Pós-Graduação em Ciência da Computação

Marcondes Ricarte da Silva Júnior

MAPAS AUTO-ORGANIZÁVEIS PROBABILÍSTICOS PARA CATEGORIZAÇÃO DE LUGARES BASEADA EM OBJETOS

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
www.cin.ufpe.br/~posgraduacao

RECIFE
2016

Marcondes Ricarte da Silva Júnior

MAPAS AUTO-ORGANIZÁVEIS PROBABILÍSTICOS PARA CATEGORIZAÇÃO DE LUGARES BASEADA EM OBJETOS

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientador: *Aluizio Fausto Ribeiro Araújo*

RECIFE
2016

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

S586m Silva Júnior, Marcondes Ricarte da
Mapas auto-organizáveis probabilísticos para categorização de lugares baseada em objetos / Marcondes Ricarte da Silva Júnior. – 2016.
115 f.: il., fig., tab.

Orientador: Aluizio Fausto Ribeiro Araújo.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2016.
Inclui referências.

1. Inteligência artificial. 2. Mapas probabilísticos. I. Araújo, Aluizio Fausto Ribeiro (orientador). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2017-24

Marcondes Ricarte da Silva Júnior

Mapas Auto-Organizáveis Probabilísticos para Categorização de Lugares Baseada em Objetos

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação

Aprovado em: 30/08/2016.

BANCA EXAMINADORA

_____

Prof. Dr. Hansenclever de França Bassani
Centro de Informática / UFPE

_____

Prof. Dr. Guilherme de Alencar Barreto
Departamento de Engenharia e Teleinformática / UFC

_____

Prof. Dr. Aluizio Fausto Ribeiro Araújo
Centro de Informática / UFPE
(**Orientador**)

Aos meus Pais,
Marcondes e Raquel Ricarte,
com muito carinho.

Agradecimentos

Inicialmente agradeço a Deus por cuidar-me tão bem de mim por toda a vida. Acredito que Ele me ama com um amor infinito e incondicional, e grato sou por isso. Obrigado dar-me a oportunidade e talentos.

Agradeço ao Profº Aluizio Araújo por ter sido meu orientador, companheiro, incentivador, carrasco e mediador. Sei que cada papel que ele executou neste tempo foi para que eu tivesse a melhor formação e desenvolvesse este trabalho da melhor maneira possível. Obrigado por me confiar este trabalho.

Não tenho palavras para agradecer aos meus pais, Marcondes e Raquel Ricarte, e minha irmã Rebbeca Ricarte. Apoiar o outro no que se acredita é fácil, contudo quando se apoia algo que aparentemente seria uma loucura ou que eu não poderia provar que daria certo, neste momento pude sentir o respeito e confiança que tinham por mim. Mãe, obrigado por confiar em mim e me apoiar mesmo quando sua intuição não apostava nisto. Pai, obrigado por toda minha vida me acompanhar e me incentivar em todas as minhas atividades. Agradeço a Rebbeca, a criatura que mais me fez *bullying* em todo a minha vida. Você faz parte do que eu sou hoje e sempre serei, sem você eu não seria o mesmo. Agradeço a Bruna Estumano pelo companheirismo, compreensão, dedicação, carinho e paciência. Obrigado a Vovó Iolanda, minhas tias, meus tios, primos e primas, vocês também fazem parte disto.

Agradeço a Elizabeth Carvalho que ficou ao meu lado no meus momentos mais insanos (literalmente). Todas as suas observações, intervenções, conselhos, sugestões foram ouvidas com muita atenção e zelo. Sou grato a Franscilê Souza pela incentivo e palavras de discernimento quando o plano para esta dissertação não passava de uma conjectura. Obrigado a Tallyta Miranda e Larissa Bertulino pela cooperação na escrita do texto, algumas vezes nem mesmo eu entendia o que eu tinha escrito.

Obrigado aos meus amigos do Laboratório: André Tiba, Orivaldo Vieira, Flávia Araújo, Juracy França, Paulo Ferreira, Monique Soares, Elaine Guerreiro, Felipe Duque, Hesdras Viana e Tadeu Souza. Pelas conversas, sugestões e vivência na realização deste trabalho. Por fim, agradeço ao SVN por ter me salvado das minhas maiores burradas.

"A local villager knows his way by wont and without reflection to the village church, to the town hall, to the shops and back home again from the personal point of view of one who lives there. But, asked to draw or to consult a map of his village, he is faced with learning a new and different sort of task: one that employs compass bearing and units of measurement. What was first understood in the personal terms of local snapshots now has to be considered in the completely general terms of the cartographer. The villager's knowledge by wont, enabling him to lead a stranger from place to place, is a different skill from one requiring him to tell the stranger, in perfectly general and neutral terms, how to get to any of the places, or indeed, how to understand these places in relation to those of other villages."

—GILBERT RYLE (Abstractions)

Resumo

Os robôs móveis estão cada vez mais inclusos na sociedade moderna podendo se locomover usando “coordenadas cartográficas”. No entanto, com o intuito de aperfeiçoar a interação homem-robô e a navegação das máquinas nos ambientes, os robôs podem dispor da habilidade de criar um Mapa Semântico realizando Categorização dos Lugares. Este é o nome da área de estudo que busca replicar a habilidade humana de aprender, identificar e inferir os rótulos conceituais dos lugares através de sensores, em geral, câmeras.

Esta pesquisa busca realizar a Categorização de Lugares baseada em objetos existentes no ambiente. Os objetos são importantes descritores de informação para ambientes fechados. Desse modo as imagens podem ser representadas por um vetor de frequência de objetos contidos naquele lugar. No entanto, a quantidade de todos possíveis tipos de objetos existentes é alta e os lugares possuem poucos destes, fazendo com que a representação vetorial de um lugar através de objetos contidos nele seja esparsa.

Os métodos propostos por este trabalho possuem duas etapas: Redutor de Dimensionalidade e Categorizador. A primeira se baseia em conceitos de Compressão de Sinais, de Aprendizagem Profunda e Mapas Auto-Organizáveis (SOMs), a fim de realizar o pré-processamento dos dados de frequência de objetos para a redução da dimensionalidade e minimização da esparsidade dos dados. Para segunda etapa foi proposto o uso de múltiplos Mapas Auto-Organizáveis Probabilísticos (PSOMs). Os experimentos foram realizados para os métodos propostos por esse trabalho e comparados com o Filtro Bayesiano, existente na literatura para solução desse problema. Os experimentos foram realizados com quatro diferentes bases de dados que variam em ordem crescente de quantidade de amostras e categorias. As taxas de acerto dos métodos propostos demonstraram ser superiores à literatura quando o número de categorias das bases de dados é alta. Os resultados para o Filtro Bayesiano degeneram para as bases com maiores quantidade de categorias, enquanto para os métodos propostos por essa pesquisa as taxas de acerto caem mais lentamente.

Palavras-chave: Categorização de Lugares. Redução de Dimensionalidade. Aprendizado Profundo. Mapas Auto-Organizáveis Probabilísticos. Dados Esparsos.

Abstract

Mobile Robots are currently included in modern society routine in which they may move around often using "cartographic coordinates". However, in order to improve human-robot interaction and navigation of the robots in the environment, they can have the ability to create a Semantic Map by Categorization of Places. The computing area of study that searches to replicate the human ability to learn, identify and infer conceptual labels for places through sensor data, in general, cameras is the Place Categorization.

These methods aim to categorize places based on existing objects in the environment which constitute important information descriptors for indoors. Thus, each image can be represented by the frequency of the objects present in a particular place. However, the number of all possible types of objects is high and the places do have few of them, hence, the vector representation of the objects in a place is usually sparse.

The methods proposed by this dissertation have two stages: Dimensionality reduction and categorization. The first stage relies on Signal Compression concepts, Deep Learning and Self-Organizing Maps (SOMs), aiming at preprocessing the data on object frequencies for dimensionality reduction and minimization of data sparsity. The second stage employs Probabilistic Self-Organizing Maps (PSOMs). The experiments were performed for the two proposed methods and compared with the Bayesian filter previously proposed in the literature. The experiments were performed with four different databases ranging considering different number of samples and categories. The accuracy of the proposed methods was higher than the previous models when the number of categories of the database is high. The results for the Bayesian filter tends to degrade with higher number of categories, so do the proposed methods, however, in a slower rate.

Keywords: Place Categorization. Dimensionality Reduction. Deep Learning. Probabilistic Self-Organizing Maps. Sparse Data.

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

PbSOM *Probabilistic Self-Organizing Map* 51

FDP Função de Densidade de Probabilidade 50

PSM Mapa Semântico Probabilístico 32

PSOM Mapa Auto-Organizável Probabilístico 24

MI-SVM Multiplas Máquinas de Vetores de Suporte

MRF *Markov Random Field* 36

NN *Nearest Neigbhbor* 33

NGN *Neural Gas Network* 31

RANSAC *Random Sample Consensus* 34

RGHSOM Robust Growing Hierarchical Self-Organizing Map 76

RVM *Relevance Vector Machines* 36

SAE *Stacked Auto-Encoders* 43

SOCEM *Self-Organizing Conditional Expectation Maximization* 52

SODAEM *Self-Organizing Deterministic Annealing Expectation Maximization* 52

SOEM *Self-Organizing Expectation Maximization* 52

SA Aproximação Estocástica 50

SLAM Localização e Mapeamento Simultâneos 20

SIFT *Scale-Invariant Feature Transform* 20

SOM Mapa Auto-Organizável 24

SURF *Speeded Up Robust Features* 20

STSOM Short Term Self Organizing Map 76

SVM Máquina de Vetores de Suporte 35

TDP *Transformed Dirichlet Process* 20

TF-IDF *Term Frequency Inverse Document Frequency* 36

TSOM *t-Student Self-Organizing Map* 51

VO Odometria Visual 31

VW *Visual Words* 31

UGM *Undirected Graphical Model* 36

Sumário

1

Introdução

Os robôs móveis possuem uma longa história que iniciou na década de 60. O Shakey foi o primeiro robô móvel do mundo de propósito geral a ser capaz de decidir individualmente em cada etapa sobre as suas próprias ações. Os robôs, antes do Shakey, teriam de ser instruídos por completo sobre cada etapa para a execução de uma tarefa maior. Já o Shakey poderia analisar comandos e dividi-los em partes básicas por si só. Ele foi desenvolvido no Artificial Intelligence Center da SRI (Stanford Research Institute) (NILSSON, 1984; NATTHARITH, 2011). O robô foi equipado com vários sensores e sua execução ficou sob responsabilidade de um programa de resolução de problemas chamado *STRIPS*. Para isto, foram usados algoritmos de percepção, modelagem de mundo e controle de atuadores. Outro exemplo de um robô móvel foi o CART, desenvolvido em 1977, na Universidade de Stanford, pelo Hans Moravec em sua tese de doutorado (MORAVEC, 1983). Contudo, CART era muito lento, devido ao alto tempo de processamento para dinâmica do mundo real. Outro pioneiro é o robô Rover, desenvolvido na Carnegie Mellon University (CMU)), no início dos anos 80, que foi equipado com uma câmera e possuía mais poder de processamento que CART. No entanto, a sua capacidade percepção do mundo real, processamento de tarefas e ativação de atuadores ainda eram muito lentos para aplicações em tempo real (GUZEL, 2013).

Os robôs pioneiros tinham algoritmos de Visão Computacional muito simples principalmente por causa do baixo poder de processamento dos computadores de então. Com o passar do tempo, as principais áreas de conhecimento envolvidas na robótica têm passado por inovações tecnológicas rápidas e expressivas (GUZEL, 2013). Particularmente, as tecnologias nas áreas de computação, telecomunicações e dispositivos eletrônicos avançaram no desenvolvimento de sensores inteligentes, de atuadores e no planejamento e tomada de decisão. Estes avanços melhoraram significativamente as capacidades e flexibilidade dos robôs móveis (GUZEL, 2013).

Os animais possuem a habilidade de se locomover através do ambiente ao seu redor de uma forma simples, robusta e autônoma seja este ambiente previamente conhecido ou não. Os humanos dotados da capacidade de criar rotas em ambientes dinâmicos e a aprender rapidamente novas rotas para ir de um lugar para outro, eles têm a capacidade de navegar pelo ambiente. Informações de rotas e representações estruturais formadas no cérebro são organizadas formando

um mapa cognitivo (TRULLIER et al., 1997). A navegação autônoma busca replicar essa habilidade de percepção de ambientes e a elaboração de mapas e rotas que os animais possuem. O objetivo é permitir que os robôs se movam de forma independente e robusta através do ambiente. Uma das áreas da computação que contribui para navegação autônoma é Visão Computacional, em particular pode ser útil para robôs móveis (BONIN-FONT; ORTIZ; OLIVER, 2008).

Navegação robótica é a capacidade que o robô tem para determinar sua própria posição em seu plano de referência e, em seguida, planejar um caminho para ponto objetivo (CHOSET et al., 2005). Diferentes sensores podem ser empregados para o propósito de navegação como infra-vermelho, sonar e câmeras. Em particular, a navegação baseada em visão recebeu muitas contribuições da comunidade científica nas últimas três décadas.

Pode-se dividir basicamente a navegação robótica de acordo com o ambiente: terrestre, aérea e aquática. Cada uma dessas navegações tem características únicas e também restrições diferenciadas. Por exemplo, os veículos aéreos que não possuem limitação de campo de visão como os terrestres e desenvolvem uma maior velocidade de deslocamento, portanto, eles necessitam de uma resposta de processamento mais eficiente para não colidirem com um obstáculo (BONIN-FONT; ORTIZ; OLIVER, 2008; GUZEL, 2013). Outra categoria de subdivisão é a navegação em ambientes fechados e abertos. Ambientes fechados tendem a ter uma maior regularidade e estrutura mais bem definida do que ambientes abertos.

O Mapeamento Semântico amplia as abordagens de construção de mapas para levarem conta não os componentes existentes no ambiente (paredes, obstáculos, piso) referenciados por coordenadas cartesianas, mas também com um semântica relacionada ao ambiente que está sendo mapeado. Aquisição das imagens para o Mapeamento Semântico pode utilizar diferentes tipos de câmeras, tais como monocular (VISWANATHAN et al., 2010; MADOKORO; UTSUMI; SATO, 2012), estéreo (CADENA et al., 2012; KOSTAVELIS; NALPANTIDIS; GASTERATOS, 2012; SENGUPTA et al., 2013), omnidirecional (WANG; LIN, 2011) e com sensor de profundidade (KOSTAVELIS; GASTERATOS, 2013; OLIVER et al., 2012). As imagens são matrizes de *pixels* que fornecem propriedades como forma, cor e textura. Para extração e transformação dos dados dessas matrizes em características, empregam-se algoritmos como Descritor de Gist (OLIVA; TORRALBA, 2001), *Scale-Invariant Feature Transform* (SIFT) (LOWE, 2004), *Speeded Up Robust Features* (SURF) (BAY et al., 2008). A partir desses conjuntos de características, representações de objetos e lugares são construídos. Para tal, alguns algoritmos empregados são *Bag of Visual Words* (BoVW) (YANG et al., 2007), *Deformable Part Models* (DPM) (FELZENSZWALB et al., 2013) e *Transformed Dirichlet Process* (TDP) (SUDDERTH et al., 2008).

1.1 Mapeamento e Navegação Semântica

O Localização e Mapeamento Simultâneos (SLAM) é um processo importante no desenvolvimento para robôs móveis. Esse processo fez grandes avanços no contexto de orientação

cartográfica, através de coordenadas, bússola ou GPS, no qual os robôs móveis atuais se comportam num sentido restrito, como máquinas cartográficas se deslocando no mundo real, incapazes de assegurar uma ligação do mapeamento/navegação pelo ambiente com as atividades para qual o robô foi projetado no contexto dos humanos. Várias abordagens de mapeamento existentes se destinam a construir um mapa métrico consistente para que o robô utilize em sua navegação. Os robôs atuais utilizam modelos de mapeamento que suprem a necessidade de descobrir ou inferir a sua própria localização no mapa construído em relação a ele. Dessa forma, é possível detectar o seu posicionamento real no mundo com boa precisão. Através dessa capacidade, os robôs podem planejar um caminho e navegar em direção a um objetivo que pode ser definido por uma posição métrica no referencial do mapa. Entretanto, para um robô aprender sobre o ambiente onde está inserido de maneira similar aos humano, faz-se necessário o entendimento do contexto em que o robô está inserido e as entidades com que pode se relacionar. Os robôs devem poder identificar ou reconhecer o contexto de uma cozinha, um escritório ou um quarto. Logo, mapas aumentados por atributos semânticos relativos a conceitos podem determinar tal contexto. Tal capacidade de reconhecimento é essencial para robôs que atuem em ambientes de humanos executando tarefas cooperativas (KOSTAVELIS; GASTERATOS, 2015).

Mapeamento Semântico pode ser definido como um processo de construção de uma representação de um ambiente que associa conceitos semânticos às entidades espaciais. O Mapa é construído através de rótulos semânticos dentro do ambiente onde o robô está imerso. Um Mapa Semântico é ilustrado na Figura 1.1. O resultado do Mapeamento Semântico deve ser idealmente uma representação completa do ambiente. Essa representação não deve conter apenas as informações semânticas, mas representar explicitamente as entidades espaciais para que a semântica seja bem estruturada para a navegação. O robô pode não ter conhecimento prévio do ambiente e as informações virem diretamente dos seus sensores de visão (câmeras) a partir de observações reais do ambiente. Ele pode incorporar conhecimento a partir de um conjunto de modelos categóricos e conceituais adquiridos em outros ambientes, ou a partir de bancos de dados (PRONOBIS, 2011).

O Mapa Semântico é uma descrição qualitativa do ambiente onde o robô está inserido, com o objetivo de aumentar as capacidades de navegação e a tarefa de planejamento e execução do robô, assim como para preencher as lacunas existentes neste campo em relação a Interação Humano-Robô (HRI) (PRONOBIS; JENSFELT, 2012; KOSTAVELIS; GASTERATOS, 2013; ZENDER et al., 2008). O mapeamento semântico tem como objetivo identificar, reconhecer e memorizar estruturas e símbolos que tenham significado no ambiente físico para os humanos, além de como essas estruturas e símbolos interagem entre si. Então, um mapa semântico possui tanto informações georreferenciadas do ambiente, quanto características qualitativas de alto nível. Projetar robôs capazes de perceber, memorizar e recordar semanticamente o ambiente em que estão inseridos com precisão é um elo fundamental de comunicação entre humanos e robôs. Portanto, para uma HRI bem sucedida, os robôs devem possuir as capacidades cognitivas de interpretação sobre o espaço, ou seja, eles devem envolver atributos semânticos sobre os objetos e

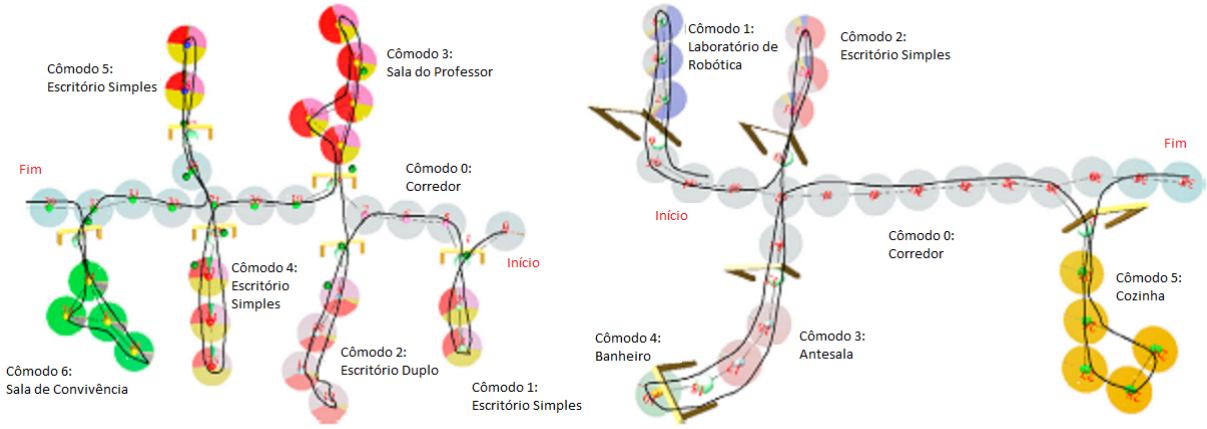

Figura 1.1: Exemplos de mapas semânticos baseados em mapas topológicos que indicam a probabilidade dos rótulos dos cômodos (PRONOBIS, 2011).

os locais encontrados, em associação com a percepção geométrica do ambiente (KOSTAVELIS; GASTERATOS, 2015).

Um mapa semântico compreende características de alto nível que modelam os conceitos humanos sobre lugares, objetos, formas e as relações de todos esses elementos. Os mapas semânticos estendem os mapas métricos, com rótulos dos lugares sobre as características geométricas que o robô deve estar ciente para navegar com segurança nos seus arredores. Contudo, há trabalhos com Mapas Semânticos que não usam um mapa métricos para determinar os tipos dos lugares, como abordagens que empregam mapas topológicos ou mapas híbridos (PRONOBIS et al., 2006; RANGANATHAN; DELLAERT, 2007; KOSTAVELIS; GASTERATOS, 2015). Mapas híbridos tentam conciliar algumas características dos mapas métricos e dos topológicos em apenas um.

Empregando Mapas Semânticos e Topológicos, a Navegação Semântica pode auxiliar atividades, como planejamento de rotas, desvio de obstáculos, exploração autônoma ou mesmo manipulação de objetos. Essa navegação permite saber o posicionamento do robô no mundo, e ser capaz de explorar o ambiente ou encontrar rotas para locais conhecidos ou desconhecidos. Várias tarefas de um robô dependem da capacidade de perceber e compreender o espaço onde estão inseridos (PRONOBIS, 2011).

Em ambientes internos, o cenário fornece informação semântica valiosa, pois pode obter informações padronizadas da arquitetura. Na verdade, a capacidade de compreender a semântica do espaço e poder rotular por termos semânticos como corredor ou escritório, possibilita uma representação mais intuitiva do ambiente em que o robô está posicionado. Se projetado com tais conceitos semânticos como forma de sala, tamanho, aparência ou presença de objetos, a representação do conhecimento espacial do robô se torna mais adequada do ponto de vista do seu desempenho em tarefas complexas e com a interação humana. Por exemplo, um robô para ambientes *indoor*, cuja tarefa seja encontrar objetos, poderia aumentar significativamente o seu nível de reconhecimento, considerando o tipo semântico do local em que estiver inserido

correlacionando com a localização do objeto a ser reconhecido (PRONOBIS, 2011).

1.1.1 Representações de Mapas

Os mapas topológicos podem ser expressos por um grafo, e são mais compactos do que os mapas métricos que gravam todas as informações métricas do ambiente. A busca em grafos tem algoritmos simples e eficientes na literatura para determinação de caminhos, como: Algoritmo Dijkstra (DIJKSTRA, 1959) e A* (HART; NILSSON; RAPHAEL, 1968). Os nós do grafo correspondem aos lugares do mapa e as arestas correspondem às vias para deslocamentos. Por meio de um mapa topológico, o ambiente deve ser mapeado de forma a manter tanto algumas informações geométricas para o conjunto de lugares mapeados, quanto informações conceituais sobre a categoria daquele lugar. Portanto, tais grafos se constituem em estruturas fundamentais para Mapeamento Semântico, pois permitem uma abstração maior que os mapas métricos (KOSTAVELIS; GASTERATOS, 2015). Pode-se ver um mapa semântico baseado em um mapa topológico na Figura 1.1.

1.1.2 Categorização de Lugares

O ponto de restrição destas abordagens de mapeamento é que os robôs entendem o ambiente como se fossem máquinas cartográficas (KOSTAVELIS; GASTERATOS, 2015) sem considerar o contexto. O Mapeamento Semântico propõe que o ambiente seja mapeado através de um contexto de alto nível se aproximando da linguagem humana para descrição de um ambiente. Para isto, podem ser empregados conceitos de Categorização de Lugares e Reconhecimento de Objetos. No contexto de Mapeamento Semântico, o reconhecimento de objetos se trata de rotular os itens (mesa, cadeira, livro, copo) existentes em um determinado cômodo. Já a Categorização de Lugares é o entendimento do contexto do que são os cômodos e propõe a rotulação destes em categorias, como: cozinha, quarto, biblioteca, escritório, dentre muitos outros. O Mapeamento Semântico emprega a Categorização de Lugares para construir a representação semântica do lugar que está sendo mapeado. Um desdobramento do Mapeamento Semântico é a Navegação Semântica, que faz uso do mapeamento para que um robô possa se deslocar pelo ambiente, neste caso, entendendo o contexto e podendo executar tarefas de alto nível de complexidade como se deslocar até a biblioteca e encontrar um livro. O entendimento semântico do ambiente viabiliza o desenvolvimento da HRI, tornando mais natural a comunicação entre humanos e robôs.

Um importante conceito em ambientes fechados é o de cômodo, que são as áreas delimitadas dos ambientes. Os cômodos tendem a compartilhar funcionalidades semelhantes que podem ser categorizadas, bem como outras propriedades espaciais. Na maioria dos casos, os cômodos são naturalmente categorizados com base em sua funcionalidade e podem ser descritos em termos dos conceitos semânticos discretos, tais como: quarto, corredor, escritório, cozinha, garagem, biblioteca ou laboratório. Cômodos também podem ser associados a conceitos que descrevem suas propriedades espaciais: área quadrada ou alongada. No entanto, as descrições

semânticas mais elaboradas podem ser usadas para refinar o processo. Reconhecimento de objetos e/ou marcadores de referência do ambiente são informações importantes que podem facilitar o processo (PRONOBIS, 2011). O processo de Categorização de Lugares é o responsável por determinar a categoria do cômodo em análise para o Mapeamento Semântico. Essa área do conhecimento será devidamente tratada no Capítulo 2 apresentando as diversas abordagens existentes na literatura para tal fim e apresentando alguns trabalhos relacionados a este.

1.2 Objetivos

O campo de estudo de Mapeamento Semântico é muito amplo, por isso, esta pesquisa restringi-se no problema de Categorização de Lugares no contexto de Mapeamento Semântico, ou seja, na determinação da categoria para os cômodos de um ambiente interno considerando as limitações de um robô móvel. A abordagem escolhida para Categorização dos Lugares nesta pesquisa foi baseada nos objetos existentes na imagem, esta abordagem será explanada no Capítulo 2. A abordagem baseada em Objetos.

Objetivo Geral:

- Categorizar de Lugares baseada em Objetos para o contexto de Mapeamento Semântico.

Objetivos Específicos:

- Reduzir a dimensionalidade e esparsidade das bases de dados que são a frequência dos tipos de objetos (atributos) existentes para os lugares (amostras);
- Realizar Engenharia Automática de Características para melhorar a combinação dos atributos que favoreça a solução pelo sistema de Categorização dos Lugares;
- Propor um novo processo de Categorização de Lugares utilizando Múltiplos Mapa Auto-Organizável Probabilístico (PSOM)s.

1.3 Organização do Documento

Os próximos capítulos desta dissertação organizam-se da seguinte forma: o Capítulo 2 faz uma revisão bibliográfica dos trabalhos sobre Categorização de Lugares. O Capítulo 3 apresenta as técnicas nas quais essa pesquisa se inspirou ou utilizou diretamente para a solução do problema: Aprendizado Profundo, Mapa Auto-Organizável (SOM) e Mapa Auto-Organizável Probabilístico (PSOM). O Capítulo 4 trata dos modelos propostos por esta pesquisa definindo e justificando as escolhas e detalhando o comportamento dos modelos e seu funcionamento. O Capítulo 5 apresenta, discute e sintetiza os resultados experimentais dos modelos propostos e compara com uma solução referenciada na literatura. Finalmente, o Capítulo 6 fecha o trabalho com o resumo sobre os resultados, limitações dos modelos propostos e trabalhos futuros.

2

Categorização de Lugares

Neste capítulo serão apresentados alguns trabalhos importantes sobre Categorização de Lugares, e as propostas e metodologias que foram usadas por eles. Aqui foram descritas mais extensivamente as partes mais originais e contribuições de cada trabalho.

2.1 Definição do Problema

Este trabalho se trata de como Categorizar Lugares (cômodos) isto serve de base para alguns processos como citado na Capítulo 1 para construção de Mapas Semânticos e consequentemente utilizado para o desenvolvimento de Robôs Móveis. Esse processo de Categorização de Lugares pode ser realizado através de diferentes sensores de varredura. Este trabalho está delimitado ao uso imagens (câmeras). O problema abordado busca que, a partir das imagens, ambientes internos possam ser extraídas informações destas para que através de métodos computacionais possam ser avaliadas e categorizadas devidamente com rótulos conceituais como: Quarto, Sala de Estar, Banheiro e Escritório.

2.2 Conceitos Gerais

Nesta seção serão abordados alguns conceitos gerais para o entendimento dos trabalhos na área de Categorização de Lugares. Esses conceitos envolvem a Categorização de Lugares considerando informações visuais que possui métodos baseados em Objetos, em Regiões e em Contexto.

2.2.1 Categorização de Lugares

Um sistema de Navegação Semântica deve ser eficiente em Categorização de Lugares. Isto é, o robô deve ser capaz de produzir rótulos de categorias para lugares sobre os quais nenhum conhecimento prévio está disponível. Em outras palavras, o sistema deve ser capaz de generalizar o conhecimento adquirido, explorando um local específico, de modo a inferir sobre o conteúdo

semântico de qualquer outro local similar (KOSTAVELIS; GASTERATOS, 2013).

O desafio é categorizar os dados de teste capturados nunca vistos anteriormente. Neste caso, os algoritmos devem enfrentar desafios adicionais resultantes da variabilidade interna da categoria. Modelos de categorização de lugares empregam como componentes provenientes da forma, do tamanho e aparência das informações dos lugares para o sistema de navegação semântica (PRONOBIS, 2011).

Nos próximos tópicos desse capítulo serão explanadas as abordagens existentes na literatura para o processo de Categorização de Lugares, que são: Métodos Baseados em Objetos, Métodos Baseados em Regiões e Métodos Baseados em Contexto.

2.2.1.1 Métodos Baseados em Objetos

Os Lugares podem ser descritos pelos objetos neles contidos. Métodos para Categorização de Lugares na ocorrência de objetos (VASUDEVAN; SIEGWART, 2008; VISWANATHAN et al., 2010; CHARALAMPOUS et al., 2014) têm sido empregados com frequência para ambientes confinados, pois nestes ambientes as estruturas como paredes, piso ou teto tendem a ser muito semelhantes em diferentes cômodos, isto é, podem não servir como discriminantes (CHARALAMPOUS et al., 2014). O reconhecimento de alguns objetos pode também ser utilizado como uma informação para a identificação de outros objetos em seu entorno, como o mouse é normalmente encontrados a direita do teclado, ou o fogão, próximo a geladeira (ROGERS III; CHRISTENSEN et al., 2012). A seleção de objetos que se deve usar na representação do lugar é um desafio para que se dê confiabilidade à inferência do lugar. Além disso, por si só, o reconhecimento de um objeto genérico já é uma tarefa desafiadora.

Os modelos baseados em *Deformable Part Models* (DPM) têm se mostrado eficazes para a detecção de objetos rígidos e deformáveis (FELZENSZWALB; MCALLESTER; RAMANAN, 2008; FELZENSZWALB et al., 2013). No entanto, o DPM requer uma grande quantidade de treinamento de dados com exemplos de objetos segmentados (VISWANATHAN et al., 2010). Os algoritmos SIFT e SURF são amplamente utilizados para a extração de características locais de objetos em imagens. Quando o cenário de visão é muito grande, o número de pontos característicos a ser detectado relevantes aos objetos é diminuído e isso pode ser tornar um problema (MADOKORO; UTSUMI; SATO, 2012). Os métodos baseados em objetos têm obtido resultados promissores especialmente em ambientes fechados (VISWANATHAN et al., 2010).

2.2.1.2 Métodos Baseados em Regiões

Em reconhecimento de lugares baseado em regiões, as características das cenas lançam mão de algoritmos de segmentação de imagem. Essas áreas possuem uma mesma característica em comum como cor ou textura. KATSURA et al. (2003) propuseram um método adaptável de categorização de cenas para clima e estações. Katsura et al. utilizaram um robô móvel para capturar imagens e subdividí-las em regiões, tais como: céu, edifícios, árvores dentre outras.

MATSUMOTO et al. (2000) compararam métodos baseados em objetos com métodos baseados em regiões mostrando que os baseados em regiões e demonstraram mais robustez. Um das vantagens desses métodos é a baixa necessidade de processamento, quando comparadas com outras abordagens. No entanto, a precisão da categorização totalmente dependente do processo de segmentação. O processo de segmentação em um ambiente real é uma tarefa complexa e desafiadora. (SHI; MALIK, 2000) propuseram uma melhoria na precisão neste processo de segmentação usando um método de corte normalizado. No entanto, o seu método requer o custo computacional um pouco mais elevado que os métodos mais tradicionais (MADOKORO; UTSUMI; SATO, 2012). Métodos baseado em regiões têm obtido um melhor desempenho em ambientes abertos.

2.2.1.3 Métodos Baseados em Contexto

Métodos baseados em contexto trabalham a ideia de que lugares inteiros podem ser descritos com poucas informações através de padrões gerais da imagem. Nessa abordagem, o efeito para a presença de objetos ou a precisão da segmentação é baixo, porque as informações da cena inteira podem ser descritas como contexto. OLIVA; TORRALBA (2006) propuseram um descritor de características, o Gist, para descrever os recursos globais de uma cena. TORRALBA (2009) propuseram um método de Categorização de Lugares usando Gist, que aloca o número de estados em Modelo Oculto de Markov (HMM) (RABINER; JUANG, 1986).

