. Uma característica desse vetor é que ele é esparsos, ou seja, muitos de seus atributos assumem o valor zero. Adicionalmente, os atributos do vetor são assimétricos, ou seja, a similaridade entre dois objetos pode ser melhor representada pela quantidade de termos compartilhados do que pelo número de termos diferentes. Assim, uma função de distância para esse tipo de dado não deveria depender dos atributos com valor zero porque dois documentos textuais, independente de serem similares, tendem a conter muitas palavras diferentes. Uma distância amplamente utilizada para comparar dois objetos de texto é a distância cosseno:

$$\cos(\vec{v}_1, \vec{v}_2) = \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1| |\vec{v}_2|} \quad (2.3)$$

onde \cdot corresponde a operação de produto escalar e $|\vec{v}_i|$ à norma do vetor de característica v_i .

O histograma *Bag-of-Words* pode ser ponderado para refletir a importância de um termo para descrever um objeto textual. Uma abordagem que pode ser utilizada para este propósito é a ponderação *TF-IDF* (*term frequency - inverse document frequency*). Sendo $TF(w)$ a frequência de um termo w em um documento, $DF(w)$ o número de objetos que contém o termo w e N o número de objetos textuais na base de dados, a estatística *TF-IDF* pode ser calculada pela seguinte equação:

$$TF-IDF = TF(w) \log \left(\frac{N}{DF(w)} \right) \quad (2.4)$$

A utilização da *TF-IDF* para ponderar as palavras na abordagem *Bag-of-Words* faz com que termos mais raros tenham um peso maior que termos frequentes entre o conjunto de documento textuais.

Uma etapa importante na extração de características textuais é o pré-processamento. Geralmente as palavras mais frequentes em uma base de dados de objetos textuais são artigos ou pronomes como “um” ou “que”, as quais possivelmente não carregam informações úteis. Assim, muitas aplicações removem essas palavras, denominadas de *stop-words* dos objetos textuais. Outra técnica de pré-processamento empregada é o *stemming*, que consiste em reduzir palavras aos seus radicais [78]. O objetivo é representar palavras correlatas por uma mesma raiz. Por exemplo, na língua portuguesa, as palavras “terra”, “terreno” e “terrinha” poderiam ser reduzidas para o radical “terr”. Um desafio para aplicação de *stemming* é que é necessário empregar um algoritmo especializado para cada língua. Ou seja, um algoritmo de *stemming* para língua inglesa pode não ser adequado para um documento escrito em português [105].

Em [35] foi avaliada a qualidade de dados textuais de serviços de mídia social. Os autores observaram que uma fração significativa dos objetos multimídia não possuíam parte dos dados de contexto textuais. Adicionalmente, foi observado que dados textuais, sobretudo *tags*, podem assumir múltiplos significados. Outro desafio é que, quando comparados com textos de jornais ou de documentos da Web, dados textuais de serviços de mídia social tendem a ser mais curtos. Como consequência, os objetos textuais frequentemente não possuem palavras o suficiente para gerar uma representação *Bag-of-Words*. Esse problema é tratado em [77] por meio da utilização de fontes de dados externas, como a Wikipedia, para complementar os dados textuais.

2.2.2 Grafo social

Uma rede social pode ser definida como um conjunto de atores (por exemplo, indivíduos ou grupos sociais) e relações de diferentes tipos como, por exemplo, amizade. Exemplos de redes sociais incluem atores que participaram em um filme ou pesquisadores que colaboraram em um artigo. Neste projeto de pesquisa o interesse são as redes sociais online de serviços de mídia social como Facebook, Flickr e Twitter. Redes sociais online fornecem aos usuários duas importantes características: a possibilidade de construção de um perfil e o gerenciamento de uma lista de contatos [13]. Esses contatos podem ser, por exemplo, amigos no caso do Facebook ou seguidores no caso do Twitter.

Redes sociais podem ser representadas como grafos sociais. Um grafo $G = (V, E)$ é composto por um conjunto de nós (vértices) V e um conjunto de arestas E . Uma maneira de gerar um grafo social a partir de uma rede social consiste em mapear indivíduos como vértices e conexões (ex. amigos) como arestas. Grafos são úteis por fornecerem um arcabouço poderoso para minerar informações de redes sociais. Por este motivo, grafos sociais são utilizados, por exemplo, para detectar comunidades [37] ou para descobrir como redes sociais evoluem com o tempo [54].

Grafos de redes sociais em geral exibem comunidades, ou seja, grupos de usuários com uma forte conexão entre si. Técnicas de detecção de comunidade podem ser usadas para encontrar usuários que compartilham um mesmo interesse ou para prever quais usuários tem maior probabilidade de juntar-se a uma comunidade no futuro [67]. Existem muitos algoritmos para detecção de comunidades em grafos [27, 70, 36]. Um exemplo é o algoritmo de Girvan-Newman [39], que é baseado na medida de centralidade *betweenness*. O *betweenness* de uma aresta é definido como se segue:

Definição 2.2. *O betweenness de uma aresta e é o número de menores caminhos entre todo o par de vértices v e u que contenha e .*

Um alto valor de *betweenness* indica que a aresta conecta duas comunidades. Desta maneira é possível encontrar comunidades removendo arestas que apresentem um valor de *betweenness* acima de um valor de limiar. Outro tipo de informação que pode ser minerada de redes sociais são as propriedades estatísticas. Por exemplo, suponha uma rede social na qual o número de conexões está correlacionado a influência dos usuários. Neste caso, a distribuição do número de conexões é uma estatística importante pois permite determinar se usuários influentes (com muitas conexões) são comuns ou raros.

A distribuição estatística dos graus dos vértices pode ser utilizada para determinar o comportamento do número de conexões dos usuários de uma rede social. O grau $d(v)$ de um vértice v corresponde ao número de arestas incidente àquele vértice. Em muitas redes sociais reais observa-se que a distribuição de graus segue uma lei potência $f(d) \propto d^{-\alpha}$, onde $f(d)$ é fração de vértices com grau d e $\alpha > 0$ [67].

Outra importante estatística do grafo social é seu diâmetro. Em uma rede social, o diâmetro mede a maior separação existente entre os indivíduos. O diâmetro de um grafo corresponde ao seu maior menor caminho e pode ser definido como se segue:

Definição 2.3. *O diâmetro de um grafo é o maior inteiro δ , onde δ , para todos os pares de nós u e v , é o menor número de arestas que precisam ser percorridas no caminho de u para v .*

Intuitivamente, o diâmetro fornece um limite superior de quantas arestas precisam ser percorridas para atingir um vértice do grafo. Uma das mais importantes características das redes sociais é a ocorrência de um pequeno diâmetro. Essa característica é conhecida como o fenômeno de mundo pequeno

(*small-world phenomeon*) [111, 110]. Por exemplo, Ugander et al. observaram que apesar do grande tamanho do grafo social do Facebook (712 milhões de usuários) 99,6% dos vértices são separados por apenas seis conexões [101].

Assim como ocorre para o Facebook, grafos sociais de serviços de mídia social em geral são grandes. Acomodar grafos desta escala em memória ou mesmo em um único disco pode ser caro ou mesmo impraticável [21]. Assim, paralelismo é uma ferramenta promissora para minerar grandes grafos [51]. O *map-reduce* é um framework de programação que provê uma solução conveniente para resolver problemas paralelizáveis utilizando grandes coleções de máquinas comuns (*commodity hardware*). No *map-reduce* os usuários precisam especificar duas funções: *map* e *reduce* e o sistema gerencia automaticamente falhas, comunicação em rede e balanço de carga.

Implementações do *map-reduce* são realizadas utilizando como base sistemas de arquivos distribuídos (*DFS - distributed file systems*). Os arquivos são divididos em múltiplos *chunks* (“pedaços”) que são replicados em diferentes *compute nodes* (nós computacionais). A replicação dos dados protege o sistema contra falhas que ocorrem com frequência devido à utilização de uma grande quantidade de máquinas comuns. Adicionalmente, ao separar grandes arquivos em múltiplos *chunks*, diferentes partes do arquivo podem ser lidas em paralelo, aumentando o *throughput* de entrada e saída. O primeiro *DFS* proposto foi Google File System (GFS) [38]. Hadoop, que é uma implementação popular e *open-source* do *map-reduce* utiliza o Hadoop Distributed File System (HDFS).

Um programa que utiliza o *map-reduce* precisa implementar ao menos as funções *map* e *reduce*. A função *map* toma como entrada uma coleção de objetos e converte cada um em zero ou mais pares chave-valor. Considere como exemplo um programa que conta a frequência de palavras em um grande arquivo de texto contido em um *DFS*. Neste caso, os objetos seriam os *chunks* do arquivo de texto e a saída pares no quais as chaves são palavras e os valores a frequência da respectiva palavra.

A saída de todas as chamadas à função *map* é particionada em um conjunto de listas. Cada lista contém os valores associados a uma dada chave. No exemplo considerado, a saída das chamadas à função *map* é particionada em uma lista para cada palavra encontrada. Cada lista contém os valores de contagem retornados por múltiplas chamadas à função *map*. Finalmente, a função *reduce* toma como entrada uma lista associada a uma chave. Os valores dessa lista são agregados, resultando em um único par de chave e valor. Para o exemplo de contagem de palavras, a função *reduce* somaria todos os valores da lista e retornaria um par no qual a chave é uma palavra e o valor a contagem da palavra. O resultado de todas as chamadas às funções *reduce* corresponde a saída do processo de *map-reduce*.

Nem todos os algoritmos paralelizáveis podem ser implementados utilizando *map-reduce*. No entanto, operações de multiplicação de matrizes podem ser implementadas eficientemente no framework de programação proposto pelo *map-reduce*. Uma vez que muitos algoritmos de mineração de grafos como *page-rank*, cálculo de diâmetro e de componentes conexos são baseados em operações de multiplicação de matrizes e vetores o framework *map-reduce* têm sido empregado para a análise de grandes grafos [51, 21].

2.3 Combinando conteúdo e contexto

Tanto conteúdo como o contexto apresentam limitações do ponto de vista de mineração de imagens. Como discutido na Seção 2.1, o conteúdo das imagens é representado por meio de características visuais de baixo nível tais como cor, textura e forma. Essas características visuais de baixo nível e a percep-

ção dos usuários são geralmente discrepantes do ponto de vista de semântica, resultando no problema conhecido como lacuna semântica (*semantic gap*) [119, 30]. Mesmo com importantes avanços observados recentemente na área de extração de características, que incluem as representações por meio de *Bag-of-Features* [60, 91], o problema da lacuna semântica continua em aberto [114]. Dados de contexto como legendas, comentários e tags podem conter informações mais próximas da semântica da imagem. No entanto, os dados de contexto também apresentam limitações. Informações fornecidas por usuários podem ser subjetivas e incompletas [69] introduzindo ruído no processo de mineração. Esse problema pode ser agravado quando se considera a existência de *spam*.

Apesar de apresentarem limitações, conteúdo e contexto contém informações complementares [4, 29]. Assim, a combinação de ambas as modalidades de informação tem o potencial de aprimorar o desempenho das técnicas de mineração de imagens sob a hipótese da correção mútua de erros [29]. Ou seja, os erros provenientes da análise dos dados de contexto podem ser corrigidos com base nas informações complementares presentes no conteúdo. Desta maneira, observa-se na literatura de mineração e recuperação de imagens diferentes abordagens para combinar conteúdo e contexto. A Figura 2.6, adaptada de [29], mostra um diagrama hierárquico de estratégias para combinação de conteúdo e contexto que serão discutidas nas seções que se seguem. A categorização apresentada no diagrama é baseada nas categorizações apresentadas em [4, 120, 20, 61].

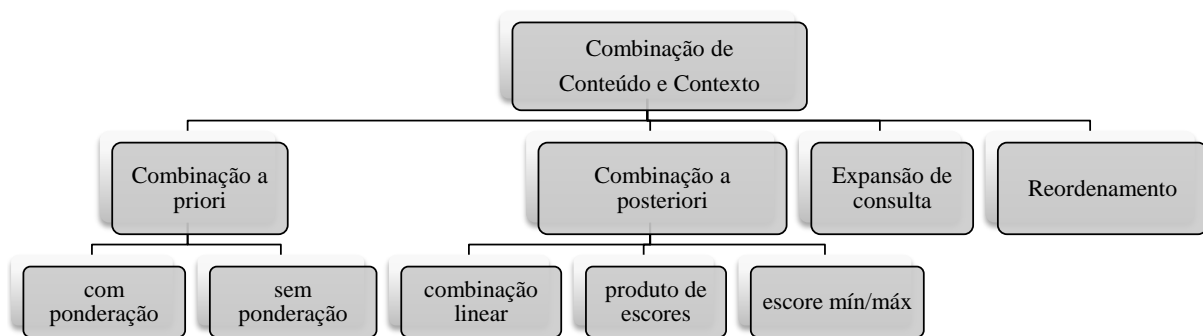


Figura 2.6: Diagrama hierárquico das estratégias de combinação de contexto e conteúdo em mineração de imagens. Adaptado de [29].

2.3.1 Combinação *a priori*

Possivelmente, a estratégia *a priori* (*early fusion*) é a mais simples para combinar conteúdo e contexto¹. Na combinação *a priori* os vetores de características que representam o conteúdo e o contexto da imagem são concatenados em um único grande vetor de características. A combinação *a priori* resulta em um único vetor de características para representar o conteúdo e o contexto de uma imagem. Ou seja, as imagens são representadas em um único espaço de características.

A representação das imagens em um único espaço de características pode ser vantajosa. Por exemplo, na tarefa de classificação de imagens, um único classificador pode ser utilizado. No entanto, a concatenação dos vetores de características de conteúdo e contexto pode gerar um espaço de características com uma dimensionalidade excessivamente alta, agravando o problema da maldição da dimensionalidade [53].

¹A estratégia *a priori* é também denominada de *attribute level fusion* [4].

A combinação *a priori* pode ser realizada com ou sem ponderação de características [33]. Em [104], Verbeek et al. propõem um método de anotação de imagens denominado *TagProp* que utiliza combinação *a priori* com ponderação. Para representação do conteúdo foram utilizadas 15 vetores de características, incluindo histograma RGB, GIST [72] e SIFT [60, 91]. O contexto das imagens foi representado por um histograma de frequência de tags de usuários. Os pesos para cada vetor de característica e histograma de frequência de tags são computados maximizando a log-verossimilhança de predizer as anotações corretas para um conjunto de treinamento de imagens utilizando o método do gradiente projetado (*projected-gradient*).

2.3.2 Combinação *a posteriori*

Na combinação *a posteriori* (*late fusion* ou *decision level fusion*) cada modalidade de informação é analisada separadamente e as decisões retornadas são combinadas para chegar a uma decisão final. A análise das modalidades de informação e a combinação das decisões são dependentes da tarefa de mineração de imagens. Por exemplo, na tarefa de classificação de imagens, a análise e combinação das informações poderia ser realizada da seguinte maneira:

Análise: A etapa de análise consiste em submeter as características de conteúdo e de contexto a classificadores distintos. Cada classificador retornará a probabilidade predita de que a imagem pertença a cada uma das classes;

Combinação das decisões: Neste caso, as decisões tomadas na etapa de análise são as probabilidades retornadas pelos classificadores. A combinação poderia ser realizada, por exemplo, por meio da média da probabilidade de cada classificador.

Uma vez que as etapas de análise e combinação das decisões intermediárias são dependentes da tarefa de mineração de imagens, as estratégias de combinação *a posteriori* são mais variadas do que as de combinação *a priori*. As estratégias de combinação *a posteriori* podem ser divididas em três categorias: combinação linear de *scores*, produto de *scores* e *score* mínimo ou máximo (Figura 2.6). Neste caso o *score* é um valor que sintetiza o resultado da análise de cada modalidade de informação e pode ser, por exemplo, a probabilidade de classe para um problema de classificação ou a relevância para um problema de recuperação de imagens.