Os métodos baseados em contexto também podem tratar o problema como um processo de minimização de energia, como proposto por FAZL-ERSI; TSOTSOS (2010). PRONOBIS et al. (2006) apresentaram um reconhecimento de lugares com abordagem discriminativa, que se norteia na categorização baseada em contexto, a qual foi empregada a técnica de *High Dimensional Composed Receptive Field Histograms* (HDCRFH) (LINDE; LINDEBERG, 2004).

2.3 Trabalhos na área Categorização de Lugares

Neste tópico foi feito uma descrição de alguns trabalhos na área Categorização de Lugares para debater as suas diversas características de cada trabalho e seus prós e contras. Os trabalhos listados seguem a abordagem baseada em objetos e em contexto, para ambientes internos a abordagem baseada em regiões não se mostra muito promissora.

2.3.1 Viswanathan et al. (2009)

O trabalho de VISWANATHAN et al. (2009) de categorização de lugares seguiu a abordagem de métodos baseados objetos. A partir da informação da presença dos objetos na cena e a quantidade que estes aparecem, este trabalho se propõe inferir o tipo de cômodo. Para isto foi utilizado base de dados com anotações sobre os objetos nelas existentes. Em um trabalho posterior (VISWANATHAN et al., 2010), foi incluída uma etapa de inferência de objetos para

que o sistema inferisse tanto o objeto e sua probabilidade quanto a quantidade de cada tipo de objetos existente na imagem. Para detecção dos objetos foi empregado o processo proposto por FELZENSZWALB; MCALLESTER; RAMANAN (2008) com base em DPM para realizar a detecção de objetos. Ele foi escolhido devido ao seu alto sucesso no *Pascal Visual Object Classes Challenge* (EVERINGHAM et al., 2015).

Com o intuito de realizar a categorização de lugares com base em objetos, Viswanathan et al. trabalharam com a informação da quantidade (frequência) de ocorrências de objetos em cada tipo de lugar. Essa informação foi obtida através de dados rotulados com as classes dos lugares e com os objetos encontrados nelas, e ajudam a construir uma estatística das ocorrências de cada objeto na cena. A tabela de contagem $ct_p(o,c)$ é a frequência que um objeto o ocorre em um típico lugar p, no qual c é a contagem. Se o número de imagens do lugar tipo p é n_p, a probabilidade de observar o objeto o no lugar de p é calculada como descrito pela Equação 2.1. Essa probabilidade é denominada como o Modelo de Contagem (CM) para o aprendizado do Modelo de Lugares (PM). A Equação 2.1 demonstra o cálculo do PM representado por $P(o,c|p)$ a partir do CM que é representado por $ct_p(o,c)$.

$$P(o,c|p) = \frac{ct_p(o,c)}{n_p} \tag{2.1}$$

Dado um Modelo de Contagem (CM), o Modelo de Lugares (PM) é usado para prever o tipo de lugar mais provável dos objetos observados. Para calcular a probabilidade a priori do lugar, emprega-se a Equação 2.2. A probabilidade do tipo de lugar p dado objeto o pode ser obtida através da Equação 2.3

$$P(p) = \frac{n_p}{\Sigma_i n_i} \tag{2.2}$$

$$P(p|o,c) = \frac{P(o,c|p)P(p)}{\Sigma_i P(o,c|p_i)P(p_i)} \tag{2.3}$$

Os testes de validação empregaram a base de dados LabelMe (RUSSELL et al., 2008) para a categorização de 4 tipos de cômodos (cozinhas, quartos, banheiros e escritórios). Esta base de dados é uma fonte de dados on-line gratuita que fornece uma grande e crescente quantidade de dados visuais referenciados com anotações por humanos, muitas contêm cenas de lugares *indoor* adequadas para rotulagem e reconhecimento de objetos. A base possui anotações da quantidade de objetos existentes nas imagens, quais o nome e quando a região onde estão segmentados os objetos nas imagens. Na Figura 2.1, há um exemplo de imagem do banco de dados do LabelMe. Viswanathan et al. não utilizaram a informação das imagens diretamente, somente as anotações sobre as imagens. Além disso, a informação do rótulo dos objetos e suas quantidade, não foram utilizadas informações sobre a região segmentada do objeto na imagem.

A figura 2.2 mostra o histogramas de frequência de objetos para cozinha e escritório. Verifica-se que podem existir objetos correlacionados ou sinônimos que são categorizados em

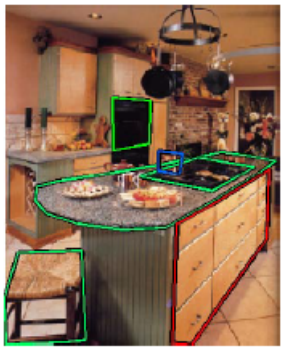

Figura 2.1: Exemplo da base LabelMe.: Uma cena de uma cozinha com polígonos com linhas coloridas delimitando os objetos (VISWANATHAN et al., 2010).

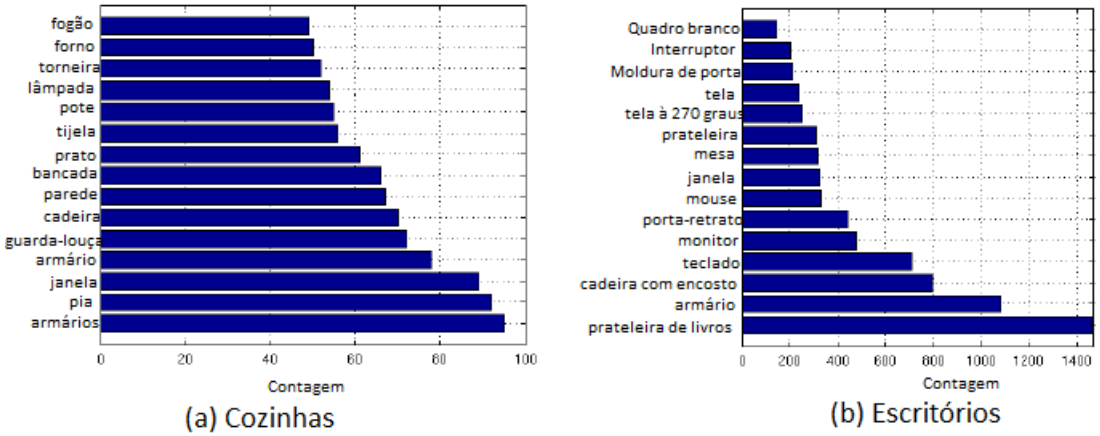

Figura 2.2: Histograma de objetos por tipo de lugar. (a) Cozinhas. (b) Escritórios (VISWANATHAN et al., 2010).

rótulos diferentes, como mesa em: sala de estar, sala de jantar e cozinha. Também foram executados testes simulando 8 ambientes fechados construindo as visualizações do espaço do ambiente através de imagens da base de dados não utilizadas no treinamento, simulando assim, a distribuição espacial dos cômodos no ambiente por completo.

Viswanathan et al. realizaram experimentos apenas para quatro cômodos e os resultados foram excelentes atingindo taxas de acerto entre 97% e 98%. Os experimentos foram realizados com bases de treinamento e testes distintas, cuja matriz de confusão é apresentada na Tabela 2.1. O trabalho de Viswanathan et al. possui ótimos resultados, contudo somente foi avaliada uma base de dados limitada e com poucas categorias. O processo desenvolvido utiliza poucos recursos computacionais para ser processado e pode ser facilmente embarcado em um robô móvel.

Categoria	Cozinha	Banheiro	Quarto	Escritório	Outros
Cozinha	**98%**	0%	0%	1%	1%
Banheiro	13%	**84%**	0%	0%	3%
Quarto	3%	0%	**94%**	4%	0%
Escritório	0%	0%	0%	**100%**	0%

Tabela 2.1: Matriz de confusão do Filtro Bayesiano para LabelMe4.

2.3.2 Zhou et al. (2014)

No recente trabalho de ZHOU et al. (2014) foi realizado um estudo sobre Reconhecimento de Lugares através de Rede Neural Convolucional (CNN) (BENGIO, 2009). Foram utilizados dua Redes Neurais Convolucionais (CNNs), uma baseada em contexto e outra baseada em objetos, Places-CNN e Image-Net respectivamente. Além disso, foi utilizada um rede híbrida a partir dessas duas redes. Zhou et al. usaram utiliza a própria CNNs tanto na formação de descritores de imagem quanto na etapa de Categorização de Lugares.

Foram utilizadas repositórios de tamanho que variaram de 100 mil até a ordem de 7 milhões de imagens, apenas com tamanho superior a 200 x 200 pixels e com o mínino de 100 amostras por categoria. Para evitar que existam imagens duplicadas ou similares foi calculado o descritor Gist entre as imagens e o algoritmos CNN é executado para identificar estes casos de imagens na base de dados. Dessa maneira se trata a alta densidade de imagens com o padrão extremamente parecido. Para que um sistema de categorização tenha um bom desempenho os dados utilizados como treinamento devem ter características de diversidade e densidade para cada categoria do banco de dados. Para execução dos testes e avaliação foram utilizados 4 bancos de dados com diferentes características: Scene15 (LAZEBNIK; SCHMID; PONCE, 2006), MIT (QUATTONI; TORRALBA, 2009), SUN (XIAO et al., 2010) e SUN Attribute (PATTERSON; HAYS, 2012). A redes foram executadas de 300 mil à 700 mil iterações utilizando computadores com GPU NVidia.

A Tabela 2.2 mostra os resultados das duas redes para as diversas bases de dados que superam as taxas de acerto para os *benchmarks*: SUN 47,20% (SÁNCHEZ et al., 2013) e MIT 66,87% (DOERSCH; GUPTA; EFROS, 2013). O trabalho de Zhou et al. obteve ótimos resultados para bases com uma quantidade muito alta de amostras e categorias, contudo o processo de treinamento é lento e não se adequa exatamente ao contexto da robótica. Mesmo assim o trabalho serve de referência pelos ótimos resultados na área.

Categoria	SUN397	MIT Indoor67	Scene15	SUN Attribute
Places-CNN feature	54.32 (0.14)	68.24	90.19 (0.34)	91.29
ImageNet-CNN feature	42.61 (0.16)	56.79	84.23 (0.37)	89.95

Tabela 2.2: Resultados obtidos para as diversas bases de dados testadas. (ZHOU et al., 2014)

2.3.3 Kostavelis e Gasteratos (2013)

No trabalho de KOSTAVELIS; GASTERATOS (2013) é proposto um modelo no qual coexistem informações de SLAM 3D e reconhecimento de lugares utilizando sensores RGB-D. O sistema proposto foi subdividido em duas partes denominadas: camada de baixo nível (navegação numérica) e camada de alto nível (interpretação semântica). A camada de baixo nível trata das informações numéricas e geométricas do local onde se está inserido. O sensor Kinect,

sensor RGB-D, é utilizado nessa etapa para estimação de movimento. Dessa forma é calculada um densa nuvem de pontos da cena. Para cada detecção de planos o algoritmo RANSAC (FISCHLER; BOLLES, 1981) é aplicado para distinguir as superfícies mais promissoras da cena, que determinam os planos estruturais do ambiente.

A camada de alto nível do sistema aprende os modelos abstratos das áreas visitadas a cada iteração e cada instância de cena é detectada e memorizada. Consequentemente, um sistema como este se depara como um excessivo número de rotações, variação de pontos de vista, mudança de escala e iluminação. Essa abundante quantidade de informações é tratada com uma representação por *Bag of Visual Words* (BoVW) (ZHANG et al., 2007).

O modelo proposto possui as duas camadas ligadas por meio de um mapa topológico semântico. Este último é formado explorando a Odometria Visual (VO) (NISTÉR; NARODITSKY; BERGEN, 2004) que é um sistema de referenciamento realizado através de imagens. A camada de baixo nível processa o sistema de odometria, enquanto o sistema também aproveita atributos semânticos do esquema de navegação para alta camada. A partir de cada*frame* obtido pelo câmera, características de descrições locais são extraídos e armazenadas. Em seguida, a *Neural Gas Network* (NGN) realiza a quantização do espaço de dados. No passo seguinte, histogramas para cada amostra de sequencia de imagens são criadas sobre as descrições de quantização e como resultado, o robô aprende uma representação abstrata para cada *frame*.

O algoritmo de extração de característica usado foi o SIFT, e BoVW como representação dos dados extraídos da imagem. Todas as características extraídas são alimentadas em uma NGN (MARTINETZ; BERKOVICH; SCHULTEN, 1993) que serve para quantização de espaço que trata um conjunto de dados esparsos dando uma representação abstrata e coesa do espaço de informações. A NGN tem como objetivo básico optimizar uma função de custo que minimize o erro de quantização.

Os vetores de quantização correspondem ao vocabulário que contém todas as *Visual Words* (VW) descrevendo o espaço inicial de uma forma abstrata. No passo seguinte, os vetores de quantização (VW) são empregados para elaboração dos histogramas que representam imagens de entrada. Então, cada imagem fornecida como entrada do processo, está sendo representada pelo histograma de aparência para treinar um categorizador. Com objetivo de tornar o vocabulário mais distinto, Kostavelis e Gasteratos compararam seu método com árvores de vocabulário (NISTER; STEWENIUS, 2006). Essa estrutura permite a recuperação rápida de imagens de grandes coleções. O sistema proposto pode utilizar o mesmo vocabulário tanto para reconhecimento quanto para categorização, atingindo alta capacidade de generalização.

Os testes no mundo real foram realizados através de uma plataforma robótica denominada MAGGIE (Mobile Autonomous riGGed Indoors Exploratory), desenvolvida pelos próprios autores. Os testes foram realizados em ambiente com variações de luminosidade: ensolarado, nublado, noite (luz artificial). Esses testes foram realizados a fim da avaliar tanto a capacidade de reconhecimento quanto de categorização do algoritmo. Para os mesmos testes também foi utilizada a base de dados COLD (ULLAH et al., 2008). Ainda foram realizados testes de

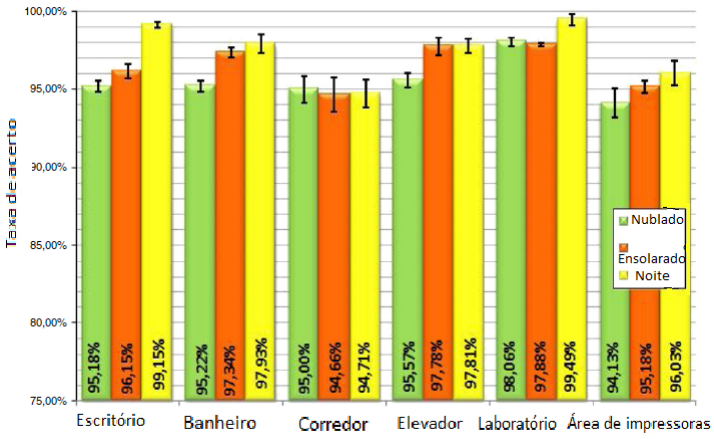

Figura 2.3: Taxa de acerto por classe de lugares em diversos tipo de iluminação: Ensolarado, nublado e noite (KOSTAVELIS; GASTERATOS, 2013).

reconhecimento de lugares com a base de dados COLD. Essa base é composta por variações de luminosidade: ensolarado, nublado, noite. Podemos verificar os resultados desses testes na Figura 2.3. No treinamento foi empregada a técnica de validação cruzada (*10-folds*). O processo proposto por Kostavelis e Gasteratos teve dificuldade em reconhecer os cômodos que tinham textura parecidas como escritório e laboratório.

O trabalho de Kostavelis e Gasteratos trata bem o problema de variações de iluminação em ambiente e dinâmica do ambiente na categorização dos lugares que é uma característica importante para o Mapeamento Semântico, obtém ótimas taxas de acerto, acima de 95%, contudo utiliza um um número reduzido de tipos de cômodos.

2.3.4 Li e Meng (2012)

O trabalho de LI; MENG (2012) tem a proposta de trabalhar pontos fracos deixados por trabalhos focados na aparência global do ambiente confinado. Os autores defendem da ideia de que a semântica a ambientes confinados está nos objetos ou que estão nesses objetos. O método de representação da informação foi o *Transformed Dirichlet Process* (TDP). Li e Meng usam *Dirichlet Process Mixture Model* (DPMM) para modelar a organização das informações extraídas das imagens.

Observa-se cada lugar (cena) como uma coleção de diferentes componentes com semântica, o que pode ser objetos e características. Portanto, o reconhecimento de objetos e extração de características são pré-requisitos para DPMM. Para cada cena existem os rótulos de detecção de objetos e para as áreas das imagens não cobertas por objetos vetores de características são gerados. Esses dois tipos de componentes compõem cada cena. As Características Gist no espaço de cor *Hue Saturation Value (Brightness)* (HSV) capturaram propriedades globais e características SIFT representaram texturas locais.

A última etapa do processo é a construção da representação de cada lugar (cena), através da construção de um Mapa Semântico Probabilístico (PSM). Após extrair os componentes

a partir das imagens, constrói-se uma representação interna de cada cena. Ela é construída através de dois processos: inicialização e atualização. Para cada nova cena são associados diretamente as características e objetos conforme demonstrado na Equação 2.4. Onde i é o i-ésimo componente e j é j-ésima imagem. Para inicialização, $w_{ij} = 1$ para descrever presença definitiva pela componente da c_i na cena s até agora.

$$f : w_{ij}c_j \rightarrow \mathbf{s}_i = 1,...,n, j = 0 \tag{2.4}$$

Em seguida, atualizar esta associação com a nova amostra j, na qual $j = 1,..,n$, isso significa o aprendizado com outras imagens e provocado por uma atualização do cenário visualizado pelo robô. Por exemplo, quando o robô detecta algum movimento em seu lugar atual que leva a mudança da cena. Um método interessante é empregar a simplificação de atualização bayesiana. A partir da definição do conhecimento prévio como uma probabilidade anterior e um novo exemplo de observação, obtêm-se a distribuição de probabilidade a posteriori. Contudo, isto exige modelagem explícita dos lugares como certa distribuição de probabilidade que geralmente é mais complexa de se obter.

Para uma nova imagem apresentada podemos ter 3 cenários de respostas do modelo: novo componente, componente restante e componente desaparecido. Um novo componente só precisa de outro processo de inicialização quando for necessário adicionar um nó para o mapa. Nos casos mais frequentes: novo componente e componente restante. Pode-se simplesmente determinar a frequência de ocorrência através da contagem nas imagens. Este método requer a memorização da frequência de cada um dos componentes. Com modelos simples de cenas e componentes é possível construir uma rede hierárquica de modo a refletir a relação entre eles. Os componentes podem ser compartilhados entre lugares como por exemplo: livros na biblioteca, no quarto, no escritório ou laboratório. Desta forma, essa relação é representada através da conexão entre cenas diferentes para objetos comuns. A relação de conjunto de contenção lugar na teoria de conjuntos pode ser expressa pela Equação 2.5.

$$min \sum_{w} \|ws_1 - s_2\| \tag{2.5}$$

Se o valor da Equação 2.5 tende a zero, então o lugar s_1 contém todos os elementos de s_2. Assim s_1 é um subconjunto de s_2. Então, os dois lugares são conectados com uma relação de está contido. Desta forma, um mapa semântico é construído sobre a relação entre os lugares e seus componentes. Os testes no trabalho de Li e Meng foram realizados através de uma base de dados (QUATTONI; TORRALBA, 2009) e de uma plataforma de um robô humanóide. Foi utilizando no processo o classificador *Nearest Neigbhbor* (NN). Os resultados das taxas de reconhecimento para diversos rótulos de lugares são apresentados na Figura 2.4. O trabalho de Li e Meng desenvolve um conceito interessante de conciliar as abordagens baseadas em Objetos e Contexto tentando fundir essas informações no mesmo modelo. A deficiência do modelo proposto por Li e Meng é a dificuldade de parametrização para construção dos modelos

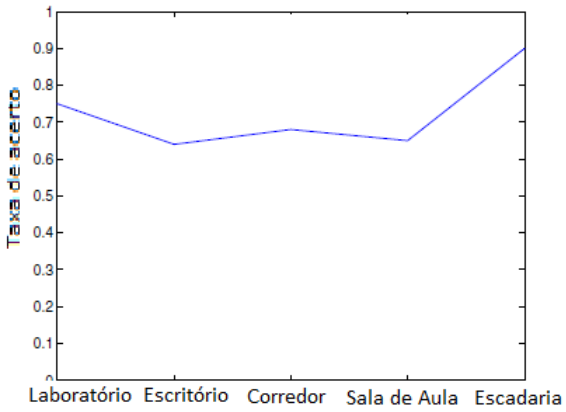

Figura 2.4: Resultados de taxa de acurácia da Categorização de Lugares (LI; MENG, 2012).

e consequentemente a escalabilidade para situações com muitas categorias de lugares. Além do que as taxas de acerto para poucas categorias, somente 5, são baixas, entre 70% e 90%, em relação a outros trabalho.

2.3.5 Charalampous et al. (2014)

O trabalho de CHARALAMPOUS et al. (2014) foi elaborado com foco em objetos. Ele emprega um sensor RGB-D, assim obtém informações de profundidade. Essas informações de profundidade da câmera podem ser usadas como entrada para o sistema de localização do robô. Através de algoritmos de reconstrução da cena, são identificadas as superfícies dominantes do ambiente. Sob essas superfícies objetos são mais prováveis de serem encontrados. Essa premissa é válida pois os ambientes são estruturados, especialmente os lugares confinados, nos quais a maioria dos objetos são postos sobre o piso, mesas e paredes. A parte nuvem de pontos, que é identificada como *outliers* desses planos, é utilizada para identificar posicionamento de prováveis objetos.

Sendo assim, os dados RGB-D são adquiridos e a cena é reconstruída em uma nuvem de pontos. Essa nuvem de pontos pode ser utilizada para inferir no deslocamento do robô através de Odometria Visual (VO) que geralmente é realizada com câmeras estéreos. A detecção de planos perpendiculares é realizada pelo *Random Sample Consensus* (RANSAC) (FISCHLER; BOLLES, 1981; HARTLEY; ZISSERMAN, 2003), focado a procurar planos perpendiculares ao eixo z. O plano dominante é o piso, enquanto que os restantes são considerados faces planas do mobiliário.

No caso dos pontos de *outliers* podem se agrupar em um volume V_i que são os volumes detectados de objetos pelo estudo de planos. Então, os dados de entrada de região desta imagem são usados para o processo de reconhecimento de objetos proposto por CHARALAMPOUS; GASTERATOS (2013), em um trabalho anterior, que é um aprendizado baseado em *Hierarchical Temporal Memory* (HTM) para reconhecimento de objetos. O mapa de saliência da imagem é

calculado a partir do espaço de entrada inicial. A rotina de classificação dentro deste esquema compreende um número de classificadores Máquina de Vetores de Suporte (SVM) igual ao número de classes de objetos. Cada classificador SVM é treinado sob a abordagem *one-vs-all*. Isso implica que para cada amostra de teste a SVM fornece um valor que designa o nível de confiança sobre a identidade da amostra. O classificador que obtiver mais pontuação é o que vai determinar a identidade do objeto.

Na etapa de categorização de lugares, o processo foi baseado no Classificador Multinomial de Bayes (MBC) (THEODORIDIS; KOUTROUMBAS, 2006). Na versão de multinomial de algoritmo de categorização cada amostra é expressa como frequências de eventos. Na abordagem proposta por Charapalpous et al., cada característica do vetor $F_j = [f_1 f_2 ... f_{N_O}]^T$ emprega o número de aparições de cada objeto antes dela ser reconstruída.

Os testes foram realizados experimentalmente na plataforma robótica. Um robô construído com um sensor Kinect como sensor RGB-D e um banco de dados de objetos foi utilizado para treinamento dos algoritmos HTM e SVM. O robô percorreu um ambiente *indoor* e classificou os lugares em cinco classes existentes: laboratório, escritório, cozinha, corredor e global. Os resultados são apresentados na Figura 2.3. A maioria das avaliações erradas do robô se dão em zonas de transição ou quando em uma pequena quantidade de objetos são encontrados cenas. O trabalho de Charampoulos el al. avança nos conceitos de Mapeamento de Lugares utilizando sensores de profundidade para mapear os objetos e coletar múltiplas visões da cena para Categorização do Lugar. Eles trabalham com um número reduzido de categorias e suas taxas de acerto ficam em torno de 90%.

	Laboratório	Escritório	Cozinha	Corredor	Global
Taxa de acerto	89.32	80.53	75.16	84.21	82.3

Tabela 2.3: Resultados de taxa de acurácia da categorização dos lugares (CHARALAMPOUS et al., 2014).

2.3.6 Rogers et al. (2012)

No trabalho de ROGERS III; CHRISTENSEN et al. (2012) utiliza-se da segmentação espacial de lugares, na separação dos cômodos num ambiente confinado, antes de fazer a categorização. Ambientes fechados são organizados em cômodos e esses contém objetos específicos, tais como pasta de dentes no banheiro e calculadoras no escritório. Os cômodos também estão dispostos geograficamente em padrões característicos para permitir que a vida seja eficiente e confortável, tais como banheiros ao lado de quartos e salas de jantar são adjacentes para cozinhas. O trabalho de Rogers et al. se utilizou dessa restrição natural desses ambientes para trabalhar o modelo de *Conditional Random Field* (CRF) (LAFFERTY; MCCALLUM; PEREIRA, 2001). Além disso, como os objetos no mesmo cômodos possuem o mesmo contexto semântico do lugar, desta forma a inferência de um determinado objeto pode ser alimentada pelo contexto dos objetos e do local onde se está.

Rogers et al. utilizaram CRF para definir o relacionamento entre os cômodos e o relacionamento cômodo-objetos para definir um rótulo para o cômodo. Modelos de CRF expressam a probabilidade das configurações das variáveis através de um conjunto de funções de compatibilidade. No caso de modelagem de adjacência de cômodos e co-ocorrência cômodo-objeto há um tipo de função de compatibilidade que favorece pares prováveis de cômodos e combinações entre cômodos e objetos. A Equação 2.6 representa o relacionamento entre os cômodos e as propriedades dos objetos. O modelo de CRF foi escolhido por Rogers et al. ao invés do modelo mais tradicional *Markov Random Field* (MRF), por ele permitir incorporar certos elementos de prova como absoluta e resolver para a distribuição das incertezas das variáveis condicionando a uma evidência absoluta. Rogers et al. utilizaram uma implementação CRF demominada *Undirected Graphical Model* (UGM) (SCHMIDT, 2012).

$$p(y|x) = \frac{1}{Z(x)} \exp \sum_{k=1}^{K} \lambda_k f_k(y,x) \tag{2.6}$$

Nessa implementação, objetos são segmentados do plano de fundo como nuvens de ponto 3D. A nuvem de pontos obtida pela câmera em profundidade é filtrada espacialmente para conter apenas porções pertinentes dentro de um volume do objeto de interesse em frente do robô. Então, são extraídos até 4 planos da nuvem de pontos usando um RANSAC (FISCHLER; BOLLES, 1981). Portanto estes planos são analisados para encontrar agrupamentos de pontos restantes que estão sob esses planos, o que indicariam a existência de possíveis objetos. A extensão destes pontos na câmera é utilizada para segmentar o candidato a objeto para região e aplicar o processo de reconhecimento de objetos.

A etapa de reconhecimento foi realizada usando o algoritmo SURF. Essas características são comparadas com características SURF de outras imagens numa base de dados. O conjunto de características correspondentes de um modelo são usados para calcular um modelo de homografia usando RANSAC. Se este conjunto de dados é de tamanho suficiente, então o objeto é dito como reconhecido e absolutamente esta informação é então enviada para o sistema de mapeamento juntamente com medições geométricas da localização do objeto.

O sistema de categorização de objetos que baseia-se em BoVW (SIVIC; ZISSERMAN, 2005). Esta técnica requer um vocabulário de palavras visuais que foi aprendido pela realização de agrupamento K-means através das características SURF como conjunto de treinamento. Características SURF são extraídas a partir de uma imagem de um objeto e são quantizados num vetor de VW em um vocabulário. Um histograma é construído pela contagem da frequência da aparência de cada palavra visual na imagem do objeto candidato, ponderada pelo *Term Frequency Inverse Document Frequency* (TF-IDF) (RAMOS, 2003). Um conjunto de *Relevance Vector Machines* (RVM) (TIPPING, 2001) é treinado para reconhecer categorias de objeto usando este histograma visual. Cada um dos RVM é treinado para reconhecer uma categoria. O saída de uma RVM é um valor real, ao contrário da saída de um SVM que é inteira.

Em um trabalho de NIETO-GRANDA et al. (2010) utilizam uma técnica simples para

representar a extensão de um espaço por uma gaussiana extraída de dados de distância a laser em um determinado lugar. A suposição feita no trabalho é a de que o robô só vê dentro de um cômodo por vez e que o laser de varredura é suficiente para cobrir a área do cômodo. A utilização deste modelo permite realizar uma avaliação da distância Mahalanobis para determinar se o robô está dentro de um determinado cômodo ou se uma nova instância de modelo gaussiano deve ser criada. Sendo assim é mantido um rastreamento dos cômodos no percurso do robô para construção de uma representação topológica de adjacência dos cômodos.

Para estabelecer a eficácia do RVM na técnica do BoVW foi realizada um experimento no conjunto de dados Caltech 101 (FEI-FEI; FERGUS; PERONA, 2007). Cinco classes de lugares foram selecionadas e objetos de interesse foram extraídos utilizando anotações fornecidas com o conjunto de dados. Metade das imagens foram usadas para treinar o modelo RVM usando validação cruzada de *3-folds* para seleção de parâmetro de raio da RVM. Já o resultado da RVM foi usado para categorizar as imagens a partir da outra metade das imagens que foram reservadas para o conjunto de teste.

Nos testes em campo (foi utilizado um robô em uma casa simulada), os objetos que foram categorizados erroneamente, pois não pertenciam a base de dados de treinamento, induziram a categorização do local também errônea. Este mesmo objeto visto em um outro contexto foi aprendido corretamente, então o robô ao retornar a uma avaliação anterior corrige a classificação do objeto. O trabalho de Rogers et al. aborda um conceito interessante de relacionar os objetos por sua localização e distribuição no ambiente, criando assim um conceito hieráquico dos objetos existentes no ambiente e a sua relação com os cômodos, porém a sua taxa de acerto é baixa, além de que o seu modelo é difícil de ser parametrizado para um problema com muitas categorias.

2.4 Resumo dos Trabalhos

Nesta sessão são abordados os problemas em aberto indicados pelos artigos ou inferidos a partir destes. Mapeamento semântico e categorização de lugares são processos compostos por vários módulos hierárquicos, com muitas variáveis espalhadas por cada etapa. Consequentemente há inúmeros problemas e lacunas podem ser exploradas. Na tabela 2.4 há um resumo das informações dos artigos descritos evidenciando suas principais características de uma forma estruturada para facilitar a visualização. A seguir serão discutidas as principais características e lacunas existentes para o problema de Categorização de Lugares para o contexto de Mapeamento Semântico.

Na parte de reconhecimento de objetos, o problema a ser tratado está na escolha de que os objetos devem ser detectados (como fazer a busca na imagem) e levados em conta para a Categorização de Lugares. Além de como as várias visões do objetos durante o mapeamento/navegação podem ser levadas em conta em todo o processo, fazendo com que as várias visões em posicionamentos diferenciados de câmera para com o objeto sejam úteis para solução. A quantidade de dados que é enorme para treinamento e processamento em tempo real para inferência de uma

Referência	Técnicas Imagens	Técnicas Aprendizagem	Métricas	*Abordagem*	Bases de Dados
VISWANATHAN et al. (2010)	SIFT Gist	DPM MI-SVM CM Filtro Bayesiano	Matriz de Confusão	Objetos	LabelMe
ZHOU et al. (2014)	CNN	CNN	Taxa de acerto	Contexto	Scene15 MIT Indoor67 Sun Database Sun Atribute
KOSTAVELIS; GASTERATOS (2013)	SIFT BoVW	NGN SVM	Avaliação desempenho por quantidade de dados Taxa de acerto Cruzamento de avaliações em iluminação diferentes	Contexto	COLD
LI; MENG (2012)	SIFT DPMM	PSM NN	Taxa de acerto	Contexto Objetos	Não disponível
CHARALAMPOUS et al. (2014)	HTM	SVM MBC	Taxa de acerto	Objetos	Não disponível
ROGERS III; CHRISTENSEN et al. (2012)	SURF	CRF	Matriz de Confusão	Objetos	Caltech 101

Tabela 2.4: Resumos dos trabalhos mais importantes e recentes.

grande quantidade de objetos e atributos da cena. Um grande problema na detecção de objetos é encontrar as regiões de existência de possíveis objetos. Um atalho que pode ser tomado para isso é empregar um sensor de imagem em profundidade e outro caminho é procurar saliências nos planos encontrados como chão, mesas, paredes. Normalmente os objetos estão sob eles. Encontrar os planos das superfícies e tratar os *outliers* como objetos é um desafio, mas traz um eficiente resultado para todo o processo por facilitar da reconhecimento de objetos no ambiente.

Para a etapa de categorização de lugares uma área que pode ser explorada é automatizar o agrupamento de objetos através dos seus rótulos. Se determinados rótulos forem sinônimos, então eles podem pertencer a mesma classe de objetos em vez de serem considerados separadamente. Além do que os atributos (cor, textura, tamanho) pertencentes ao objetos podem ser levados em conta na categorização do lugar no processo como um todo. A relação entre os objetos de um lugar pode ajudar na inferência de reconhecimento dos outros objetos de um cômodo, pois os objetos de um lugar geralmente estão no mesmo contexto. A informação da distribuição de como estão os cômodos é uma informação relevante na inferência da semântica dos objetos e dos cômodos. As estruturas e objetos inferem a semântica do lugar e a semântica ajuda na definição das estruturas e objetos, estão mutuamente ligadas. Existe também uma dificuldades na Categorização de Lugar e que possua poucos objetos discriminantes no local (baseado em objetos) ou texturas e dimensões muitos semelhantes entre os ambientes (baseado em contexto). Além disso o robô pode aumentar ou diminuir a quantidade de objetos analisados na avaliação dependendo da incerteza de inferência da Classificação de Lugares, como por exemplo, examinando objetos

desconhecidos com maiores detalhes.

Os sistemas de Categorização de Lugares para um Mapeamento Semântico processam um quantidade de dados muito grande. Ter uma representatividade nos modelos do mapa para que minimizem a quantidade de dados e maximizem a quantidade de informações é um desafio, a forma da representação do mapa semântico tem um impacto grande na viabilidade da aplicação do robô e como todo o sistema é elaborado. Verifica-se que é desafiador o processo de categorizar corretamente ambiente que possuam texturas semelhantes (no piso, paredes, teto). Outra dificuldade aparece em zonas de transição dos lugares, causando categorizações errôneas. Auto aprendizado é uma característica muito importante que um sistema com este deve ter para se adequar as diversas situações e minimizar os erros com a experiência. Um resumo das conclusões de problemas em aberto está na Tabela 2.5.

Os trabalhos relatados neste Capítulo tem seus pontos fortes, fracos e seus contexto de aplicação. Para escolha do trabalho que servirá como base de comparação levamos em conta os seguintes aspectos: Disponibilidade de Bases de Dados, aderência com aplicação robótica, resultados para poucas categorias (até 10 categorias), resultados para muitas categorias (mais de 100 categorias) e escalabilidade para processamento de muitas categorias. Essa análise está apresentada na Tabela 2.6. Ao final do resumo de cada trabalho fizemos considerações que levam ao preenchimento da Tabela 2.6. Os trabalhos mais interessantes visualizando os critérios de avaliação foram o trabalho de Viswanathan et al., Kostavelis e Gasteratos. O trabalho de Viswanathan et al. foi escolhido em detrimento do outro por ter apresentado um melhor resultado de taxa de acertos. A abordagem escolhida para este trabalho é a baseada em objetos e será comparada com o trabalho de Viswanathan et al.

Foram listadas algumas características do problema de Categorização de Lugares baseado em Objetos que será tratado nesta pesquisa. Este trabalho foi delimitado para focar em alguns aspectos do processo de categorização de lugares: relacionamentos entre os objetos (7), relacionamento dos objetos com o lugar (9), hierarquia de informações dos objetos (12) e esparsidade dos objetos por lugar (13) da Tabela 2.5. Esses itens tratam em foco da relação objetos e lugares. O próximo capítulo contempla um revisão na literatura de métodos que foram utilizados no modelo proposto ou que servem como base de para o desenvolvimento deste.

Ordem	Etapa	Característica	Descrição
1	Objetos	Detecção e Classificação de Objetos	Como tratar problemas como: oclusão, variação de luminosidade e variação de posicionamento da câmera.
2	Objetos	Busca de Objetos em profundidade	Determinação de planos principais das superfícies e determinação de volumes de *outliers* representativos à objetos
3	Objetos	Busca de Objetos em profundidade e detecção simultâneos	Reconhecimento de Objetos e detecção por profundidade mútua. Os dois processos aprenderem um através dos resultados do outro.
4	Objetos	Atributos dos objetos	Como enriquecer o modelo de dados e fornecer mais informações sobre os objetos como: textura, material, tamanho.
5	Objetos	Busca de Objetos na Imagem	Algoritmos de detecção de características na imagem e casamento de padrões.
6	Objetos	Classificação de objetos	Grande quantidade de classes e dados para se encontrar os padrões.
7	Objetos	Classificação de objetos sinônimos	Tratar objetos com funcionalidade semelhantes como um atributo semelhantes.
8	Objetos Lugares	Relacionamento entre objetos	Criar um relacionamento entre os objetos: espacial na imagem e funcional de contexto.
9	Objetos Lugares	Relacionamento dos objetos com o lugar	Como a relação entre os objetos (quantidades e características destes) podem ajudar a inferir informações sobre o lugar.
10	Objetos Lugares	Categorizar lugares usando atributos dos objetos	Como as informações de atributos dos objetos pode ajudar a melhorar o processo de categorização do lugar refinado-o.
11	Objetos Lugares	Categorização de objetos e lugares simultâneos	Como levar em conta a categorização dos objetos como o lugar e simultaneamente fazer o inverso ajustando o aprendizado dos dois lados.
12	Objetos Lugares	Hierarquia de informações dos objetos	Como hierarquizar os dados obtidos para que sejam melhores aproveitados e possam se atribuir pesos para eles.
13	Objetos Lugares	Esparsidade dos objetos por lugar	Muitas vezes os lugares a serem analisados possuem poucos objetos (às vezes somente 1) os processos de inferência têm de ser robustos com respeito a esparsidade dos objetos.
14	Lugares	Uso de informações globais na categorização dos lugares	Como usar as informações globais como textura, cores, dimensionalidade do local para contribuir para o processo de categorização em conjunto com objetos.
15	Lugares Mapa Semântico	Classificação de lugares de múltiplas vistas.	Um mesmo local visto de pontos de vista diferentes precisam ser robustas as variações de ambiente como: ensolado, nublado e iluminação.
16	Lugares Mapa Semântico	O processo deve ser atender ao reconhecimento e categorização de lugares.	Reconhecimento e categorização são dois processos importantes para mapeamento semântico aplicado a robôs móveis é interessante que possa ter as duas habilidades citadas.
17	Mapa Semântico	Identificação de zonas de transição	As zonas de transição entre os cômodos tendem a confundir o categorizador por causa da transição de características.
18	Mapa Semântico	Categorização de lugares adicionando a disposição de cômodos	Como levar em conta o conhecimento de estrutura de ambientes *indoor* sobre a decisão de Categorização de Lugares.
19	Objetos Lugares Mapa Semântico	Categorização de Lugares adicionando disposição de cômodos	Como levar em conta o conhecimento de estrutura de ambientes *indoor* (disposição dos cômodos) na decisão de Categorização de Lugares, e lugares e objetos mutuamente.

Tabela 2.5: Resumo de problemas em aberto.

Referência	Disponibilidade de Bases de Dados	Aderência com aplicação em robótica	Resultados para poucas categorias	Resultados para muitas categorias	Escalabilidade para processamento de muitas categorias
VISWANATHAN et al. (2009)	Sim	Sim	Sim	Não	Sim
ZHOU et al. (2014)	Sim	Não	Não	Sim	Sim
KOSTAVELIS; GASTERATOS (2013)	Sim	Sim	Sim	Não	Sim
LI; MENG (2012)	Não	Sim	Sim	Não	Não
CHARALAMPOUS et al. (2014)	Não	Sim	Sim	Não	Não
ROGERS III; CHRISTENSEN et al. (2012)	Sim	Sim	Sim	Não	Não

Tabela 2.6: Escolha de trabalho para comparação abordando os seguintes critérios Disponibilidade de Bases de Dados (DBD), Aderência com aplicação robótica (APR), Resultados para poucas categorias (APC), Resultados para muitas categorias (RMC), Escalabilidade para processamento de muitas categorias (EPMC).

3

Redes de Aprendizado Profundo e SOMs Probabilísticos

Neste capítulo serão detalhados os modelos que servem de inspiração ou que estão presentes diretamente no método proposto neste trabalho. O modelo proposto é dividido em duas etapas: redução de dimensionalidade e categorizador, respectivamente. As Redes de Aprendizagem Profunda servem de inspiração para a primeira etapa do modelo proposto, já os Mapas Auto-Organizáveis Probabilísticos irão compor a solução para a segunda etapa do modelo proposto, realizando agrupamento.

3.1 Redes de Aprendizagem Profunda

As técnicas convencionais de Aprendizagem de Máquina são limitados na sua capacidade de processar dados naturais em sua forma bruta. Durante décadas, construção de um sistema de reconhecimento de padrões e de Aprendizagem de Máquina foram dependentes de uma engenharia cuidadosa e de considerável experiência de domínio para projetar extratores de características. Elas transformam os dados brutos (como os valores de pixels de uma imagem) em uma representação ou vetor de características adequado para servir como entrada a um sistema de aprendizagem (LECUN; BENGIO; HINTON, 2015).

Métodos de Aprendizagem Profunda são métodos de representação de aprendizagem com vários níveis de representação, obtido pela composição de módulos simples, no qual aplicam transformações não-lineares. Com composições suficientes de tais transformações, funções de alta complexidade podem ser aprendidas. Para tarefas de categorização, as camadas mais altas da representação amplificam os aspectos da entrada que são importantes para a discriminação e suprimirem variações irrelevantes. Uma imagem consiste de uma matriz de valores de pixel (entrada), já as características aprendidas na primeira camada de nodos podem representam a presença ou ausência de bordas em determinadas orientações locais da imagem. A segunda camada tipicamente detecta arranjos particulares linhas, contornos, independentemente de pequenas variações nas posições de borda. A terceira camada pode montar combinações que

correspondem a partes de objetos familiares e as camadas subsequentes servem para detectar objetos como combinações destas partes. O aspecto fundamental da aprendizagem profunda é que essas camadas que constroem as representações características não são projetadas por humanos: elas são aprendidas a partir de dados usando um algoritmo de aprendizagem (LECUN; BENGIO; HINTON, 2015).

Métodos de Aprendizagem Profunda visam aprender uma hierarquia de características com variações de complexidade formada pela composição de características de nível mais baixo. Os resultados de métodos de Aprendizagem Profunda apresentam melhor desempenho que modelos de aprendizado raso (com uma ou duas camadas ocultas). Em muitas instâncias de Aprendizagem Profunda, a função objetivo é não-convexa, e possui muitos mínimos locais distintos no espaço dos parâmetros do modelo (LECUN; BENGIO; HINTON, 2015). A evidência de estratégias de formação eficazes para a Arquitetura Profunda veio com os algoritmos para redes de *Deep Belief Network* (DBN) (HINTON; OSINDERO; TEH, 2006) e *Stacked Auto-Encoders* (SAE) (MARC'AURELIO RANZATO; CHOPRA; LECUN, 2007). Cada camada é pré-treinada com um algoritmo de aprendizagem não-supervisionado, aprendendo uma transformação não-linear da sua entrada, a saída da camada anterior, que captura as principais variações na sua entrada. O pré-treinamento não-supervisionado prepara os dados para uma fase de treinamento final que ajusta dos parâmetros em relação a um critério de treinamento supervisionado baseado no gradiente (ERHAN et al., 2010). A fase de treinamento não-supervisonado transforma o espaço de parâmetros para uma representação mais adequada à optimização mais eficaz para alcançar um valor menor na função de custo do problema em questão (BENGIO et al., 2007).

Aprendizagem Profunda está fazendo grandes avanços na resolução de problemas que resistiram as melhores tentativas das técnicas de Inteligência Computacional antes dela. Ela se mostrou ser muito eficaz em descobrir estruturas complexas em dados de alta dimensionalidade, e portanto, aplicabilidade a um vasto domínio de problemas (LECUN; BENGIO; HINTON, 2015). As Redes Neurais Artificiais são treinadas com o objetivo de minimizar funções objetivo para resolução de um problema. Um gradiente de correção pode ser gerado para orientar o ajuste dos pesos dos nodos como por exemplo o algoritmo de *backpropagation* em uma rede *Multilayer Perceptron* (MLP). Em geral os algoritmos de treinamento consistem em mostrar o vetor de entrada, calculando resultados e taxa de erro. Assim, para diminuir essa taxa de erro são ajustados os pesos da rede. O processo é repetido para as amostras do conjunto de treinamento até um convergência do erro. Após o treinamento, o desempenho do sistema é medido com um conjunto diferente de amostras chamados um conjunto de teste. Isto serve para medir a capacidade de generalização do sistema (LECUN; BENGIO; HINTON, 2015).

Uma rede neural de múltiplas camadas pode ter dificuldade para fazer os ajustes de pesos nas camadas mais distantes da saída. Assim redes neurais com o treinamento de ajustes de pesos passando por muitas camadas se torna ineficiente pelo acúmulo de erro das derivadas do gradiente de ajuste. Aprendizagem Profunda tenta resolver essa limitação treinando cada

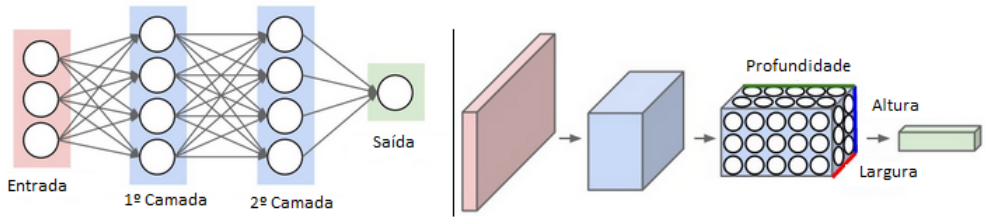

Figura 3.1: Esquerda: uma rede neural padrão com 3 camadas. Direita: uma CNN que organiza seus nodos em três dimensões (largura, altura, profundidade) como visualizado em uma das camadas. Cada camada da CNN transforma o volume de entrada 3D em um volume de saída de ativações de nodos 3D. Neste exemplo, a camada de entrada contém a imagem em vermelho, onde a sua largura e altura são equivalentes as dimensões da imagem e a profundidade seria a terceira dimensão (KARPATHY, 2016).

camada independentemente uma a uma. Esta é a principal vantagem aprendizagem profunda. A arquitetura de Aprendizado Profunda é uma pilha de camadas simples, assim essas redes podem implementar funções altamente complexas e ser sensível a ajustes de pesos. A seguir serão apresentados os conceitos da Rede Neural Convolucional (CNN) na qual será inspirada a primeira etapa do modelo a ser proposto no Capítulo 4.

3.1.1 Rede Neural Convolucional

As CNNs foram projetadas por uma necessidade de construir filtros adaptativos para imagens que aprendessem representações de dados existentes neles. Antes destes tipos de redes surgirem a imagem era processada por um conjunto de filtros; determinados inicialmente que extraíam informações das imagens e estas informações serviam de entrada para algum um classificador. Uma CNN propõe que esses filtros sejam treinados pela própria rede, como em uma cascata de filtros com complexidade gradativa. A entrada e a saída entre cada camada são mapas de características. Este processo é análogo para sinais de áudio, no qual cada mapa de característica é representado por um vetor (LECUN; BENGIO, 1995).

CNNs tiram proveito da propriedade de características contíguas da entrada (imagem, sinais de áudio), assim que restringem a arquitetura a terem uma menor quantidade de pesos. Em particular, as camadas de uma CNN têm nodos dispostos em 3 dimensões: largura, altura, profundidade. Note que que a palavra profundidade aqui refere-se para a terceira dimensão de um volume de ativação, não para a profundidade de uma rede neural total, que pode referir-se ao número total de camadas de uma rede. Como será apresentado a seguir, os nodos em uma camada só serão ligados a uma pequena região da próxima camada, ao invés de se ligarem a todos os nodos desta. Ao final da arquitetura da CNN, esta irá reduzir a imagem completa num único vector de classe, disposto ao longo da dimensão de profundidade. A Figura 3.1 mostra-se uma representação e comparação entre uma rede neural padrão e uma CNN.

Uma CNN possui uma sequência de camadas de modo que cada uma transforma um volume de ativações para outra camada através de uma função diferenciável. São usadas três

tipos principais de camadas para construir arquiteturas CNN: Camada Convolucional, Camada *Pooling* e Camada *Fully-Connected*. Essas camadas são empilhadas para formar uma arquitetura CNN completa. Para exemplificar o tamanho dos dados nas camadas será assumida uma imagem de entrada com 32x32 pixels (KRIZHEVSKY; SUTSKEVER; HINTON, 2012).

- INPUT [32x32]: manterá os valores de pixel da imagem, neste caso, uma imagem de largura 32 e altura 32.
- CONV: esta camada irá calcular a saída de nodos que estão ligados a regiões locais na entrada, cada cálculo de um produto de pontos entre os seus pesos e uma pequena região que estão ligados no volume de entrada. Isso resulta em um volume como [32x32x12] para um exemplo de 12 filtros.
- POOL: é aplicada uma função de ativação elemento a elemento, como por exemplo max(0, X). O tamanho da profundidade permanece inalterado ([32x32x12]), a camada vai realizar uma operação de subamostragem ao longo das dimensões espaciais (largura, altura) resultando em volume por exemplo [16x16x12].
- FC: *Fully-Connected* irá calcular os scores de classe, resultando em volume de tamanho [1x1x12], onde cada um dos 12 valores correspondem a um score. Tal como acontece com Redes Neurais clássicas, cada nodo nesta camada será conectado a todos os nodos no volume anterior.

Ao lidar com entradas de alta dimensionalidade, como imagens, é impraticável conectar todos os nodos para todos os nodos no volume entre os volumes. Em vez disso, conecta-se cada nodo em apenas uma região local do volume de entrada. A extensão espacial dessa conectividade é um parâmetro chamado o campo receptivo do nodo (inspirado biologicamente nas células nervosas da visão). A extensão da ligação ao longo do eixo de profundidade é sempre igual à profundidade do volume de entrada. As conexões são locais no espaço (ao longo de largura e altura), mas são sempre completas ao longo de toda a profundidade do volume de entrada. Na camada de CONV é realizada uma convolução entre a região de entrada ligada ao filtro e este. A saída desta operação construirá o volume de ativações para a próxima camada. A Figura 3.2 exemplifica este processo na camada de convolução. As CNNs tem tido muitos bons resultados em diversos campos quando se trabalha com de imagens, vídeo ou áudio.

É comum inserir periodicamente uma camada de *Pooling*, como mostrado na Figura 3.3, em entre camadas sucessivas CONV numa arquitetura CNN. A sua função é reduzir progressivamente o tamanho espacial da representação para reduzir a quantidade de parâmetros e computação da rede, e, portanto, também para controlar o superajuste. A camada de *Pooling* opera de forma independente em cada plano de nodos do volume da entrada para redimensioná-lo espacialmente, usando a operação MAX. A forma mais comum é uma camada de *Pooling* com filtros de tamanho 2x2 aplicados com um passo de 2, de forma a reduzir a resolução a cada fatia

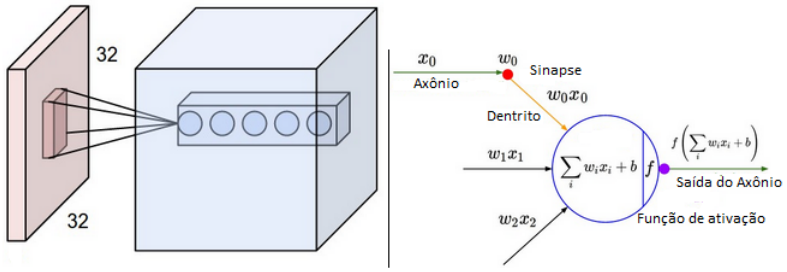

Figura 3.2: Esquerda: Um imagem de entrada (vermelho). Cada nodo na camada convolutional está ligado apenas a uma região local no volume de entrada espacialmente. Existem alguns nodos ao longo da profundidade, todos ligados a mesma região de entrada. Direita: Função de convolução para as entradas pelo filtro (KARPATHY, 2016).

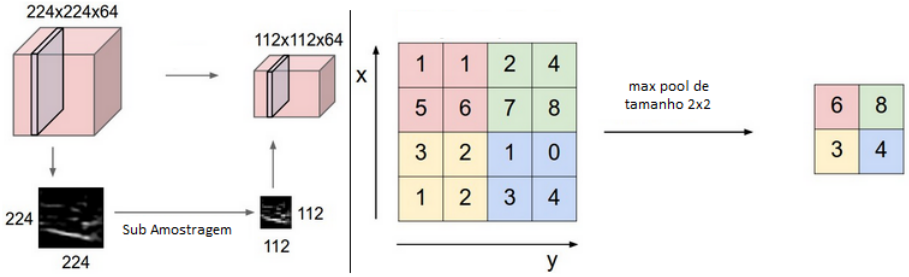

Figura 3.3: A contribuição desta camada é reduzir a resolução do volume espacialmente, independentemente, em cada fatia profundidade do volume de entrada. Esquerda: Neste exemplo, o volume de entrada de tamanho [224x224x64] é reunido com o tamanho do filtro 2, passo 2 em volume de tamanho [112x112x64] saída. Note-se que a profundidade de volume é preservada. Direita: A operação de redução da resolução mais comum é max (max pooling), mostrado aqui com um passo de 2 (KARPATHY, 2016).

de profundidade na entrada por 2 ao longo de largura e altura, descartando 75% das ativações. Cada operação MAX iria, neste caso, tomar um máximo de 4 valores (região 2x2). Assim, a dimensão de profundidade permanece inalterada. O ajuste dos pesos é realizado pelo algoritmo de *Backprogagation* atualizando os pesos para cada camada. A camada de *Pooling* não interfere no gradiente do *Backprogagation*.

3.2 Mapa Auto-Organizável Probabilístico (PSOM)

O Mapa Auto-Organizável Probabilístico (PSOM) (LóPEZ-RUBIO, 2010) é baseado no bem conhecido Mapa Auto-Organizável (SOM), nesta seção será feita uma revisão sobre Mapa Auto-Organizável (SOM), a apresentação dos conceitos gerais de Mapa Auto-Organizável Probabilístico (PSOM) além da apresentação de alguns modelos escolhidos para uso desse trabalho.

3.2.1 Mapa Auto-Organizável (SOM)

O Mapa Auto-Organizável (SOM) é um modelo de rede neural que encontrou ampla aplicação em várias áreas do conhecimento, como: Análise de Imagens (MAIA; BARRETO; COELHO, 2011), Análise de Vídeo (LÓPEZ-RUBIO; LUQUE-BAENA; DOMINGUEZ, 2011), Reconhecimento de Fala (WANG; HAMME, 2011), Processamento de Sinais, Telecomunicações, Física, Robótica (WALTER; SCHULTEN et al., 1993), Design de Circuitos Elétricos (GOSER, 1997), Análise de Série Temporais (BARRETO; ARAUJO, 2001), Análise de Séries Financeiras (DEBOECK; KOHONEN, 2013), Análise de Clima (MORIOKA; TOZUKA; YAMAGATA, 2010), Bioinformática (ARAUJO; BASSANI; ARAUJO, 2013), dentre outros . Uma das vantagens importantes dessa família de modelos de Redes Neurais Artificiais (RNAs) é que elas são capazes de aprender de uma forma não-supervisionada. O SOM foi projetado inicialmente utilizando uma única camada de nodos para realização de agrupamento e visualização dos dados e era utilizado como um quantizador de informação. Com frequência, o seu treinamento é não-supervisionado e é representado em um espaço cuja dimensão é, geralmente, menor (tipicamente de magnitude 2 ou 3) que os dados de treinamento. Além disso, se o SOM procura também preservar as propriedades topológicas do espaço de entrada (ASTUDILLO; OOMMEN, 2014).

O objetivo do mapeamento dentro do SOM é representar todos os casos dos dados de entrada definida por um conjunto (geralmente) menor de protótipos que são os nodos, de tal maneira que os pontos que estão próximos do espaço de entrada são representados pelos nodos que também estão próximos, e vice-versa, ou seja, pontos que estão distantes no espaço de entrada são representados por nodos que estão distantes no espaço-alvo (ASTUDILLO; OOMMEN, 2014).

Alguns conceitos importantes para o entendimento do SOM e são apresentados abaixo (KOHONEN; HARI, 1999; KOHONEN; SCHROEDER; HUANG, 2001):

- Estímulos de entrada, sinais de entrada ou apenas entrada da rede é um vetor de dados n-dimensional, $\mathbf{x} = [x_1\ x_2\ \ldots\ x_n]^\top$, isto é, uma lista de números que representa os valores do estímulo em cada dimensão;
- Nodo ou unidade n_i possui um conjunto de valores numéricos ou pesos sinápticos, w_i. O vetor $\mathbf{w}_i = [w_{i1}\ w_{i2}\ \ldots\ w_{in}]^\top$ possui a mesma dimensão de $\mathbf{x}$ e pode ser considerado uma posição no espaço de entrada;
- O nodo vencedor s possui vetor sináptico $\mathbf{w}_s$, também conhecido como nodo mais adaptado. Este é o que possui o maior grau de semelhança para um dado estímulo de entrada;
- Arestas ou conexão é um conceito comum nos mapas auto-organizáveis. Elas unem os nodos para formar a sua vizinhança;

O SOM usa uma malha A com N nodos. As posições, interconexões e pesos dos nodos são determinados em função do conjunto de entrada $\chi \subseteq \mathbb{R}^d$. O algoritmo de treinamento tenta organizar os nodos de maneira tal que a distribuição e a topologia dos dados de entradaM_A seja preservada. Durante este processo de formação, um vetor de pesos $w_i \subseteq \mathbb{R}^d$ de cada nodo $i \subseteq \mathbb{R}^d$ é ajustado: O vetor $\mathbf{w}_i$ pode ser percebido como a posição do nodo i no espaço de características. Em cada passo do treinamento, um estímulo x, que é um exemplo do conjunto de treinamento, é apresentado à rede e os nodo competem entre si, de modo a identificar qual é o nodo vencedor, *Best Matching Unit* (BMU) (ASTUDILLO; OOMMEN, 2014).

O treinamento do algoritmo SOM padrão inicia por definir uma estrutura topológica da rede cuja estrutura neural permanece estática durante o treinamento. O objetivo desse treinamento SOM é ajustar os pesos w_i, de modo a preservar, o máximo possível, as propriedades topológicas do espaço de entrada M_A. A cada iteração os pesos w_i são atualizados pela chamada regra de atualização dos pesos dado pela Equação 3.1, na qual $w_i(t+1)$ é o próximo peso para o nodo i, w_i é o peso atual para o nodo i, $\alpha(t)$ é a função de taxa de aprendizagem e $h_{j,v(x)}$ é a função de vizinhança (ASTUDILLO; OOMMEN, 2014). Esse processo de atualização será descrito adiante.

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t) h_{j,v(x)}(t)(\mathbf{x} - \mathbf{w}_i(t)) \tag{3.1}$$

A determinação do BMU é uma das mais cruciais etapas do algoritmo SOM, nela é identificado o nodo mais adequado de uma determinada entrada $x \subseteq \chi$ em um processo denominado Aprendizado Competitivo (AC). Seguindo a ideia do AC, o vetor x é apresentado ao SOM e os nodos competem entre si para verificarem qual responde mais fortemente ao estímulo do padrão de entrada. Desta forma, apenas um único vencedor é selecionado, BMU, denotado por $s(x)$, através da Equação 3.2, na qual o argumento da exponencial é dado pelo negativo da distância euclidiana.

$$s(\mathbf{x}) = arg \max_i exp^{-\|\mathbf{x}-\mathbf{w}_i\|} \tag{3.2}$$

A notação pode ser estendida para se referir ao nodo mais próximo $s_1(x)$, o segundo nodo mais próximo $s_2(x)$ e sucessivamente. A Equação 3.2 é um componente fundamental no processo de competição entre os nodos. A partir desta perspectiva, é possível classificar as RNAs em duas sub-categorias. No primeiro grupo, apenas saída do nodo é ativada, *hard* AC. No segundo grupo, tanto o BMU quanto os outros nodos vizinhos estão envolvidos no processo, *soft* AC. O SOM padrão está no grupo de *soft* AC (ASTUDILLO; OOMMEN, 2014).

Uma vez determinado o nodo vencedor alguns nodos além deste também sofrem ajuste. Esse subconjunto de nodos é conhecido como a vizinhança do BMU. Este conceito envolve determinar quais nodos devem ser atualizados em cada etapa de aprendizagem de modo a alcançar a convergência. Em geral, o SOM inicia com um vizinhança larga e vai diminuindo de tamanho para um valor que tende a zero, quando o número de iterações cresce, como descrito na Equação

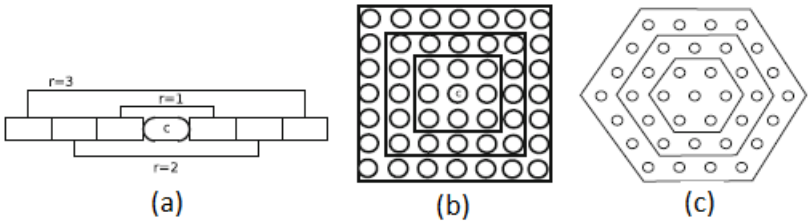

Figura 3.4: (a) Grade Linear. (b) Grade Retangular. (c) Grade Hexagonal.

3.3. Na Figura 3.4 há exemplos de diferentes arranjos de nodos: Grade Linear, Retangular ou Hexagonal (ASTUDILLO; OOMMEN, 2014).

A função de vizinhança determina que tanto o BMU quanto os nodos de sua vizinhança serão atualizados. No SOM padrão, a função de vizinhança é a gaussiana e é apresentada pela Equação 3.3, na qual $v(x)$ é o nodo vencedor, $d_{j,v(x)}$ é a distância entre o nodo vencedor e nodo atual e σ é o fator que decai em função das n épocas. Uma vez que essa função de vizinhança é calculada, o próximo passo do processo consiste na atualização dos nodos na vizinhança. Os pesos w_i do BMU e os nodos em sua vizinhança são atualizados utilizando a Equação 3.1. O conceito implícito dessa atualização em grupo é que esses nodos nas proximidades do BMU vão aprender parte das informações fornecidas pelo estímulo da entrada $\mathbf{x}$ (ASTUDILLO; OOMMEN, 2014).

A grandeza pela qual os nodos são movidos em direção ao estímulo é determinada pela chamada taxa de aprendizagem, $\alpha(t) \in [0,1]$, calculada através da Equação 3.4, na qual α_0 éa taxa de aprendizagem inicial, n é a época das iterações e τ é da taxa de decaimento da exponencial. Grandes valores da taxa de aprendizagem, ou seja, os valores próximos da 1 irão fazer com que a dinâmica dos nodos seja maior. Normalmente, maiores as taxas de aprendizagem são associadas a um comportamento mais caótico. Por outro lado, valores menores para α, ou seja, valores próximos de zero são responsáveis por modificações mais refinadas na posição dos pesos e como resultado se obtém alterações mais suaves nas posições. Em geral, a taxa de aprendizagem sofre um decaimento durante as épocas tendendo a um valor próximo a zero conforme na Equação 3.4 (ASTUDILLO; OOMMEN, 2014). O Algoritmo do SOM padrão é descrito pelo Algoritmo 1 (KOHONEN, 1990).

$$h_{j,v(x)}(n) = \exp^{\left[-d_{j,v}/2\sigma^2(n)\right]} \tag{3.3}$$

$$\alpha(n) = \alpha_0 \exp^{-\frac{n}{\tau}} \tag{3.4}$$

3.2.2 Conceitos Gerais Mapa Auto-Organizável Probabilístico (PSOM)

Os PSOMs usam funções de densidade de probabilidade em seu processo. Cada abordagem de PSOM propõe os seus próprios critérios e métodos para o treinamento da rede. Em se

Algoritmo 1: Pseudo-Algoritmo SOM

```
   /* Definição de variáveis                                   */
 1 X_i: vetor de entrada i;
 2 m: quantidade de atributos do padrão X_i;
 3 i: quantidade de nodos da malha;
 4 η_0: taxa de aprendizado inicial;
 5 w_j: vetor de pesos;
 6 n: quantidade de épocas do algoritmo;
   /* Fase de inicialização                                    */
 7 Inicialize os pesos da malha de nodos com valores aleatórios próximos a zero [-0.01
   +0.01];
 8 for cada época n do
 9     for cada X_i do
           /* Fase de competição                               */
10         Apresente o padrão de entrada X_i à rede;
11         for cada nodo i do
12             calcule a distância entre X_i e peso w_j;
13         Escolha o BMU através da Equação 3.2 ;
           /* Fase de Adaptação                                */
14         Atualize a função de vizinhança através da Equação 3.3;
15         for cada nodo da vizinhaça do BMU do
16             Atualize os pesos conforme a equação 3.1 ;
17     Atualize a taxa de aprendizado 3.4 ;
```

tratando de PSOM, pode-se considerar agrupar em dois tipos no quesito treinamento: aqueles que se baseiam nos algoritmos de Maximização da Expectativa (EM) (DEMPSTER; LAIRD; RUBIN, 1977) e os que utilizam Aproximação Estocástica (SA) (ROBBINS; MONRO, 1951). Estas duas estratégias são muito diferentes. Enquanto EM tenta ajustar o modelo ao conjunto de dados de treinamento (que é uma aprendizagem por lote), a SA assume que as amostras, uma a uma, e o seu erro devem ser calculados a partir da média (LóPEZ-RUBIO, 2010). Os critério de classificação dos PSOM se refere ao tipo de auto-organização: Kohonen KOHONEN (1990), Latente (BISHOP; SVENSéN; WILLIAMS, 1998) e Heskes (HESKES, 2001). Alguns conceitos básicos para PSOM e, posteriormente, dois modelos de PSOM e as suas características particulares, serão apresentados a seguir.