Na estratégia de combinação linear, os *scores* retornados na etapa de decisão para cada modalidade m_i são combinados de acordo com a seguinte expressão:

$$S_{\text{Comb}} = a_1 S(m_1) + a_2 S(m_2) + \dots + a_N S(m_N), \quad \sum_{i=1}^N a_i = 1 \quad (2.5)$$

onde $S(m_i)$ é o *score* retornado para a modalidade m_i e S_{Comb} é o *score* combinado. Por exemplo, para o caso no qual temos uma modalidade de contexto e uma modalidade de conteúdo, a equação 2.5 pode ser reescrita como:

$$S_{\text{Comb}} = a S(m_{\text{conteúdo}}) + (1 - a) S(m_{\text{contexto}}) \quad (2.6)$$

Em [120], a estratégia de combinação linear é utilizada para a tarefa de anotação de imagens. Para predizer a relevância de uma anotação para uma imagem, três classificadores SVM são treinados: um

para as características de conteúdo da imagem e dois para os dados de contexto. A saída de cada classificador SVM é um *score* que prediz a relevância da anotação para a imagem. Os *scores* retornados pelos três classificadores são combinados de acordo com a Equação 2.5. Para determinar os pesos da combinação linear os autores treinam um SVM linear utilizando os *scores* como atributos.

Outra estratégia de combinação *a posteriori* consiste em combinar os *scores* de decisão resultantes da etapa de análise por meio de um produtório:

$$S_{\text{Comb}} = \prod_{i=1}^N S(m_i) \quad (2.7)$$

Essa estratégia foi adotada em [66] para a tarefa de anotação de imagens. A utilização do produto entre *scores* faz com que anotações sejam consideradas relevantes para uma imagem somente se um alto *score* foi obtido tanto para o contexto como para o conteúdo. De maneira oposta, as estratégias de *score* mínimo ou máximo realizam a etapa de combinação de decisões escolhendo somente um dos *scores* obtidos [106].

2.3.3 Expansão e reordenamento de consulta

Duas outras estratégias de combinação de conteúdo e contexto voltadas para a recuperação de imagens são o reordenamento e expansão de consulta. Na estratégia de reordenamento (*re-ranking*) [20], uma consulta por termos textuais é realizada e as imagens que apresentem um contexto relevante são retornadas. As imagens retornadas estão inicialmente ordenadas de acordo com a relevância de contexto em relação aos termos de consulta. O processo de reordenamento toma como entrada o ordenamento inicial e o modifica utilizando o conteúdo das imagens como informação adicional para aprimorar o resultado da consulta.

No processo de reordenamento a consulta por termos textuais é utilizada como um filtro. Ou seja, dado o conjunto completo de imagens, somente as N imagens com o contexto mais relevante são retornadas. Assim, somente o conteúdo de um pequeno subconjunto de imagens precisa ser analisado durante a fase de reordenamento. Esse é um comportamento desejável porque realizar consultas por termos textuais podem ser processadas de maneira escalável e são empregadas por sistemas de busca como Google e Bing. Já consultas por conteúdo envolvem cálculos de distância em espaços de alta dimensionalidade, o que pode ser indesejável devido ao problema da maldição da dimensionalidade.

A estratégia de expansão de consulta [19] consiste em, inicialmente, obter um conjunto das N imagens visualmente mais similares a uma imagem de consulta. O contexto das N imagens retornadas é analisado para encontrar termos textuais relevantes. Esses termos textuais são utilizados para realizar novas consultas, expandindo a consulta inicial.

2.4 Considerações finais

Conforme discutido na seção 2.3, tanto representações de conteúdo como de contexto apresentam limitações do ponto de vista de mineração de imagens. Para o conteúdo existe o problema da lacuna semântica, que refere-se ao fato de características visuais de baixo nível não serem capazes de descrever a semântica da imagem. Já os dados de contexto podem ser ambíguos ou incompletos. Por este motivo, existe grande interesse no desenvolvimento de métodos para combinar dados de conteúdo e de contexto. De

fato, resultados empíricos indicam que a combinação de conteúdo e contexto contribui para a melhoria do desempenho da mineração de imagens [29].

Contudo a combinação de conteúdo e contexto ainda apresenta desafios em aberto para serem tratados. Por exemplo, na abordagem *a priori* (seção 2.3.1), a combinação de conteúdo e contexto pode gerar espaços de características de alta dimensionalidade, contribuindo para que o problema da maldição da dimensionalidade seja agravado. Para as abordagens *a posteriori*, existe o desafio de se determinar a contribuição de cada modalidade (conteúdo e contexto) na análise das imagens. Por exemplo, na estratégia de combinação linear (seção 2.3.2) não existe um consenso sobre como definir de maneira adequada a ponderação para cada modalidade de informação [29].

Capítulo 3

Mineração de imagens

A mineração de imagens tem como objetivo descobrir conhecimento, padrões e relações não explicitamente armazenados em bases de imagens [46]. A mineração de imagens emprega técnicas de diferentes disciplinas como mineração de dados, bancos de dados, recuperação de informação e visão computacional [86, 107, 9]. Na mineração de dados tradicional os dados estão organizados na forma de tabelas, nas quais as linhas são exemplos e as colunas a medida de um atributo. Já na mineração de imagens existe o desafio de extrair representações das imagens, uma vez que os valores de pixels individuais não têm significado semântico útil para a mineração de dados. Por este motivo, o processo de mineração de imagens considerando o conteúdo e contexto pode ser dividido em três etapas:

1. Extração de características: nesta etapa são extraídos os vetores de características que representam o conteúdo das imagens;
2. Combinação de conteúdo e contexto: nesta etapa os vetores de características são integrados aos dados de contexto da imagem utilizando as técnicas discutidas na seção 2.3;
3. Mineração: nesta etapa são aplicados algoritmos de mineração de dados para extrair conhecimento da base de imagens.

Conforme ilustrado no diagrama da Figura 3.1, uma vez combinado o conteúdo e contexto das imagens é realizada a mineração de dados. Segundo [74], as tarefas de mineração de dados podem ser categorizadas como sendo preditivas ou descritivas. As tarefas preditivas têm como objetivo prever o valor de um determinado atributo alvo com base no valor dos demais atributos. Já as tarefas descritivas, procuram extrair padrões tais como correlações, tendências e agrupamentos que podem ser utilizados para descrever os dados sendo analisados. Algumas das principais tarefas de mineração de dados são:

Classificação: Trata-se de uma tarefa de predição utilizada para atributos alvo do tipo discreto. A classificação pode ser utilizada, por exemplo, para se prever o diagnóstico de um paciente.

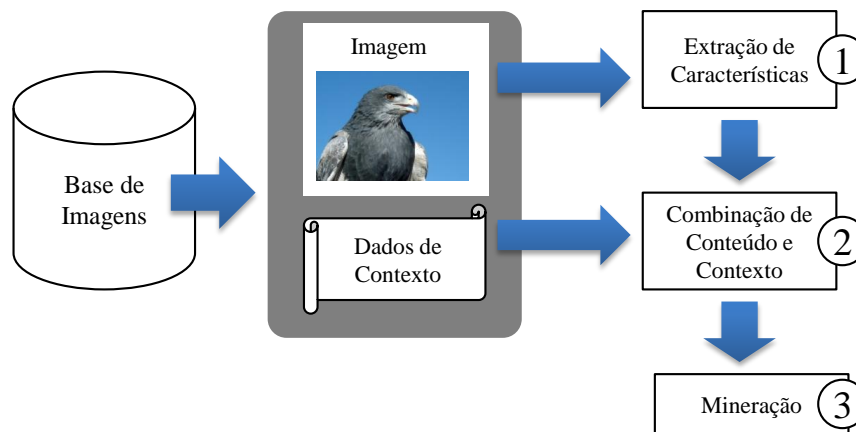


Figura 3.1: As etapas do processo de mineração de imagens contemplando o conteúdo e contexto.

Regressão: Assim como a classificação, a regressão é uma tarefa preditiva. No entanto, o atributo alvo é do tipo contínuo como, por exemplo, na previsão da temperatura em um determinado local e dia.

Análise de Agrupamentos: O objetivo desta tarefa é encontrar grupos de objetos de modo que aqueles que pertençam a um mesmo grupo são mais similares entre si que objetos que pertençam a grupos diferentes [49]. A análise de agrupamentos pode ser utilizada, por exemplo, para encontrar grupos de clientes que apresentem comportamentos similares.

Outra importante tarefa de mineração de dados é a mineração de itemsets frequentes. Seu objetivo é encontrar padrões em bases de dados organizadas na forma de transações (cestos de itens). Neste projeto de doutorado pretende-se explorar a mineração de itemsets frequentes para realizar a mineração de imagens. Assim, na seção que se segue será discutida a tarefa de mineração de itemsets frequentes.

3.1 Mineração de *itemsets* frequentes

Redes de supermercado coletam diariamente dados na forma de transações que correspondem a itens de uma compra em uma de suas lojas. A Tabela 3.1 ilustra a organização deste tipo de dado. Cada linha corresponde a uma transação que possui um identificador denominado *TID* (*Transaction Identifier*) e um conjunto de itens comprado por um cliente denominado *itemset*. A mineração de *itemsets* frequentes foi proposta inicialmente em [1] com o propósito de analisar e descobrir itens que aparecem com frequência em uma mesma compra. Por exemplo, na Tabela 3.1 em três das cinco compras os itens fralda e cerveja aparecem juntos. Neste caso, a mineração de *itemsets* frequentes poderia ser utilizada para evidenciar esse padrão. Desta maneira, encontrar *itemsets* frequentes permite a um lojista descobrir quais produtos são comumente comprados em conjunto. A mineração de *itemsets* frequentes, no entanto, não é limitada a análise de compras de supermercado, podendo ser aplicada no domínio de mineração de imagens (seção 3.1.6).

Um conceito importante na análise de associações é o de *itemset* (conjunto de itens). Sendo $\mathcal{I} = \{i_1, i_2, \dots, i_d\}$ o conjunto de todos os itens que podem ocorrer em uma transação, um *itemset* é definido como um conjunto de zero ou mais itens de \mathcal{I} , sendo que um *itemset* que possua k itens é denominado

Tabela 3.1: Exemplo de dados organizados na forma de transações.

TID	Transação
1	{Pão, Leite}
2	{Pão, Fraldas, Cerveja, Ovos}
3	{Leite, Fraldas, Cerveja, Refrigerante}
4	{Pão, Leite, Fraldas, Cerveja}
5	{Pão, Leite, Fraldas, Refrigerante}

k-itemset. Por exemplo, o *itemset* {Pão, Leite} é um *2-itemset*, pois contém dois itens. Sendo $\mathcal{T} = \{t_1, t_2, \dots, t_N\}$ o conjunto de todas as transações da base de dados, define-se o suporte de um *itemset* X como sendo o número de transações $t_j \in \mathcal{T}$ que contêm X . Formalmente, o suporte de um *itemset* X pode ser expresso da seguinte maneira [74]:

$$\text{sup}(X) = |\{t_i | X \subseteq t_i, t_i \in \mathcal{T}\}| \quad (3.1)$$

sendo que $|\cdot|$ denota o número de elementos do conjunto. No caso da tabela 3.1, o *itemset* {Pão, Leite} tem suporte igual a três, uma vez que três das cinco transações da tabela contêm todos os dois itens.

3.1.1 Regras de associação

Uma maneira de representar os padrões encontrados por meio da mineração de *itemsets* frequentes consiste em utilizar regras de associação. Regras de associação são expressões de implicação na forma “se-então” e podem ser formalmente definidas como:

Definição 3.1. *Uma regra de associação é uma expressão de implicação na forma $X \Rightarrow Y$, onde X e Y são dois itemsets disjuntos e X e Y são chamados, respectivamente, de antecedente e consequente da regra.*

Adicionalmente, existem medidas para quantificar a utilidade de uma regra. Duas dessas medidas são o suporte e a confiança. O suporte mede a frequência de uma regra no conjunto de dados e a confiança corresponde à fração das transações que contêm todos os itens do antecedente que também contêm todos os itens do consequente. Sendo N o total de transações existentes na base de dados, as medidas de suporte e confiança podem ser definidas da seguinte maneira:

$$\text{Suporte: } \text{sup}(X \Rightarrow Y) = \frac{\text{sup}(X \cup Y)}{N} \quad (3.2)$$

$$\text{Confiança: } \text{conf}(X \Rightarrow Y) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)} \quad (3.3)$$

Tomando como exemplo a Tabela 3.1 e a regra $\{\text{Fraldas}\} \Rightarrow \{\text{Cerveja}\}$, temos que $X = \{\text{Fraldas}\}$, $Y = \{\text{Cerveja}\}$ e $X \cup Y = \{\text{Fraldas}, \text{Cerveja}\}$. Uma vez que o número de transações na Tabela 3.1 é cinco, o suporte da regra é dado por:

$$\text{sup}(\{\text{Fraldas}\} \Rightarrow \{\text{Cerveja}\}) = \frac{\text{sup}(\{\text{Fraldas}, \text{Cerveja}\})}{N} = \frac{3}{5} \quad (3.4)$$

A medida de confiança da regra é calculada como:

$$\text{conf}(\{\text{Fraldas}\} \Rightarrow \{\text{Cerveja}\}) = \frac{\text{sup}(\{\text{Fraldas}, \text{Cerveja}\})}{\text{sup}(\{\text{Fraldas}\})} = \frac{3}{4} \quad (3.5)$$

Usualmente, o problema de minerar regras de associação consiste em encontrar em uma base de dados todas as regras de associação que apresentem valores de suporte e confiança maiores que um limiar mínimo. Formalmente, sendo \mathcal{T} o conjunto de todas as transações, deseja-se encontrar todas as regras de associação com suporte maior que minSup e confiança maior que minConf , onde minSup e minConf são valores de limiar mínimos para as métricas de suporte e confiança. Assim, a mineração de regras de associação pode ser resolvida em duas etapas:

1. Encontrar todos os *itemsets* que apresentem suporte maior do que minSup . *Itemsets* com suporte maior do que minSup são denominados de *itemsets* frequentes.
2. Gerar regras a partir dos *itemsets* frequentes que apresentem confiança maior do que minConf .

Em termos de custo computacional, a geração dos *itemsets* frequentes domina o tempo de execução dos algoritmos de mineração de regras de associação. De fato, o método utilizado para gerar os *itemsets* frequentes tem um papel fundamental no desempenho do algoritmo. Supondo que em uma base de dados existam d itens, então o total de *itemsets* que podem ser gerados, excluindo o *itemset* vazio, é $2^d - 1$. Uma vez que o total de *itemsets* cresce exponencialmente com relação ao número de itens na base de dados, computar o suporte para cada um desses *itemsets* torna-se inviável mesmo para bases de dados de tamanho moderado. Na seção que se segue é discutido o algoritmo *Apriori* proposto por [2]. Explorando propriedades da medida de suporte, o algoritmo é capaz de encontrar *itemsets* frequentes sem a necessidade de gerar todos os possíveis *itemsets* existentes na base de dados.

Uma vez aplicado o algoritmo *Apriori*, a próxima etapa do processo de mineração consiste em gerar regras a partir dos *itemsets* frequentes encontrados. É possível extrair uma regra de associação realizando o particionamento de um *itemset* freqüente Y em dois conjuntos não vazios, X e $Y - X$, tal que a regra $X \Rightarrow Y - X$ tenha confiança maior que minConf . Apesar da medida de confiança não satisfazer a propriedade de monotonicidade, se uma regra $X \Rightarrow Y - X$ não satisfizer o critério da confiança mínima, então qualquer regra $X' \Rightarrow Y - X'$, onde X' é um subconjunto de X , não satisfará o limiar mínimo de confiança. Essa propriedade permite realizar o descarte de possíveis regras candidatas, tornando a etapa de geração de regras mais eficiente.

3.1.2 O Algoritmo *Apriori*

Uma importante propriedade apresentada pela medida de suporte é a anti-monotonicidade: o valor de suporte de um *itemset* nunca excede o valor de suporte de seus subconjuntos. Sendo $J = 2^{\mathcal{I}}$ o conjunto potência de \mathcal{I} (conjunto de todos os subconjuntos de \mathcal{I}), pode-se dizer que a medida de suporte é anti-monotônica pois:

$$\forall X, Y \in J : (X \subseteq Y) \implies \text{sup}(X) \geq \text{sup}(Y) \quad (3.6)$$

A anti-monotonicidade do suporte é também conhecida como princípio *apriori*. Esse princípio consiste no fato de que, se um *itemset* é freqüente, então todos os seus subconjuntos também o são. O algoritmo

Apriori, proposto em [2], faz uso deste princípio para controlar a quantidade de *itemsets* candidatos gerados. *Itemsets* que possuam subconjuntos sabidamente infreqüentes podem ser descartados, dispensando o cálculo da medida de suporte. Tomando como exemplo o látice de *itemsets* da figura 3.2, se o *itemset* $\{ab\}$ não atender o critério do suporte mínimo, então, pelo princípio apriori, pode-se descartar os *itemsets* $\{abc\}$, $\{abd\}$ e $\{abcd\}$, devido a estes terem $\{ab\}$ como subconjunto.