3.2.2.1 Auto-Organização

A auto-organização é um conceito que parece explicar várias estruturas neurais do cérebro que executam detecção de características invariantes (FUKUSHIMA, 1999). O estudo dessas estruturas proporcionou um impulso no desenvolvimento de mapas computacionais criados para explorar e analisar dados multidimensionais. Todos os modelos de PSOM compartilham desta definição de característica, ou seja, a definição de uma Função de Densidade de Probabilidade

(FDP) na representação do nodo. Essa função é definida como uma mistura em que cada unidade probabilística do mapa é associada a um componente de mistura, definindo que os dados de entrada tenham uma dimensão espacial D. A probabilidade dos dados observados $t \in \mathbb{R}^D$ é dada pela mistura da FDP segundo a Equação 3.5, no qual H é o número de componentes da mistura (unidades), π_i é a probabilidade prévia ou a proporção de mistura da unidade i e $p(t|i)$ é uma FDP associada à unidade i (LóPEZ-RUBIO, 2010).

$$p(\mathbf{t}) = \sum_{i=1}^{H} \pi_i p(\mathbf{t}|i) \tag{3.5}$$

Em modelos baseados em agrupamento as amostras de dados são agrupadas através da aprendizagem de um modelo de mistura (geralmente um modelo de Mistura gaussianas) na qual cada componente de mistura representa um grupo de dados. Existem dois métodos principais para aprendizagem para esses modelos: (i) Mistura de Verossimilhança, onde a probabilidade de cada amostra de dados é uma mistura de todas das probabilidades de componentes da amostra de dados; (ii) Classificação da Verossimilhança na qual a probabilidade de cada uma das amostras de dados é gerada por sua componente onde somente uma é a vencedora (SYMONS, 1981; FRALEY; RAFTERY, 2007). Nessas abordagens a estimativa ótima global dos parâmetros do modelo não pode ser obtida analiticamente, algoritmos de aprendizagem iterativos que só garantem em obter soluções localmente ótimas.

3.2.2.2 Componentes da Mistura (MC)

Os métodos baseados em Misturas de Densidades (TITTERINGTON; SMITH; MAKOV, 1985) utilizam Múltiplas Funções de Densidade de Probabilidade combinadas entre si para construir um função hipotética. Essas múltiplas funções são chamadas de Componentes da Mistura (MC), em geral, seu conjunto de parâmetros é desconhecido a princípio. Embora as componentes da misturas possam ter formatos paramétricos distintos, as suas distribuições são gaussianas e não raramente representadas pelos mesmos conjuntos de parâmetros (JAIN; MURTY; FLYNN, 1999), também são utilizadas distribuições *t-Student* (LóPEZ-RUBIO, 2010) Esta hipótese se deve a grande complexidade computacional da estimativa desses valores que são métodos iterativos baseados na ideia de Máxima Verossimilhança (ML) dos vetores de parâmetros das componentes da mistura (JAIN; DUBES, 1988). Existem duas abordagens para o cálculo dos MC que são Maximização da Expectativa (EM) e Aproximação Estocástica (SA).

3.2.3 Modelos de Mapa Auto-Organizável Probabilístico

Os principais modelos de PSOM listados na literatura podem ser vistos na Tabela 3.1. As características dos PSOM que são evidências são: Componentes, Treinamento e Tipo de Auto-Organização. Dois modelos foram escolhidos para serem utilizados e neste trabalho. A escolha foi feita por *Probabilistic Self-Organizing Map* (PbSOM) e *t-Student Self-Organizing*

Map (TSOM) por apresentarem melhores resultados para bases de dados do UCI (LóPEZ-RUBIO, 2010; BLAKE; MERZ, 1998) e por contemplar variações de conceituais existentes no conjunto de modelos.

Modelo	Referência	Componentes	Treinamento	Tipo de Auto-Organização
GMT	BISHOP; SVENSéN; WILLIAMS (1998)	Gaussiana-Esférica	EM	Latente
t-GMT	VELLIDO (2006)	*t-Student*-Esférica	EM	Latente
SOMM	VERBEEK; VLASSIS; KRÖSE (2005)	Gaussiana-Esférica	EM	Heskes
MLTM	VAN HULLE (2005)	Gaussiana-Esférica	EM	Kohonen
PbSOM	CHENG; FU; WANG (2009)	Gaussiana	EM	Kohonen
KBTM	VAN HULLE (2002)	Gaussiana-Esférica	SA	Kohonen
SOMN	YIN; ALLINSON (2001)	Gaussiana	SA	Heskes
PPCASOM	LÓPEZ-RUBIO; LAZCANO-LOBATO; LÓPEZ-RODRÍGUEZ (2009)	Gaussiana-PPCA	SA	Heskes
TSOM	LÓPEZ-RUBIO (2009)	*t-Student*	SA	Heskes
PSOG	LÓPEZ-RUBIO et al. (2011)	Gaussiana	SA	Heskes
GHPSOG	LÓPEZ-RUBIO; PALOMO (2011)	Gaussiana	SA	Heskes

Tabela 3.1: Resumo dos principais modelos de PSOM relatadas na literatura.

3.3 *Probabilistic Self-Organizing Map* (PbSOM)

A elaboração do PbSOM (CHENG; FU; WANG, 2009) foi motivada pelo conceito de Energia de Acoplamento (SUM et al., 1997). Nessa rede se utiliza um modelo de Misturas de Acoplamento de Verossimilhança através de distribuições gaussianas multivariadas. No PbSOM, os locais de energia de acoplamento entre os nodos e suas vizinhanças estão expressos em termos de probabilidades e cada uma das componentes de misturas expressa a probabilidade de acoplamento local entre um nodo e sua vizinhança. Foram desenvolvidos três algoritmos de aprendizagem para o PbSOM; o *Self-Organizing Conditional Expectation Maximization* (SOCEM), *Self-Organizing Expectation Maximization* (SOEM) e *Self-Organizing Deterministic Annealing Expectation Maximization* (SODAEM) (CHENG; FU; WANG, 2009) que foram inspirados nos algoritmos *Conditional Expectation Maximization* (CEM) (CELEUX; GOVAERT, 1992), EM (BILMES et al., 1998; MCLACHLAN; KRISHNAN, 2007) e *Deterministic Annealing Expectation Maximization* (DAEM) (UEDA; NAKANO, 1998), respectivamente. Como herdam as propriedades dos algoritmos CEM, EM e DAEM esses algoritmos do PbSOM são caracterizados por confiança de convergência, de baixo custo por iteração, a economia de armazenamento dos dados e facilidade de programação.

O PbSOM pode ser definido como um SOM com G nodos $R = \{r_1, r_2, .., r_G\}$ em uma rede com função de vizinhança h_{kl} que define a intensidade da interação lateral entre dois nodos r_l e r_k, para $k, l \in \{1, 2, .., G\}$. Cada nodo é associado com um modelo de referência que representa

alguma distribuição de probabilidade no espaço de dados. O algoritmo de aprendizagem de Kohonen pode ser interpretado em termos de maximização das correlações locais (energias) de acoplamento entre os nodos e suas vizinhanças com os dados de entrada (SUM et al., 1997). Dada uma amostra de dados. A partir de dados de entrada $\mathbf{x}_i \in \chi = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, a energia de acoplamento local entre r_k e a vizinhança é definida pela Equação 3.6. No qual $r_k(\mathbf{x}_i; \boldsymbol{\theta}_k)$ indica a resposta do nodo r_k ao vetor $\mathbf{x}_i$ e a distribuição de probalidade do nodo $\boldsymbol{\theta}_k$. Assim, a energia de acoplamento da rede é definida pela Equação 3.7. A função a ser maximizada é dada pela Equação 3.8. O termo $\sum_{l=1}^{G} h_{kl} r_l(\mathbf{x}_i; \boldsymbol{\theta}_l)$, na Equação 3.6 pode ser considerada como a resposta da vizinhança de r_k, na qual a conjunção entre as respostas de nodos é implementada usando a operação somatória.

$$E_{xi|k} = r_k(\mathbf{x}_i; \boldsymbol{\theta}_k) \sum_{l=1}^{G} h_{kl} r_l(\mathbf{x}_i; \boldsymbol{\theta}_l) \tag{3.6}$$

$$E_{x_i} = \sum_{k=1}^{G} E_{x_i|k} \tag{3.7}$$

$$C = \sum_{i=1}^{N} \log E_{x_i} \tag{3.8}$$

Para cada vetor de entrada $\mathbf{x}_i$ é definida uma energia de acoplamento local entre r_k e a vizinhança como uma verossimilhança acoplada expressa pela Equação 3.9, na qual Θ é um conjunto modelos referência e h denota a função de vizinhança e p_s é a probabilidade de acoplamento local de cada nodo. Quando definido o acoplamento de verossimilhança de $\mathbf{x}_i$ através da rede, representa-se a mistura de verossimilhanças pela Equação 3.10, na qual $w_s(k)$ varia com $k = 1, 2, ..., G$.

$$p_s(\mathbf{x}_i|k; \Theta, h) = \exp\left(\sum_{k=1}^{G} h_{kl} \log r_l(\mathbf{x}_i; \theta_l)\right) \tag{3.9}$$

$$p_s(\mathbf{x}_i; \Theta, h) = \sum_{k=1}^{G} w_s(k) p_s(\mathbf{x}_i|k; \Theta, h) \tag{3.10}$$

Assim, os pesos associados às misturas podem ser aprendidos automaticamente. O o processo de aprendizagem se dá em maximizar a probabilidade de acoplamento local para cada nodo, ou seja, entre nodo e sua vizinhança dos dados fornecidos amostra $\mathbf{x}_i$. Pode-se comparar a estrutura de rede do modelo de Mistura de Acoplamento de Verossimilhança, representado pela Equação 3.10 com modelo de mistura de gaussianas (GMM), como mostrado na Figura 3.5. O PbSOM insere uma camada de acoplamento-verossimilhança entre a camada gaussiana-verossimilhança e da camada de mistura-verossimilhança para ter avaliação do acoplamento entre os nodos e suas vizinhanças. CHENG; FU; WANG (2009) propuseram três algoritmos de aprendizagem: *Self-Organizing Conditional Expectation Maximization* (SOCEM), *Self-*

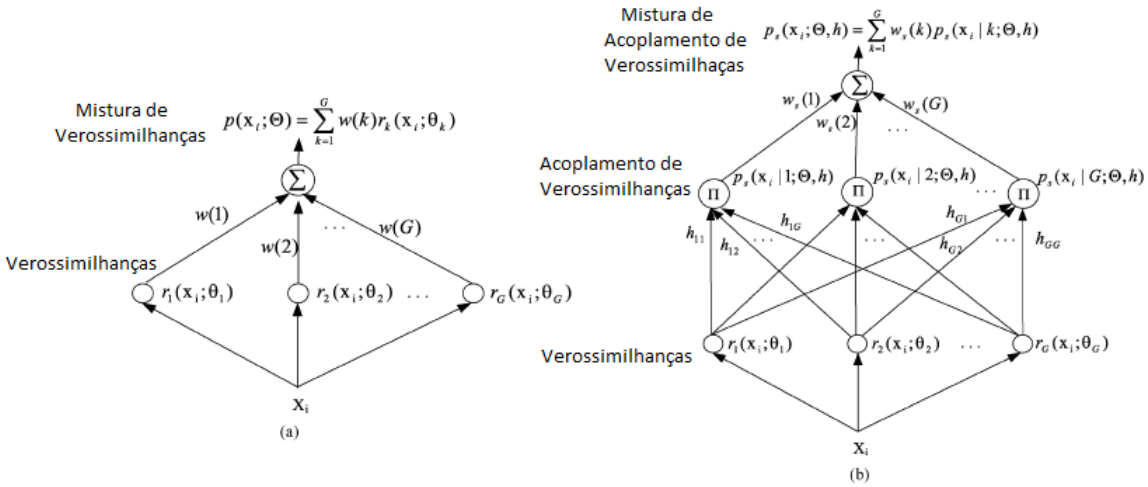

Figura 3.5: (a) Rede com modelo de mistura Gaussiana. (b) Rede com modelo de acoplamento de verossimilhança (CHENG; FU; WANG, 2009).

Organizing Expectation Maximization (SOEM) e *Self-Organizing Deterministic Annealing Expectation Maximization* (SODAEM) que serão descritos abaixo.

3.3.1 *Self-Organizing Conditional Expectation Maximization* (SOCEM)

O processo de auto-organização de PbSOM pode ser descrito como um procedimento de agrupamento de dados baseado em modelo que preserva a relação espacial entre os agrupamentos na rede. Baseado no critério de classificação da verossimilhança para agrupamento de dados (CELEUX; GOVAERT, 1992), o cálculo da probabilidade de uma amostra de dados é restringida ao nodo vencedor. Assim, o objetivo é o de estimar partição χ, $\hat{P}=\{\hat{P}_1,\hat{P}_2,...,\hat{P}_G\}$ e o conjunto de modelos de referência $\hat{\Theta}$, com o objetivo de maximizar o logaritmo da verossimilhança sobre as entradas, através da Equação 3.11, na qual h_{kl} é a função de vizinhança e C_s é a função objetivo.

$$C_s(\Theta,\hat{P}^{(t)},\chi,h)=\sum_{l=1}^{G}\sum_{k=1}^{G}\sum_{\mathbf{x}_i\in\hat{P}_k^{(t)}} h_{kl}\log r_l(\mathbf{x}_i,\theta_l)+Const. \tag{3.11}$$

Semelhante ao algoritmos de Kohonen, os algoritmos do PbSOM são aplicados em duas fases. A primeira etapa é aplicada a um grande vizinhança para formar um mapa ordenado perto do centro das amostras de dados. Em seguida, os modelos de referência dos nodos são adaptados para ajustar a distribuição das amostras dos dados gradualmente diminuindo o raio da vizinhança. Sem perda de generalidade, a função de vizinhança é adotada como gaussiana, Equação 3.12. O valor do σ é alto inicialmente vai decaindo com a convergência do algoritmo.

$$h_{kl}=\exp\left(-\frac{\|r_k-r_l\|^2}{2\sigma^2}\right) \tag{3.12}$$

Os pseudo-algoritmos do SOCEM, SOEM e SODAEM podem ser visualizados em:

Algoritmo 2 é a inicialização das variáveis e estruturas de dados, já o Algoritmo 3 é a parte do algoritmo responsável pelo processo de aprendizado. O algoritmo de aprendizagem SOCEM possui 3 passos: E, C e M. A Etapa E é responsável pelos cálculos das funções de verossimilhança em relação entrada, o passo C define quais são os nodos ganhadores e o passo M atualiza as representações dos nodos, mu_l e Σ_l.

- Passo E: Dado um conjunto de modelos $\Theta^{(t)}$, calcule a probabilidade a posteriori de cada componente da mistura $p_s = (\mathbf{x}_i; \Theta^{(t)}, h)$ para $\mathbf{x}_i$ na Equação 3.13.

$$\gamma_{k|i}^{(t)} = p_s(k|\mathbf{x}_i; \Theta^{(t)}, h) = \frac{\exp\left(\sum_{l=1}^{G} h_{kl} \log r_l(\mathbf{x}_i, \theta_l^{(t)})\right)}{\sum_{j=1}^{G} \exp\left(\sum_{l=1}^{G} h_{kl} \log r_l(\mathbf{x}_i, \theta_l^{(t)})\right)} \tag{3.13}$$

- Passo C: Para cada entrada $\mathbf{x}_i$ existe um agrupamento que corresponde a maior probabilidade para a entrada, como $\mathbf{x}_i \in \hat{P}_j^{(t)}$, se $j = \arg\max_k \gamma_{k|i}^{(t)}$.

- Passo M: A partição de χ é formada pela função objetivo definida pela Equação 3.11. Fazendo a derivação do passo M do algoritmo EM para aprender um modelo de mistura gaussiana (BILMES et al., 1998), a média é definida pela Equação 3.14 e a covariância pela Equação 3.15 podem ser deduzidas através da equação da distribuição gaussiana.

$$\mu_l^{(t+1)} = \frac{\sum_{k=1}^{G} \sum_{\mathbf{x}_i \in \hat{P}_k^{(t)}} h_{kl} \mathbf{x}_i}{\sum_{k=1}^{G} \left|\hat{P}_k^{(t)}\right| h_{kl}} \tag{3.14}$$

$$\Sigma_l^{(t+1)} = \frac{\sum_{k=1}^{G} \sum_{\mathbf{x}_i \in \hat{P}_k^{(t)}} h_{kl} (\mathbf{x}_i - \mu_l^{(t+1)})(\mathbf{x}_i - \mu_l^{(t+1)})^\top}{\sum_{k=1}^{G} \left|\hat{P}_k^{(t)}\right| h_{kl}} \tag{3.15}$$

3.3.2 *Self-Organizing Expectation Maximization* (SOEM)

No algoritmo SOCEM apenas as probabilidades de acoplamento locais associadas com os nodos vencedores são considerados. Alternativamente, calcula-se a probabilidade de acoplamento de usar a probabilidade de mistura definida pela Equação 3.10 e aplicar o algoritmo EM para maximizar a função objetivo de log-verossimilhança. O algoritmo de aprendizagem SOEM possui 2 passos: passo E e M. Estes serão descritos abaixo:

- Passo E:

  Com o modelo de mistura dado pela Equação 3.10, reescreve-se a Equação 3.16, no qual γ a probabildade a posteriori.

$$Q_s(\Theta, \Theta^{(t)}) = \sum_{l=1}^{N} \sum_{i=1}^{G} \sum_{k=1}^{N} \gamma_{k|i}^{(t)} h_{kl} \log r_l(\mathbf{x}_i, \theta_l) + Const. \tag{3.16}$$

- Passo M:

  Substituindo a resposta $r_l(\mathbf{x}_i, \theta_l)$ na Equação 3.16 com a função de densidade gaussiana multivariada derivando a partir de Q_s a média pode ser definida pela Equação 3.17 e a covariância pela Equação 3.18.

$$\mu_l^{(t+1)} = \frac{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\gamma_{k|i}^{(t)}h_{kl}\right)\mathbf{x}_i}{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\gamma_{k|i}^{(t)}h_{kl}\right)} \tag{3.17}$$

$$\Sigma_l^{(t+1)} = \frac{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\gamma_{k|i}^{(t)}h_{kl}\right)(\mathbf{x}_i-\mu_l^{(t+1)})(\mathbf{x}_i-\mu_l^{(t+1)})^\top}{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\gamma_{k|i}^{(t)}h_{kl}\right)} \tag{3.18}$$

3.3.3 *Self-Organizing Deterministic Annealing Expectation Maximization* (SODAEM)

Seguindo a derivação do algoritmo de aprendizagem do*Deterministic Annealing Expectation Maximization* (DAEM) em relação ao GMM (UEDA; NAKANO, 1998) tem-se o DAEM em relação ao PbSOM que é SODAEM. Com a probabilidade de mistura definida pela Equação 3.10, DAEM deriva a densidade a posteriori no passo E usando a máxima entropia.

- Passo E: A derivação da probabilidade a posteriori (UEDA; NAKANO, 1998) com conjunto de parâmetros do modelo atual, obtém-se a probabilidade a posteriori do k-ésimo componente de mistura apresentado na Equação 3.19. A função objetivo auxiliar para minimização está descrita pela Equação 3.20.

$$\tau_{k|i}^{(t)} = \frac{\exp\left(\beta\sum_{l=1}^{G}h_{kl}\log r_l(\mathbf{x}_i,\theta_l^{(t)})\right)}{\sum_{j=1}^{G}\exp\left(\beta\sum_{l=1}^{G}h_{kl}\log r_l(\mathbf{x}_i,\theta_l^{(t)})\right)} \tag{3.19}$$

$$U_s(\Theta,\Theta^{(t)}) = \sum_{l=1}^{N}\sum_{i=1}^{G}\sum_{k=1}^{N}\tau_{k|i}^{(t)}h_{kl}\log r_l(\mathbf{x}_i,\theta_l) + Const. \tag{3.20}$$

- Passo M: As médias e covariâncias são derivadas através da função de minimização e podem ser descritas pela Equação 3.21 e 3.22, respectivamente.

$$\mu_l^{(t+1)} = \frac{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\tau_{k|i}^{(t)}h_{kl}\right)\mathbf{x}_i}{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\tau_{k|i}^{(t)}h_{kl}\right)} \tag{3.21}$$

$$\Sigma_l^{(t+1)} = \frac{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\tau_{k|i}^{(t)}h_{kl}\right)(\mathbf{x}_i-\mu_l^{(t+1)})(\mathbf{x}_i-\mu_l^{(t+1)})^\top}{\sum_{i=1}^{N}\left(\sum_{k=1}^{G}\tau_{k|i}^{(t)}h_{kl}\right)} \tag{3.22}$$

$$Const = \log(((2\pi)^{(d/2)}\|\sigma\|)^{1/2}) \tag{3.23}$$

$$r(x_i, \theta_j) = -\frac{1}{2}\frac{\left\|x_i - \mu_j\right\|}{\sigma_j^2} - Const \tag{3.24}$$

$$\sigma_{ij}^2 = \min\left\{\left\|\mu_i - \mu_j\right\|\right\} \tag{3.25}$$

Algoritmo 2: Pseudo-Algoritmo PbSOM - Etapa Inicialização

```
   /* Definição de variáveis                                          */
 1 x_[i×d]: dados de treinamento;
 2 n: quantidade de padrões de entrada;
 3 d: dimensão do espaço dos dados de entrada;
 4 k: quantidade de nodos na grade;
 5 x_[k×i]: malha de nodos;
 6 σ: parâmetro de entrada sigma;
 7 σ_min: parâmetro de entrada sigma mínima admitida nodos;
 8 σ_step: parâmetro de entrada de decréscimo do sigma;
 9 minVar: valor mínimo da variância (valor empírico);
10 h_kl: matriz de distância entre os nodos (l=k);
11 β: fator exponencial (SODAEM);
12 β_max: valor teto do fator exponencial (SODAEM);
13 C_[k×i]: valores de verossimilhança (SOCEM);
14 Q_[k×i]: valores de verossimilhança (SOEM);
15 U_[k×i]: valores de verossimilhança (SODAEM);
   /* Criação das Estruturas                                          */
16 Crie uma malha de nodos distribuída uniformemente: r_k = 1,2,...,K;
   /* Inicialização das variáveis e matrizes                          */
17 Inicialize cada nodo com um vetor médio dos dados de treinamento;
18 switch tipo de aprendizado do
19     case SOCEM || SOEM
20         Inicialize cada variância dos nodos com o valor unitário;
21     case SODAEM
22         A variância do vetor será baseada na menor norma entre os vetores
           representativos dos nodos através da Equação 3.25;
```

Algoritmo 3: Pseudo-Algoritmo PbSOM - Etapa Aprendizagem

```
/* Processo de Aprendizado                                          */
while σ > σ_min do
    Calcule h através da Equação 3.12;
    Calcule o valor da função r através da Equação 3.24;
    switch tipo de aprendizado do
        case SOCEM
            Calcule C entre todos os nodos e todos as entradas através da Equação
            3.11;
            Atualize o vetor de cada nodo através da Equação 3.14;
            Atualize as covariâncias através da Equação 3.15;
            Verifique para cada entrada qual é o nodo vencedor para elas a partir da
            maior saída da matriz C;
            Calcule o somatório de todos os elementos da matriz C para score (σ) de
            andamento de algoritmo;
        case SOEM
            Calcule Q entre todos os nodos e todos as entradas através da equação
            3.16;
            Atualize o vetor de cada nodo através da Equação 3.17;
            Atualize as covariâncias através da Equação 3.18;
            Verifique o maior valor para cada nodo em Q;
            Calcule o somatório de todos os elementos da matriz Q para score (σ) de
            andamento de algoritmo;
        case SODAEM
            Calcule U entre todos os nodos e todos as entradas através da Equação
            3.20;
            Atualize o vetor de cada nodo através da Equação 3.21;
            Atualize as covariâncias através da Equação 3.22;
            Verifique o maior valor para cada nodo em U;
            Calcule o somatório de todos os elementos da matriz U para score (σ) de
            andamento de algoritmo;
    switch tipo de aprendizado do
        case SOCEM || SOEM
            Atualize σ, σ = σ − σ_step;
        case SODAEM
            if β > β_max then
                Atualize β, β = β ∗ const (const > 1);
            else if β < β_max then
                Atualize σ, σ = σ − σ_step;
```

3.4 *t-Student Self-Organizing Map* (TSOM)

O TSOM (LÓPEZ-RUBIO, 2009) é baseado na distribuição *t-Student*. Essa distribuição possui benefício principalmente se as distribuição de probabilidade tiver caudas longas (PEEL; MCLACHLAN, 2000). Elas também são interessantes para representar uma pequena quantidade de dados. Este tipo de mistura, *t-Student*, pode ser definida pela Equação 3.5, onde H é o número de componentes da mistura, e π_i (inicialmente definido pela Equação 3.27) são as probabilidades a priori ou proporções de mistura. A probabilidade *t-Student* Multivariada associada a cada componente de mistura i é dada pela Equação 3.26, onde Γ é a função gama (uma extensão da função fatorial para o conjunto dos números reais e complexos), D é a dimensão das amostras de entrada $\mathbf{t}$ e $\boldsymbol{\mu}_i$ é o vetor média, ν_i (inicialmente definido pela Equação 3.28) é o grau de liberdade e o $\boldsymbol{\Sigma}_i$ (inicialmente definido pela Equação 3.30) é denominada de matriz de escala$\mathbf{C}_i$ (inicialmente definido pela Equação 3.29) e não deve ser confundida com a matriz de covariância (LÓPEZ-RUBIO, 2009).

$$p(\mathbf{t}|i) = \frac{\frac{\Gamma(\nu_i+D)}{2}}{|\boldsymbol{\Sigma}_i|^{1/2}(\pi\nu_i)^{D/2}\Gamma\left(\frac{\nu_i}{2}\right)} \times \left(1 + \frac{(\mathbf{t}-\boldsymbol{\mu}_i)^\top)|\boldsymbol{\Sigma}_i|^{-1}(\mathbf{t}-\boldsymbol{\mu}_i))}{\nu_i}\right)^{\frac{-\nu_i-D}{2}} \tag{3.26}$$

$$\pi_i(0) = \frac{1}{H} \tag{3.27}$$

$$\boldsymbol{\mu}_i(0) = \frac{1}{K}\sum_{n=1}^{K}\mathbf{t}_n \tag{3.28}$$

$$\mathbf{C}_i(0) = \frac{1}{K}\sum_{n=1}^{K}(\mathbf{t}_n - \boldsymbol{\mu}_i(0))(\mathbf{t}_n - \boldsymbol{\mu}_i(0))^\top \tag{3.29}$$

$$\boldsymbol{\Sigma}_i(0) = \frac{\nu_i(0)-2}{\nu_i(0)}\mathbf{C}_i(0) \tag{3.30}$$

Foi assumido que $\nu_i > 2$ uma vez que esto garante convergência para as expectativas que aparecem nas equações acima. Note que, nestas condições, tanto $\boldsymbol{\Sigma}_i$ e quanto $\mathbf{C}_i$ são simétricas e positivas. A *t-Student* multivariada é uma generalização da gaussiana multivariada, dado o limite de $\nu \to \infty$ reduz a uma gaussiana com média $\boldsymbol{\mu}_i$ e a matriz de covariância $\mathbf{C}_i$ (LÓPEZ-RUBIO, 2009).

Quando como $\nu_i \to 2$ a distribuição *t-Student* tem caudas mais pesadas, o que fornece uma melhor capacidade de modelar as amostras periféricas (longe da média $\boldsymbol{\mu}_i$). A estimativa do parâmetro de graus livres é bem conhecida na literatura como um problema difícil, o que retarda a convergência (PEEL; MCLACHLAN, 2000). Quando aplicada a uma mistura de distribuições para a estimativa de máxima verossimilhança dos parâmetros de mistura, o Método Maximização

da Expectativa (EM) oferece soluções de forma fechada para a atualização iterativa de π_i, μ_i e Σ_i (LÓPEZ-RUBIO, 2009).

No entanto, não existe uma solução de forma fechada para a atualização ν_i, e sim uma equação não-linear de ν_i para ser resolvida. Os métodos *Peel* (PEEL; MCLACHLAN, 2000) e *Shoham* (SHOHAM, 2002), trazem propostas dessas equações não-lineares e pode-se considerar ambos os métodos como opções para estimar ν_i. Eles são métodos iterativos que dependem de um valor de inicialização para ν_i, uma vez que as equações de atualização para ν_i dependem $\mathbf{\Sigma}_i$. Uma terceira alternativa, de um método não-iterativo é método *Direct* (LÓPEZ-RUBIO, 2009) que estima ν_i sem a necessidade de uma estimativa anterior de ν_i. Isto reduz os efeitos indesejáveis das estimativas iniciais pobres de ν_i e propagação de erros entre ν_i e $\mathbf{\Sigma}_i$. Então as variações do TSOM são *Peel*, *Shoham* e *Direct*. Os pseudo-algoritmos do TSOM e suas respectivas equações utilizadas em cada etapa foram descritas no algoritmo de inicialização (Algoritmo 4) e na etapa de aprendizado (Algoritmo 5).

$$\mathbf{m}_i(0) = \frac{1}{H}\boldsymbol{\mu}(0) \tag{3.31}$$

$$\mathbf{M}_i(0) = \frac{1}{H}\mathbf{C}_i(0) \tag{3.32}$$

$$\omega_i(0) = \frac{1}{HK}\sum_{n=1}^{K} \times \left[\log\left(\frac{\nu_i(0)+D}{\nu_i(0)+(\mathbf{t}_n-\boldsymbol{\mu}_i(0))^T\mathbf{\Sigma}_i(0)^{-1}(\boldsymbol{t}_n-\boldsymbol{\mu}_i(0))}\right) - \left(\frac{\nu_i(0)+D}{\nu_i(0)+(\mathbf{t}_n-\boldsymbol{\mu}_i(0))^T\mathbf{\Sigma}_i(0)^{-1}(\boldsymbol{t}_n-\boldsymbol{\mu}_i(0))}\right)\right] \tag{3.33}$$

$$\omega_i(0) = \frac{1}{HK}\sum_{n=1}^{K} \times \left[\log\left(\frac{\nu_i(0)+D}{\nu_i(0)+(\mathbf{t}_n-\boldsymbol{\mu}_i(0))^T\mathbf{\Sigma}_i(0)^{-1}(\boldsymbol{t}_n-\boldsymbol{\mu}_i(0))}\right) - \left(\frac{2}{\nu_i(0)+(\mathbf{t}_n-\boldsymbol{\mu}_i(0))^T\mathbf{\Sigma}_i(0)^{-1}(\boldsymbol{t}_n-\boldsymbol{\mu}_i(0))}\right)\right] \tag{3.34}$$

$$\omega_i(0) = \frac{1}{HK}\sum_{n=1}^{K} \log\sqrt{(\mathbf{t}_n-\boldsymbol{\mu}_i(0))^T\mathbf{C}_i^{-1}(0)(\mathbf{t}_n-\boldsymbol{\mu}_i(0))} \tag{3.35}$$

$$-\psi(\frac{\nu_i(n)}{2})+\log(\frac{\nu_i(n)}{2})+1+\gamma_i(n)+\psi(\frac{\nu_i(n)+D}{2})-\log(\frac{\nu_i(n)+D}{2})=0 \tag{3.36}$$

$$\log(\frac{\nu_i(n)}{2})+1-\psi(\frac{\nu_i(n)}{2})+\gamma_i(n)=0 \tag{3.37}$$

$$\gamma_i(n) - \frac{1}{2}(\psi(\frac{D}{2}) - \psi(\frac{\nu_i(n)}{2}) - \log(\frac{1}{\nu_i(n)-2})) = 0 \tag{3.38}$$

$$\Sigma_i(0) = \frac{\nu_i(0)-2}{\nu_i(0)} C_i(0) \tag{3.39}$$

$$\pi_i(n) = (1-\varepsilon(n))\pi_i(n-1) + \varepsilon(n) R_{ni} \tag{3.40}$$

$$\mathbf{m}_i(n) = (1-\varepsilon(n))\mathbf{m}_i(n-1) + \varepsilon(n) R_{ni} \mathbf{t}_n \tag{3.41}$$

$$\boldsymbol{\mu}_i(n) = \frac{\mathbf{m}_i(n)}{\pi_i(n)} \tag{3.42}$$

$$\mathbf{M}_i(n) = (1-\varepsilon(n))\mathbf{M}_i(n-1) + \varepsilon(n) R_{ni} (\mathbf{t}_n - \mu_i(n)) \times (\mathbf{t}_n - \mu_i(n))^\top \tag{3.43}$$

$$\mathbf{C}_i(n) = \frac{\mathbf{M}_i(n)}{\pi_i(n)} \tag{3.44}$$

$$\begin{aligned} \omega_i(n) = (1-\varepsilon(n))\omega_i(n-1) + \varepsilon(n) R_{ni} \times \\ \left[\log\left(\frac{\nu_i(n-1)+D}{\nu_i(n-1) + (\mathbf{t}_n - \boldsymbol{\mu}_i(n))^T \boldsymbol{\Sigma}^{-1}(n-1)(\boldsymbol{t}_n - \boldsymbol{\mu}_i(n))} \right) \right. \\ \left. - \left(\frac{\nu_i(n-1)+D}{\nu_i(n-1) + (\mathbf{t}_n - \boldsymbol{\mu}_i(n-1))^T \boldsymbol{\Sigma}^{-1}(n-1)(\boldsymbol{t}_n - \boldsymbol{\mu}_i(n))} \right) \right] \end{aligned} \tag{3.45}$$

$$\begin{aligned} \omega_i(n) = (1-\varepsilon(n))\omega_i(n-1) + \varepsilon(n) R_{ni} \times \\ \left[\log\left(\frac{\nu_i(n-1)+D}{\nu_i(n-1) + (\mathbf{t}_n - \boldsymbol{\mu}_i(n))^T \boldsymbol{\Sigma}^{-1}(n-1)(\boldsymbol{t}_n - \boldsymbol{\mu}_i(n))} \right) \right. \\ \left. - \left(\frac{2}{\nu_i(n-1) + (\mathbf{t}_n - \boldsymbol{\mu}_i(n-1))^T \boldsymbol{\Sigma}^{-1}(n-1)(\boldsymbol{t}_n - \boldsymbol{\mu}_i(n))} \right) \right] \end{aligned} \tag{3.46}$$

$$\omega_i(n) = (1-\varepsilon(n))\omega_i(n-1) + \varepsilon(n) R_{ni} \log \sqrt{(\mathbf{t}_n - \boldsymbol{\mu}_i(n))^T \mathbf{C}_i^{-1}(n)(\mathbf{t}_n - \boldsymbol{\mu}_i(n))} \tag{3.47}$$

$$\gamma_i(n) = \frac{\omega_i(n)}{\pi_i(n)} \tag{3.48}$$

$$\Sigma_i(n) = \frac{\nu_i(n)-2}{\nu_i(n)} C_i(n) \tag{3.49}$$

$$r_{nj}(z_n = i) = P(z_n = i|r_{nj}) \exp\left(-\frac{\delta(h_i,h_j)}{\Delta(n)}\right) \tag{3.50}$$

$$\Delta(n) = \Delta(0)\left(1 - \frac{n}{steps}\right) \tag{3.51}$$

$$\delta(i,j) = \left\|(i_x,i_y) - (j_x,j_y)\right\| \tag{3.52}$$

$$q_{nji} = \exp^{\left[-\frac{1}{2}\frac{\delta(i,j)}{\Delta(n)}\right]} \tag{3.53}$$

$$p(\mathbf{t}_n|q_{nji}) = \sum_{i=1}^{H} q_{nji} p(\mathbf{t}_n|i) \tag{3.54}$$

$$Winner(n) = arg\max_j \left\{p(q_{nj}|\mathbf{t}_n)\right\} \tag{3.55}$$

$$\varepsilon(n) = \frac{1}{\varepsilon_0 n + \frac{1}{\varepsilon_0}}, \quad 0 < \varepsilon_0 < 1 \tag{3.56}$$

$$R_{ni} = P(i|\mathbf{t}_n, Q_n) = P(i, q_{nh}) = q_{nhi}, ondeh = Winner(n) \tag{3.57}$$

$$p(t|i) = \frac{\Gamma\left(\frac{\nu+D}{2}\right)}{|\Sigma_i|^{1/2}(\pi\nu_i)^{D/2}\Gamma\left(\frac{\nu_i}{2}\right)} \times \left(1 + \frac{(t-\mu_i)^\top \Sigma_i^{-1}(t-\mu_i)}{\nu_i}\right)^{\left(\frac{-\nu_i-D}{2}\right)} \tag{3.58}$$

$$\mathbf{\Sigma}_i = \frac{\nu_i(n)-2}{\nu_i(n)} C_i(n) \tag{3.59}$$

3.5 Discussão

As técnicas que foram apresentadas neste capítulo vão servir de inspiração ou de uso direto no modelo proposto no próximo capítulo. A primeira etapa do modelo foi inspirada em ideias de Aprendizagem Profunda que tem tido bons resultados nos últimos anos conforme relatos presentes na literatura revisada. Na segunda etapa serão utilizados modelos de Mapa Auto-Organizável Probabilístico (PSOM), mais especificamente *Probabilistic Self-Organizing Map* (PbSOM) e *t-Student Self-Organizing Map* (TSOM). Esses modelos foram escolhidos de acordo com o desempenho (LÓPEZ-RUBIO, 2009) e na variedade dos métodos de funcionamento destes. No próximo capítulo, os modelos propostos serão especificado em detalhes.

Algoritmo 4: Pseudo-Algoritmo TSOM - Etapa Inicialização

```
   /* Definição de variáveis                                          */
 1 t_[n×D]: dados de treinamento;
 2 n: quantidade de padrões de entrada;
 3 D: dimensão do espaço dos dados de entrada;
 4 K: quantidade de elementos na partição;
 5 H: quantidade de mistura de componentes (nodos);
 6 steps: quantidade total de iterações;
 7 π_i: Probabilidade a priori das misturas;
 8 μ_i: Média das distribuições;
 9 m_i: Média das distribuições (auxiliar);
10 C_i: Matriz de covariância das distribuições;
11 M_i: Matriz de covariância das distribuições (auxiliar);
12 ω_i: Parâmetro (auxiliar);
13 γ_i: Parâmetro ponderado (auxiliar);
14 ε: Taxa de aprendizado;
15 ν_i: Grau de liberdade ;
16 Σ_i: Matrix Escala;
   /* Criação das Estruturas e inicialização de dados                 */
17 Crie uma malha de i nodos: i = 1,2,...,H;
18 Crie partições dos dados com uma cardinalidade K para cada nodo i;
   /* Inicialização dos nodos                                         */
19 Calcule π_i através da Equação 3.27;
20 Calcule μ_i através da Equação 3.28 (Sugestão: K=10D);
21 Calcule C_i através da Equação 3.29;
22 Calcule m_i através da Equação 3.31;
23 Calcule M_i através da Equação 3.32;
24 Setar ν(0) com uma constante inicial. (Sugestão: 10^3);
25 Calcule Σ_i(0) através da Equação 3.30;
26 for até a convergência de ν do
       /* Sugestão: 1000 iterações ou variação de ν menor
          que 0.001                                                   */
27     switch tipo de ajuste do
28         case Peel
29             Calcule ω_i(0) através da Equação 3.33;
30         case Shoham
31             Calcule ω_i(0) através da Equação 3.34;
32         case Direct
33             Calcule ω_i(0) através da Equação 3.35;
34 Calcule γ_i(0) através da Equação 3.48;
35 switch tipo de ajuste do
36     case Peel
37         Calcule ν_i(n) através da Equação 3.36;
38     case Shoham
39         Calcule ν_i(n) através da Equação 3.37;
40     case Direct
41         Calcule ν_i(n) através da Equação 3.38;
```

Algoritmo 5: Pseudo-Algoritmo TSOM - Etapa Aprendizagem

```
   /* Definição de variáveis                                          */
 1 Δ: Raio de vizinhança;
 2 R_ni: Probabilidade a posteriori - Responsabilidade;
 3 Q = q_n1, ..., q_nh: Conjunto de distribuições;
 4 steps: Quantidade de iterações do algoritmo;
   /* Inicialização                                                   */
 5 Inicialize o Δ(0) com H_cols + H_rows/Const;
   /* Aprendizado                                                     */
 6 for n para steps do
 7     Selecione um padrão t_n;
 8     Atualize o valor do raio Δ(n) através da Equação 3.51 ;
 9     Calcule δ(i, j) através da Equação 3.52;
10     for todo nodo i do
11         Calcule p(t_n|i) através da Equação 3.58;
12     for todo nodo j do
13         for todo nodo i do
14             Calcule q_nji através da Equação 3.53;
15             Calcule p(t_n|q_nji) através da Equação 3.54;
16     Calcule o nodo ganhador através da Equação 3.55;
17     Calcule R_ni através da Equação 3.57;
18     Calcule ε através da Equação 3.56;
19     for todo nodo i do
20         Calcule π_i(n) através da Equação 3.40;
21         Calcule m_i(n) através da Equação 3.41;
22         Calcule μ_i(n) através da Equação 3.42;
23         Calcule M_i através da Equação 3.43;
24         Calcule C_i através da Equação 3.44;
25         switch tipo de ajuste do
26             case Peel
27                 Calcule ω_i(n) através da Equação 3.45;
28             case Shoham
29                 Calcule ω_i(n) através da Equação 3.46;
30             case Direct
31                 Calcule ω_i(n) através da Equação 3.47;
32         Calcule γ_i(n) através da Equação 3.48;
33         switch tipo de ajuste do
34             case Peel
35                 Calcule ν_i(n) através da Equação 3.36;
36             case Shoham
37                 Calcule ν_i(n) através da Equação 3.37;
38             case Direct
39                 Calcule ν_i(n) através da Equação 3.38;
40         Calcule Σ_i através da Equação 3.59;
```

4

Categorizador Não-supervisionado de Lugares

Neste capítulo serão apresentados os modelos propostos por esta dissertação baseados nos conceitos apresentados no Capítulo 3 para solucionar o problema de Categorização de Lugares através de Objetos expostos no Capítulo 2. Inicialmente será evidenciado o processo geral desse problema explicando o que cada etapa representa, além de explanar as contribuições dos novos modelos e como esta pesquisa atua para solucionar o problema.

4.1 Introdução

A separação ou segmentação de objetos ou características (componentes) isoladas podem não descrever por completo a representação de um lugar em ambiente confinado, um lugar interno. Se for possível encontrar componentes mais representativos do lugar, deve-se usá-los para caracterizar de uma forma compacta e eficaz os ambientes. A exemplo disso, pode-se considerar um cômodo. Ele pode ser imaginado como um espaço com quatro paredes no qual há componentes como janela e porta, que não são significativos, pois estes componentes podem estarem presentes na maioria dos tipos de cômodos. No caso da cozinha, representa-se essencialmente com forno, geladeira, panela, que dentre outros são significativos. Além disso, pode-se atribuir pesos distintos para diferentes componentes de um local (LI; MENG, 2012).

Os objetos em ambientes internos são discriminantes para a Categorização dos Lugares onde estes objetos estão inseridos, para isso a transformação de informação de imagem (matriz de *pixels*) para um vetor de frequência de objetos (histograma) é uma redução conveniente. Esse vetor é composto por informações de tipos e quantidades de objetos em uma dada imagem. Contudo, a incidência de tipos de objetos por imagem em geral é muito baixa e a variedade de objetos alta, por isso esses vetores de objetos caracterizam uma representação esparsa que, tipicamente, determina alta dimensionalidade dos dados.

Os novos modelos propõem uma redução na dimensionalidade de dados esparsos e um categorizador para a solução do problema abordado. Um diagrama de blocos do fluxo completo

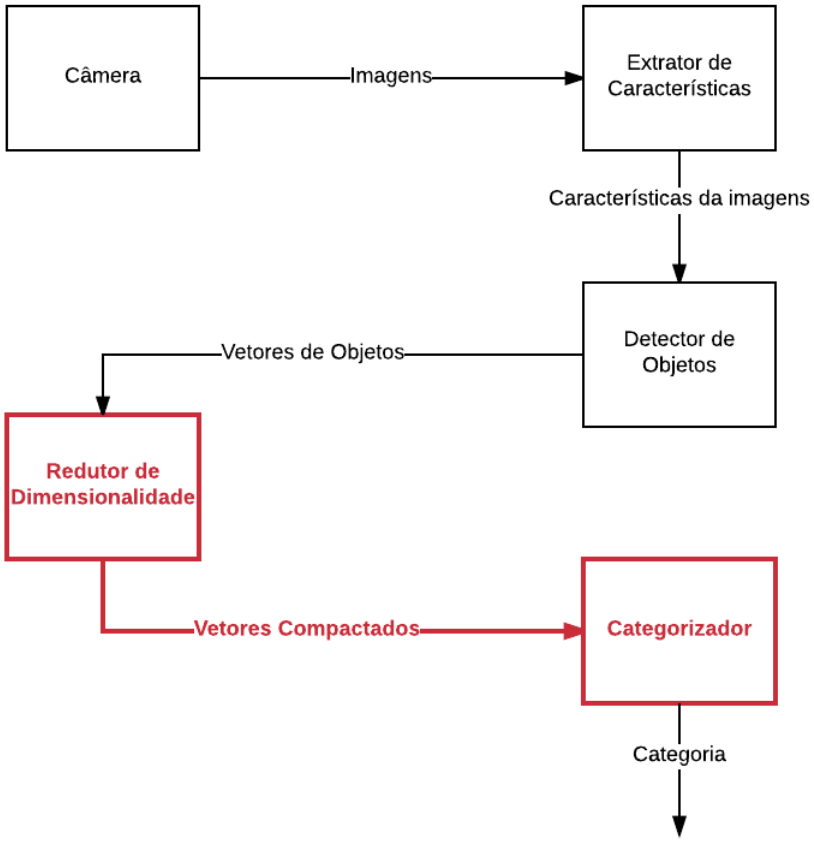

Figura 4.1: Diagrama de blocos do problema de Categorização de Lugares baseada em Objetos. O destaque em vermelho representa as contribuições deste trabalho.

do problema pode ser vista na Figura 4.1. As partes em vermelho são os módulos nos quais este trabalho possui contribuições.

A Figura 4.1 apresenta 5 módulos: câmera, extrator de características, detector de objetos, redutor de dimensionalidade e categorizador. O sensor de entrada é uma câmera que obtém as imagens. Essas imagens são as entradas para a etapa do extrator de características. Esse módulo irá aplicar algoritmos que obtenham pontos e descritores das imagens que sejam interessantes para a detecção de objetos e a saída desse módulo são essas características da imagem. Para a terceira etapa serão detectados os objetos existentes nas imagens através das características das imagens e dos modelos de objetos aprendidos por esta etapa (FELZENSZWALB; MCALLESTER; RAMANAN, 2008).

A quarta etapa não é obrigatória. A saída do terceiro módulo poderia ser conectada diretamente ao quinto módulo: o categorizador. Todavia, os resultados com o categorizador diretamente ligado a saída do detector de objetos não foram adequados, pois os dados obtidos através deste são esparsos. A quarta etapa, o redutor de dimensionalidade, tem como função melhorar e viabilizar os resultados do categorizador. Uma redução de dimensionalidade dos dados diminui o tempo de processamento para o categorizador, além de que se a taxa de esparsidade dos dados for minimizada o Categorizador terá maior eficácia. Em geral, o categorizador utiliza

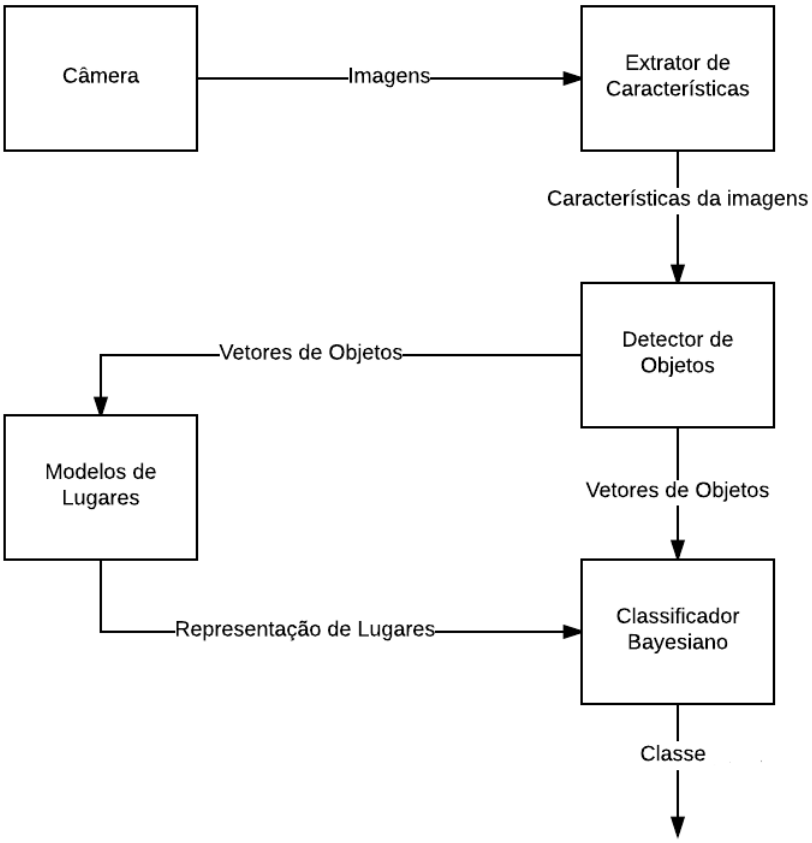

Figura 4.2: Diagrama de blocos do problema de Categorização de Lugares baseada em Objetos para o trabalho de Viswanathan.

algum tipo de distância vetorial em seu processo. As taxas de esparsidade dos dados serão apresentados no Capítulo 5 e o seu comparativo levou em consideração os métodos utilizados nessa pesquisa. Para finalizar o processo de Categorização de Lugares, o categorizador recebe como entrada os vetores compactados pelo redutor de dimensionalidade e é treinado com essas entradas. A saída do categorizador é a categoria cuja imagem está sendo processada.

O diagrama do processo proposto nesta pesquisa pode ser comparado com o diagrama do método de referência utilizando filtro bayesiano (VISWANATHAN et al., 2009, 2010). As 3 primeiras etapas são idênticas, se diferenciando somente nas duas últimas etapas. No método de referência são construídos modelos de lugares através do conjunto de dados de cada categoria e após isto é aplicado Filtro Bayesiano. Os detalhes desses processos foram explanados no Capítulo 2. No final do processo temos uma categoria para cada amostra avaliada.

4.2 Redutor de Dimensionalidade

Alguns problemas possuem uma representação com alta dimensionalidade e de dados esparsos. A alta dimensionalidade dos dados dificulta a comparação entre as amostras e protótipos, pois as métricas de distância são sensíveis a elas, reduzindo, portanto, o seu poder discriminante.

Esse poder está ligado diretamente ao conceito de distância entre os dois vetores. Conceitos como a proximidade ou distância entre dois vetores tornam-se menos significativos quando se aumenta a dimensionalidade dos dados (HINNEBURG; AGGARWAL; KEIM, 2000). BELLMAN (1960) argumenta que a distância relativa desde o ponto mais distante e mais próximo converge para zero à medida que aumenta a dimensionalidade dos dados, maldição da dimensionalidade, como mostrado pelo Teorema 1 (BEYER et al., 1999), no qual d é a dimensionalidade do vetor, X_d é um vetor em d dimensões, $Dmax_d$ é a máxima distância de um vetor à origem $(0,...,0)$, $Dmin_d$ é a máxima distância de um vetor à origem, $E[X]$ esperança matemática de X, $var[X]$ variância de X e $Y_d \rightarrow_p c$ a sequência de vetores $Y_1,...$ converge para um vetor constante c. Como consequência, a discriminação entre o vizinho mais próximo e o mais distante tende a torna-se bastante pobre no espaço de alta dimensão. O teorema mostra que esta situação independe da métrica de distância utilizada entre os vetores. Por esse motivo a redução de dimensionalidade torna-se visada por estarem presentes em vários problemas na literatura, como: detecção de fraude (HE; GRACO; YAO, 1998) e detecção de tumores em imagens médicas (KORN et al., 1998).

Teorema 1.

$$Se \lim_{d\to\infty} var\left(\frac{||X_d||}{E||X_d||}\right) = 0 \rightarrow, \ então \ \frac{Dmax_d - Dmin_d}{Dmin_d} \rightarrow_p 0.$$

Para a redução de dimensionalidade de dados pode-se lançar mão do conceito de compressão de sinais. Em processamento de sinais digitais, decimação é o processo de redução da taxa de amostragem de um sinal. Ela é a operação em complemento a interpolação, que aumenta a taxa de amostragem. Decimação utiliza filtragem para atenuar a distorção, que pode ocorrer quando os dados são apenas subamostrados diretamente. Decimação por um fator inteiro, M, pode ser explicada como um processo de duas etapas, com uma implementação equivalente que é mais eficiente (MILIC, 2009):

- Reduzir componentes de sinal de alta frequência com um filtro passa-baixa digital.
- Subamostragem do sinal filtrado por M, ou seja, manter apenas uma amostra a cada M.

A subamostragem como operação única faz com que componentes de sinal de alta frequência possam ser mal interpretados, que é uma forma de distorção chamada *aliasing*. Então o primeiro passo, se necessário, é para suprimir o serrilhado do sinal para um nível aceitável. O filtro é chamado um filtro *anti-aliasing* (MILIC, 2009). O filtro passa-baixa age como um integrador calculando a média para a região do sinal aplicada ao filtro. A etapa de subamostragem reduz o tamanho do sinal. Esses conceitos serão utilizados para a proposta de arquitetura de Agrupamento Fixo. A Figura 4.3 apresenta um sinal submetido ao processo de decimação, após o processo a quantidade de dados é reduzida de 120 para 30, porém a distribuição da função continua similar a original (MILIC, 2009).

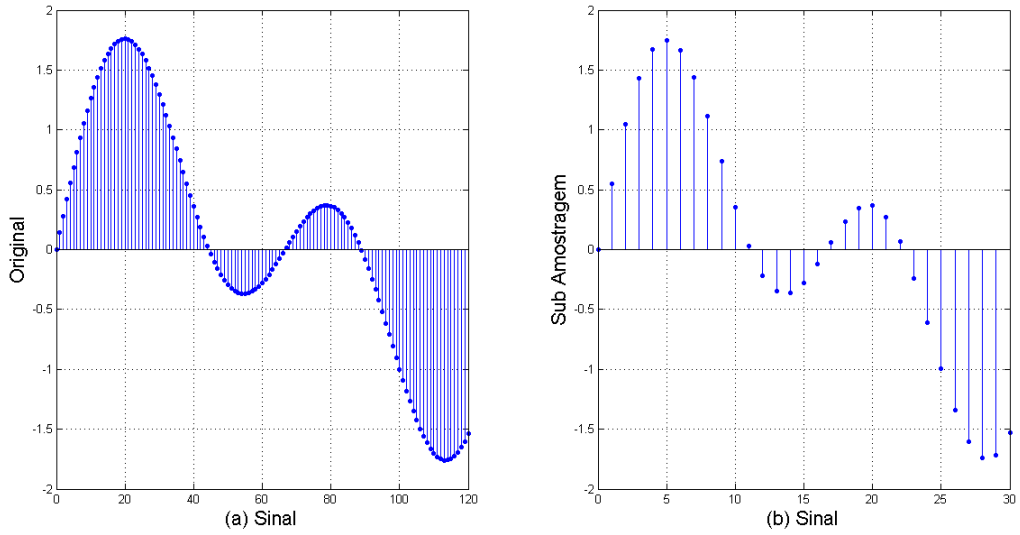

Figura 4.3: Os gráficos foram gerados a partir do sinal discreto $x = sin(2\pi 30t) + sin(2\pi 60t)$. (a) Sinal x original. (b) Sinal x com Decimação aplicada.

4.3 Categorizador Não-supervisionado

A categorização é feita utilizando-se o método PSOM (LÓPEZ-RUBIO, 2009), porém, para aumentar a taxa de acerto de categorização pode se utilizar um PSOM para cada categoria, de tamanho pequeno em vez de um único PSOM grande, isso maximiza a probabilidade $p(\mathbf{t}_n)$. Então o processo consiste em treinar vários PSOMs, um para cada categoria da base de dados, e, em seguida, as amostras de teste serem atribuídas a categoria correspondente ao PSOM através da apresentação a cada um desta redes. Para esse processo podem ser definidas algumas métricas para avaliação de categorizadores. A métrica utilizada nesta dissertação para categorização é definida pela Equação 4.1, na qual N_C é a quantidade de categorias, que também é o mesmo número de PSOMs, N é o número total de amostras de entrada e N_{ij} é número de amostras atribuídas a categoria i que pertencem à classe de j. O valor do *score* está no intervalo [0, 1] (MOSCHOU; VERVERIDIS; KOTROPOULOS, 2007).

$$Score = \frac{\sum_{i=1}^{N_c} N_{ii}}{N} \tag{4.1}$$

$$N = \sum_{i=1}^{N_c} \sum_{j=1}^{N_c} N_{ij} \tag{4.2}$$

O processo de avaliação pelo categorização se inicia separando os conjuntos de dados em dados de treinamento e dados de testes. Os dados de testes são agrupados por categoria e cada PSOM é treinada com os dados pertencentes a sua categoria. Após a fase de treinamento é realizada a fase de testes, na qual todos os dados de testes são apresentados a todas da instâncias de PSOM. O nodo vencedor de cada PSOM para cada amostra é calculado. Finalmente, a

categoria de cada amostra é escolhida pela maior resposta dentre os nodos vencedores de cada PSOM através da Equação 4.1 no qual o maior *score* das instâncias determina a categoria da amostra.

Nesta pesquisa utilizamos dois modelos PSOM para instâncias de múltiplas redes do categorizador: PbSOM e TSOM. Como esses dois modelos possuem três variações de algoritmos de aprendizagem cada um, então foi escolhida uma dessas variações para de processo do aprendizado para cada um dos dois modelos a partir dos resultados descritos na literatura referentes a cada modelo (CHENG; FU; WANG, 2009; LÓPEZ-RUBIO, 2009). Para o PbSOM foi utilizada a variação SODAEM e para o TSOM a variação *Shoham*.

4.4 Arquiteturas

Este trabalho propõem três formas para redução de dimensionalidade dos dados esparsos. A primeira baseia-se em um Agrupamento Fixo de variáveis, a segunda é um SOM Raso, somente uma camada, e na terceira serão utilizados SOMs em múltiplas camadas (arquitetura profunda) para transformação dos atributos (originalmente objetos existentes na imagem) numa codificação esparsa para um domínio de variáveis de dimensão e percentual de dados esparsos reduzidos.

4.4.1 Arquitetura de Agrupamento Fixo

Para reduzir a dimensionalidade dos dados a primeira proposta é agrupar os atributos de uma forma cega formando agrupamentos de atributos de tamanho fixo. Primeiramente define-se a dimensão c da saída dos dados. O algoritmo de Agrupamento Fixo foi inspirado no processo de decimação para processamento de sinais. Para a formação do agrupamento a operação de somatório substitui a de média, da etapa filto passa-baixa. Como os dados serão normalizados e o tamanho da janelas de somatório é constante, então esse processo é análogo ao filtro passa-baixa. Para a etapa de subamostragem o processo é realizado de uma forma que não haja intersecção entra valores subamostrados, o deslize da janela de subamostragem possui a mesma dimensão que a janela.

O algoritmo do Agrupamento Fixo está descrito no Algoritmo 6. Os atributos de saída serão a soma dos atributos de entrada e grupos de tamanho fixo. Inicialmente define-se o parâmetro c. Divide-se a quantidade total de atributos de base original por c, desta maneira obtêm-se o tamanho N de cada agrupamento. Os atributos de saída são gerados a partir dos agrupamentos sem sobreposição com o tamanho N seguindo a ordem dispostas inicialmente pelos atributos. Por exemplo, se a base de dados original tem 1000 atributos e a quantidade de atributos de saída for 50. Os atributos de saída serão a soma dos intervalos de 20 em 20 atributos da base original (1000 atributos originais/50 agrupamentos = 20 atributos por agrupamento), Algoritmo 6 Linha 7. Os atributos das bases de dados originais são objetos e estão ordenados pela ordem alfabética, então existe um relação tênue de semântica e localidade nos atributos, como

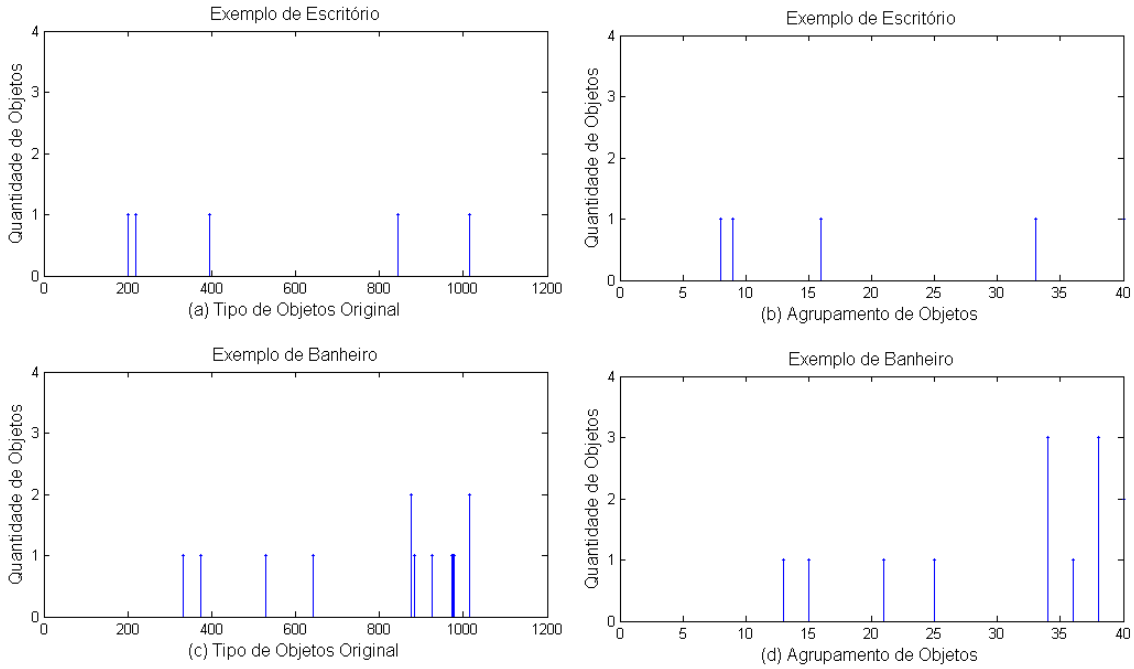

Figura 4.4: Exemplo de histogramas de objetos para dados originais (1069 atributos) e Agrupamento Fixo (40 atributos) para base LabelMe4: (a) Original para um Escritório; (b) Agrupamento Fixo para um Escritório; (c) Original para um Banheiro; (d) Agrupamento Fixo para um Banheiro.

por exemplo: jarro, jarro ocluso, jarro quebrado. A Figura 4.4 apresenta exemplos de redução de variáveis para amostras da base LabelMe4. Já a Figura 4.5 mostra a média das amostras de cada categoria da base de dados LabelMe4 sendo tanto para os dados de originais quanto para os dados reduzidos pelo Agrupamento Fixo.

Algoritmo 6: Pseudo-Algoritmo Agrupamento Fixo

1 **X**: vetores de entrada;
2 M: quantidade atributos na entrada;
3 N: quantidade atributos a serem agrupados;
4 c: quantidade atributos na saída (grupos);
5 N = M/c;
6 **for** c **do**
7 some o valor do atributo $X_{[1+c.(N-1)]}$ até $X_{[c.N]}$ e crie o nome atributo de saída;

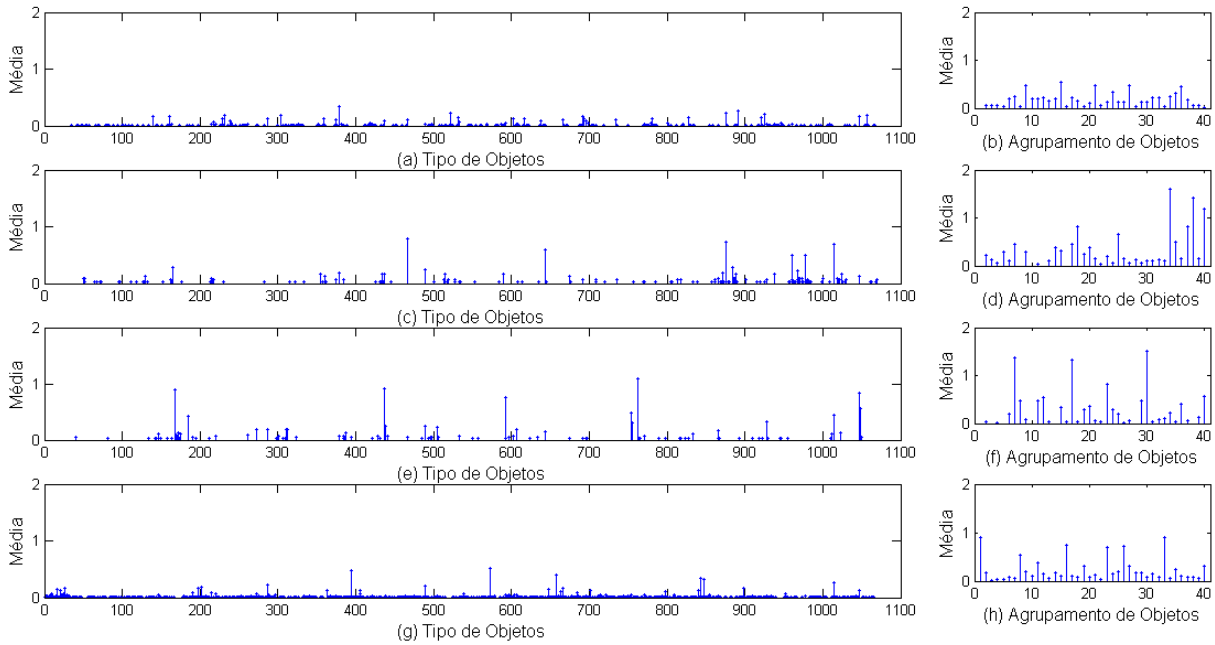

Figura 4.5: Histogramas da média dos atributos por categoria para LabelMe4 para dados originais (1069 atributos) e Agrupamento Fixo (40 atributos): (a) Categoria Cozinha (dados originais); (b) Categoria Cozinha (Agrupamento Fixo); (c) Categoria Banheiro (dados originais); (d) Categoria Banheiro (Agrupamento Fixo); (e) Categoria Quarto (dados originais); (f) Categoria Quarto (Agrupamento Fixo); (g) Categoria Escritório (dados originais); (h) Categoria Escritório (Agrupamento Fixo).