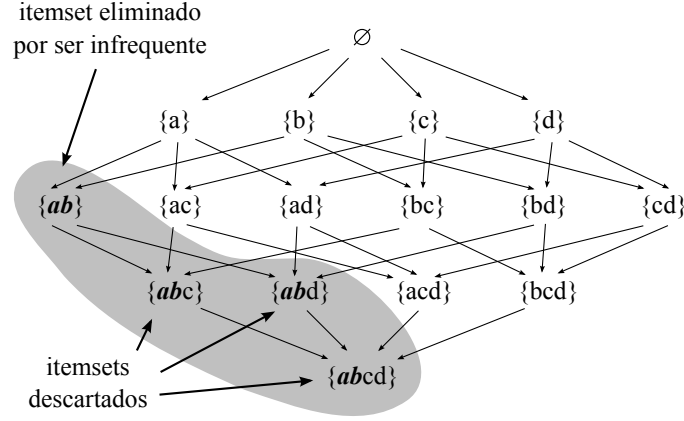


Figura 3.2: O princípio apriori pode ser utilizado para descartar *itemsets* compostos por subconjuntos infreqüentes. No exemplo, $\{ab\}$ é infreqüente e, por este motivo, os *itemsets* $\{abc\}$, $\{abd\}$ e $\{abcd\}$ podem ser descartados, por também serem considerados infreqüentes.

O algoritmo para geração de *itemsets* frequentes do algoritmo *Apriori* consiste em, inicialmente, encontrar todos os 1-*itemsets* e seus respectivos valores de suporte. É então, inicializada a variável k com valor 1 e as três etapas que se seguem são repetidas até que nenhum *itemset* seja identificado:

Geração de Itemsets Candidatos: A partir dos k -*itemsets*, são gerados os $(k + 1)$ -*itemsets* candidatos.

Descarte: Os $(k + 1)$ -*itemsets* candidatos que tenham como subconjunto k -*itemsets* infreqüentes são descartados.

Eliminação: É computado o valor de suporte para os $(k + 1)$ -*itemsets* candidatos restantes, eliminando aqueles que não atendam o critério do suporte mínimo. O valor da variável k é incrementado em uma unidade.

Para gerar os *itemsets* candidatos, o algoritmo *Apriori* utiliza o método $F_{k-1} \times F_{k-1}$, onde F_{k-1} denota o conjunto de todos os $(k - 1)$ -*itemsets* frequentes. O método consiste em manter ordenados os itens dentro de um *itemset* e então agrupar dois $(k - 1)$ -*itemsets* somente se os seus $k - 2$ itens forem idênticos. Ou seja, sendo $X = \{x_1, x_2, \dots, x_{(k-1)}\}$ e $Y = \{y_1, y_2, \dots, y_{(k-1)}\}$ dois $(k - 1)$ -*itemsets* frequentes, o algoritmo $F_{k-1} \times F_{k-1}$ realiza seu agrupamento somente se a condição que se segue é satisfeita:

$$x_i = y_i \text{ para } i = 1, 2, \dots, k - 2 \text{ e } x_{k-1} \neq y_{k-1} \quad (3.7)$$

É importante notar que método $F_{k-1} \times F_{k-1}$ pode gerar *itemsets* candidatos infreqüentes. Por este motivo, realiza-se em seguida a etapa de descarte dos *itemsets* candidatos que tenham subconjuntos infreqüentes. Por fim, na etapa de eliminação, é feita a contagem do suporte dos *itemsets* candidatos não descartados, eliminando aqueles que não atendam ao critério do suporte mínimo.

A geração de candidatos freqüentes pelo algoritmo *Apriori* é descrita no algoritmo 3.1. Nas linhas 1-2 são encontrados os 1-*itemsets* e seus respectivos valores de suporte e inicializada a variável k . A geração de *itemsets* candidatos juntamente com o descarte daqueles que contêm subconjuntos infreqüentes ocorre na linha 5. A atualização do valor de suporte dos *itemsets* candidatos é realizada nas linhas 6-11. Ao invés de se comparar cada transação t com todos os *itemsets* candidatos, é feita na linha 7 uma enumeração dos *itemsets* candidatos contidos em t que são atribuídos à variável C_t . A enumeração pode ser feita de maneira eficiente se os itens de cada transação t forem armazenados de maneira ordenada. Nas linhas 8-10 os *itemsets* candidatos enumerados têm seu valor de suporte atualizado. Por fim, na linha 12, os *itemsets* candidatos que não atendam o critério do suporte mínimo são eliminados.

Algoritmo 3.1 Geração de candidatos freqüentes do algoritmo *Apriori*.

Entrada: Conjunto de itens $\mathcal{I} = \{i_1, i_2, \dots, i_d\}$ e conjunto de transações $\mathcal{T} = \{t_1, t_2, \dots, t_N\}$.

Saída: Conjunto de *itemsets* freqüentes $\bigcup F_k$.

```

1:  $k \leftarrow 1$ 
2:  $F_k \leftarrow \{i \mid i \in \mathcal{I} \wedge \text{sup}(\{i\}) \geq N \times \text{minSup}\}$     // Atribui a  $F_k$  todos os 1-itemsets freqüentes.
3: repetir
4:    $k \leftarrow k + 1$ 
5:    $C_k \leftarrow \text{geraItemsetsCandidatos}(F_{k-1})$     //  $F_{k-1} \times F_{k-1}$  seguido por descarte.
6:   para cada transação  $t \in \mathcal{T}$  faça
7:      $C_t \leftarrow \{c \mid c \subseteq t \wedge c \subseteq C_k\}$     //  $C_t$  são os candidatos que pertencem a  $t$ .
8:     para cada itemset candidato  $c \in C_t$  faça
9:        $\text{sup}(c) \leftarrow \text{sup}(c) + 1$ 
10:    fim para
11:  fim para
12:   $F_k \leftarrow \{c \mid c \in C_k \wedge \text{sup}(c) \geq N \times \text{minSup}\}$     // Atribui a  $F_k$  todos os  $k$ -itemsets freqüentes.
13: até que  $F_k = \emptyset$ 
14: retorna  $\bigcup F_k$ 

```

3.1.3 Medidas de interesse para regras de associação

A propriedade de anti-monotonicidade do suporte é útil para minerar regras de associação de maneira eficiente. No entanto, tanto a medida de suporte quanto a medida de confiança apresentam limitações do ponto de vista de avaliação de utilidade de regras. Considere o seguinte exemplo:

Exemplo 3.1. Suponha que a seguinte regra de associação frequente $\{\text{Sabonete}\} \Rightarrow \{\text{Arroz}\}$ foi minerada em uma base de dados de compras em um supermercado brasileiro. Suponha ainda que a regra tenha uma medida de confiança de 75% e que o *itemset* $\{\text{arroz}\}$ tenha um suporte de 80%. Uma análise não cuidadosa da regra pode levar a impressão de que pessoas que comprem sabonete tendem a comprar arroz. No entanto, o produto arroz é comprado por 80% das pessoas independentemente do produto sabonete ter sido comprado ou não. De fato, ao comprar o produto sabonete, observa-se uma queda de 80% para 75% na probabilidade de se comprar arroz. Assim, existem situações nas quais a confiança e o suporte não são suficientes para caracterizar a utilidade de uma regra de associação.

A principal limitação da confiança está no fato de que a medida não considera o suporte do conseqüente da regra. Por este motivo, muitos autores propuseram medidas alternativas ao suporte e a

confiança para avaliar a utilidade de regras de associação [94]. Um exemplo é a métrica *lift* descrita pela seguinte expressão:

$$\text{Lift: } \text{lift}(X \Rightarrow Y) = \frac{\text{conf}(X \Rightarrow Y)}{\text{sup}(Y)} \quad (3.8)$$

O *lift* corresponde a razão entre a confiança da regra e o suporte de seu consequente. No caso da regra do exemplo 3.1 teríamos:

$$\text{lift}(\{\text{Sabonete}\} \Rightarrow \{\text{Arroz}\}) = \frac{\text{conf}(\{\text{Sabonete}\} \Rightarrow \{\text{Arroz}\})}{\text{sup}(\{\text{Arroz}\})} = \frac{0,75}{0,80} = 0,9375 \quad (3.9)$$

O *lift* pode ser visto como uma medida de correlação entre o antecedente e o consequente de uma regra. Assim, se $\text{lift}(X \Rightarrow Y) > 1$ tem-se que X e Y são positivamente correlacionados. Se $\text{lift}(X \Rightarrow Y) < 1$ então X e Y são negativamente correlacionados (caso do Exemplo 3.1). Uma outra medida de utilidade de regras de associação que considera o suporte do consequente é o *fator de interesse*:

$$\text{Fator de Interesse: } \text{Int}(X \Rightarrow Y) = \text{conf}(X \Rightarrow Y) - \text{sup}(Y) \quad (3.10)$$

As medidas de *lift* e fator de interesse podem ser interpretadas como a comparação da confiança de uma regra contra o suporte do consequente. Assim, uma regra é considerada interessante se sua confiança difere significativamente do valor esperado para o suporte. Em [62] um modelo de máxima entropia é utilizado para prever o suporte de *itemsets* baseado no suporte e confiança de *itemsets* já conhecidos. Um *itemset* é considerado útil se existe uma grande diferença entre a predição e o valor real de seu suporte.

3.1.4 Minerando *itemsets* frequentes em grandes bases de dados

Nesta seção são discutidos dois fatores que afetam o desempenho da mineração de *itemsets* frequentes em grandes bases de dados: a contagem de pares de itens (*2-itemsets*) e o acesso a disco. A contagem de pares de itens (*itemsets* de tamanho dois) domina o tempo de execução e o uso de memória [81] por dois motivos:

1. A quantidade de *itemsets* de tamanho um é muito menor do que *itemsets* de tamanho dois. De fato, se N é o número de itens na base de dados, o número de 1-*itemsets* é N e o número de 2-*itemsets* é $(N^2 - N)/2$.
2. A quantidade de *itemsets* frequentes com mais de dois itens tende a ser menor do que a quantidade de *itemsets* de tamanho dois. A mineração de regras de associação assume que somente um pequeno número de *itemsets* frequentes devem ser encontrados¹ [74]. Assim, o valor do limiar mínimo de suporte deve ser alto o suficiente para garantir que somente um pequeno número de *itemsets* frequentes seja encontrado. Ao mesmo tempo, a propriedade da anti-monotonicidade (Equação 3.6) garante que se houver um *itemset* frequente de tamanho três, então haverá também três *itemsets* frequentes de tamanho dois contidos nele. Por este motivo, existe um número significativamente maior de pares frequentes do que de *itemsets* frequentes de tamanho maior do que dois.

¹Caso o número de *itemsets* frequentes não fosse pequeno, seria impraticável analisar, considerar a significância ou mesmo ler os *itemsets* frequentes.

Uma vez que o algoritmo *Apriori* é capaz de descartar *itemsets* não frequentes, o tempo de execução e uso de memória é dominado pela contagem de *itemsets* frequentes. Por este motivo, é necessário garantir que: (i) exista espaço em memória para contar os pares de itens e (ii) que essa memória seja utilizada de maneira eficiente. Assim é preciso representar os itens de maneira eficiente. Uma solução consiste em representar os itens com números inteiros. Caso os itens sejam representados por cadeias de caracteres na base de dados, pode-se utilizar uma estrutura *hash* para converter o nome do item para o respectivo valor inteiro.

Adicionalmente, o processo de contar o suporte de um par de itens exige que, ao encontrá-lo na base, um contador respectivo seja atualizado. Uma maneira de acessar de maneira eficiente os contadores de suporte dos pares consiste na utilização de matrizes nas quais as entradas correspondem a pares de itens. Ou seja, dado um par de itens $\{i, j\}$, seu suporte poderia ser armazenado na entrada $\text{Cont}[i, j]$ de uma matriz, com $i < j$. O problema desta estratégia é que metade da matriz Cont não seria utilizada, desperdiçando memória. Uma alternativa consiste em utilizar matrizes triangulares [81]. Uma matriz triangular é um vetor de uma dimensão que armazena o par $\{i, j\}$ em sua k -ésima posição, onde k é dado por:

$$k = (i - 1) \left(n - \frac{i}{2} \right) + j - 1 \quad (3.11)$$

Outro modo de melhorar o desempenho da mineração de *itemsets* frequentes consiste em amostrar a base de dados. Sendo f um fator de amostragem (fração das transações amostradas da base), a estratégia de amostragem consiste em:

1. Amostrar aleatoriamente $N \times f$ transações, onde N corresponde ao número de transações na base de dados;
2. Ajustar o valor do suporte mínimo de minSup para $\text{minSup} \times f$;
3. Aplicar o algoritmo de mineração de *itemsets* frequentes utilizando as transações amostradas e suporte mínimo ajustado.

A vantagem da amostragem é que se a amostra escolhida for suficientemente pequena, é possível executar o algoritmo em memória principal. Assim, a base só é lida em disco para realizar a amostragem dos *itemsets*. Caso exista uma garantia de que as transações estejam em ordem aleatória, basta carregar do disco as primeiras $N \times f$ transações.

A aplicação da amostragem, no entanto, pode gerar falsos positivos e falsos negativos. Falsos positivos são *itemsets* que não são frequentes na base mas que são frequentes na amostra. Os falsos negativos são *itemsets* frequentes na base mas que não são frequentes na amostra. É importante notar, no entanto, que *itemsets* com suporte muito maior ou muito menor do que minSup tem uma baixa probabilidade de gerar falsos negativos ou falsos positivos. Adicionalmente, falsos positivos podem ser removidos por meio de uma leitura extra da base de dados. Ou seja, após obter o conjunto de *itemsets* frequentes a partir de uma amostra da base, é possível contar o suporte dos *itemsets* obtidos para toda a base de dados e remover aqueles que apresentem o suporte menor que minSup . Falsos negativos podem ser reduzidos, porém não eliminados. Para tanto, é necessário reduzir o suporte mínimo para a base amostrada. Ou seja, ao invés de ajustar o valor do suporte mínimo de $\text{minSup} \times f$ para $a \times \text{minSup} \times f$, onde $0 < a < 1$.

O algoritmo SON (Savasere, Omiecinski e Navathe) [88] utiliza amostragem para realizar a mineração de *itemsets* frequentes em memória e garante que não haverão falsos negativos ou falsos positivos. Para tanto, são realizadas duas leituras da base de dados. O algoritmo SON consiste nas seguintes etapas:

1. Dividir a base de dados em c *chunks* (pedaços);
2. Minerar os *itemsets* frequentes em cada um dos c *chunks*, ajustando a valor do suporte mínimo para $(1/c)minSup$;
3. Unir os *itemsets* frequentes encontrados para cada chunk em um único conjunto;
4. Contar o suporte dos *itemsets* frequentes obtidos para toda a base, removendo os falsos positivos;

A etapa 4 garante que o algoritmo SON não gera falsos positivos. Adicionalmente, não haverão falsos negativos pois se um *itemset* não for frequente em nenhum *chunk* então, necessariamente, seu suporte deverá ser menor que $minSup$, pois:

$$\underbrace{c}_{\text{chunks}} \times \underbrace{(1/c)minSup}_{\text{minSup (chunk)}} = minSup \quad (3.12)$$

Uma vez que a mineração de *itemsets* frequentes nos *chunks* são independentes entre si, a formulação do algoritmo SON permite sua implementação em computação paralela utilizando o paradigma map-reduce.

3.1.5 Classificadores Associativos

A classificação associativa é uma abordagem em mineração de dados na qual associações fortes entre padrões de características e classes são mineradas para construir classificadores, denominados de classificadores associativos. Desta maneira, a classificação associativa pode ser vista como a integração entre duas tarefas de mineração de dados: a de classificação e a de mineração de regras de associações. Esta integração é realizada por meio de um subconjunto especial de regras de associação nas quais o consequente está restrito ao atributo classe. Na literatura, tais regras são denominadas de regras de associação de classificação (*Classification Association Rules - CARs*).