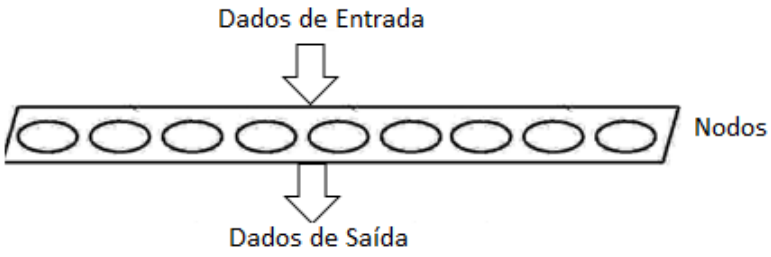

Figura 4.6: Diagrama do modelo SOM Raso.

4.4.2 Arquitetura de SOM Raso

O SOM original tem sido muito utilizado para agrupamento de dados e redução de dimensionalidade (ASTUDILLO; OOMMEN, 2014). A redução de dimensionalidade deSOM padrão é restrita à forma de arranjo dos nodos do próprio SOM. Por exemplo: o SOM em forma bidimensional realiza uma redução de dimensionalidade para duas dimensões. Este trabalho propõe uma modificação no SOM para se obter uma redução de dimensionalidade de tamanho flexível. Nossa proposta, chamada de SOM Raso, o tamanho final da representação das amostras será a quantidade de nodos existentes na malha unidimensional.

O algoritmo proposto é baseado no SOM padrão, Algoritmo 1, com algumas modificações, apresentados no Algoritmo 7. Os pesos são inicializados com valores próximos à zero, no intervalo [-0,01 +0,01]. A malha de nodos é definida com o formato unidimensional, como na Figura 4.6, e de tamanho m, número de nodos, na qual também será a dimensão da representação dos dados. Antes da etapa de treinamento os dados são normalizados por amostra. O treinamento do SOM é executado para n épocas e para todas as amostras, linha 10 a 15 do Algoritmo 7.

A função de vizinhança é um progressão geométrica decrescente, dada pela Equação 4.3, na qual τ é o fator de decaimento da função geométrica, $d_{j,v}$ é a distância vetorial entre nodo em questão e nodo vencedor, e w é o tamanho máximo da vizinhança. A função de decaimento da taxa de aprendizagem é dada pela Equação 4.4, na qual $\alpha(0)$ é a taxa inicial e β é o decaimento da taxa de aprendizagem.

$$h_{j,v}(n) = \frac{1}{\tau^{d_{j,v}}}, se - w < d_{j,v} < w \tag{4.3}$$

$$\alpha_i(n) = \alpha_i(0).\beta_i^n \tag{4.4}$$

A nova representação dos dados é gerada através da ativação dos nodos pela apresentação de cada amostra, linha 16 do Algoritmo 7. Como os dados são esparsos, tem-se uma ativação intermediária sempre presente que não corresponde a uma ativação representativa entre os vetores, então foi proposto uma etapa de filtragem que é descrita pela Equação 4.5 para o ponto de corte. O limiar de ponto de corte é determinado pela Equação 4.6, no qual μ é a média das ativações para a amostra, σ é o desvio padrão das ativações e k é o parâmetro de ganho do desvio

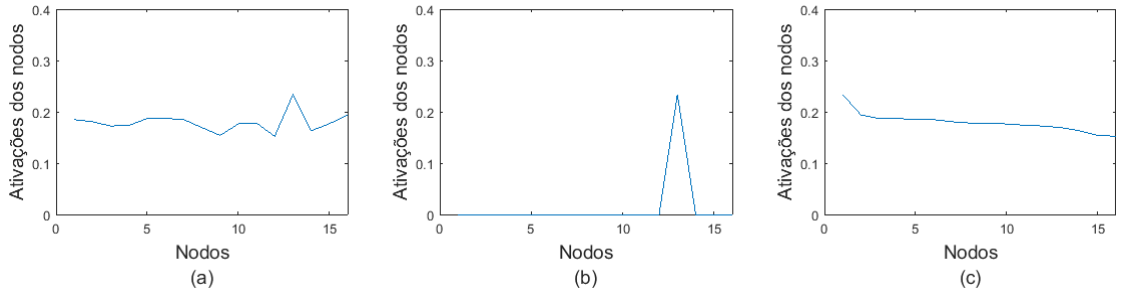

Figura 4.7: Ativações de uma amostra em um camada de 16 nodos: (a) Ativações; (b) Ativações com filtradas com o ponto de corte de k = 1; (b) Ativações ordenadas (para visualização).

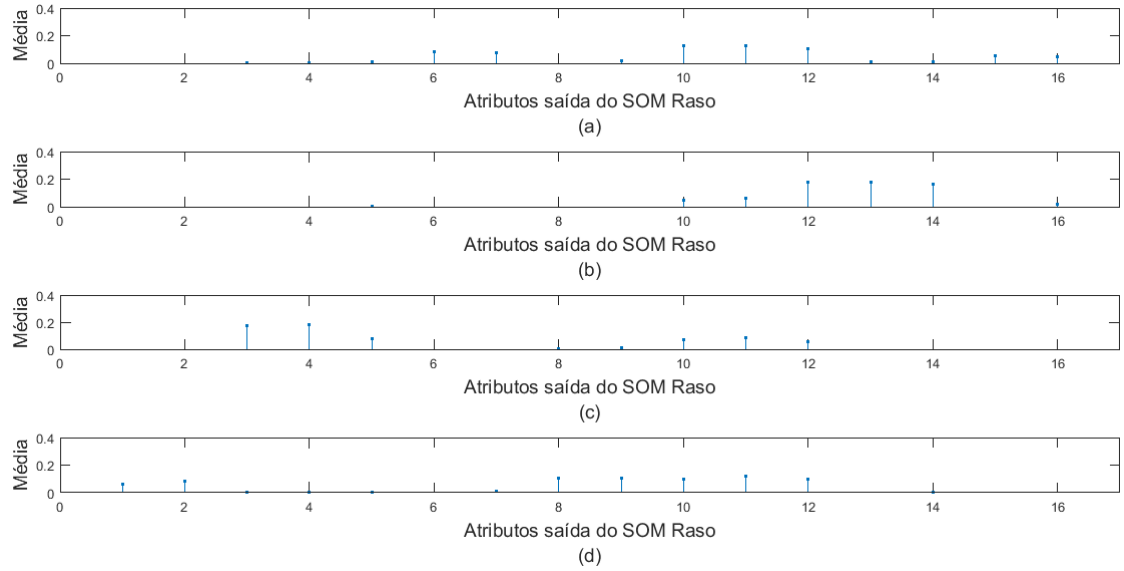

Figura 4.8: Histogramas da média dos atributos por categoria para LabelMe4 (o tamanho da saída são 16 atributos): (a) Saída da categoria Cozinha; (b) Saída da categoria Cozinha; (c) Saída da categoria Banheiro; (d) Saída da categoria Banheiro.

padrão, na linha 17 do Algoritmo 7. As ativações filtradas serão a nova representação da amostra. Exemplos das ativações geradas para nova representação dos dados é apresentada na Figura 4.7. A Figura 4.8 apresenta a média das saída para cada uma das categorias para base de dados LabelMe4 demonstrando como ocorrem as distribuições médias para cada categoria.

$$\mathbf{x}' = \begin{cases} se\ exp^{-\|\mathbf{x}-\mathbf{w}_i\|} >= limiar,\ exp^{-\|\mathbf{x}-\mathbf{w}_i\|} \\ se\ exp^{-\|\mathbf{x}-\mathbf{w}_i\|} < limiar,\ zero \end{cases} \tag{4.5}$$

$$limiar_x = \mu(exp^{-\|\mathbf{x}-\mathbf{w}_i\|}) + k.\sigma(exp^{-\|\mathbf{x}-\mathbf{w}_i\|}) \tag{4.6}$$

Algoritmo 7: Pseudo-Algoritmo SOM Raso

X: dados de entrada;
m: quantidade de nodos na camada;
n: épocas de treinamento;
α: taxa de aprendizado inicial;
β: fator de decaimento da aprendizagem;
τ: fator da função de vizinhança (progressão geométrica);
w: tamanho da janela máxima de ativação de nodos;
k: fator k para limiar de ativações de nodos;
Normalização de **X** por amostra;
Execute o Algoritmo 1:;
- Para todas as amostras;
- Para *m* nodos;
- Para *n* épocas;
- Para a função de vizinhança, progressão geométrica, através da Equação 4.3 com τ e *w*;
- Para a função de taxa de aprendizado através da Equação 4.4 com α e β;
Calcule as ativações de todos os nodos para cada amostra;
Filtre todas as amostras através da Equação 4.5 utilizando um *k*;
As ativações filtradas são a nova representação dos dados.

4.4.3 Arquitetura de SOM Profundo Compartimentado

Este trabalho propõe uma arquitetura profunda baseada em conceitos de SOM. Na concepção original o SOM possui somente uma camada. Este novo modelo é uma rede de múltiplas camadas SOM. Existem alguns modelos na literatura que utilizam modelos múltiplas camadas de SOM como: Growing Hierarchical Self-Organizing Map (GHSOM) (DITTENBACH; MERKL; RAUBER, 2000), Robust Growing Hierarchical Self-Organizing Map (RGHSOM) (MORENO et al., 2005) e Short Term Self Organizing Map (STSOM) (CARPINTEIRO, 1999).

O GHSOM utilizada várias camadas de pequenas SOMs. A primeira camada passa pelo treinamento e os neurônios são adicionados para que o erro de quantização seja reduzido. A próxima camada do GHSOM é feito com a quantidade de SOMs igual ao número de nodos da primeira camada. Os SOMs da segunda camada serão treinados somente com as amostras referentes que são representadas com o nodo da primeira camada correspondente. Esse processo é realizado iterativamente para quantas camadas se queira. Desta maneira, a primeira camada cria agrupamentos maiores e as camadas posteriores fazem constroem as agrupamentos mais refinados. Muitos métodos de agrupamento tem uma sensibilidade a dados ruidosos. Então a ideia do GHSOM foi ampliada para o RGHSOM no qual propõem uma robustez à dados ruidosos. O RGHSOM se utiliza de uma adaptação no cálculo na distância vetorial entre os nodos e a amostra através de um divisão pela variância. A proposta do RGHSOM faz com que sejam necessário menos camadas e neurônios para a representação dos dados.

A arquitetura STSOM foi proposta inicialmente para processar composições musicais. Ela é composta por dois níveis de camadas SOMs. Antes de cada camada SOM temos um etapa de integração das dados no espaço temporal, este módulo faz um integração entre as entrada já apresentadas no passado. A nova representação enviada para a segunda bloco do processo é dada pela distância entre o nodo vencedor e cada um dos outros nodos. O modelo doSOM Profundo Compartimentado tem diferenças entre esses modelos apresentados, pois a cada camada é feita um redução do tamanho das camadas, além do que forma da geração da ativação dos neurônios que será a nova representação ser diferente e será apresentada adiante. O GHSOM e o RGHSOM não fazem nenhuma modificação na representação dos dados somente agrupam dos dados em multiníveis.

Como os dados esparsos são difíceis de serem comparados entre si, o treinamento será feito em múltiplas camadas, construindo-os gradativamente. Essa ideia é inspirada no conceito de Aprendizagem Profunda de treinar um pouco cada camada de forma gradativa. Inicialmente temos uma pilha de camadas SOM, a fase de único *pipeline*, a primeira camada, única, tem como objetivo organizar uma topologia nas amostras dos dados de entrada, pode-se visualizar exemplos de amostra na Figura 4.10 das ativações das camadas para uma determinada amostra. A Figura 4.11 apresenta a média das amostras para cada categoria da base LabelMe4.

A arquitetura do SOM Profundo Compartimentado possui duas fase: único *pipeline* e múltiplos *pipelines*. O algoritmo do SOM Profundo Compartimentado é apresentado no

Algoritmo 8. A primeira fase é composta por camadas SOMs únicas empilhadas, na qual todas as amostras serão treinadas, passando a informação camada após camada. Cada uma dessas camadas é treinada isoladamente como uma camada SOM com Algoritmo 1 com os seus devidos ajustes e parametrizações. As camadas SOMs empilhadas da primeira fase devem possuir uma quantidade de alta de nodos em uma malha unidimensional e a medida que as camadas se aprofundam a quantidade de nodos nelas diminuem realizando uma redução de dimensionalidade nas saída das redes e criando topologias nas distribuições de saída.

As entradas iniciais são normalizadas por amostra para o formato das distribuições iniciais de cada amostra seja conservado, Algoritmo 8 na Linha 11. Não é realizada normalização na entrada de nenhuma outra camada, além da primeira camada, pois as saídas das camadas já estão dentro do intervalo de operação e esse processo não se mostrou efetivo em nenhuma camada após a inicial. A literatura sugere a ideia de não ser necessário normalizar as ativações entre as camadas para arquiteturas profundas (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). Cada camada SOM da primeira fase pode ser parametrizada com uma quantidade m_i de nodos, sendo treinada por n_i épocas, os nodos são inicializados com valores no intervalo [-0.01 +0.01]. Para o treinamento é utilizada uma função de vizinhança em progressão geométrica decrescente através da Equação 4.3 utilizando o fator τ_i em uma janela de atuação w_i, além de um taxa de aprendizado calculada a partir da Equação 4.4, com a taxa de aprendizagem inicial $\alpha_i(o)$ e β_i que é o fator de decaimento da taxa de aprendizagem. Esse passos são demonstrados no Algoritmo 8 da Linha 13 à Linha 18.

Esse treinamento das camadas do único *pipeline* tem como objetivo criar uma distribuição representativa das amostras da mesma categoria ou de agrupamentos de amostras representantes de uma categoria. Assim, criando uma topologia de agrupamento que responda as amostras que tem uma determinada semelhança na mesma região da camada. A saída para a camada posterior é obtida através da ativação de todos os nodos para cada amostra. Após o cálculo de todas as ativações de todos os nodos para todas as amostras é realizado um filtro executando um ponto de corte para selecionar os nodos que tenham um resposta significativa dos que não têm. A aplicação do filtro é necessária porque como os dados são esparsos e a inicialização dos pesos dos nodos estão no intervalo [-0.01 +0.01], então muitos nodos não são representativos. Para uma determinada amostra são obtidas muitas ativações nas quais não existem praticamente casamento entre a amostra e os nodos, sendo que as ativações representativas para categoria da amostra ficam levemente acima do patamar das ativações não significativas. Este fato é explicado pelo Teorema 1, quando a Dimensionalidade é muito alta a significância entre as métricas de distâncias entre vetores tende a ser ínfima, ou seja, esse fato ocorre principalmente nas saídas das camadas com mais nodos.

Dessa maneira, um ponto de corte é executado sobre as ativações, para que as ativações não representativas não sejam passadas para as próximas camadas do modelo. Um exemplo da aplicação do filtro para uma ativação de uma amostra em uma camada grande é demonstrado na Figura 4.9, na qual somente as ativações acima do limiar são válidas. O ponto de corte do limiar

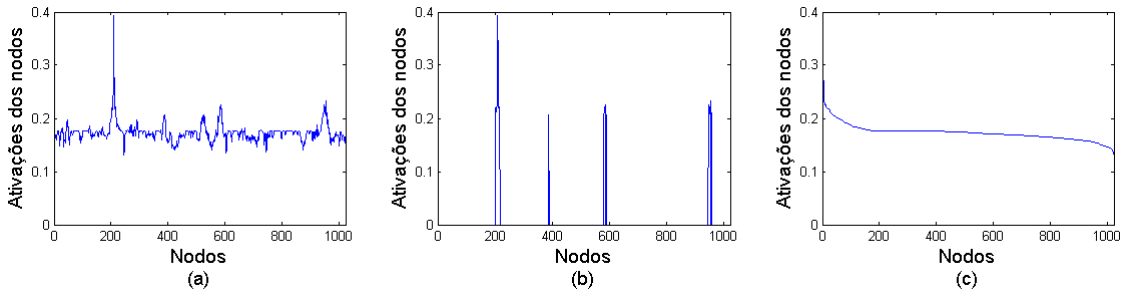

Figura 4.9: Ativações de uma amostra em um camada de 1024 nodos: (a) Ativações; (b) Ativações com filtradas com o ponto de corte de k = 2; (b) Ativações ordenadas (para visualização).

é a média mais o desvio padrão multiplicado por um fator k_i. Os limiares para cada amostra são determinadas pela Equação 4.6 e o filtro de ativações é aplicado pela Equação 4.5. Essa etapa de filtragem de ativações está no Algoritmo 8 na Linha 19 e Linha 20. A Figura 4.10 apresenta a saída da primeira camada, única, com 1024 nodos para amostras da base LabelMe4 apresentada no Capítulo 5. As funções de ativação filtradas para cada amostra construirão a nova representação da amostra para a próxima camada.

A segunda fase do algoritmo é composta de múltiplos *pipelines* de camadas SOMs empilhadas, uma pilha para cada categoria. Cada *pipeline* é treinado somente com as amostras correspondentes a categoria, baseado em uma arquitetura análoga à Sistemas de Múltiplos Classificadores (WOŹNIAK; GRAÑA; CORCHADO, 2014), desta forma cada classificador é especialista em uma determinada classe, no casa cada *pipeline* se torna especialista em uma categoria. A composição da respostas desses classificadores determina a representação final dos dados. Assim cada *pipeline* induz a relação daquela amostras com o tipo de categoria treinada por ela. As camadas são treinadas de forma semelhante as camadas da primeira etapa, sendo que a Função de Vizinhança, neste caso é uma gaussiana, Equação 4.7, na qual $v(x)$ é o nodo vencedor $d_{j,v}$ é a distância entre o nodo j e $d_{j,v}$ nodo vencedor e σ é um parâmetro para a exponencial. Esse treinamento está descrito no Algoritmo 8 da Linha 30 à Linha 35.

$$h_{j,v(x)}(n) = \exp^{\left[-d_{j,v}/2\sigma^2\right]} \tag{4.7}$$

Ao final do treinamento são calculadas as ativações de todos os nodos para todas as amostas (não somente as amostras correspondentes ao *pipeline* em questão). As ativações entre as camadas do *pipeline* são filtradas obedecendo o ponto de corte da Equação 4.5. Assim constrói-se os dados de entrada pra camada posterior. As camadas diminuem de tamanho a medida que vão se aprofundando e são bem menores que as camadas da fase com SOMs únicos. O objetivo desta fase é que cada *pipeline* se especialize na resposta de uma saída para cada categoria. As amostras são filtradas da mesma forma que as camadas da primeira fase. O Algoritmo 8 na Linha 36 e 37 representam esse processo. O processo de filtro pode ser optativo quando a camada tem uma dimensão baixa, pois nesse caso não ocorre o problema de ativações de neurônios não

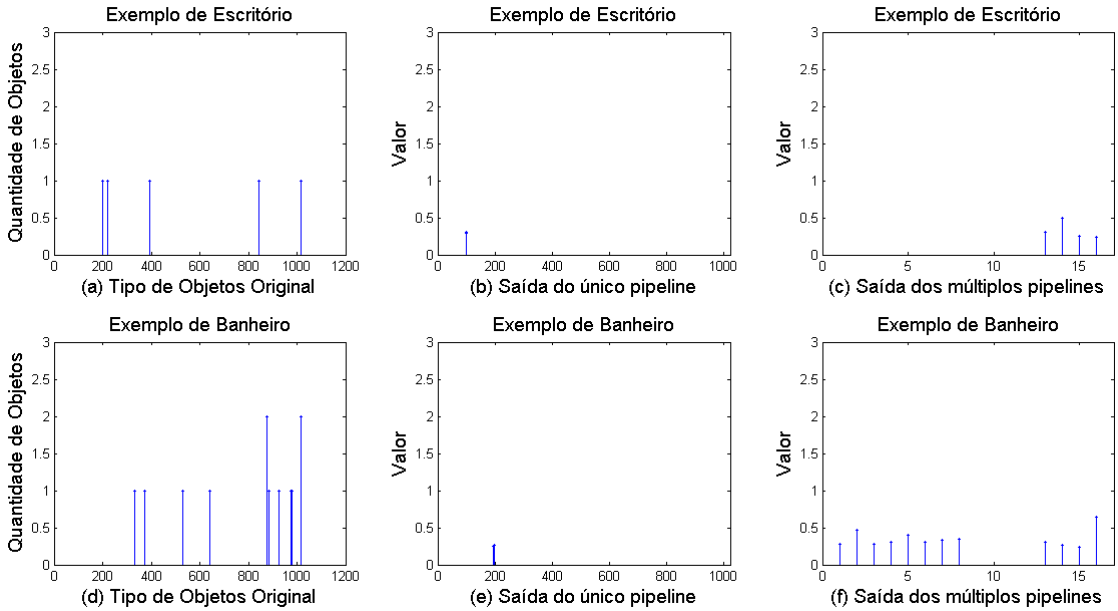

Figura 4.10: Histogramas de exemplos dos atributos para dados originais e SOM Profundo Compartimentado para base LabelMe4: (a) Dado original para um Escritório (1069 atributos); (b) Saída do único *pipeline* para um Escritório (1024 atributos); (c) Saída dos múltiplos *pipelines* para um Escritório (16 atributos); (d) Dado original para um Banheiro (1069 atributos); (e) Saída do único *pipeline* para um Banheiro (1024 atributos); (f) Saídas concatenadas dos múltiplos *pipelines* para um Banheiro (16 atributos).

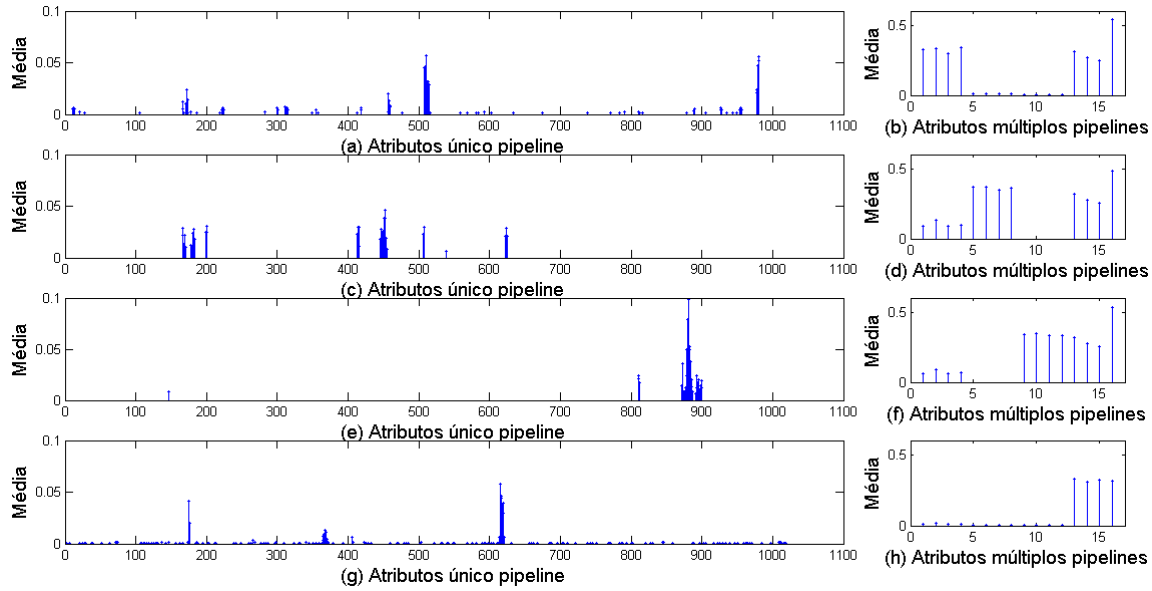

Figura 4.11: Histogramas da média dos atributos por categoria para LabelMe4 (o tamanho da saída do único *pipeline* são 1024 atributos e o da saída de múltiplos *pipelines* são 16 atributos): (a) Saída da categoria Cozinha do único *pipeline*; (b) Saída da categoria Cozinha múltiplos *pipelines*; (c) Saída da categoria Banheiro do único *pipeline*; (d) Saída da categoria Banheiro múltiplos *pipelines*; (e) Saída da categoria Quarto do único *pipeline*; (f) Saída da categoria Quarto múltiplos *pipelines*; (g) Saída da categoria Escritório do único *pipeline*; (h) Saída da categoria Escritório múltiplos *pipelines*.

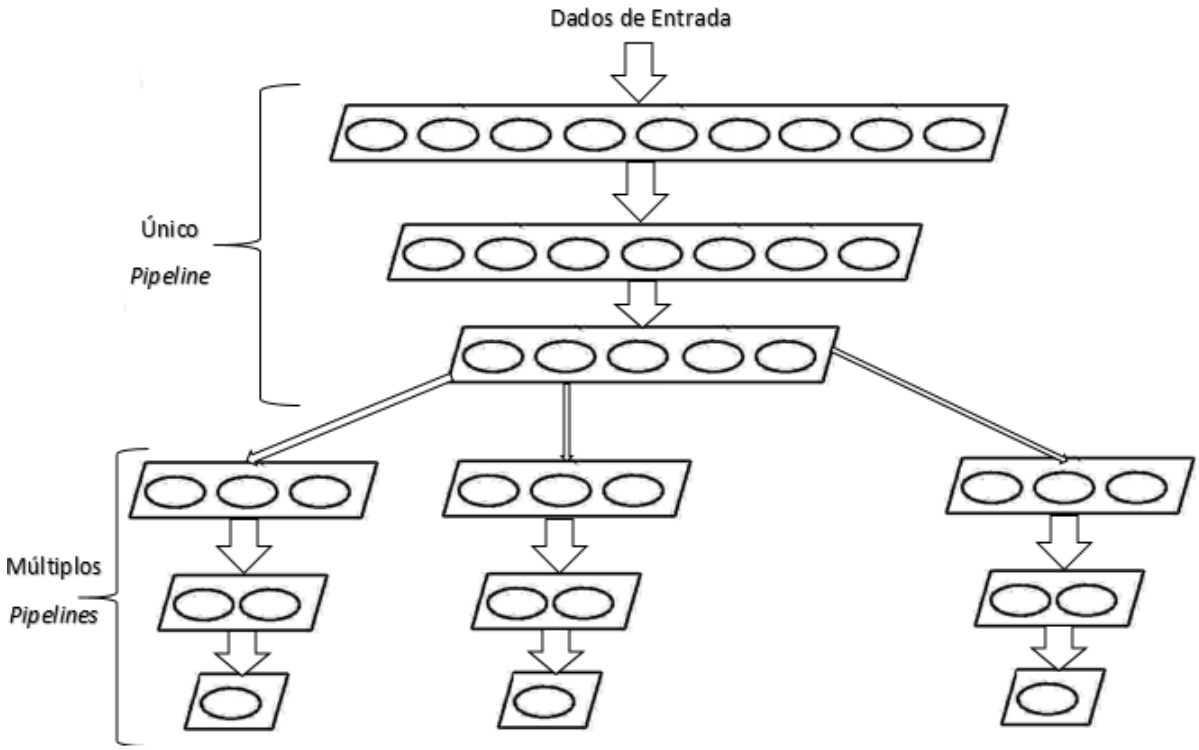

Figura 4.12: Diagrama do modelo SOM Profundo Compartimentado.

representativos.

Na saída de cada *pipeline* é gerada uma quantidade de ativações q que serão a representação de cada amostra para aquele *pipeline*. Esse processo é realizado para todos os *pipelines*. No final é realizada uma concatenação das saídas de cada *pipeline* para cada uma das amostras montando uma representação, na qual o tamanho da representação de cada amostra é igual a $q.N$ (Algoritmo 8 na Linha 39). Essa nova representação das amostras será a entrada do Categorizador. A Figura 4.12 mostra a estrutura para o SOM Profundo Compartimentado: único *pipeline* (primeira fase) e múltiplos *pipelines* (segunda fase). A Figura 4.10 apresenta a redução de atributos para exemplos de amostras utilizando SOM Profundo Compartimentado e a Figura 4.11 apresenta a média para as amostras de cada categoria na nova representação de dados.

Neste capítulo foram descritos os modelos propostos por este trabalho, além da justificativa para sua elaboração. No próximo capítulo serão apresentados os experimentos e comparação entre os modelos propostos e a literatura.

Algoritmo 8: Pseudo-Algoritmo SOM Profundo Compartimentado

X: dados de entrada;
c: quantidade de categorias;
/* Fase único *pipeline* */
l_i: quantidade de camadas SOM únicas;
m_i: quantidade de nodos na camada;
n_i: épocas de treinamento;
α_i: taxa de aprendizado inicial;
β_i: fator de decaimento da aprendizagem;
τ_i: fator da função de vizinhança (progressão geométrica);
w_i: tamanho da janela máxima de ativação de nodos;
k_i: fator k para limiar de ativações de nodos;
Normalização de **X** por amostra;
for *cada i camada única* l_i **do**
    Execute o Algoritmo 1:;
    - Para todas as amostras;
    - Para m_i nodos;
    - Para n_i épocas;
    - Para a função de vizinhança, progressão geométrica, através da Equação 4.3 com τ_i e w_i;
    - Para a função de taxa de aprendizado através da Equação 4.4 com α_i e β_i;
    Calcule as ativações de todos os nodos para cada amostras;
    Filtre todas as amostras através da Equação 4.5 utilizando um k_{ij};
    Atribua as funções filtradas entrada pra a próxima camada;
/* Fase múltiplos *pipelines* */
l_{ij}: quantidade de camadas SOM múltiplas;
d_i: quantidade de nodos na camada;
n_i: épocas de treinamento;
α_i: taxa de aprendizado inicial;
β_i: fator de decaimento da aprendizagem;
k_{ij}: fator k para limiar de ativações de nodos;
for *cada i categoria* l_i **do**
    for *cada j camada camada múltipla* l_{ij} **do**
        Execute o Algoritmo 1:;
        - Para as amostras da categoria específica do *pipeline*;
        - Para m_i nodos;
        - Para n_i épocas;
        - Para a função de vizinhança, Função Gaussiana, através da Equação 4.7;
        - Para a função de taxa de aprendizado através da Equação 4.4 com α_{ij} e β_i;
        Calcule as ativações de todos os nodos para cada amostras;
        Filtre todas as amostras através da Equação 4.5 utilizando um $k_{i,j+1}$;
        Atribua as funções filtradas entrada pra a próxima camada;
Concatene a saída de cada *pipeline* de SOM para cada amostra;

5

Experimentos

Neste capítulo serão descritos como foram realizados os experimentos, as bases de dados utilizadas, os resultados dos métodos e comentários sobre estes. Será apresentado como foram organizadas as bases de dados, quais são as informações que elas apresentam e a metodologia dos experimentos.

5.1 Bases de Dados

Para se fazer a avaliação sobre a Categorização de Lugares baseada em Objetos emprega-se algumas bases de dados para efeito de comparação. Assim se faz necessário se realizar avaliações com vários conjuntos de dados realistas para o desenvolvimento e avaliação dos algoritmos elaborados. Os experimentos foram realizados considerando-se 4 bases de dados denominadas: LabelMe4, LabelMe8, RIS62 e SUN407. As bases de dados originais contém, por cada amostra (imagem do lugar), informações de segmentação dos objetos na imagem, contagem dos objetos e nome dos objetos. Os dados de entrada dos experimentos se restringem a anotações das ocorrência dos objetos e não são utilizadas as imagens dos lugares propriamente ditas. Assim, a representação de cada lugar é a quantidade de objetos existentes por amostra.

As duas primeiras bases de experimentos foram montadas a partir da base de dados LabelMe (RUSSELL et al., 2008). A base de dados LabelMe4 é a base utilizada trabalho de Viswanathan (VISWANATHAN et al., 2009, 2010), na qual há 4 categorias (cozinha, banheiro, quarto e escritório) como cômodos. A base LabelMe8 tambem foi montada a partir da base LabelMe e possui 8 categorias (cozinha, banheiro, quarto, escritório, sala de reunião, corredor, sala de jantar e sala de estar), lembra-se que não foram utilizados conjuntos relativos a imagens omnidirecionais.

A base de dados RIS62 foi obtida da base de dados RIS (QUATTONI; TORRALBA, 2009). Essa base contém originalmente 15.620 amostras, contudo, apenas 2.738 delas possuem anotações dos objetos. Das 67 categorias, 5 delas possuem menos de 5 exemplos, então, foram descartadas, tendo assim uma base de dados com 62 categorias consideradas na RIS62. A quarta base de dados, SUN407, está baseada nos dados SUN (XIAO et al., 2010). Essa base de dados

Figura 5.1: Exemplos de amostras (imagens) com as regiões dos objetos segmentados para cada uma das categorias da LabelMe8. (a) Cozinha (17 objetos); (b) Banheiro (23 objetos); (c) Quarto (17 objetos); (d) Escritório (16 objetos); (e) Sala de Conferência (33 objetos); (f) Corredor (15 objetos); (g) Sala de Jantar (4 objetos); (h) Sala de Estar (14 objetos).

possui no total 131.072 imagens em 908 categorias, porém, somente 407 categorias possuem mais do que 10 amostras com anotações, totalizando um total de 17.567 imagens. Os detalhes das bases de dados estão contidos na Tabela 5.1.

5.1.1 Apresentação dos Dados

As bases de dados originais completas possuem todas mesma formatação, definida inicialmente pela base LabelMe, com respeito as imagens e anotações. Exemplos da base de dados LabelMe8 podem ser visualizados na Figura 5.1, uma amostra para cada categoria da base. Cada imagem tem os objetos segmentados em destaque. A Figura 5.2 indica os histogramas de objetos. Cada histograma corresponde a imagem da Figura 5.1. Ao todo, a base LabelMe8 possui 1380 tipos de objetos. O histograma indica a quantidade de objetos do mesmo tipo na mesma imagem.

5.1.2 Análise das Bases de Dados

As bases de dados possuem diferentes níveis de complexidade: quantidade de categorias e a relação quantidade de objetos por quantidade de categorias. Elas foram escolhidas e organizadas para que os métodos pudessem ser avaliados em diferentes contextos, e para que a robustez das soluções fossem avaliadas. As quatro bases de dados possuem a mesma natureza de informação (frequência de tipo de objetos na amostra). A Tabela 5.1 descreve as bases de dados levando em

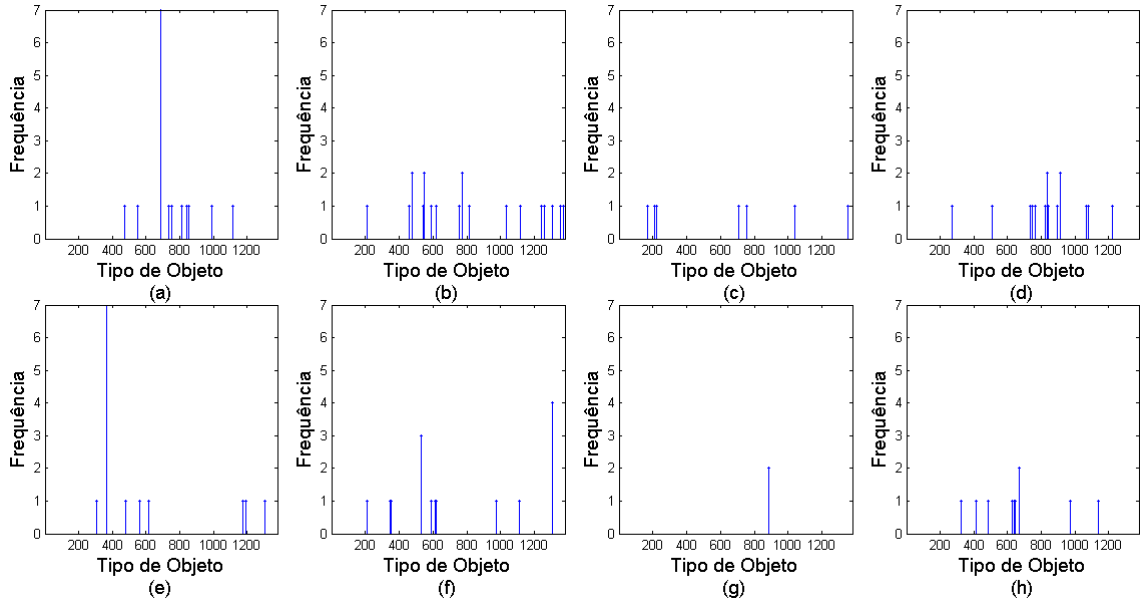

Figura 5.2: Histogramas das amostras da base de dados LabelMe8. (a) Cozinha (17 instâncias e 11 tipos de objetos); (b) Banheiro (23 instâncias e 15 tipos de objetos); (c) Quarto (17 instâncias e 13 tipos de objetos); (d) Escritório (16 instâncias e 14 tipos de objetos); (e) Sala de Conferência (33 instâncias e 17 tipos de objetos); (f) Corredor (15 instâncias e 10 tipos de objetos); (g) Sala de Jantar (4 instâncias e 4 tipos de objetos); (h) Sala de Estar (14 instâncias e 13 tipos de objetos).

consideração as seguintes informações: quantidade de amostras; de tipo de objetos; de categorias; quantidade de objetos por quantidade de categorias; o mínimo, o máximo, a média e o desvio padrão tipo de objetos por amostra; a menor, a maior, a média e o desvio padrão do número de amostras por categoria.

As bases foram ordenadas por ordem crescente de categorias para fins de organização da avaliação e comparação. Uma informação importante é que existem algumas amostras que possuem somente um tipo de objeto: Tabela 5.1 (Mínimo de tipo de Objetos por Amostra). Essa informação também pode ser visualizada na Figura 5.4.

A Figura 5.3, mostra através dos gráficos, a quantidade de amostras por categoria. Os gráficos evidenciam que em todas as bases existem somente algumas categorias com grande quantidade de amostras, o que é um aspecto importante para verificar e entender a dificuldade para o processo de aprendizado dos algoritmos. A média de amostras por categoria diminui de $238,25$ para $43,16$ (Tabela 5.1), da primeira até a quarta base. Uma menor quantidade de amostras por categoria dificulta o processo de generalização do categorizador.

A Figura 5.4 apresenta o histograma de amostras pela quantidade de tipo de objetos. Esses gráficos apresentam que a quantidade de tipos de objetos é pequena. O pico do gráfico (quantidade de amostras por quantidade de objetos) para todas as bases é menor que 10 objetos. Além do que, a maioria das amostras possuem 20 ou menos tipos de objetos: $97,48\%$ para LabelMe4, $98,06\%$ para LabelMe8, $93,72\%$ para RIS62 e $82,32\%$ para SUN407.

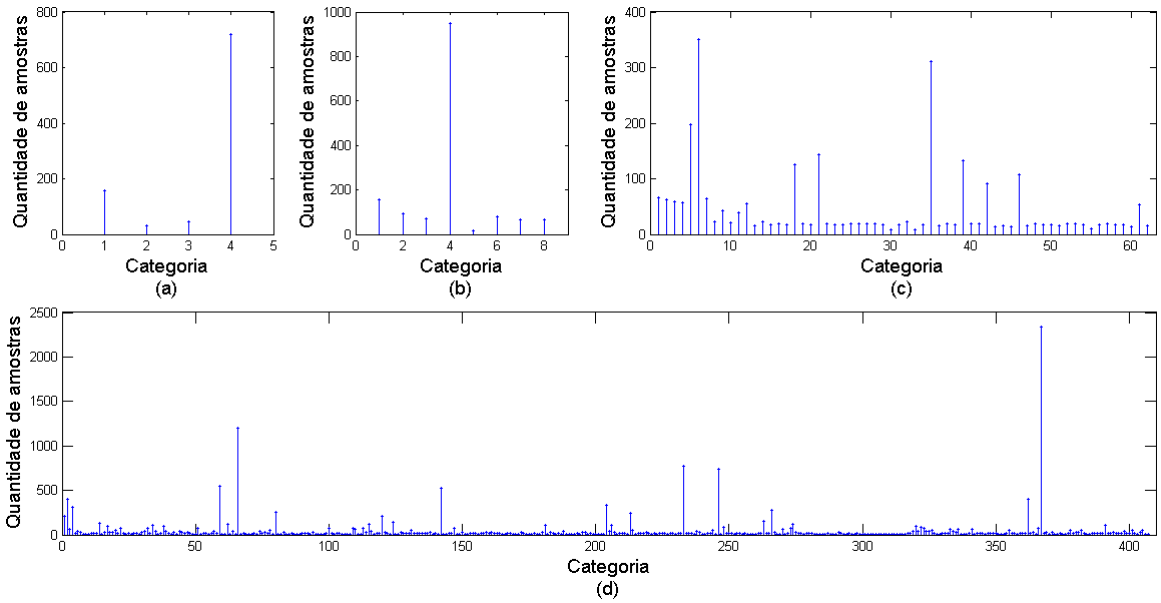

Figura 5.3: Gráficos de quantidade de amostras por suas categorias para cada base de dados: (a) LabelMe4; (b) LabelMe8; (c) RIS62; (d) SUN407.

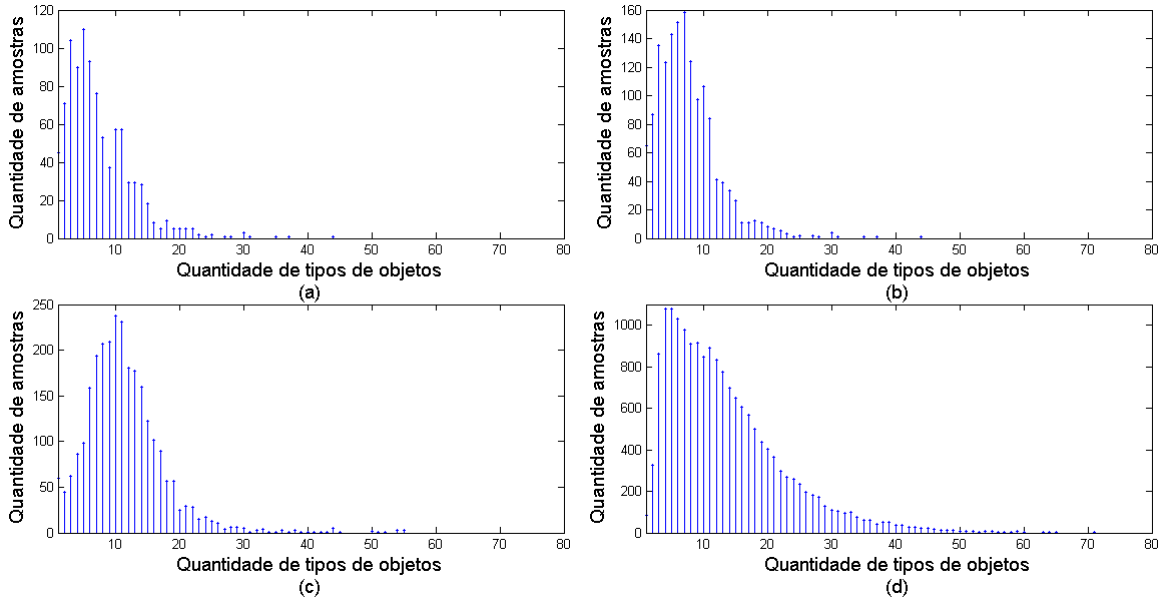

Figura 5.4: Histogramas de quantidade de amostras de base de dados por quantidade de tipos de objetos: (a) LabelMe4; (b) LabelMe8; (c) RIS6;. (d) SUN407.

Informações	LabelMe4	LabelMe8	RIS62	SUN407
Amostras	953	1494	2725	17567
Tipo de Objetos	1069	1380	2945	3716
Categoria	4	8	62	407
Quantidade de Objetos por Quantidade de Categorias	267,25	172,5	47,5	9,13
Mínimo de tipos de Objetos por Amostra	1	1	1	1
Máximo de tipos de Objetos por Amostra	44	44	55	71
Média de tipos de Objetos por Amostra	7,38	7,53	11,35	13,20
Desvio padrão de tipos de Objetos por Amostra	5,19	4,78	6,36	8,87
Menor categoria (número de amostras)	32	15	8	10
Maior categoria (número de amostras)	718	947	350	2334
Média categoria (número de amostras)	238,25	186,75	43,95	43,16
Desvio padrão por categoria (número de amostras)	324,68	309,68	64,36	149,43

Tabela 5.1: Dados estatísticos sobre as bases de dados utilizadas. Referências para bases de dados: LabelMe4 e LabelMe8 [http://labelme.csail.mit.edu/];
RIS62 [web.mit.edu/torralba/www/indoor.html];
SUN407 [groups.csail.mit.edu/vision/SUN/]

5.1.3 Esparsidade dos Dados

No Capítulo 4, foi comentado que a modelagem original dos dados de Categorização de Lugares baseada em Objetos tem um problema de esparsidade inerente para as bases de dados escolhidas. Conforme indicam as Figuras 5.2, 5.4 e a Tabela 5.1, a quantidade de tipos de objetos por amostra é bem menor que a quantidade total de tipos de objetos. A Tabela 5.2 evidencia que o percentual de dados esparsos por base de dados é sempre maior que 99%, desta forma, fazer um Redução de Dimensionalidade dos dados e se organizar os atributos de uma forma mais representativa, se torna um processo fundamental para solucionar o problema.

Essa Tabela também apresenta a Redução de Taxa de Esparsidade e dimensionalidade dos dados. O Agrupamento Fixo foi sempre reduzido para 40 ou 60 atributos, o SOM Raso foi reduzido para 24, 24, 64 e 64 atributos nas respectivas bases de dados e o SOM Profundo Compartimentado foi reduzido para 16, 16, 124 e 407 atributos em cada uma das respectivas bases, pois a saída da representação das amostras por este processo é dependente da quantidade de categorias da base de dados (como foi tratado no Capítulo 4). A dimensão para qual foi realizada a redução das bases na Tabela 5.2 foi escolhida de acordo com a redução de dimensionalidade demonstrada nos experimentos.

Bases de dados	Original	Agrupamento Fixo	SOM Raso	SOM Profundo Compartimentado
LabelMe4	99,33%	89,96% (40 atributos)	37,97% (24 atributos)	67,03% (16 atributos)
LabelMe8	99,45%	89,30% (60 atributos)	57,42% (24 atributos)	76,56% (16 atributos)
RIS62	99,61%	84,38% (60 atributos)	31,88% (64 atributos)	93,60% (124 atributos)
SUN407	99,64%	82,22% (60 atributos)	21,42% (64 atributos)	91,50% (407 atributos)

Tabela 5.2: Taxa de Esparsidade dos dados.

5.2 Resultados dos Experimentos

Os resultados dos experimentos estão relatados e discutidos nesta seção. Os primeiros experimentos foram realizados para validação dos métodos PbSOM e TSOM, já os experimentos posteriores são relativos ao problema de Categorização de Lugares baseada em Objetos. Os códigos foram implementados em Matlab e executados uma máquina com processador i5 1.6Ghz com 4Gb de RAM.

5.2.1 Replicando Resultados PSOMs

Os experimentos de validação do PSOM e TSOM replicam e fazem variações nos experimentos existentes para os dois métodos em suas referências (CHENG; FU; WANG, 2009; LÓPEZ-RUBIO, 2009). Os experimentos escolhidos para os dois métodos foram os que analisavam as topologias das redes.

5.2.1.1 Resultados Experimentais PbSOM

Alguns experimentos foram realizados para validar implementação do PbSOM. Os experimentos replicados têm como objetivo agrupamento de dados e visualização de um problema existente no banco de dados da UCI *Machine Learning* (BLAKE; MERZ, 1998): o conjunto de teste da *Image Segmentation*, denominado como ImgSeg, que consiste de 2100 vetores de características com 19 dimensões. Para o algoritmos de aprendizagem SOCEM do PbSOM foram utilizadas cinco configurações para a estrutura da rede, são elas: 3x3, 4x4, 5x5, 6x6, 7x7 e 8x8 em uma grade retangular com nodos igualmente espaçados utilizando um função gaussiana como função de vizinhança, a visualização dos padrões atribuídos a cada nodo pode ser conferida na Figura 5.5.

5.2.1.2 Resultados Experimentais TSOM

Os experimentos foram executados com 10.000 amostras de treinamento da distribuição uniforme, o qual é utilizado como um problema padrão porque permite avaliar facilmente o desdobramento do mapa auto-organizável e ter uma noção da aprendizagem da rede. Os algoritmos foram executados em 100.000 épocas (uma amostra por época), foram testadas as configurações 1D (16, 32 e 64 nodos) e 2D (16, 32 e 64 nodos) como variação de topologia do mapa. As Figuras 5.6 e 5.7 mostram as organizações das redes e as elipses são nodos utilizando a distância Mahalanobis para cada componente da mistura. A Figura 5.6 apresenta a topologia da TSOM unidimensional para os experimentos e também defini as regiões do espaço bidimensional de influência de cada nodo. A Figura 5.7 apresenta a mesma informação da Figura 5.6 só que para um TSOM bidimensional.

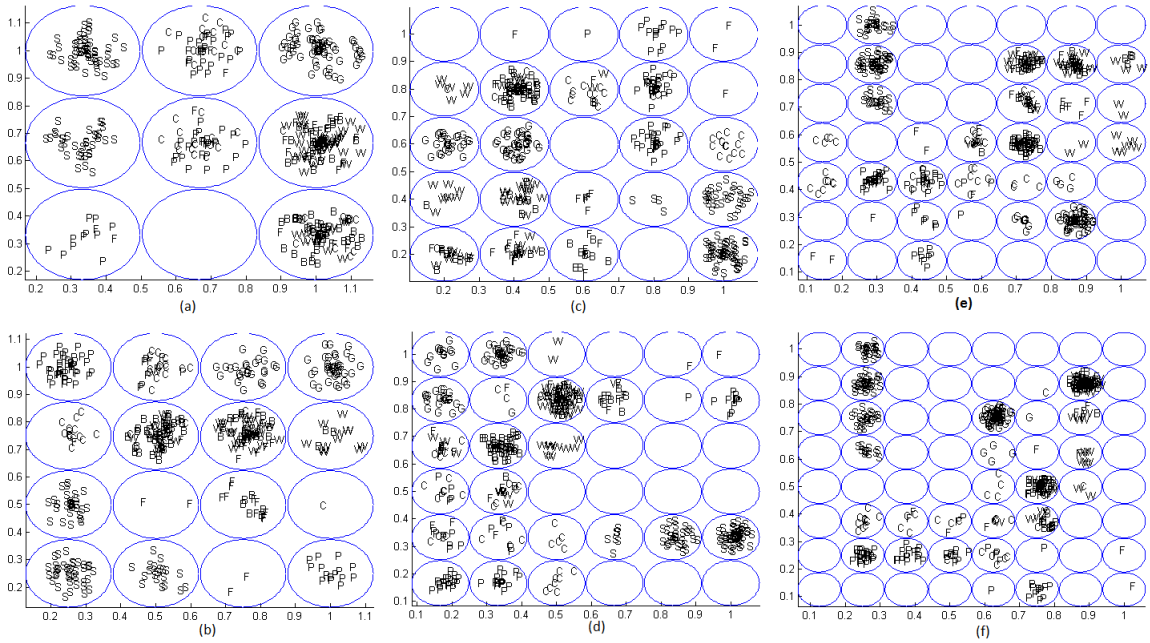

Figura 5.5: Rede PbSOM com algoritmo de aprendizagem SODAEM para (a) 3x3 nodos (b) 4x4 nodos (c) 5x5 nodos (d) 6x6 nodos (e) 7x7 nodos (f) 8x8 nodos.

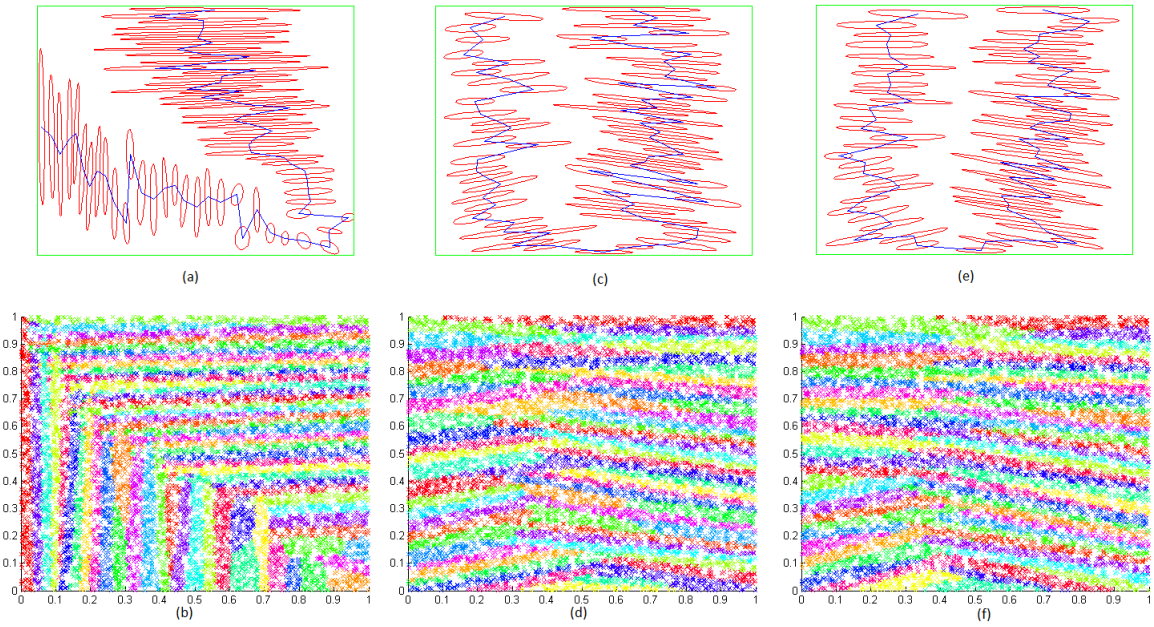

Figura 5.6: Rede TSOM com 64 nodos em 1D, visualização das organização da rede e a zonas de fronteiras de padrões (a) Rede com algoritmo *Peel* (b) Distribuição dos padrões classificados pela Rede com algoritmo *Peel* (c) Rede com algoritmo *Shoham* (d) Distribuição dos padrões classificados pela Rede com algoritmo *Shoham* (e) Rede com algoritmo *Direct* (f) Distribuição dos padrões classificados pela Rede com algoritmo *Direct*.

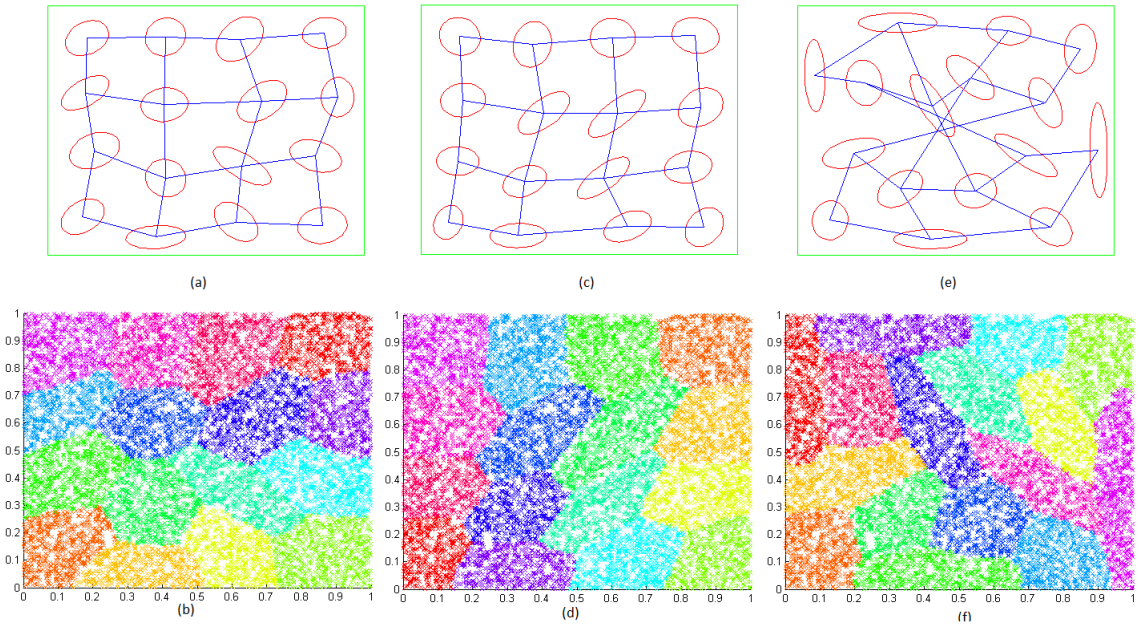

Figura 5.7: Rede TSOM com 16 nodos em 2D, visualização da organização da rede e as zonas de fronteiras de padrões (a) Rede com algoritmo *Peel* (b) Distribuição dos padrões classificados pela Rede com algoritmo *Peel* (c) Rede com algoritmo *Shoham* (d) Distribuição dos padrões classificados pela Rede com algoritmo *Shoham* (e) Rede com algoritmo *Direct* (f) Distribuição dos padrões classificados pela Rede com algoritmo *Direct*.

5.2.2 Experimentos de Categorização de Lugares

Os experimentos foram executados para 7 variações de métodos. O primeiro é o Filtro Bayesiano que foi proposto para a solução de Categorização de Lugares por Viswanathan (VISWANATHAN et al., 2009, 2010). As propostas de solução deste trabalho são as composições de solução para duas fases do processo: Redução de Dimensionalidade e Categorizador. Para Redução de Dimensionalidade é utilizado o Agrupamento Fixo, SOM Raso ou SOM Profundo Compartimentado. Para categorização são empregados o *Probabilistic Self-Organizing Map* (PbSOM) ou o *t-Student Self-Organizing Map* (TSOM). Então, os 7 métodos para os quais experimentos foram executados são: Filtro Bayesiano, Agrupamento Fixo + PbSOM, Agrupamento Fixo + TSOM, SOM Raso + PbSOM e SOM Raso + TSOM, SOM Profundo Compartimentado + PbSOM e SOM Profundo Compartimentado + TSOM. Os experimentos foram executados para as 4 bases de dados: LabelMe4, LabelMe8, RIS62 e SUN407. Isso determina um total de 28 composições de resultados diferentes (7 métodos e 4 bases de dados).

Os parâmetros dos modelos baseados em SOM foram definidos seguindo sugestões existentes na literatura. Os valores para os parâmetros do modelo de Agrupamento c que é a quantidade de atributos de saída foram: 40 para LabelMe4 e 60 para LabelMe8, RIS62 e SUN407. Para o SOM Raso a quantidade de nodos são: 24, 24, 64 e 64 respectivamente para as bases de dados. Para o SOM Profundo Compartimentado o *pipeline* da primeira fase tem somente uma camada e de tamanho: 1024 nodos para LabelMe4 e LabelMe8; 2048 nodos para RIS62 e

SUN407. Para etapa de múltiplos *pipelines* o experimentos foram executados com 3 camadas no qual o número de *pipelines* é a quantidade de categorias da base. A primeira camada de cada *pipeline* tem 64 nodos, a segunda 16 nodos, já a última tem quantidades de nodos diferentes para cada base: 4 para LabelMe4; 2 para LabelMe e RIS62; e 1 para SUN407. A Tabela 5.3 apresenta os parâmetros para as camadas SOM. Para o Categorizador foram utilizados Múltiplos PSOMs, um para cada categoria da base de dados de dimensão entre 2x2 e 4x4. Para PbSOM é realizado de 100 à 300 época (1 época = todas as amostra da categoria) e para TSOM é realizado de 1.000 à 2.000 épocas (1 época = 1 amostra).

Bases de dados	n	$\alpha(0)$	β	τ	w	k	σ
SOM Raso	[20 80]	[0.01 0.15]	[0.7 0.9]	[2 5]	1	[1 2]	-
Fase *Pipeline* único	[5 10]	[0.2 0.4]	[0.9 0.95]	[2 5]	$\lfloor nodos/100 \rfloor$	[1 2]	-
Fase *Pipelines* múltiplos	[5 20]	[0.2 0.4]	[0.9 0.95]	-	-	[1 2]	[1.5 2.5]

Tabela 5.3: Os parâmetros das camadas SOMs são: n que é o número de épocas; $\alpha(0)$ a taxa de aprendizagem inicial; β o fator de decaimento da taxa de aprendizagem; τ fator de função de vizinhança em Progressão Geométrica (somente necessário para a fase de único*pipeline*); *w* raio máximo da função de vizinhança (somente existente necessário para a fase de único*pipeline*); *k* fator da função ponto de corte; σ raio máximo da função de vizinhança (somente existente necessário para a fase de múltiplos *pipelines*).

Todos os experimentos foram executados 10 vezes com os conjuntos de amostras disjuntos: treinamento 90% e testes 10% das amostras. As métricas de avaliação para os diversos experimentos são: a taxa média de acerto e desvio padrão, melhor taxa de acerto, taxa média de acerto por categoria e matriz de confusão. A Tabela 5.4 apresenta a taxa de acerto médio e o desvio padrão para as 28 composições e a Tabela 5.5 apresenta o melhor resultado para cada uma das composições. Os melhores resultados para cada base estão em negrito. A análise dos resultados será realizada nos próximos tópicos por cada base de dados.

Bases de dados	Filtro Bayesiano (%)	Agrupamento Fixo + PbSOM (%)	Agrupamento Fixo + TSOM (%)	SOM Raso + PbSOM (%)	SOM Raso + TSOM (%)	SOM Profundo Compartimentado + PbSOM (%)	SOM Profundo Compartimentado + TSOM (%)
LabelMe4	96,56 (1,57)	76,60 (10,28)	85,26 (3,26)	79,69 (9,51)	94,12 (9,69)	**96,80 (1,49)**	96,49 (1,96)
LabelMe8	81,34 (2,81)	49,15 (9,91)	74,31 (4,30)	72,42 (6,58)	**89,90 (3,51)**	85,49 (3,08)	87,25 (2,18)
RIS62	22,90 (2,94)	29,01 (3,36)	43,92 (2,66)	36,24 (2,18)	48,24 (3,01)	60,34 (2,35)	**61,30 (3,08)**
SUN407	3,74 (0,26)	11,25 (0,55)	26,46 (1,04)	14,05 (1,33)	25,98 (0,70)	16,39 (0,58)	**32,46 (1,30)**

Tabela 5.4: Taxa de acerto média e desvio padrão para método x base de dados.

Bases de dados	Filtro Bayesiano	Agrupamento Fixo + PbSOM	Agrupamento Fixo + TSOM	SOM Profundo Compartimentado + TSOM	SOM Raso + PbSOM	SOM Profundo Compartimentado + PbSOM	SOM Raso + TSOM
LabelMe4	**97,94%**	88,54%	88,66%	91,75%	96,91%	98,97%	**100%**
LabelMe8	84,31%	65,33%	76,47%	79,74%	**91,50%**	89,54%	90,20%
RIS62	25,26%	33,45%	44,37%	40,27%	53,92%	63,82%	**66,89%**
SUN407	3,92%	13,16%	28,55%	16,22%	27,10%	17,35%	**34,07%**

Tabela 5.5: Melhor taxa de acerto para método x base de dados.

5.2.2.1 Análise dos Resultados com a Base LabelMe4

Para a base de dados LabelMe4, os melhores resultados são para o SOM Profundo Compartimentado: $96,56\%$. As soluções com o Filtro Bayesiano e SOM Profundo Compartimentado + TSOM possuem uma variação de menos de 1% para o melhor resultado. O resultado do SOM Profundo Compartimentado + PbSOM tem menor variabilidade dentre esses resultados tendo somente $1,49\%$ de desvio padrão em comparação a $1,57\%$ do Filtro Bayesiano e $1,96\%$ do SOM Profundo Compartimentado + TSOM. Os resultados para o Agrupamento Fixo não foram bons para LabelMe4 ficaram mais de 11% abaixo da média do melhor, além do que, os desvios padrão para o Agrupamento Fixo foram muito altos, chegando a $10,28\%$ no caso Agrupamento Fixo + PbSOM. Os resultados com o SOM Raso obtiveram desempenho entre o Agrupamento Fixo e o SOM Profundo Compartimentado.

A Figura 5.8 apresenta de forma gráfica as taxas de acerto por categoria para LabelMe4 de forma a facilitar a visualização dos resultados. As matrizes de confusão que estão descritas pelas Tabelas 5.6 a 5.12 enriquecem análise dos resultados. Quando os três melhores resultados são comparados (Filtro Bayesiano, SOM Profundo Compartimentado + TSOM e SOM Profundo Compartimentado + PbSOM), pode-se evidenciar que o Filtro Bayesiano possui as melhores taxas de acerto para as classes majoritárias 1 e 4. Essas duas categorias compõem $91,82\%$ da base, assim acertando as categorias majoritárias elevando a taxa de acerto geral, Tabelas 5.11 e 5.12. Contudo, esses dois métodos obtêm uma taxa de acerto baixa para as categorias minoritárias. Já as soluções com SOM Profundo Compartimentado acertam em 100% as categorias minoritárias e acerta um pouco menos as classes majoritárias do que os dois métodos com maior acerto global, Tabelas 5.11 e 5.12.