O problema de se construir um classificador associativo pode ser dividido em duas etapas principais: (i) minerar as *CARs* e (ii) determinar a classe de um dado objeto de teste por meio das regras geradas. Uma vez que uma *CAR* está na forma $X \Rightarrow \{c'\}$, onde $X = \{x_1, x_2, \dots, x_n\}$ representa um conjunto de itens derivados dos atributos do conjunto de dados e c' um valor para o atributo classe, uma abordagem que pode ser empregada para mineração consiste em encontrar todos os *itemsets* no formato $\{x_1, x_2, \dots, x_n, c'\}$. Para este propósito, pode-se empregar algoritmos de mineração de *itemsets* frequentes tradicionais, como o *Apriori*. De fato, esta abordagem é amplamente adotada por métodos de indução de classificadores associativos [58, 85].

Quando o número de atributos do conjunto de dados é muito alto, sobretudo quando o limiar mínimo de suporte $minSup$ é baixo, o custo computacional do algoritmo *Apriori* é excessivamente alto. Para tornar o processo de mineração de *CARs* mais eficiente, algoritmos como o L^3 [8] empregam o método *FP-Growth*. O método *FP-Growth* (*Frequent Pattern Growth*) [41] minera *itemsets* frequentes por meio de uma árvore de prefixos denominada de *FP-Tree*. Na *FP-Tree* cada *itemset* é representado por um

caminho na árvore, sendo que dois *itemsets* diferentes podem compartilhar parte de um dado caminho. Esta propriedade da árvore permite que o conjunto de dados seja representado utilizando menos memória. Adicionalmente, a construção da *FP-Tree* exige que o conjunto de dados seja percorrido somente duas vezes, reduzindo o número de leituras a disco.

Existem ainda métodos de treinamento de classificadores associativos que não empregam algoritmos tradicionais de regras de associação. Um exemplo é o algoritmo *CPAR (Classification based on Predictive Associations Rules)*[43] que adota uma estratégia gulosa na qual a construção de uma regra consiste em crescer o antecedente da regra um item por vez, sendo que o item a ser adicionado à regra é aquele que maximiza a confiança da regra. A principal desvantagem do método de mineração do algoritmo *CPAR* é seu alto tempo de execução quando comparado com os demais métodos [95].

Outro aspecto relevante a respeito dos classificadores associativos é o conceito de ordenação entre regras. Por exemplo, em [58], as regras mineradas são ordenadas com respeito à sua métrica de confiança, sendo que o suporte e a ordem de geração da regra são empregados com critérios de desempate. Essa ordenação é utilizada pelo algoritmo tanto para descartar regras desnecessárias como também para classificar um dado objeto de teste.

Uma vez obtido o conjunto de regras final, existem duas abordagens principais para se determinar a classe de um objeto de teste. A primeira delas, empregada em [58, 8], envolve definir uma ordem entre as regras geradas com base nas medidas de suporte e confiança das regras. A classe predita é aquela que apresenta a maior precedência dentre todas as que são aplicáveis ao objeto de teste. Uma segunda estratégia, adotada em [43, 3, 85], consiste em encontrar um valor de escore para cada uma das classes a partir do conjunto de regras que são aplicáveis ao objeto sendo classificado. A classe que apresentar o maior valor de escore é aquela que será retornada pelo classificador.

A tabela 3.2 apresenta um resumo dos principais aspectos de diferentes métodos de classificação associativa. Na primeira coluna, (*Mineração de Regras*), é apresentado o algoritmo de mineração de *CARs* empregado em cada método. Na coluna *ordenação de regras* são apresentados os critérios, por ordem de precedência, empregados para ordenar as regras mineradas. Caso ocorra empate para um determinado critério, o seguinte é utilizado para realizar o desempate. Por fim, a coluna *Modelo de Predição* corresponde à estratégia empregada para classificar um *itemset*.

3.1.6 Mineração de imagens utilizando *itemsets* frequentes

A mineração de *itemsets* frequentes é uma tarefa amplamente empregada para realizar mineração de imagens. Contudo, a mineração de *itemsets* tem sido relativamente pouco explorada pela comunidade de mineração de imagens [34]. Ainda assim, importantes resultados em mineração de imagens utilizando *itemsets* foram obtidos recentemente. Esta seção tem como objetivo apresentar esses resultados.

A obtenção de *itemsets* para representar imagens pode ser dividida em três etapas:

1. Extração de representações de imagens;
2. Conversão das representações para *itemsets*;
3. Mineração de *itemsets* frequentes e úteis;

A extração de representações consiste na aplicação de extratores de características (ver seção 2.1) para representar o conteúdo visual das imagens. Existem métodos que empregam dados de contexto para

Tabela 3.2: Comparação entre métodos de classificação associativa.

Nome	Mineração de Regras	Ordenação de Regras	Modelo de Predição	Referência
<i>CBA</i>	<i>Apriori</i>	1. Confiança; 2. Suporte; 3. Ordem de geração da regra.	Única regra	[58]
<i>L³</i>	<i>FP-Growth</i>	1. Confiança; 2. Suporte; 3. Número de itens na regra; 4. Ordem lexicográfica do antecedente da regra.	Única regra	[8]
<i>CPAR</i>	Gulosa	1. Confiança; 2. Suporte; 3. Número de itens na regra.	Múltiplas regras	[43]
<i>ARC-PAN</i>	<i>Apriori</i>	1. Confiança.	Múltiplas regras	[3]
<i>IDEA</i>	<i>Apriori</i>	Não utiliza ordenação de regras.	Múltiplas regras	[85]

complementar as representações de imagem [84]. Na etapa de conversão, os vetores de características ou histogramas de palavras visuais são convertidos para itens, gerando os *itemsets* que irão representar imagens. Por fim, na terceira etapa, são aplicadas técnicas de mineração de *itemsets* frequentes para encontrar padrões úteis.

A motivação para utilizar *itemsets* frequentes para mineração de imagens está na hipótese de que padrões frequentes, derivados das características de imagem, podem ser semanticamente mais representativos do que as representações isoladamente. Por exemplo, um conjunto de palavras visuais que ocorre com frequência em imagens de uma mesma classe poderia ser utilizado para identificar imagens similares de maneira mais precisa do que as palavras visuais isoladamente.

Em [71], é proposta a geração de *itemsets* de imagens a partir de representações baseadas na abordagem *Bag-of-Features* (seção 2.1.4). Assim, se uma palavra visual está presente no histograma de palavras visuais da imagem, então essa palavra visual também estará presente no *itemset* da imagem. Para encontrar *itemsets* frequentes os autores aplicam mineração de *itemsets* fechados. Os *itemsets* fechados minerados são utilizados para gerar uma nova representação de imagens. Essa representação consiste em um vetor binário $\mathbf{V} = \{V_1, V_2, \dots, V_n\}$, onde V_i indica se o i -ésimo *itemset* está contido no conjunto de palavras visuais da imagem. As representações de imagens são aplicadas na tarefa de classificação de imagens utilizando um classificador *SVM* linear.

Em [34] *itemsets* de imagens são gerados a partir de histogramas de palavras visuais locais. Ou seja, de cada imagem são extraídos vários histogramas de palavras visuais denominados de *LBOW* (*local bag-of-words*). Para gerar *itemsets* a partir dos *LBOW* os autores desenvolveram o método *FLH* (*frequent local histograms*). De modo similar a [71], os *itemsets* são gerados verificando a presença de palavras visuais nos *LBOW*. No entanto, para poder preservar a informação de frequência das palavras visuais dos

histogramas de palavras visuais, os itens são codificados como pares. Cada um dos pares é composto por uma palavra visual e a frequência dessa palavra no *LBOW*. A mineração de *itemsets* frequentes tem duas etapas. Inicialmente são minerados *itemsets* fechados. Na segunda etapa de mineração, os autores empregam as métricas *discriminativity*, representividade e de redundância para selecionar um subconjunto de *itemsets*. Os *itemsets* minerados são utilizados como uma nova representação de imagem e são aplicados na tarefa de classificação. Para tanto, os autores propõem uma função de distância entre conjuntos de *itemsets* que é utilizada para treinar um classificador *SVM*.

A mineração de *itemsets* frequentes é empregada em [112] para gerar representações de imagens denominadas de *grouplets*. Os *grouplets* são compostos por unidades de características (*feature units*), que correspondem a uma palavra visual e sua localização espacial na imagem. O autores empregam uma modificação do algoritmo *Apriori* para minerar os *grouplets*, que são aplicados na tarefa de reconhecimento de atividades em fotografias.

Em [84] é proposto o método *IDEA*, que aplica regras de associação para auxiliar o diagnóstico médico. Para representar as imagens são utilizados extratores de características globais. Adicionalmente, são utilizadas anotações médicas associadas aos exames médicos como dados de contexto. A conversão das representações de imagem emprega o algoritmo *Omega*, que consiste em discretizar os atributos dos vetores de características de cada imagem. Assim, cada um dos intervalos de discretização é mapeado para um item. O *itemset* de imagem é composto por intervalos de discretização e por itens que correspondem às anotações médicas. O conjunto de *itemsets* obtidos é submetido ao algoritmo *Apriori* adotando-se duas restrições às regras que podem ser geradas: o antecedente é composto por intervalos de discretização e o consequente por anotações médicas. As regras mineradas são utilizadas para prever a relevância de cada uma das anotações médicas para um novo exame médico.

A Tabela 3.3 sumariza os métodos discutidos nesta seção quanto às estratégias de representação de imagens, geração de *itemsets* e mineração de padrões.

Tabela 3.3: Comparação dos métodos de representação de imagens por *itemsets* discutidos na seção 3.1.6.

Algoritmo / Ref.	Representação de Imagem	Geração de <i>Itemsets</i>	Mineração
<i>IDEA</i> [84]	Características globais e anotações médicas	Discretização de atributos	<i>Apriori</i> com restrições no formato das regras
[71]	<i>Bag-of-Features</i>	Itens como palavras visuais	<i>itemsets</i> fechados e seleção de <i>itemsets</i> discriminativos
[112]	<i>Bag-of-Words</i> com informações espaciais	Palavras visuais espacialmente próximas	<i>Apriori</i>
<i>FLH</i> [34]	<i>Bag-of-Words</i> locais	Itens como palavras visuais e frequência no histograma de palavras visuais.	<i>itemsets</i> fechados e seleção de <i>itemsets</i> discriminativos

3.2 Mineração de imagens em serviços de mídia social

3.2.1 Detecção de eventos e pontos turísticos

Usuários de serviços de mídia social frequentemente têm interesse em encontrar fotografias de um determinado ponto turístico ou de eventos sociais. Em [93] é realizada a detecção de imagens relacionadas a eventos utilizando imagens do serviço Twitter. O método proposto inicialmente utiliza agrupamento de dados baseado nos textos associados às imagens para agrupá-las em grupos de tópicos similares. As imagens são então ordenadas de acordo com seu tempo de criação para gerar um fluxo temporal de imagens (*image stream*). Os eventos são detectados aplicando, para cada tópico, algoritmos de *burst detection* para identificar períodos de maior atividade.

Em [75] é proposto um método para detectar eventos e pontos turísticos. São utilizadas imagens do Flickr para gerar um grafo de similaridade no qual as imagens são nós e as arestas indicam a similaridade entre as imagens. As arestas são criadas com base em dados de conteúdo e de contexto. Cada imagem é conectada aos 20 vizinhos visualmente mais similares e também a outras imagens com as quais compartilha *tags*. Para medir a similaridade visual, foi empregado o extrator *SIFT*, adotando a abordagem *Bag-of-Features*. Sobre o grafo gerado é então aplicado agrupamento de dados. Cada grupo de imagens gerado foi classificado como sendo um evento ou um ponto turístico. Para realizar a classificação foram utilizados como atributos:

1. Duração do grupo (intervalo de tempo entre a criação da primeira e última imagem);
2. Razão entre o número de usuários que criaram imagens e o número de imagens no grupo;
3. Conjunto das *tags* mais frequentes no grupo;

A classificação foi realizada utilizando um classificador treinado a partir de um conjunto manualmente anotado de grupos. O método proposto foi aplicado em um conjunto de 207 mil imagens provenientes do Flickr localizadas geograficamente na cidade de Barcelona na Espanha.

3.2.2 Classificação de imagens

A tarefa de classificação de imagens tem como objetivo atribuir imagens a uma dentre várias categorias [28]. Um modo de melhorar o desempenho da classificação de imagens consiste em desenvolver métodos de extração capazes de melhor representar o conteúdo visual das imagens [99]. Contudo, devido à riqueza dos dados de contexto, imagens provenientes de serviços de mídia social abrem um novo caminho para aprimorar a classificação de imagens.

Em [107] são propostas características de imagens baseadas em dados de contexto. Essas características são utilizadas para classificação de imagens. A hipótese dos autores é a de que em um conjunto suficientemente grande de imagens, sempre haverá imagens altamente similares a uma imagem de referência. Tomando como base essa hipótese, é proposto um método de extração de características textuais para imagens que utiliza um conjunto auxiliar de 10 milhões de imagens obtidas do site Flickr. *Tags* são empregadas como dados de contexto. O algoritmo de extração de representações textuais para imagens consiste em buscar os K vizinhos mais similares no conjunto obtido do Flickr. Para medir a similaridade entre as imagens são considerados três diferentes características visuais: *SIFT*, *Gist* e histograma

de cores. A estrutura da representação textual é um histograma que mede a ocorrência de cada uma das possíveis *tags* entre os K vizinhos.

Em [108] os autores utilizam uma base de imagens auxiliar proveniente do *website* Flickr para expandir o conjunto de treinamento de imagens. Para tanto, é criado um grafo de similaridade do conjunto auxiliar a partir de características visuais e de *tags* fornecidas por usuários. Para tornar o processo de construção do grafo escalável, os autores empregam o *framework map-reduce*. O conjunto de imagens de treinamento é expandido por imagens visualmente similares no grafo gerado. Adicionalmente, as representações das imagens de treinamento bem como das imagens a serem classificadas, são enriquecidas com os dados de contexto das imagens visualmente similares do grafo. A classificação de imagens é realizada adotando a abordagem *a posteriori* (discutida na seção 2.3.2). As representações obtidas são utilizadas para treinar dois classificadores: um baseado nos dados de conteúdo e outro baseado nos dados de contexto. Os resultados dos dois classificadores são combinados para retornar a classe de uma imagem.

3.2.3 Anotação automática de imagens

A anotação automática de imagens é uma tarefa de mineração de imagens que tem como objetivo fornecer descrições textuais para uma determinada imagem baseado na análise de conteúdo visual [114]. Tradicionalmente, pesquisas em recuperação de imagens tratam de dois tipos de consultas: por exemplo e por palavras chaves [28]. Quando realiza consultas por exemplo, o usuário submete uma imagem de consulta e o sistema de busca retorna um conjunto de imagens visualmente similares. Uma desvantagem da consulta por exemplo é que em determinadas aplicações não é possível ou viável obter uma imagem para expressar a consulta. Adicionalmente, existe o problema da lacuna semântica, que pode prejudicar o desempenho do sistema de recuperação de imagens.

Nas consultas por palavras chaves, as imagens são recuperadas de maneira similar a documentos de texto. A vantagem é que palavras chaves textuais apresentam uma relação próxima da semântica da consulta [47, 35]. Contudo, as imagens devem possuir anotações textuais para que a consulta por palavras chaves seja possível. Uma vez que anotar manualmente grandes volumes de imagens pode ser impraticável e também porque as anotações fornecidas por humanos tendem a ser subjetivas, existe grande interesse da comunidade científica em desenvolver métodos capazes de fornecer descrições textuais para imagens de maneira automática [29].

O objetivo de um método de anotação de imagens é, dado uma imagem não anotada, retornar uma lista de anotações textuais que melhor descrevem o conteúdo da imagem. Cada anotação a_i retornada tem um escore de relevância associado. Um maior valor para o escore significa que a anotação é mais relevante em relação ao conteúdo visual da imagem.

Uma das abordagens existentes para anotação automática de imagens é o treinamento cruzado (*cross-training*) [12]. O treinamento cruzado consiste em treinar um classificador binário para cada anotação. Os classificadores são então empregados para predizer quais anotações são relevantes para uma imagem não anotada. Cada classificador é treinado utilizando as imagens que contém a respectiva anotação como exemplos positivos e as demais imagens como exemplos negativos.

Por exemplo, considere um problema de anotação no qual existam três diferentes anotações: “noite”, “manhã” e “tarde”. Neste problema o treinamento cruzado requer que três classificadores binários sejam treinados, um para cada anotação. Para treinar o classificador para a anotação “noite”, imagens que

contém a anotação “noite” são utilizadas como exemplos positivos e as demais são utilizadas como exemplos negativos.

A saída de um anotador de treinamento cruzado para uma imagem não anotada é uma lista contendo todas as possíveis anotações. Cada uma dessas anotações tem um escore associado que corresponde a probabilidade retornada pelo respectivo classificador binário. Alguns autores removem da lista de anotações retornadas aquelas que apresentem um escore menor que um determinado limiar.

Encontrar os parâmetros ótimos para os classificadores na abordagem de treinamento cruzado é uma tarefa que pode ser computacionalmente cara. Grande parte dos classificadores apresentam um conjunto de parâmetros que precisam ser ajustados pelos usuários. Por exemplo, para treinar um classificador *SVM* (*Support Vector Machine*) [23], é necessário selecionar o tipo de função de kernel, os parâmetros da função de kernel (p. ex. o expoente para o kernel de função de base radial) e o custo de erro de classificação. De fato, a seleção dos parâmetros por meio de validação cruzada é essencial para obter resultados aceitáveis para classificadores *SVM*.

Outra desvantagem do treinamento cruzado é que as imagens devem ser tratadas como exemplos positivos ou negativos durante a fase de treinamento. Uma vez que o número de exemplos positivos é significativamente menor do que o número de exemplos negativos, o problema de classificação binária torna-se não balanceado, degradando a acurácia do classificador [45].

Outra abordagem para anotação de imagens consiste em utilizar modelos de vizinhos mais próximos para anotar imagens baseando-se nas anotações de imagens visualmente similares [115, 11]. A similaridade visual é estimada calculando a distância entre os vetores de característica das imagens. Alguns modelos baseados em vizinhos mais próximos apresentaram resultados promissores. Por exemplo, em [104], é proposto o método de anotação denominado *TagProp*, que calcula a relevância das anotações por meio de uma soma ponderada baseada nas anotações dos vizinhos mais próximos. O treinamento do anotador *TagProp* consiste em maximizar o número de anotações corretas no conjunto de treinamento utilizando um algoritmo de gradiente projetado.

3.3 Considerações

Conforme discutido neste Capítulo, o processo de mineração de imagens considerando conteúdo e contexto contempla três etapas: (i) extração de características, (ii) combinação de conteúdo e contexto e (iii) mineração. As etapas (i) e (ii) são fundamentais, pois nelas são obtidas representações de imagens que possibilitam a aplicação de técnicas de mineração de dados.

Um exemplo de tarefa de mineração discutida foi a mineração de *itemsets* frequentes, utilizada para encontrar padrões em grandes bases de dados organizadas na forma de transações (*itemsets*). Na seção 3.1.6 foram discutidos métodos que utilizam a mineração de *itemsets* frequentes para encontrar padrões frequentes a partir das características visuais das imagens. Para tanto, esses métodos convertem as representações visuais das imagens para *itemsets*. Esses padrões frequentes sintetizam conjuntos de características visuais que aparecem com frequência em uma base de dados de imagem. Assim, características visuais derivadas destes padrões têm o potencial de reduzir o problema da lacuna semântica (*semantic gap*).

No entanto, um aspecto pouco explorado na representação de imagens por meio de *itemsets* é a combinação de conteúdo e contexto. Por exemplo, nos métodos estudados neste Capítulo, somente o

método *IDEA* [84] considera dados de contexto para gerar os *itemsets* que descrevem as imagens. Ainda assim, o método *IDEA* somente considera dados de contexto na forma de anotações textuais.

Assim, uma linha de pesquisa que pode ser explorada na área de mineração de imagens é a mineração de *itemsets* frequentes para encontrar padrões e combinar características de conteúdo e contexto. No entanto, existem desafios que precisam ser resolvidos. Por exemplo, é necessário desenvolver métodos capazes de gerar itens a partir de dados de contexto complexos como grafo social ou dados textuais que não estejam organizados na forma *tags* ou anotações.

Capítulo 4

Resultados preliminares

Como resultado preliminar deste projeto de doutorado foi desenvolvido o método *MFS-Map* (*Multi-Feature Space Map*). O *MFS-Map* é utilizado para converter as representações dos dados de conteúdo e contexto de imagens para *itemsets*. A motivação para representar imagens como *itemsets* está na possibilidade de minerar padrões frequentes derivados das características de imagens. Os resultados obtidos indicam que esses padrões frequentes, que combinam conteúdo e contexto, podem ser utilizados para melhorar o desempenho da anotação automática de imagens. A hipótese é que os padrões frequentes são capazes de melhor capturar a semântica das imagens do que as características de conteúdo e de contexto separadas.

Este capítulo descreve o *MFS-Map* (seção 4.1) e sua aplicação para a tarefa de anotação automática de imagens (seções 4.2 e 4.3). Na seção 4.4, são apresentados os experimentos para anotação automática de imagens empregando o *MFS-Map*. Para tanto foram utilizados dois conjuntos de dados disponíveis publicamente: o *MIR-Flickr* e o *Image CLEF 2011*.

4.1 *MFS-Map*

O *MFS-Map* é um método desenvolvido neste projeto de doutorado que gera representações de imagem na forma de *itemsets*. Essas representações, denominadas de *itemsets de imagens*, descrevem tanto o conteúdo como os dados de contexto das imagens. Os itens que compõem o *itemset* de imagem são denominados de *itens de característica*.

Para explorar as características de cada modalidade de informação, o processo de obtenção dos itens de característica de dados de contexto é diferente do processo de obtenção dos itens para os dados de conteúdo. Deste modo, na seção 4.1.1 é apresentado o processo de obtenção dos itens de características a partir de vetores de características visuais. Na seção 4.1.2 é descrita a obtenção de itens para dados de contexto representados por meio da abordagem *Bag-of-Words*.

4.1.1 Itens a partir do conteúdo

Para obter os itens de características utilizados para representar o conteúdo das imagens, o *MFS-Map* utiliza o conceito de subespaços de características. Os subespaços são definidos pelas características extraídas por um determinado extrator. Os itens são rótulos de centróides gerados a partir da aplicação de agrupamento de dados *k-means* nos diferentes espaços de característica. Para estimar o parâmetro k foi empregado o critério da silhueta simplificada [103], variando os valores de k e escolhendo o agrupamento resultante que minimiza a silhueta.

A Figura 4.1 ilustra a extração da representação de *itemset* para um cenário com dois subespaços de características gerados por dois extratores. Na Figura 4.1(a), os extratores são utilizados para gerar os vetores de característica das imagens. Neste exemplo, os subespaços estão representados como espaços tridimensionais e cada vetor é representado por um ponto. Na Figura 4.1(b) são mostrados os centróides obtidos a partir da aplicação do algoritmo *k-means* nos subespaços de características A e B. Para o subespaço A, dois centróides foram obtidos: c_1 e c_2 . Para o subespaço B, três centróides foram obtidos: c_3 , c_4 e c_5 .

Os *itemsets* de imagem são obtidos encontrando os centróides mais próximos dos vetores de característica da imagem em cada um dos subespaços. No exemplo da Figura 4.1(b), a imagem, indicada por um ponto, tem sua representação de *itemset* composta pelos centróides $\{c_1, c_5\}$.

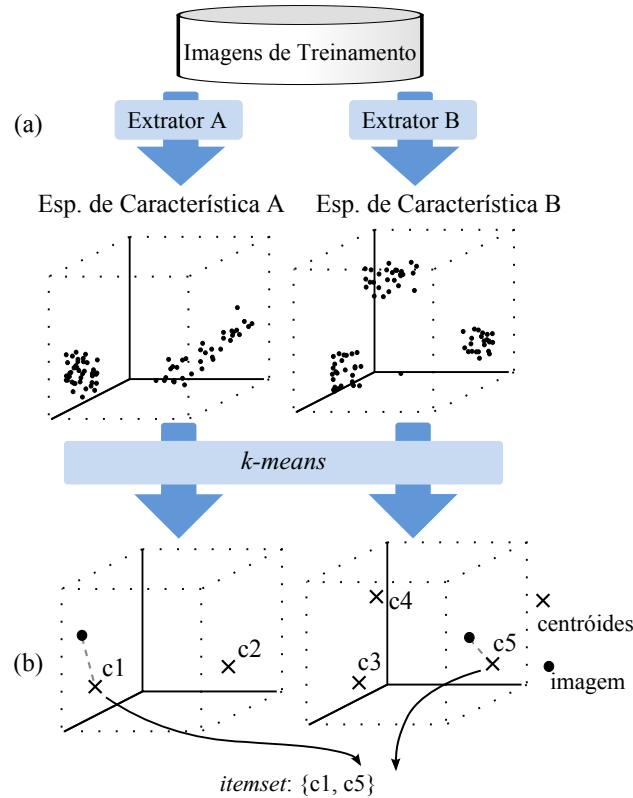


Figura 4.1: Obtenção da representação de *itemset* para características visuais no método *MFS-Map*. (a) Extração de vetores de características das imagens. (b) Centróides obtidos após a execução do algoritmo *k-means* nos subespaços de característica A e B. A representação por *itemset* de uma imagem é obtida encontrando os centróides mais próximos em cada subespaço.

4.1.2 Itens a partir de dados contexto

Nos experimentos desenvolvidos com o *MFS-Map* foram estudadas maneiras de extrair representações de *itemsets* a partir de *tags* de usuários do *website* Flickr. Uma maneira de obter essas representações seria utilizar a mesma abordagem adotada para as características visuais. Ou seja, representar as *tags* como um vetor por meio da abordagem *Bag-of-Words* e aplicar o algoritmo *k-means* para encontrar centróides.

Contudo, esta abordagem não explora dois aspectos importantes da representação *Bag-of-Words*. Primeiramente, as características são esparsas, ou seja, grande partes dos valores de atributo são zero. O segundo aspecto é que a similaridade entre dois objetos pode ser melhor representada pela quantidade de palavras (*tags*) compartilhadas do que pelo número de palavra diferentes.

Assim, a estratégia adotada para extração de *itemsets* para dados de contexto é diferente da estratégia adotada para dados de conteúdo. Para dados de contexto representados pela abordagem *Bag-of-Words* o *MFS-Map* atribui um item para cada *tag* presente nos dados de contexto da imagem. Ou seja, se a imagem possui a *tag* como dado de contexto, então o item correspondente será incluído no *itemset* da imagem. Uma vez que a geração de itens a partir de representações *Bag-of-Words* não requer agrupamento, sua obtenção é computacionalmente mais eficiente do que a abordagem adotada para os itens de conteúdo. Adicionalmente, essa mesma estratégia pode ser aplicada a representações baseadas na abordagem *Bag-of-Features* para características visuais.

4.2 Geração das regras

As anotações geradas pelo *MFS-Map* foram utilizadas para a tarefa de anotação automática de imagens. Ou seja, dada uma imagem não anotada, foi desenvolvido um algoritmo que retorna uma lista de anotações textuais que melhor descrevem o conteúdo da imagem. Cada anotação retornada tem um escore de relevância associado. Um maior valor para o escore significa que a anotação tende a ser mais relevante em relação ao conteúdo visual da imagem.

Para prever a relevância das anotações, o método de anotação desenvolvido minera regras de associação que descrevem relações entre anotações e regiões de espaços de características. O processo de mineração das regras, ilustrado na Figura 4.2, consiste em três etapas:

1. Obtenção dos *itemsets* de imagem por meio do algoritmo *MFS-Map*;
2. Inserção das anotações nos *itemsets* das imagens do conjunto de treinamento;
3. Mineração de regras na forma $\{f_i\} \Rightarrow \{a_j\}$ a partir dos *itemsets* de imagens, onde f_i e a_j correspondem, respectivamente, a um item de característica e a uma anotação de imagem.

Para minerar as regras de associação, o algoritmo proposto utiliza o conceito de confiança $\text{conf}(\{f_i, a_j\})$ de um par composto por um item de característica f_i e uma anotação a_j . A confiança $\text{conf}(\{f_i, a_j\})$ do par é similar à utilizada para regras de associação na seção 3.1.1 e é definida como:

Definição 4.1. A confiança $\text{conf}(\{f_i, a_j\})$ de um par item de característica f_i e anotação a_j é dada por:

$$\text{conf}(\{f_i, a_j\}) = \text{freq}(\{f_i, a_j\}) / \text{freq}(\{f_i\}), \quad (4.1)$$

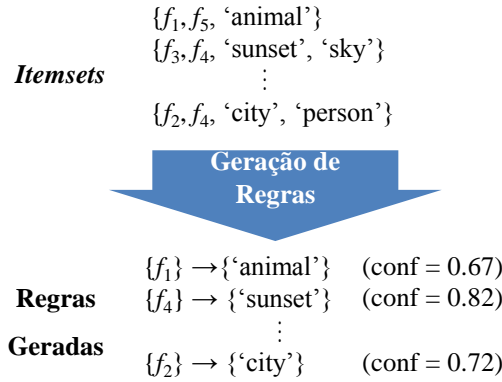


Figura 4.2: Entrada e saída do algoritmo de geração de regras de associação. Os *itemsets* são compostos de itens de característica e anotações. A saída consiste nas regras mineradas no formato $\{\text{item de característica}\} \Rightarrow \{\text{anotação}\}$.

onde $\text{freq}(\{f_i, a_j\})$ é o número de vezes que o par de característica f_i e anotação a_j ocorrem no mesmo itemset e $\text{freq}(\{f_i\})$ é o número de vezes que f_i aparece em um itemset.

O algoritmo de geração de regras calcula a confiança para cada par $\{f_i, a_j\}$. A confiança é utilizada como uma estimativa da utilidade do item de característica f_i em prever a relevância da anotação a_j . As regras são geradas de todos os pares $\{f_i, a_j\}$ que apresentem confiança maior que um limiar mínimo, denotado por minConf . Por exemplo, se o par $\{f_1, \text{'animal'}\}$ tem uma confiança de 0,8 e o valor de minConf é 0,7, então a regra $\{f_1\} \rightarrow \{\text{'animal'}\}$ será gerada.

O algoritmo de geração de regras requer uma única passagem sobre o conjunto de *itemsets*. Nessa passagem é contada a frequência dos pares formados por itens de característica e anotações. Adicionalmente, é contada a frequência de cada item de característica. Uma vez que o número de pares que ocorrem no conjunto *itemsets* tende a ser significativamente menor do que todos os possíveis pares, a frequência dos pares é armazenada em uma tabela *hash* para reduzir o uso de memória.

4.3 Anotação das imagens

A fase de anotação do *MFS-Map* toma como entrada o conjunto de regras mineradas e prediz as anotações para uma imagem não anotada. Cada anotação retornada tem um escore de relevância associado. A primeira parte da fase de anotação consiste em extrair as representações de *itemsets* da imagem de entrada utilizando o método *MFS-Map*. Para as imagens não anotadas a representação de *itemset* não contém anotações e é, portanto, composta somente por itens de característica. É importante notar que para a fase de anotação é necessário utilizar os mesmos extratores empregados para representar as imagens de treinamento. Uma vez que os centróides foram calculados durante a etapa de treinamento, a anotação de imagens dispensa o uso de algoritmos de agrupamento.

A segunda etapa da fase de anotação consiste em selecionar as regras que tem em seu antecedente algum item presente na representação de *itemset* da imagem a ser anotada. O escore de relevância de uma anotação a_i é dado pela confiança média de todas as regras selecionadas que apresentem a_i no consequente. Por exemplo, se a representação de *itemset* da imagem de entrada fosse $\{f_2, f_4\}$ e o conjunto de regras mineradas fosse:

1. $\{f_1\} \rightarrow \{\text{'animal'}\} (\text{conf} = 0.67)$

2. $\{f_4\} \rightarrow \{\text{'sunset'}\}(\text{conf} = 0.82)$
3. $\{f_2\} \rightarrow \{\text{'city'}\}(\text{conf} = 0.72)$
4. $\{f_4\} \rightarrow \{\text{'sunset'}\}(\text{conf} = 0.93)$

então as regras 2, 3 e 4 seriam selecionadas porque seus antecedentes contêm o item de característica f_4 and f_2 respectivamente. A regra 1, contudo, não seria selecionada pois o item de característica f_1 não está presente na representação de *itemset* da imagem de entrada.

O escore de relevância da anotação ‘sunset’, considerando as regras selecionadas, é a média das confianças das regras que contêm ‘sunset’ em seu conseqüente, ou seja, as regras 2 e 4, resultando em um escore de relevância de $(0,82 + 0,93)/2 = 0,875$. Para anotações que não aparecem no conseqüente de nenhuma das regras selecionadas, como por exemplo a anotação ‘animal’, o escore de relevância retornado é zero.

4.4 Experimentos

Para avaliar o *MFS-Map* e o algoritmo de anotação proposto foram empregados dois conjuntos de imagens disponíveis publicamente: o *MIR Flickr* [48] e o *Image CLEF 2011*. Ambos os conjuntos são amplamente empregados para a avaliação de métodos de anotação automática de imagens [48] e são compostos por imagens provenientes do serviço de mídia social Flickr. O *MIR Flickr* e o *Image CLEF 2011* contém, respectivamente, 25.000 e 18.000 imagens.

Cada imagem em ambos os conjuntos são manualmente anotadas. O tamanho do vocabulário (número de diferentes anotações) é de 25 para o *MIR Flickr* e de 98 para o *Image CLEF 2011*. As anotações descrevem objetos presentes nas imagens (por exemplo, “car”, “dog”, “flower”) e descrição da cena (p. ex. “sunset”, “indoor”, “night”). Adicionalmente, cada imagem contém *tags* sugeridas por usuários do Flickr. Apesar de apresentarem ruídos e serem subjetivas, as *tags* introduzem informação valiosa no processo de anotação. Assim, as *tags* foram utilizadas como dados de contexto pelo *MFS-Map* e demais métodos de anotação avaliados. A Figura 4.3 ilustra um exemplo de imagem do conjunto *MIR Flickr* com suas *tags* e anotações.

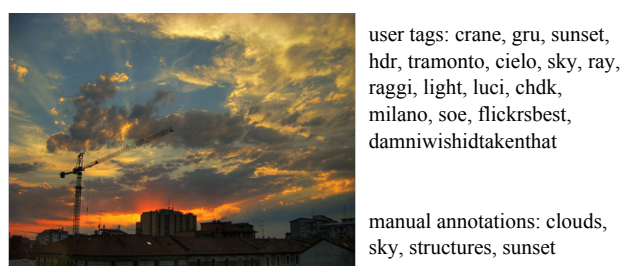


Figura 4.3: Exemplo de imagem do conjunto *MIR Flickr*. A direita são mostradas as *tags* sugeridas por usuários e as anotações que foram utilizadas no processo de anotação e avaliação.

Para quantificar o desempenho foram empregadas duas métricas: precisão média (*average precision* - *AP*) e *break even point precision* (*BEP*). Ambas são calculadas por anotação, mas a média pode ser calculada para obtenção de um único valor. Para calcular a *AP* e *BEP* para uma determinada anotação a as imagens são ordenadas por sua relevância, sendo que a relevância é quantificada pelo escore de

anotação. A métrica AP (precisão média) consiste em calcular a média da precisão em cada uma das posições da lista ordenadas de imagens. A precisão em uma dada posição n corresponde à fração de imagens ranqueadas abaixo ou igual a n que foram corretamente anotadas. Já a métrica BEP corresponde à precisão na posição n' , onde n' é o número de imagens anotadas com a anotação sendo avaliada. Os valores para a AP e BEP foram obtidos por meio de dez repetições de validação cruzada com dez partições.

4.4.1 Extração de características

Para a extração de características foram utilizados os extratores apresentados na tabela 4.1. Os vetores de características Gist e SFTA foram normalizados para o intervalo entre zero e um. Os demais vetores de características foram normalizados para que os atributos somassem um (normalização L_1).

Tabela 4.1: Métodos de extração de características empregados no experimento.

Extrator	Descrição
Histograma RGB	Histograma gerado requantizando cada canal de cor RGB para 7 bins, resultando em um vetor de características com $7^3 = 343$ dimensões.
Histograma HSV	Histograma gerado requantizando cada canal de cor HSV para 7 bins, resultando em um vetor de características com $7^3 = 343$ dimensões.
Histograma HSV com informações espaciais	Vetor de característica gerado a partir da divisão da imagem em três faixas horizontais de mesma altura. De cada uma das faixas é calculado um histograma HSV local. O histograma HSV local é gerado requantizando cada canal de cor HSV para 5 bins resultando em um vetor de características com $3 \times 5^3 = 375$ dimensões.
SIFT	Histograma de palavras visuais (<i>bag-of-visual-words</i>) com 100 bins. O histograma é gerado por meio do extrator SIFT [60].
Gist	Vetor de características com 512 dimensões gerado utilizando o extrator Gist [72].
SFTA	Vetor de características de 21 dimensões gerado utilizando o extrator SFTA para textura [24].

Adicionalmente, foram empregados como dados de contexto *tags* de usuários associadas a cada uma das imagens do Flickr. Primeiramente foram selecionadas as *tags* nos conjuntos de dados com uma frequência mínima de 25. Em seguida, para cada imagem foi gerado um vetor binário onde cada atributo corresponde a uma *tag*. Se a *tag* está presente nos dados de contexto da imagem, então o atributo correspondente toma o valor um, caso contrário, o valor do atributo é zero. Uma vez que os dados de contexto utilizados neste experimento estão na forma de texto, nos resultados as características de contexto são apresentadas como características textuais.

4.4.2 Métodos de anotação

O *MFS-Map* foi comparado com os seguintes métodos de anotação de imagens: (i) classificador utilizando combinação a posteriori (**LF-SVM**); (ii) classificador utilizando combinação a priori (**EF-SVM**)

e (iii) modelo de vizinho mais próximo que anota a imagem com as anotações da imagem mais similar no conjunto de treinamento (**1-NN**).

Para o algoritmo *MFS-Map* o parâmetro *minConf* foi configurado para 0,5 por ter apresentado o melhor desempenho. Adicionalmente, para as características textuais (dados de contexto) e as características baseadas na abordagem *Bag-of-Features* o *MFS-Map* empregou a estratégia de geração de itens para dados de contexto (descrita na seção 4.1.2).

Para as abordagens baseadas em classificadores (EF-SVM e LF-SVM) foram empregados SVM com kernel do tipo *RBF* (*radial basis function*). Para encontrar os melhores parâmetros dos SVM (custo e grau do kernel *RBF*) foi empregada validação cruzada para cada SVM treinado. Adicionalmente, foi utilizado regressão linear para mapear os escores dos SVM para probabilidades. Essas probabilidades foram utilizadas para prever as relevâncias das anotações.

Para a abordagem 1-NN foi empregada a distância Euclidiana para todos os vetores de característica exceto o SIFT e as *tags* de usuários. Como ambas as representações são esparsas, a dissimilaridade foi calculada utilizando a distância cosseno. Todos os métodos de anotação foram implementados em C++. Para implementação das abordagens baseadas em classificadores foi utilizada a biblioteca LibSVM [16].

4.4.3 Precisão da anotação

Nesta seção são descritos os experimentos realizados com o propósito de avaliar a precisão do *MFS-Map* para a tarefa de anotação de imagens. A tabela 4.2 mostra os valores de AP e BEP obtidos considerando cada método para diferentes configurações: T, V e T+V. A coluna T mostra os resultados obtidos somente com características textuais. A coluna V mostra os resultados obtidos utilizando somente características visuais. A coluna V+T mostra os resultados obtidos utilizando a combinação das características textuais e visuais. O desvio padrão de cada medida é mostrado entre parênteses.

Tabela 4.2: Valores de AP e BEP obtidos por cada método para diferentes configurações: T, V e T+V. T corresponde aos resultados obtidos utilizando somente características textuais. V corresponde aos resultados obtidos utilizando somente características visuais. T+V corresponde aos resultados obtidos utilizando a combinação das características visuais e textuais. O desvio padrão das medidas é indicado entre parênteses.

Dataset	Método	T		V		V+T	
		AP (%)	BEP (%)	AP (%)	BEP (%)	AP (%)	BEP (%)
MIR Flickr	<i>MFS-Map</i>	54,0 (0,69)	47,3 (0,66)	41,1 (0,42)	39,3 (0,42)	55,1 (0,66)	52,0 (0,37)
	EF-SVM	29,9 (0,26)	26,9 (0,35)	28,5 (0,89)	25,0 (1,28)	29,9 (0,26)	27,0 (0,44)
	LF-SVM	29,9 (0,27)	26,9 (0,35)	29,9 (0,73)	27,0 (0,86)	29,9 (0,31)	26,9 (0,47)
	1-NN	30,8 (2,92)	28,3 (4,36)	35,1 (0,43)	37,7 (0,53)	30,9 (3,19)	28,5 (4,80)
Image CLEF	<i>MFS-Map</i>	48,1 (0,27)	45,5 (0,35)	42,9 (0,33)	40,2 (0,34)	49,4 (0,58)	47,6 (0,59)
	EF-SVM	37,7 (0,22)	37,2 (0,22)	36,8 (0,59)	35,7 (0,73)	37,8 (0,27)	37,2 (0,32)
	LF-SVM	37,8 (0,21)	37,2 (0,22)	37,7 (0,18)	37,2 (0,19)	37,8 (0,33)	37,2 (0,31)
	1-NN	38,2 (1,36)	37,9 (2,10)	40,2 (0,29)	41,6 (0,53)	38,1 (0,98)	37,7 (1,59)

Os resultados obtidos por cada método foram comparados considerando as configurações de características (V, T e V+T) e conjunto de dados (*MIR-Flickr* e *Image CLEF*). Para cada combinação de

configuração de característica e conjunto de dados (p. ex. V e *MIR-Flickr*), foi realizado um teste t de Student com duas caudas e significância $p=5\%$. Os melhores resultados são indicados em negrito. Para ambos os conjuntos de dados o *MFS-Map* obteve os melhores resultados com uma única exceção. O 1-NN obteve o melhor BEP para o conjunto *Image CLEF* utilizando características visuais (configuração V).

A Figura 4.4 mostra a precisão média (AP) obtida para cada anotação do conjunto *MIR Flickr*. Os resultados por anotação não são mostrados para o conjunto *Image CLEF* porque as 98 diferentes anotações não poderiam ser visualizadas apropriadamente em um único gráfico. O *MFS-Map* obteve a maior precisão média para todas as anotações com exceção de ‘Baby’, a qual é a anotação menos frequente no conjunto de dados. Esses resultados indicam que o maior valor de AP observado para o *MFS-Map* na Tabela 4.2 é também observado consistentemente para as anotações individualmente.

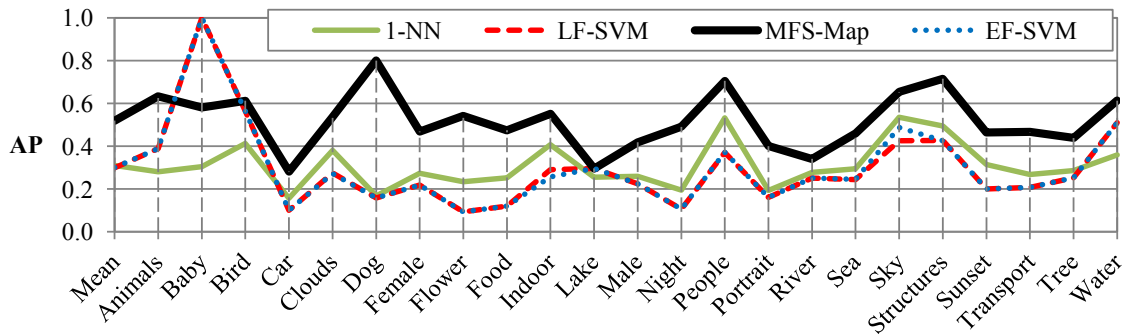


Figura 4.4: Precisão média (AP) obtida para cada anotação individualmente para o conjunto *MIR Flickr*.

4.4.4 Tempo de treinamento e anotação

Na Figura 4.5 é mostrado o gráfico do tempo total gasto por cada método para treinar e anotar as imagens. O tempo total é indicado no eixo vertical e é comparado com a precisão média (AP) no eixo horizontal. As medidas de tempo foram obtidas usando amostras aleatórias de 20% do conjunto *MIR Flickr* e de 40% do conjunto *Image CLEF*. Os experimentos foram executados em um computador com um processador Intel i7 de 2,66GHz, 8GB de RAM executando o sistema operacional Windows 64-bit. Os valores de AP podem diferir daqueles obtidos na seção 4.4.3 devido a amostragem das imagens.

Para ambos os conjuntos de dados o *MFS-Map* foi o método mais rápido. Adicionalmente, assim como ocorreu na seção 4.4.3, o *MFS-Map* apresentou a maior precisão média (AP). O método 1-NN não tem uma fase de treinamento, desta maneira o tempo medido para o 1-NN corresponde somente ao tempo de teste (anotação). Contudo, para os demais métodos o tempo de treinamento correspondeu a pelo menos 95% do tempo total. Para o *MFS-Map* o tempo de treinamento foi ao menos 48 vezes mais rápido que os métodos EF-SVM e LF-SVM para conjunto *Image CLEF*. Adicionalmente, para o conjunto *MIR-Flickr* o *MFS-Map* foi 3,6 vezes mais rápido que o EF-SVM e LF-SVM.

O maior tempo de treinamento das abordagens baseadas em classificadores (EF-SVM e LF-SVM) pode ser atribuído à necessidade de treinar um classificador para cada anotação. Assim, quando o tamanho do vocabulário de anotações é grande - 98 para o *Image CLEF* e 25 para o *MIR Flickr* - o tempo de treinamento aumenta significativamente.

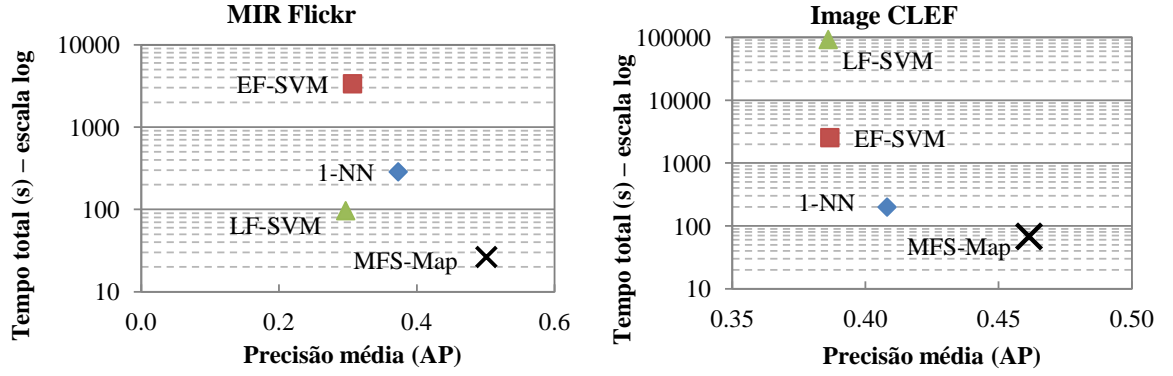


Figura 4.5: Precisão média (AP) contra o tempo de treinamento para cada método de anotação. O método desenvolvido (*MFS-Map*) é indicado pelo símbolo \times em preto.

4.5 Considerações

Nesta primeira etapa do projeto de pesquisa foi desenvolvido o método *MFS-Map*. O *MFS-Map* é utilizado para converter representações de conteúdo e de contexto de imagens para *itemsets*. Adicionalmente, foi descrito neste capítulo um método, também desenvolvido como parte do projeto de doutorado, para geração de regras de associação a partir dos *itemsets* do *MFS-Map*. As regras geradas foram aplicadas para a tarefa de anotação automática de imagens.

O *MFS-Map* foi avaliado empregando dois conjuntos de dados disponíveis publicamente: o *MIR-Flickr* (25,000 imagens) e o *Image CLEF* (18,000 imagens). Os resultados obtidos foram comparados com abordagens baseadas em classificadores utilizando combinação *a priori* e *a posteriori* e também um modelo de anotação baseado em vizinhos mais próximos. O *MFS-Map* obteve maior precisão e menores tempo de execução que os demais métodos, demonstrando ser um método promissor para abordar o problema proposto nesse projeto de doutorado.

Capítulo 5

Proposta de pesquisa

Devido a sua crescente importância, serviços de mídia social como Twitter, Flickr e Facebook são uma valiosa fonte de informações a partir da qual se pode realizar mineração de imagens. A mineração de imagens pode ser realizada empregando o conteúdo visual das imagens e dados de contexto associados. No entanto, tanto o conteúdo como contexto apresentam limitações. Em mineração de imagens o conteúdo das imagens é representado por meio de vetores de característica que sintetizam características visuais de baixo nível. Essas características de baixo nível nem sempre correspondem à semântica da imagem, acarretando o problema conhecido como lacuna semântica (*semantic gap*). Já os dados de contexto, fornecidos por usuários do serviço de mídia social, podem ser subjetivos ou incompletos. Contudo, dados de conteúdo e contexto carregam informações complementares. Assim, as informações dos dados de contexto poderiam ser utilizadas para complementar e corrigir erros provenientes da análise de dados conteúdo e vice-versa. Por este motivo, existe grande interesse em combinar dados de contexto e conteúdo para aprimorar a mineração de imagens em serviços de mídia social.

Contudo, a combinação de conteúdo e contexto apresenta desafios a serem tratados. Por exemplo, a abordagem de combinação *a priori* (*early fusion*) gera espaços de características de alta dimensionalidade. Assim, devido a maldição da dimensionalidade, os métodos de mineração de imagens têm seu desempenho prejudicado. Uma alternativa é a abordagem de combinação *a posteriori* (*late fusion*), que consiste em analisar o conteúdo e contexto das imagens separadamente e combinar os resultados da análise em uma etapa final da mineração. No entanto, como as duas modalidades de informação são analisadas separadamente, possíveis correlações úteis entre dados de contexto e conteúdo podem não ser detectadas.

5.1 Objetivos

Trabalhos recentes têm empregado a mineração de *itemsets* frequentes como uma ferramenta para encontrar correlações entre características visuais [34, 84, 71, 112]. Tais correlações são empregadas para gerar representações de imagens mais próximas da semântica das imagens. Este projeto de pesquisa tem como hipótese que a mineração de *itemsets* frequentes pode ser empregada para encontrar padrões não

somente entre características visuais, mas também entre dados de conteúdo e de contexto. Assim, este projeto de doutorado tem como objetivo desenvolver técnicas de combinação de dados de conteúdo e contexto de imagens utilizando como arcabouço teórico a mineração de *itemsets* frequentes. Para tanto, a pesquisa a ser realizada será dividida em três etapas principais:

1. Desenvolvimento de técnicas para conversão de representações de conteúdo e contexto para *itens de característica* e realizar a sua combinação empregando *itemsets de imagem*;
2. Aplicação de técnicas de mineração de *itemsets* frequentes e de regras de associação para analisar os *itemsets* de imagem. O objetivo é encontrar padrões entre os itens de característica derivados do conteúdo e do contexto da imagem;
3. Aplicação dos *itemsets* frequentes e regras de associação encontrados para minerar imagens de serviços de mídia social;

Os conceitos de *itens de característica* e de *itemsets de imagem* são definidos da seguinte forma:

Definição 5.1. *Itemsets de imagem* $\{f_1, f_2, \dots, f_n\}$ são representações do conteúdo e de contexto de imagens por meio de *itemsets*. Os *itemsets de imagem* são o resultado da combinação de itens de característica de conteúdo e contexto.

O restante deste capítulo apresenta o projeto de pesquisa. Na seção 5.2 é apresentada a metodologia a ser adotada para alcançar os objetivos propostos. Nas seções 5.6 e 5.7, são discutidas a metodologia de avaliação e cronograma de atividades.

5.2 Metodologia

Na seção 5.3 é discutida a abordagem a ser adotada para a extração dos itens de características e sua combinação em *itemsets* de imagem. Na seção 5.4 é apresentada a abordagem a ser adotada para realizar a mineração de *itemsets* frequentes e sua aplicação em mineração de imagens em serviços de mídia social. A seção 5.5 descreve o estágio de pesquisa a ser realizado com o propósito de minerar dados de contexto de grafos para mineração de imagens.

5.3 Representações por meio de *itemsets*

O objetivo desta etapa de pesquisa será o desenvolvimento de técnicas para representar imagens por meio de *itemsets*. A abordagem proposta, denominada de *itemsets de imagem*, tem como objetivo gerar itens a partir das representações dos dados de conteúdo e de contexto. A abordagem proposta para obtenção dos *itemsets* de imagem será composta das seguintes etapas:

1. Extração das representações de conteúdo e contexto das imagens;
2. Geração dos itens de característica a partir dos dados de conteúdo;
3. Geração dos itens de característica a partir dos dados de contexto;
4. Combinação dos itens de característica de conteúdo e contexto para gerar os *itemsets* de imagem.

Assim, uma característica importante da abordagem proposta é que os itens relativos aos dados de conteúdo são gerados separadamente dos itens relativos aos dados de contexto. A motivação para essa separação é a possibilidade de utilizar técnicas especializadas para gerar itens de contexto e de conteúdo, explorando as características de cada modalidade de informação.

Conforme discutido no Capítulo 4, a geração de itens de características a partir de dados de conteúdo está parcialmente desenvolvida. Para tanto foi proposto o método *MFS-Map* que explora o conceito de análise de subespaços de característica. Ou seja, ao invés de concatenar os vetores de características obtidos por diversos extratores em um único grande vetor, o *MFS-Map* divide as características visuais em subespaços. Esses subespaços são então divididos em regiões por meio de agrupamento de dados e os itens de características são gerados de acordo com sua região no subespaço de característica. Pretende-se aprimorar o *MFS-Map* estudando novas estratégias de geração de subespaços de características.

Para extração de características das imagens serão utilizadas implementações disponíveis em bibliotecas públicas, algoritmos desenvolvidos pelo doutorando durante seu projeto de doutorado e extratores desenvolvidos no GBdI do ICMC-USP, que é o grupo de pesquisa onde este projeto de doutorado está sendo realizado. O *MFS-Map* está sendo desenvolvido utilizando a linguagem de programação C++.

5.4 Mineração de *itemsets* frequentes e aplicação na mineração de imagens

O objetivo desta etapa de pesquisa será encontrar relações úteis entre itens de característica de conteúdo e de contexto. Adicionalmente, a mineração de *itemsets* frequentes será utilizada para descartar ruídos e correlações não úteis nos dados. As relações serão representadas tanto na forma de *itemsets* frequentes como também na forma de regras de associação. Pretende-se ainda avaliar e estudar quais métricas são mais adequadas para medir a utilidade dos *itemsets* minerados no contexto de mineração de imagens.

Para minerar grandes bases de imagens serão empregados algoritmos distribuídos. Para este propósito serão explorados algoritmos de mineração de *itemsets* frequentes distribuídos. Como plataforma de desenvolvimento pretende-se empregar o *framework map-reduce*).

5.5 Estágio e pesquisa no exterior

Um aspecto de grande importância dos serviços de mídia social é a possibilidade dos usuários conectarem entre si. Assim, um dado de contexto valioso para mineração de imagens são as redes sociais desses serviços. Com o propósito de desenvolver técnicas de mineração de imagens que possam explorar as informações contidas nas redes sociais, o aluno irá realizar um estágio de pesquisa na Carnegie Mellon University sob supervisão do Prof. Christos Faloutsos. O objetivo do estágio será o desenvolvimento de algoritmos escaláveis para descoberta de informações úteis a partir de redes sociais de serviços de mídia social. As informações mineradas serão utilizadas como dados de contexto para mineração de imagens.

A análise das redes sociais será realizada utilizando teoria dos grafos. Uma rede social online pode ser modelada como um grafo representando os usuários como vértices e relações de amizade como arestas. Deste grafo é possível encontrar informações úteis como grupos de usuários com forte conexões entre si (detecção de comunidades).

Grafos oriundos de serviços de mídia social são frequentemente compostos de milhões de vértices e bilhões de arestas [101]. Para realizar a análise de grafos desta escala serão desenvolvidos algoritmos que fazem uso de computação paralela utilizando o framework *map-reduce*. O desenvolvimento de algoritmos para mineração de grafo social utilizando *map-reduce* é importante porque muitos algoritmos não paralelos precisam ser reprojados para serem implementados utilizando funções map e reduce [21]

5.6 Validação e conjuntos de dados

A avaliação dos métodos de mineração de imagens desenvolvidos utilizará conjuntos de dados que contenham conteúdo e contexto de imagens provenientes de serviços de mídia social. Serão considerados tanto conjuntos de dados públicos como dados coletados diretamente de serviços de mídia social. Os conjuntos de dados públicos permitirão comparar os métodos desenvolvidos com os resultados publicados por outros autores. Exemplos de conjuntos de dados públicos relevantes para o desenvolvimento deste projeto incluem:

SNAP: O conjunto de grandes redes de Stanford (*Stanford Large Network Dataset Collection*¹) possui conjuntos de dados provenientes de serviços de mídia social.

MIR Flickr: Conjunto com 1 milhão de imagens provenientes do Website Flickr. Cada imagem possui *tags* (etiquetas) de usuários. Apesar de serem úteis para o processo de mineração de imagens, essas *tags* nem sempre são adequadas como *ground truth* para avaliação por apresentarem ruído. Para *ground truth* o MIR Flickr apresenta um subconjunto de 25 mil imagens anotadas manualmente. As anotações descrevem objetos presentes (ex. “carros”, “cão”, “flor”) e descrição da cena (ex. “por do sol”, “noite”).

Para coletar dados diretamente dos serviços de mídia social serão desenvolvidos *crawlers*. *Crawlers* são programas que realizam o download sistemático de dados disponibilizados publicamente por serviços de mídia social. Para o desenvolvimento serão utilizadas as *APIs* (*application programming interfaces*) dos serviços de mídia social.

Na etapa atual do projeto de doutorado está em desenvolvimento um *crawler* que utiliza a *API* pública do Twitter. O Twitter é um serviço de rede social no qual os usuários compartilham *tweets*: pequenos textos de no máximo 140 caracteres. Muitos usuários do serviço compartilham imagens, sobretudo fotografias capturadas por meio de dispositivos móveis.

O *crawler* em desenvolvimento coleta as imagens e dados de contexto que incluem informações do usuário, localização geográfica e texto dos *tweets*. Adicionalmente, o *Twitter* permite que usuários sigam uns aos outros, gerando um grafo social no qual os vértices correspondem aos usuários e as arestas às relações de seguir ou ser seguido.

Para percorrer o grafo social, o *crawler* toma como entrada um nome de usuário a partir do qual se iniciará a busca. Para cada vértice visitado são adicionados os vizinhos em uma fila de prioridades utilizada para determinar o próximo usuário a ser visitado. A prioridade dos vértices é calculada a partir do número de *tweets* enviados recentemente e grau de entrada e saída.

Os textos dos *tweets* coletados são processados para buscar por URLs de imagens. Caso uma URL de imagem seja encontrada, é feito o armazenamento local da imagem para análise do conteúdo. Em um

¹<http://snap.stanford.edu/data/>

teste inicial do *crawler* em desenvolvimento foram coletadas 20 mil imagens juntamente com os dados de contexto. A Figura 5.1 ilustra exemplos de imagens obtidas bem como o texto associado.



Figura 5.1: Exemplos de imagens obtidas por meio do *crawler* desenvolvido. Para cada imagem são mostrados seus respectivos *tweets* e palavras chaves.

5.7 Cronograma

Nesta seção é apresentado um resumo das principais atividades previstas este projeto de doutorado. Na Figura 5.2 é mostrado o cronograma das atividades planejadas bem como as atividades desenvolvidas até o momento. A seguir é descrita cada uma das atividades:

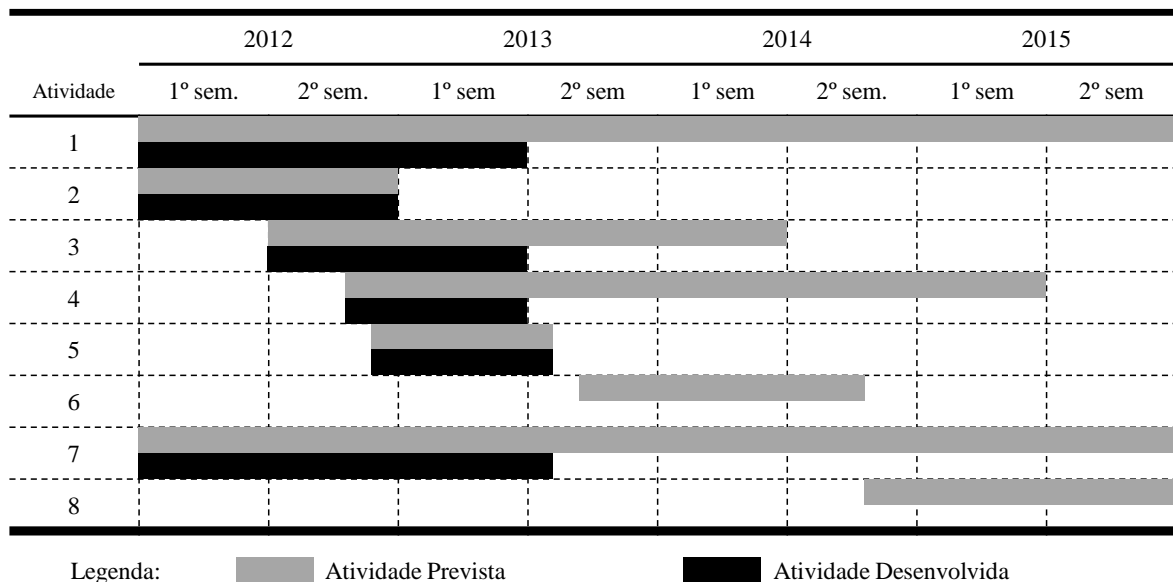


Figura 5.2: Cronograma de atividades planejadas e desenvolvidas.

1. Pesquisa bibliográfica e estudo da literatura relativa ao projeto. Esta atividade será realizada durante todo o período do projeto para acompanhar a literatura e trabalhos correlatos. As áreas

estudadas até o momento incluem anotação automática de imagens, mineração de dados na Web e aplicações de redes complexas para mineração de dados em redes sociais;

2. Cursar as disciplinas necessárias para o Programa de Doutorado do ICMC-USP. O doutorando já cursou as disciplinas necessárias;
3. Desenvolvimento de técnicas para obtenção de *itemsets* de imagens a partir de dados de conteúdo e de contexto;
4. Aplicação de técnicas de mineração de *itemsets* frequentes para combinar e encontrar padrões entre o conteúdo e o contexto de imagens. Utilização dos *itemsets* frequentes encontrados para minerar imagens de serviços de mídia social;
5. Elaborar a monografia para o Exame de Qualificação e submeter-se ao exame;
6. Realização de estágio de pesquisa no exterior na Carnegie Mellon University (CMU) sob supervisão do Prof. Christos Faloutsos;
7. Redação e submissão de artigos para veículos nacionais e internacionais seletivos e representativos da área;
8. Elaborar e defender a tese de doutorado.

Referências Bibliográficas

- [1] R. Agrawal, T. Imielinki, and A. Swami. Mining association rules between sets of items in large databases. *ACM SIGMOD Record*, pages 207–216, 1993.
- [2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *20th International Conference on Very Large Data Bases, VLDB*, pages 487–499, Santiago de Chile, Chile, 1994. Morgan Kaufmann.
- [3] M.L. Antonie and O.R. Zaïane. An associative classifier based on positive and negative rules. In *Proceedings of the 9th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 64–69, Paris, France, 2004. ACM.
- [4] Pradeep K. Atrey, M. Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, April 2010.
- [5] U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger. X-ray Categorization and Retrieval on the Organ and Pathology Level, Using Patch-Based Visual Words. *IEEE Transactions on Medical Imaging*, 30(3):733–746, 2011.
- [6] A. Backes and O. Bruno. Shape Skeleton Classification Using Graph and Multi-scale Fractal Dimension. *Image and Signal Processing*, (i):448–455, 2010.
- [7] A. G. R. Balan, A. J. M. Traina, C. Traina Jr., and P. M. A. Marques. Fractal analysis of image textures for indexing and retrieval by content. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 581–586, 2005.
- [8] E. Baralis and P. Garza. A lazy approach to pruning classification rules. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pages 35–42, IEEE Computer Society, 2002.
- [9] H. Becker, M. Naaman, and L. Gravano. Learning similarity metrics for event identification in social media. In *Proceedings of the third ACM international conference on Web search and data mining*, pages 291–300. ACM, 2010.
- [10] S. Beucher and C. Lantuejoul. Use of Watersheds in Contour Detection. In *International Workshop on Image Processing: Real-time Edge and Motion Detection/Estimation, Rennes, France.*, September 1979.
- [11] O. Boiman, E. Shechtman, and M. Irani. In defense of Nearest-Neighbor based image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, number i, pages 1–8. Ieee, June 2008.
- [12] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, September 2004.

- [13] Danah M. Boyd and Nicole B. Ellison. Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication*, 13(1):210–230, October 2007.
- [14] O. M. Bruno, R. O. Plotze, M. Falvo, and M. Castro. Fractal dimension applied to plant identification. *Information Sciences*, 178(12):2722–2733, June 2008.
- [15] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, November 1986.
- [16] C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1—27:27, 2011.
- [17] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikäinen, X. Chen, and W. Gao. WLD: a robust local image descriptor. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1705–20, September 2010.
- [18] Y. Q. Chen and M. S. Nixon. Statistical geometrical features for texture classification. *Pattern Recognition*, 28(4):537–552, 1995.
- [19] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval. In *IEEE 11th International Conference on Computer Vision (ICCV 2007)*, pages 1–8, Rio de Janeiro, 2007.
- [20] S. Clinchant, J. Ah-Pine, and G. Csurka. Semantic combination of textual and visual information in multimedia retrieval. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 44:1–44:8, Trento, Italy, 2011. ACM.
- [21] J. Cohen. Graph Twiddling in a MapReduce World. *Computing in Science & Engineering*, 11(4):29–41, July 2009.
- [22] M. L. Comer and E. J. Delp. Parameter estimation and segmentation of noisy or textured images using the EM algorithm and MPM estimation. *Image Processing, 1994. Proceedings.*, 2(31 7):650–654, 1994.
- [23] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.
- [24] A. F. Costa, G. Humpire-Mamani, and A. J. M. Traina. An Efficient Algorithm for Fractal Analysis of Textures. In *SIBGRAPI 2012 (XXV Conference on Graphics, Patterns and Images)*, pages 39–46, 2012.
- [25] A. F. Costa, J. Tekli, and A. J. M. Traina. Fast Fractal Stack: Fractal Analysis of Computed Tomography Scans of the Lung. In *MMAR’11: International ACM Workshop on Medical Multimedia Analysis and Retrieval*, pages 13–18, Scottsdale, AZ, USA, 2011. ACM.
- [26] L. F. Costa and R. M. Cesar Jr. *Shape Analysis and Classification: Theory and Practice*. CRC Press, Boca Raton, Florida, USA, 2001.
- [27] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 09(P09008), 2005.
- [28] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Image Retrieval: Ideas, Influences, and Trends of the New Age. *ACM Computing Surveys*, 40(2):1–60, 2008.
- [29] A Depeursinge and H Müller. Fusion techniques for combining textual and visual information retrieval. In *ImageCLEF: Experimental Evaluation in Visual Information Retrieval*, chapter 6, pages 95–114. Springer Berlin Heidelberg, 2010.
- [30] T. M. Deserno, S. Antani, and R. Long. Ontology of gaps in content-based image retrieval. *Journal of digital imaging*, 22(2):202–15, April 2009.

- [31] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.
- [32] A. X. Falcão, L. F. Costa, and B. S. Cunha. Multiscale skeletons by image foresting transform and its application to neuromorphometry. *Pattern Recognition*, 35:1571–1582, 2002.
- [33] M. Ferecatu and H. Sahbi. TELECOM ParisTech at ImageClefphoto 2008: Bi-Modal Text and Image Retrieval with Diversity Enhancement. In *Working notes in the CLEF 2008 Workshop*, pages 1–14, 2008.
- [34] B. Fernando, E. Fromont, and T. Tuytelaars. Effective use of frequent itemset mining for image classification. In *European Conference on Computer Vision (ECCV 2012)*, pages 214–227, 2012.
- [35] F. Figueiredo, H. Pinto, F. Belém, J. Almeida, M. Gonçalves, D. Fernandes, and E. Moura. Assessing the quality of textual features in social media. *Information Processing & Management*, 49(1):222–247, January 2013.
- [36] S. Fortunato, V. Latora, and M. Marchiori. Method to find community structures based on information centrality. *Physical review E*, 70(5):13, 2004.
- [37] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, February 2010.
- [38] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google file system. In *19th ACM Symposium on Operating Systems Principles*, pages 29–43, 2003.
- [39] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.
- [40] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, 3rd editio edition, 2007.
- [41] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. *ACM SIGMOD Record*, 29(2):1–12, 2000.
- [42] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, January 2007.
- [43] Y. Han. CPAR: Classification based on predictive association rules. In *Proceedings of SIAM international conference on data mining*, pages 331–335, 2003.
- [44] R. M. Haralick. Statistical and structural approaches to texture. In *Proceedings of the IEEE*, volume 67, pages 786–804, 1979.
- [45] H. He and E. A. Garcia. Learning from Imbalanced Data. *IEEE Trans. on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
- [46] W. Hsu, M. L. Lee, and J. Zhang. Image mining: trends and developments. *Journal of Intelligent Information Systems*, pages 7–23, 2002.
- [47] X. Hu. Text analytics in social media. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 385–414. Springer US, Boston, MA, 2012.
- [48] 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*, pages 39–43, Vancouver, Canada, 2008. ACM.
- [49] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666, June 2010.

- [50] U. Kandaswamy, D.A. Adjeroh, and M.C. Lee. Efficient Texture Analysis of SAR Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2075–2083, 2005.
- [51] U. Kang, C. E. Tsourakakis, and C. Faloutsos. PEGASUS: A Peta-Scale Graph Mining System Implementation and Observations. *2009 Ninth IEEE International Conference on Data Mining*, pages 229–238, December 2009.
- [52] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- [53] H. P. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1–58, March 2009.
- [54] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over Time: Densification Laws , Shrinking. In *Eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (KDD '05)*, pages 177–187, 2005.
- [55] J. Li and N. M. Allinson. A comprehensive review of current local features for computer vision. *Neuro-computing*, 71(10-12):1771–1787, June 2008.
- [56] X. Li and C. G. M. Snoek. Learning tag relevance by neighbor voting for social image retrieval. In *1st ACM International Conference on Multimedia information Retrieval*, 2008.
- [57] T. Lindeberg. Feature Detection with Automatic Scale Selection. *International Journal of Computer Vision*, 30(2):79 – 116, 1998.
- [58] B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 80–86, New York, NY, USA, 1998. AAAI Press.
- [59] D. G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV '99)*, pages 1150 – 1157, 1999.
- [60] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [61] Antonio Luz Jr. *Deteção de vídeos não colaborativos com base no conteúdo visual em redes sociais para compartilhamento de vídeo*. PhD thesis, Universidade Federal de Minas Gerais (UFMG), 2012.
- [62] M. Mampaey, N. Tatti, and J. Vreeken. Tell Me What I Need to Know: Succinctly Summarizing Data with Itemsets. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 573–581, New York, NY, USA, 2011. ACM.
- [63] B. S. Manjunath and W. Y. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.
- [64] R. Maree, P. Geurts, J. Piater, and L. Wehenkel. Random subwindows for robust image classification. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 34–40, 2005.
- [65] D. Marr and E. Hildreth. Theory of Edge Detection. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 207(1167):187–217, 1980.
- [66] J. L. Martínez-Fernández, A. García-Serrano, J. Villena, and V. Méndez-Sáenz. MIRACLE at ImageCLEF 2004. In *CLEF'04 Proceedings of the 5th conference on Cross-Language Evaluation Forum: multilingual Information Access for Text, Speech and Images*, pages 699–708. Springer-Verlag, 2004.

- [67] M. Mcglohon and C. Faloutsos. Statistical Properties of Social Networks. In Charu C. Aggarwal, editor, *Social Network Data Analytics*, chapter 2, pages 17–41. Springer US, Boston, MA, 2011.
- [68] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *International Journal of Computer Vision*, 60(1):63 – 86, 2004.
- [69] M. Naaman. Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications. *Multimedia Tools and Applications*, 56(1):9–34, May 2010.
- [70] M. E. J. Newman. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):16, 2004.
- [71] S. Nowozin, K. Tsuda, T. Uno, T. Kudo, and G. Bakir. Weighted Substructure Mining for Image Analysis. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, June 2007.
- [72] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
- [73] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.
- [74] T. Pang-Ning, M. Steinbach, and K. Vipin. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, first edit edition, 2005.
- [75] S. Papadopoulos, C. Zgkollis, Y. Kompatsiaris, and A. Vakali. Cluster-Based Landmark and Event Detection for Tagged Photo Collections. *IEEE Multimedia*, 18(1):52–63, January 2011.
- [76] G. V. Pedrosa and A. J. M. Traina. From Bag-of-Visual-Words to Bag-of-Visual-Phrases using n-Grams. In *SIBGRAPI 2013 (XXVI Conference on Graphics, Patterns and Images)*, pages 1–8, Arequipa, Peru, 2013.
- [77] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *17th international conference on World Wide Web (WWW '08)*, pages 91–100, New York, New York, USA, April 2008. ACM Press.
- [78] M. F. Porter. An algorithm for suffix stripping. In *Program: Electronic Library and Information Systems*, volume 14, pages 211–218. MCB UP Ltd, December 1980.
- [79] M. Rabbath, P. Sandhaus, and S. Boll. Automatic creation of photo books from stories in social media. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 7S(1):27:1–27:18, 2011.
- [80] M. Rabbath, P. Sandhaus, and S. Boll. Analysing Facebook features to support event detection for photo-based Facebook applications. In *2nd ACM International Conference on Multimedia Retrieval (ICMR '12)*, pages 11:1–11:8, 2012.
- [81] A. Rajaraman, J. Leskovec, and J. D. Ullman. *Mining of Massive Datasets*. 2013.
- [82] R. M. Rangayyan and T. M. Nguyen. Fractal analysis of contours of breast masses in mammograms. *Journal of Digital Imaging*, 20(3):223–37, September 2007.
- [83] N. Rasiwasia, J. Costa-Pereira, E. Coviello, G. Doyle, G. R.G. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. *Proceedings of the international conference on Multimedia - MM '10*, page 251, 2010.
- [84] M. X. Ribeiro, Pedro H. Bugatti, C. Traina Jr., P. M. A. Marques, N. A. Rosa, and A. J. M. Traina. Supporting content-based image retrieval and computer-aided diagnosis systems with association rule-based techniques. *Data & Knowledge Engineering*, 68(12):1370–1382, December 2009.

- [85] M. X. Ribeiro, A. J. M. Traina, C. Traina Jr., N. A. Rosa, and P. M. A. Marques. How to Improve Medical Image Diagnosis through Association Rules: The IDEA Method. In *21th IEEE International Symposium on Computer-Based Medical Systems*, pages 266–271, Jyväskylä, Finland, June 2008. IEEE.
- [86] X. Rui, M. Li, Z. Li, W. Y. Ma, and N. Yu. Bipartite graph reinforcement model for web image annotation. In *Computer Vision and Pattern Recognition, 2004 (CVPR 2004)*, page 585, New York, New York, USA, 2007. ACM Press.
- [87] E. R. S. Santos, A. P. B. Lopes, E. A. Valle, and A. A. Araújo. Vocabulários visuais para para recuperação de informação multimídia. In *Simpósio Brasileiro em Sistemas Multimídia e Web (WebMedia)*, pages 21–24, 2010.
- [88] A. Savasere, E. Omiecinski, and S. B. Navathe. An Efficient Algorithm for Mining Association Rules in Large Databases. In *21th International Conference on Very Large Data Bases (VLDB)*, pages 432–444. Morgan Kaufmann Publishers Inc., September 1995.
- [89] L. Shen, R. M. Rangayyan, and J. E. L. Desautels. Calcifications, Detection And Classification Of Mammographic. In K. W. Bowyer and S. Astley, editors, *State of the Art in Digital Mammographic Image Analysis*, pages 198–212. World Scientific Publishing Co., Inc., 1994.
- [90] B. Shevade, H. Sundaram, and L. Xie. Modeling personal and social network context for event annotation in images. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries (JCDL)*, pages 127–134, 2007.
- [91] J. Sivic and A. Zisserman. Video Google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477. IEEE, 2003.
- [92] Ingrid Sluimer, Arnold Schilham, Mathias Prokop, and Bram van Ginneken. Computer analysis of computed tomography scans of the lung: a survey. *IEEE Transactions on Medical Imaging*, 25(4):385–405, April 2006.
- [93] S. Tamura, K. Tamura, H. Kitakami, and K. Hirahara. Clustering-based burst-detection algorithm for web-image document stream on social media. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 703–708. IEEE, October 2012.
- [94] P. N. Tan, V. Kumar, and J. Srivastava. Selecting the right interestingness measure for association patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, page 32, New York, New York, USA, 2002. ACM Press.
- [95] Fadi Thabtah. A review of associative classification mining. *The Knowledge Engineering Review*, 22(01):37, May 2007.
- [96] F. Timm and T. Martinetz. Statistical Fourier Descriptors for Defect Image Classification. In *International Conference on Pattern Recognition*, volume 1, pages 4198–4201, Istanbul, Turkey, 2010. IEEE.
- [97] R. S. Torres and A. X. Falcão. Content-based image retrieval: theory and applications. *Revista de Informática Teórica e Aplicada*, 2(13):161–185, 2006.
- [98] R. S. Torres, A. X. Falcão, and L. F. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, June 2004.
- [99] T. Tuytelaars and K. Mikolajczyk. Local Invariant Feature Detectors: A Survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.

- [100] Y. Uchiyama, S. Katsuragawa, H. Abe, J. Shiraishi, F. Li, Q. Li, C. T. Zhang, K. Suzuki, and K. Doi. Quantitative computerized analysis of diffuse lung disease in high-resolution computed tomography. *Medical Physics*, 30(9):2440, 2003.
- [101] J. Ugander, B. Karrer, L. Backstrom, and C. Marlow. The Anatomy of the Facebook Social Graph. *Computer Research Repository (CoRR) - arXiv*, pages 1–17, 2011.
- [102] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *IEEE transactions on pattern analysis and machine intelligence*, 31(11):2032–2047, November 2009.
- [103] L. Vendramin, R. J. G. B. Campello, and E. R. Hruschka. Relative clustering validity criteria: A comparative overview. *Statistical Analysis and Data Mining*, 3(4):209–235, 2010.
- [104] J. Verbeek, M. Guillaumin, T. Mensink, and C. Schmid. Image annotation with tagprop on the mirflickr set. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pages 537–546. ACM, 2010.
- [105] A. F. G. Viera and J. Virgil. Uma revisão dos algoritmos de radicalização em língua portuguesa. *Information Research*, 12(3):8, 2007.
- [106] J. Villena-Román, S. Lana-Serrano, and J. C. González-Cristóbal. MIRACLE at ImageCLEFmed 2007: Merging textual and visual strategies to improve medical image retrieval. In *Working Notes of the 2007 CLEF Workshop*, Budapest, Hungary, 2007.
- [107] G. Wang, D. Hoiem, and D. Forsyth. Building text features for object image classification. In *19th International Conference on Pattern Recognition*, pages 1367–1374, Tampa, Florida, USA, June 2009. Citeseer.
- [108] S.-Y. Wang, W.-S. Liao, L.-C. Hsieh, Y.-Y. Chen, and W. H. Hsu. Learning by expansion: Exploiting social media for image classification with few training examples. *Neurocomputing*, 95(null):117–125, October 2012.
- [109] X. Wang, X. Ding, and C. Liu. Gabor filters-based feature extraction for character recognition. *Pattern Recognition*, 38(3):369–379, 2005.
- [110] D. J. Watts. *Small Worlds: The Dynamics of Networks between Order and Randomness*. Princeton University Press, Princeton, NJ, USA, 2003.
- [111] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.
- [112] B. Yao and L. Fei-Fei. Grouplet: A structured image representation for recognizing human and object interactions. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 9–16. IEEE, June 2010.
- [113] D. Zhang. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.
- [114] D. Zhang, Md. M. Islam, and G. Lu. A review on automatic image annotation techniques. *Pattern Recognition*, 45(1):346–362, May 2011.
- [115] H. Zhang, A.C. Berg, M. Maire, and J. Malik. SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2126–2136. IEEE, 2006.
- [116] Hui Zhang, Jason E. Fritts, and Sally a. Goldman. Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260–280, May 2008.

- [117] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, 73(2):213–238, 2005.
- [118] Shaoting Zhang, J Huang, Yuchi Huang, and Yang Yu. Automatic image annotation using group sparsity. In *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3312–3319. Ieee, June 2010.
- [119] Q. Zhuang, J. Feng, and H. Bao. Measuring Semantic Gap: An Information Quantity Perspective. In *5th IEEE International Conference on Industrial Informatics*, volume 100101, pages 669–674, Vienna, 2007. IEEE.
- [120] A. Znaidia, A. Shabou, A. Popescu, H. le Borgne, and C. Hudelot. Multimodal Feature Generation Framework for Semantic Image Classification. In *ICMR '12 Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*, number c, pages 38:1—38:8, 2012.