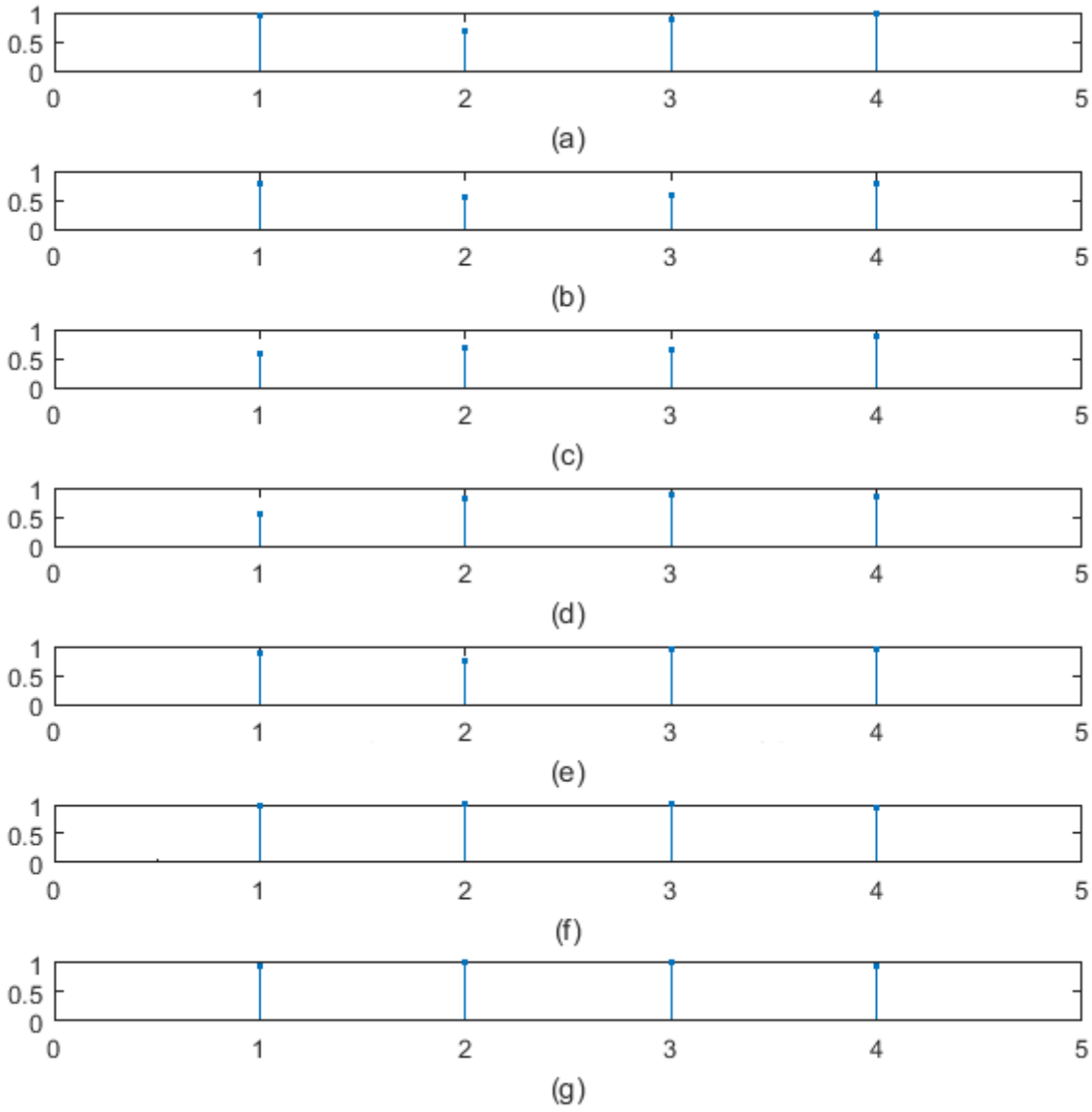

Figura 5.8: Gráfico de taxa de acerto, no intervalo de [0, 1], para cada categoria na LabelMe4. (a) Filtro Bayesiano; (b) Agrupamento Fixo + PbSOM; (c) Agrupamento Fixo + TSOM; (d) SOM Raso + PbSOM; (e) SOM Raso + TSOM; (f) SOM Profundo Compartimentado + PbSOM; (g) SOM Profundo Compartimentado + TSOM.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**95,83%**	25,00%	0%	0,93%
Banheiro	0%	**66,67%**	0%	0%
Quarto	0%	0%	**86,67%**	0%
Escritório	4,17%	8,33%	13,33%	**99,07%**

Tabela 5.6: Matriz de confusão do Filtro Bayesiano para LabelMe4.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**79,37%**	22,50%	30,00%	13,19%
Banheiro	1,25%	**55,00%**	2,00%	0,56%
Quarto	5,00%	5,00%	**58,00%**	7,78%
Escritório	14,37%	17,50%	10,00%	**78,47%**

Tabela 5.7: Matriz de confusão do Agrupamento Fixo + PbSOM para LabelMe4.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**58,12%**	10,00%	10,00%	4,58%
Banheiro	4,37%	**67,50%**	0%	2,505
Quarto	3,75%	7,50%	**64,00%**	2,91%
Escritório	33,75%	15,00%	26,00%	**90,00%**

Tabela 5.8: Matriz de confusão do Agrupamento Fixo + TSOM para LabelMe4.

Os resultados do Filtro Bayesiano e do SOM Profundo Compartimentado estão na ordem de 96%. Para analisar se os resultados são estatisticamente diferentes foram utilizados os testes de *Friedman* e *Wilcoxon.* Os valores obtidos pelos métodos variam de 0 a 1 e tem um forte grau de similaridade quanto a mais próximos de 1. Os resultados são apresentados na Tabela 5.13. Os resultados indicam que existe uma correlação forte entre Filtro Bayesiano e o SOM Profundo Compartimentado + PbSOM. Já o SOM Compartimentado Profundo + TSOM possui uma leve correlação com os dois primeiros métodos analisados.

Bases de Dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**56,25%**	15,00%	0%	1,67%
Banheiro	32,50%	**80,00%**	4,00%	10,42%
Quarto	8,13%	0%	**90,00%**	3,75%
Escritório	3,13	5,00%	6,00%	**84,17%**

Tabela 5.9: Matriz de confusão do SOM Raso + PbSOM para LabelMe4.

Bases de Dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**89,38%**	25,00%	4,00%	2,92%
Banheiro	6,88%	**75,00%**	0%	0,42%
Quarto	0,63%	0%	**94,00%**	0,42%
Escritório	3,13%	0%	2,00%	**96,25%**

Tabela 5.10: Matriz de confusão do SOM Raso + TSOM para LabelMe4.

Bases de Dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**96,88**	0%	0%	2,92%
Banheiro	3,13%	**100%**	0%	0,56%
Quarto	0%	0%	**100%**	0,14%
Escritório	0%	0%	0%	**96,39%**

Tabela 5.11: Matriz de confusão do SOM Profundo Compartimentado + PbSOM para LabelMe4.

Bases de Dados	Cozinha	Banheiro	Quarto	Escritório
Cozinha	**95,63%**	0%	0%	3,06%
Banheiro	2,50%	**100%**	0%	0,69%
Quarto	1,87%	0%	**100%**	0%
Escritório	0%	0%	0%	**96,25%**

Tabela 5.12: Matriz de confusão do SOM Profundo Compartimentado + TSOM para LabelMe4.

Métodos	*Friedman*	*Wilcoxon*
Filtro de Bayesiano / SOM Profundo Compartimentado + PbSOM	1	1
Filtro de Bayesiano / SOM Profundo Compartimentado + TSOM	0,4795	0,6996
SOM Compartimentado Profundo + PbSOM / SOM Profundo Compartimentado + TSOM	0,4795	0,7082

Tabela 5.13: Teste de hipóteses para resultados do LabelMe4.

5.2.2.2 Análise dos Resultados com a Base LabelMe8

Os resultados para LabelMe8 foram melhores para o método SOM Raso + TSOM com a taxa média de acerto de $89,90\%$, que foi ao menos 2% mais alta do que as outras soluções. Todos os resultados para o LabelMe8 tiveram um desvio padrão acima de 2%, variando de $2,18\%$ a $9,91\%$ (Tabela 5.4). O SOM Profundo Compartimentado também obteve resultados expressivos para a LabelMe8. Os resultados para o Agrupamento Fixo foram os piores para o LabelMe8, como ocorreu também para o LabelMe4, e ficaram abaixo dos resultados dos outros modelos.

A Figura 5.9 apresenta as taxas de acerto por categoria do LabelMe8 para visualização em forma de gráfico. As matrizes de confusão para esses resultados estão descritas nas Tabelas 5.14 a 5.20. Um padrão que se repete em todos os gráficos de todos os métodos é que a categoria 8, Sala de Jantar, possui uma taxa de acerto sempre baixa. As categorias que são atribuídas erroneamente para as amostras de Sala de Jantar são principalmente: Cozinha , Escritório, Sala de Conferência e Quarto, nesta ordem decrescente de taxa de erro nas matrizes de confusão. A Sala de Jantar é um ambiente que possui objetos que também aparecem em outras categorias, o que o torna sua identificação mais difícil.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**76,56%**	0%	9,37%	2,36%	0%	0%	0%	3,57%
Banheiro	14,06%	**95,00%**	0%	0%	0%	0%	0%	3,57%
Quarto	1,56%	0%	**68,75%**	1,05%	0%	0%	0%	21,42%
Escritório	1,56%	0%	0	**81,31%**	0%	6,25%	0%	0%
Sala de Conferência	6,25%	2,50%	15,62%	1,84%	**87,50%**	9,37%	0%	17,85%
Corredor	0%	2,50%	3,12%	12,10%	12,50%	**81,25%**	0%	7,14%
Sala de Estar	0%	0%	0%	0,78%	0%	0%	**100%**	17,85%
Sala de Jantar	0%	0%	3,12%	0,52%	0%	3,12%	0%	**28,57%**

Tabela 5.14: Matriz de confusão do Filtro Bayesiano para LabelMe8.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**52,50%**	5%	5%	3,05%	5%	1,25%	11,42%	21,42%
Banheiro	15,62%	**76%**	6,25%	4,31%	0%	15%	4,28%	4,28%
Quarto	0,62%	4%	**37,50%**	1,26%	5%	5%	1,42%	5,71%
Escritório	13,12%	10%	28,75%	**85,57%**	20%	18,75%	5,71%	35,71%
Sala de Conferência	0%	0%	1,25%	0,10%	**60%**	0%	0%	0%
Corredor	0,62%	3%	1,25%	0,94%	10%	**51,25%**	0%	2,85%
Sala de Estar	10%	1%	3,75%	0,84%	0%	2,50%	**72,85%**	4,28%
Sala de Jantar	7,50%	1%	16,25%	3,89%	0%	6,25%	4,28%	**25,71%**

Tabela 5.15: Matriz de confusão do Agrupamento Fixo + PbSOM para LabelMe8.

Os resultados do SOM Raso + PbSOM e do SOM Profundo Compartimentado são próximos. Os resultados de correlação são apresentados na Tabela 5.21. Os resultados indicam que existe uma correlação forte entre Filtro Bayesiano e o SOM Profundo Compartimentado +

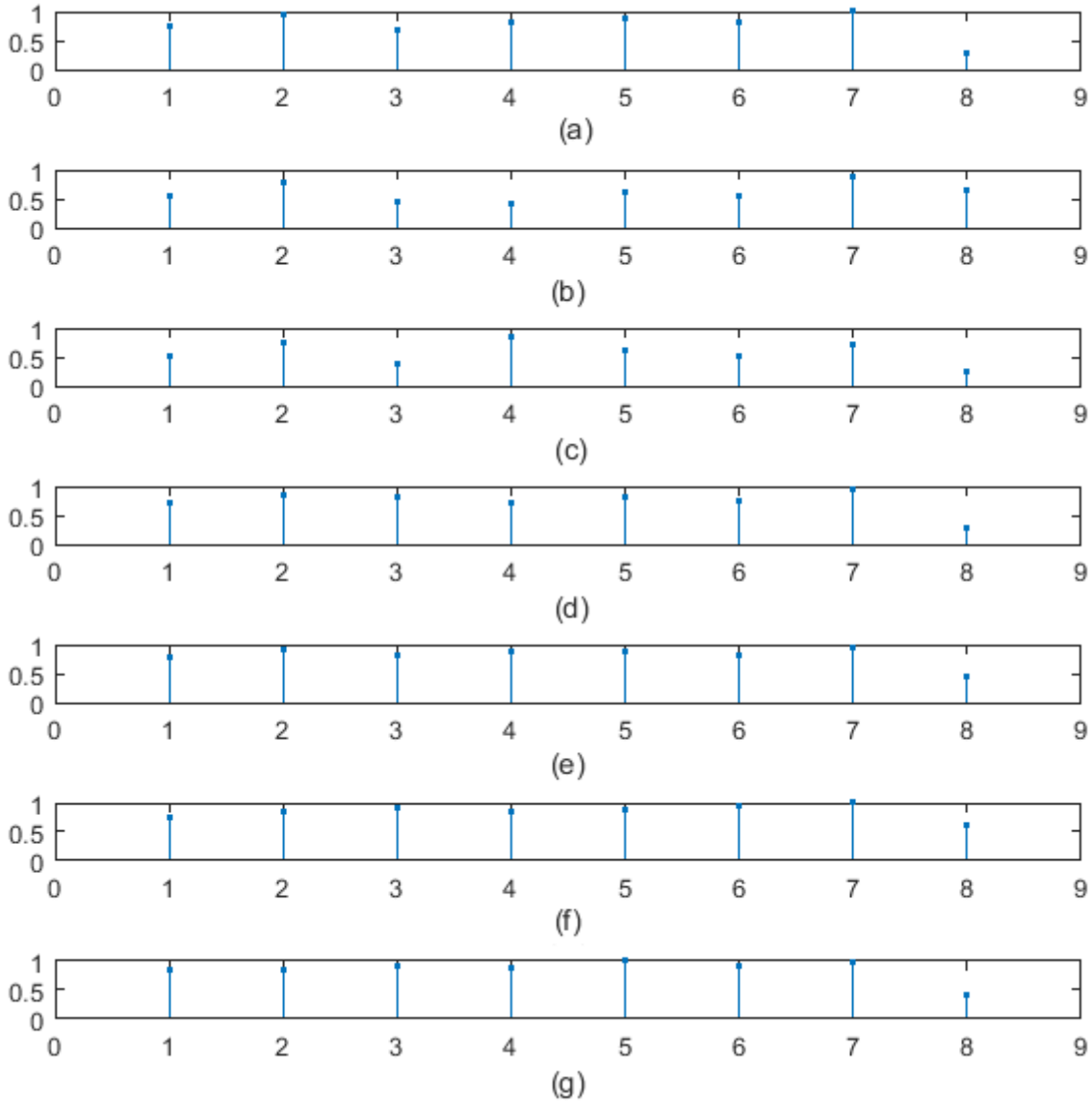

Figura 5.9: Gráfico de taxa de acerto, no intervalo de [0, 1], para cada categoria na LabelMe8. (a) Filtro Bayesiano; (b) Agrupamento Fixo + PbSOM; (c) Agrupamento Fixo + TSOM; (d) SOM Raso + PbSOM; (e) SOM Raso + TSOM; (f) SOM Profundo Compartimentado + PbSOM; (g) SOM Profundo Compartimentado + TSOM.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**53,75%**	5,00%	8,75%	8,42%	5,00%	6,25%	4,29%	17,14%
Banheiro	8,75%	**78,00%**	23,75%	3,68%	0%	6,25%	0%	2,86%
Quarto	3,13%	4,00%	**45,00%**	3,58%	0%	5,00%	0%	2,86%
Escritório	4,37%	0%	1,25%	**41,05%**	0%	2,50%	0%	4,29%
Sala de Conferência	0%	0%	0%	0,32%	**60,00%**	1,25%	0%	0%
Corredor	0,63%	4,00%	0%	1,68%	15,00%	**55,00%**	0%	2,86%
Sala de Estar	1,25%	0%	1,25%	6,53%	0%	2,50%	**87,14%**	5,71%
Sala de Jantar	28,13%	9,00%	20,00%	34,74%	20,00%	21,25%	8,57%	**64,29%**

Tabela 5.16: Matriz de confusão do Agrupamento Fixo + PbSOM para LabelMe8.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**73,12%**	5,00%	3,75%	5,15%	0%	3,75%	1,42%	7,14%
Banheiro	9,37%	**84,00%**	2,50%	5,36%	0%	1,25%	2,85%	5,71%
Quarto	2,50%	1,00%	**80%**	2,10%	0%	7,50%	0%	15,71%
Escritório	3,12%	1,00%	0%	**71,78%**	0%	6,25%	0%	4,28%
Sala de Conferência	0,62%	1,00%	3,75%	3,47%	**80,00%**	0%	1,42%	28,57%
Corridor	3,75%	3,00%	1,25%	7,36%	20,00%	**73,75%**	0%	7,14%
Sala de Estar	0%	3,00%	0%	1,05%	0%	1,25%	**94,28%**	2,85%
Sala de Jantar	7,50%	2,00%	8,75%	3,68%	0%	6,25%	0%	**28,57%**

Tabela 5.17: Matriz de confusão do SOM Raso + PbSOM para LabelMe8.

PbSOM. Já o SOM Compartimentado Profundo + TSOM possui uma leve correlação com os dois primeiros métodos analisados. Desta forma, os testes de hipóteses sugerem que as distribuições não são as mesmas para os resultados dos diversos métodos.

Bases de Dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**77,50%**	3,00%	6,25%	3,47%	0%	2,50%	0%	14,28%
Banheiro	7,50%	**91,00%**	0%	0,31%	0%	0%	0%	2,85%
Quarto	3,12%	2,00%	**81,25%**	1,15%	0%	1,25%	1,42%	22,85%
Escritório	2,50%	0%	0%	**88,21%**	0%	3,75%	0%	5,71%
Sala de Conferência	0%	0%	0%	0,84%	**90,00%**	7,50%	0%	7,14%
Corridor	0,62%	2,00%	1,25%	2,00%	10,00%	**81,25%**	0%	1,42%
Sala de Estar	0%	0%	0%	0%	0%	0%	**95,71%**	1,42%
Sala de Jantar	8,75%	2,00%	11,25%	4,00%	0%	3,75%	2,85%	**44,28%**

Tabela 5.18: Matriz de confusão do SOM Raso + TSOM para LabelMe8.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**78,12%**	11,00%	0%	1,89%	0%	0%	0%	10,00%
Banheiro	3,75%	**77,00%**	0%	0,31%	0%	1,25%	0%	4,28%
Quarto	3,12%	2,00%	**83,75%**	1,15%	0%	0%	0%	7,14%
Escritório	0,62%	0%	1,25%	**78,42%**	10,00%	0%	0%	4,28%
Sala de Conferência	3,12%	3,00%	1,25%	1,89%	**85,00%**	0%	2,85%	20,00%
Corredor	1,25%	2,00%	1,25%	4,00%	5,00%	**93,75%**	0%	1,42%
Sala de Estar	0%	0%	0%	0,42%	0%	0%	**97,14%**	4,28%
Sala de Jantar	10,00%	5,00%	12,50%	11,89%	0%	5,00%	0%	**48,57%**

Tabela 5.19: Matriz de confusão do SOM Profundo Compartimentado + PbSOM para LabelMe8.

Bases de dados	Cozinha	Banheiro	Quarto	Escritório	Sala de Conferência	Corredor	Sala de Estar	Sala de Jantar
Cozinha	**85,00%**	11,00%	3,75%	2,94%	0%	0%	0%	28,57%
Banheiro	5,00%	**84,00%**	0%	0,31%	0%	0%	0%	0%
Quarto	0,62%	2,00%	**91,25%**	0,73%	0%	0%	0%	4,28%
Escritório	1,87%	0%	0%	**88,94%**	0%	6,25%	0%	14,28%
Sala de Conferência	2,50%	0%	0%	1,78%	**100,00%**	1,25%	0%	7,14%
Corridor	0,62%	3,00%	1,25%	3,57%	0%	**92,50%**	0%	2,85%
Sala de Estar	0%	0%	0%	0%	0%	0%	**97,14%**	1,42%
Sala de Jantar	4,37%	0%	3,75%	1,68%	0%	0%	2,85%	**41,42%**

Tabela 5.20: Matriz de confusão do SOM Profundo Compartimentado + TSOM para LabelMe8.

Métodos	*Friedman*	*Wilcoxon*
SOM Raso + PbSOM / SOM Profundo Compartimentado + PbSOM	0,3173	0,6748
SOM Raso + PbSOM / SOM Profundo Compartimentado + TSOM	0,2059	0,1722
SOM Compartimentado Profundo + PbSOM / SOM Profundo Compartimentado + TSOM	0,2059	0,2868

Tabela 5.21: Teste de hipóteses para resultados do LabelMe8.

5.2.2.3 Análise dos Resultados com a Base RIS62

As bases de dados RIS62 e SUN407 possuem uma quantidade maior de categorias e a relação de quantidade de objetos por quantidade de categorias é menor do que o LabelMe4 e LabelMe8. Esses são fatores essenciais para que a taxa de acerto para estas bases tenham piores resultados em comparação as duas com menos categorias. Os melhores resultados para RIS62 foram obtidos através do SOM Profundo Compartimentado + TSOM e SOM Profundo Compartimentado + PbSOM com $61,30\%$ e $60,34\%$, respectivamente. Os resultados médios do SOM Profundo Compartimentado foram melhores do que Filtro Bayesiano, Agrupamento Fixo e SOM Raso sendo superior ao melhor resultado médio destes. Os experimentos de Agrupamento Fixo foram melhores que o Filtro Bayesiano: $29,01\%$ para o Agrupamento Fixo + PbSOM e $43,92\%$ para Agrupamento Fixo + TSOM; $36,24\%$ para o SOM Raso + PbSOM e $48,24\%$ para o SOM Raso + TSOM, enquanto o Filtro Bayesiano obteve somente 21% de taxa de acerto médio.

Os resultados para a base RIS62 são apresentados detalhadamente pela Figura 5.10 (taxa de acertos por categoria) e a Figura 5.11 (matriz de confusão). A diagonal da matriz é a taxa de acerto das categorias: o branco corresponde a 0% e preto a 100%, sendo os tons de cinza intermediários as gradações entre as duas cores em tons de cinza. A categorização realizada pelo Filtro Bayesiano possuem 27 categorias sem nenhum acerto, correspondendo a $43,55\%$ das categorias, e somente três categorias com taxa de acertos maiores que 50%, o que corresponde a $4,84\%$ das categorias (Tabela 5.22). Os resultados para as composições com Agrupamento Fixo e SOM Raso nesse tipo de avaliação são um pouco melhores que o Filtro Bayesiano, sendo as melhores composições com SOM Profundo Compartimentado.

Bases de dados	Filtro Bayesiano	Agrupamento Fixo+ PbSOM	Agrupamento Fixo + TSOM	SOM Raso +PbSOM	SOM Raso + TSOM	SOM Profundo Compartimentado +PbSOM	SOM Profundo Compartimentado + TSOM
Quantidade de Categorias com taxa de acerto zerada	27	5	9	7	7	**0**	5
Percentual de Categorias com taxa de acerto zerada	43,55%	8,06%	14,52%	11,29%	11,29%	**0%**	8,06%
Quantidade de Categorias com taxa de acerto acima de 50%	3	10	11	16	13	32	**57**
Percentual de Categorias com taxa de acerto acima de 50%	4,84%	16,13%	17,74%	25,81%	20,97%	51,61%	**91,94%**

Tabela 5.22: Análise de taxas de acertos das categorias da base de dados RIS62.

O SOM Profundo Compartimentado + PbSOM e SOM Profundo Compartimentado + TSOM obtiveram os melhores resultados nestes experimentos que foram uma e cinco categorias com taxa de acerto zeradas, correspondendo a 0% e $8,06\%$ das categorias, com 32 e 57 categorias

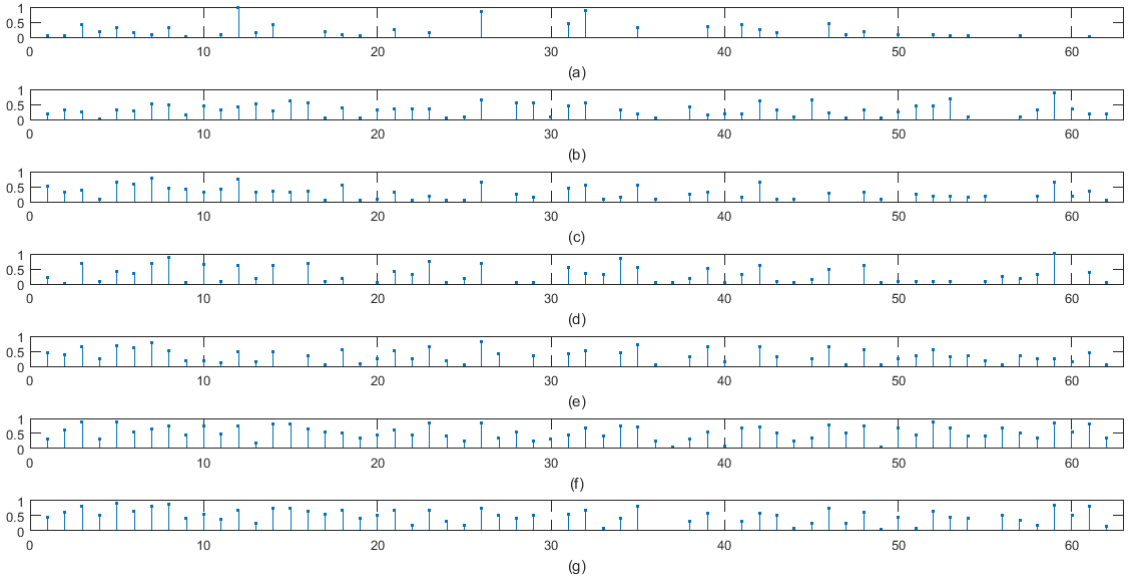

Figura 5.10: Gráfico de taxa de acerto, no intervalo de [0, 1], para cada categoria na RIS62: (a) Filtro Bayesiano; (b) Agrupamento Fixo + PbSOM; (c) Agrupamento Fixo + TSOM; (d) SOM Raso + PbSOM; (e) SOM Raso + TSOM; (f) SOM Profundo Compartimentado + PbSOM; (g) SOM Profundo Compartimentado + TSOM.

com média acima de 50%, o que equivale a $51,61\%$ e $91,94\%$ das categorias (Tabela 5.22). Os processos que utilizam SOM Profundo Compartimentado além de melhorar a performance dos resultados para base RIS62 também diminui a quantidade de categorias que possuem taxas de acertos zeradas. A matriz de confusão do Filtro Bayesiano, Figura 5.11, apresenta as categorias de número 10 e 12 com uma alta taxa de rotulação pelo categorizador, ou seja, muitos exemplos da RIS62 são categorizados como esses dois tipos, $71,30\%$ no total (Tabela 5.22). Assim, o resultados para o Filtro Bayesiano é baixo por essa tendência na categorização dessa base.

Os dois resultados utilizando o SOM Profundo Compartimentado são próximos. Os resultados de correlação entre essas duas técnicas utilizando o teste de *Friedman* e *Wilcoxon* são $0,0016$ e $0,0001$ respectivamente. Esses resultados indicam que as distribuições de resultados entre esses dois métodos não possuem nenhuma relação, desta forma as distribuições dos resultados dos métodos são distintas.

5.2.2.4 Análise dos Resultados com a Base SUN407

O melhor resultado para a base de dados SUN407 foi o SOM Profundo Compartimentado + TSOM, $32,46\%$. Esse resultado foi 8 vezes maior do que o Filtro Bayesiano. As composições com PbSOM tiveram resultados inferiores de 17%. As taxas de acerto por categoria para a base de dados SUN407 são demonstradas pela Figura 5.12 e a matriz de confusão pela Figura 5.13. Para os resultados do Filtro Bayesiano há muitas categorias coma taxa zerada. O Filtro Bayesiano apresenta para SUN407 356 categorias com a taxa de acerto zerada, $87,47\%$ das categorias, e somente 5 categorias com essa taxa acima de 50%, o que corresponde a $1,23\%$ das categorias (Tabela 5.23).

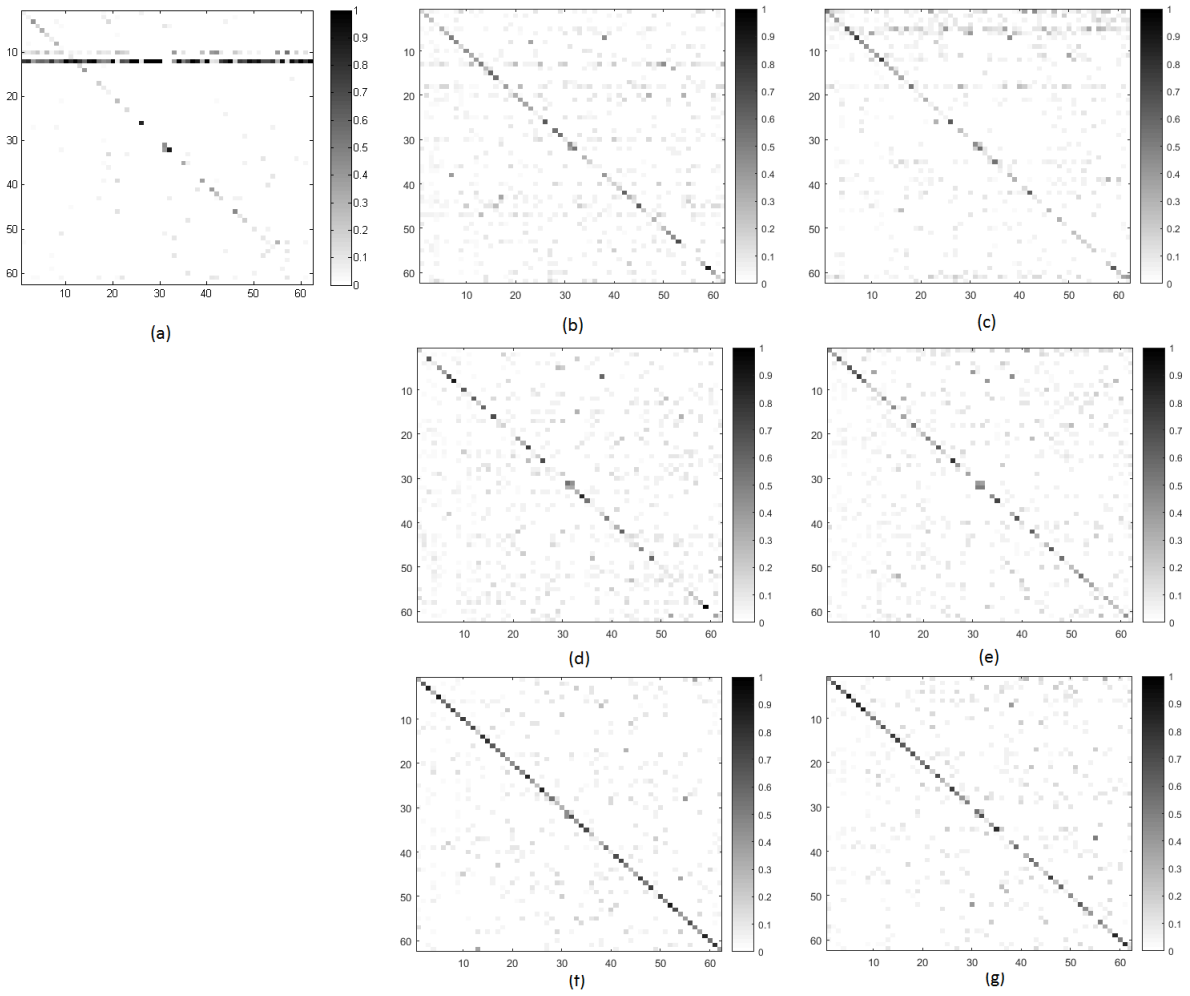

Figura 5.11: Matriz de confusão para RIS62: (a) Filtro Bayesiano; (b) Agrupamento Fixo + PbSOM; (c) Agrupamento Fixo + TSOM; (d) SOM Raso + PbSOM; (e) SOM Raso + TSOM; (f) SOM Profundo Compartimentado + PbSOM; (g) SOM Profundo Compartimentado + TSOM.

A matriz de confusão, Figura 5.13, apresenta uma concentração na categorização dos rótulos de número 270 e 273, o que totaliza 77,67% das categorizações das amostras. Os resultados dos métodos propostos nesse trabalho melhoraram as taxas de acerto em comparação ao Filtro Bayesiano e também diminuíram o número de categorias com taxa de acerto zeradas (Tabela 5.23). A relação da quantidade de tipos de objetos por quantidade de categorias (Tabela 5.1), decresce a medida que as bases ficam mais complexas tendo uma relação muito baixa para a base de SUN407, com somente 9, em relação a 267, 172 e 47 das outras bases.

Bases de dados	Filtro Bayesiano	Agrupamento Fixo+ PbSOM	Agrupamento Fixo + TSOM	SOM Raso +PbSOM	SOM Raso + TSOM	SOM Profundo Compartimentado +PbSOM	SOM Profundo Compartimentado + TSOM
Quantidade de Categorias com taxa de acerto zerada	356	**127**	185	185	264	153	226
Percentual de Categorias com taxa de acerto zerada	87,47%	**31,20%**	45,45%	45,45%	64,86%	37,59%	55,53%
Quantidade de Categorias com taxa de acerto acima de 50%	5	18	9	8	9	**18**	**18**
Percentual de Categorias com taxa de acerto acima de 50%	1,23%	4,41%	2,21%	1,97%	2,21%	**44,2%**	**44,2%**

Tabela 5.23: Análise de taxas de acertos das categorias da base de dados SUN407.

O método que possui o resultado com a menor quantidade de categorias zeradas é o Agrupamento Fixo + PbSOM com 127 categorias zeradas, o total de 31,20% das categorias. Já os melhores resultados em relação a quantidade de categorias com a taxa de acerto maior que 50% foi para dois métodos do SOM Profundo Compartimentado com 18 categorias totalizando 44,20%, da categorias para os dois métodos (Tabela 5.23).

5.3 Resumo dos Resultados

Em resumo os métodos propostos por este trabalho tiveram desempenho superiores em relação ao método comparado, Filtro Bayesiano, para as bases de dados com muitas categorias e com uma relação de quantidade de objetos por quantidade de categorias menor. Além de atingir resultados próximos ao método comparado para a menor base. A análise da base de dados LabelMe8 mostra que algumas categorias podem ser bem descritas somente com a frequência de tipos de objetos na imagem, o que pode acontecer com muitas categorias das base de dados maiores. Isso abre oportunidades para que mais informações possam enriquecer o modelo de dados das bases como a inclusão da informação dos objetos como textura, dimensão e localização na imagem.

A medida que as bases de dados crescem em quantidade de categorias os resultados do

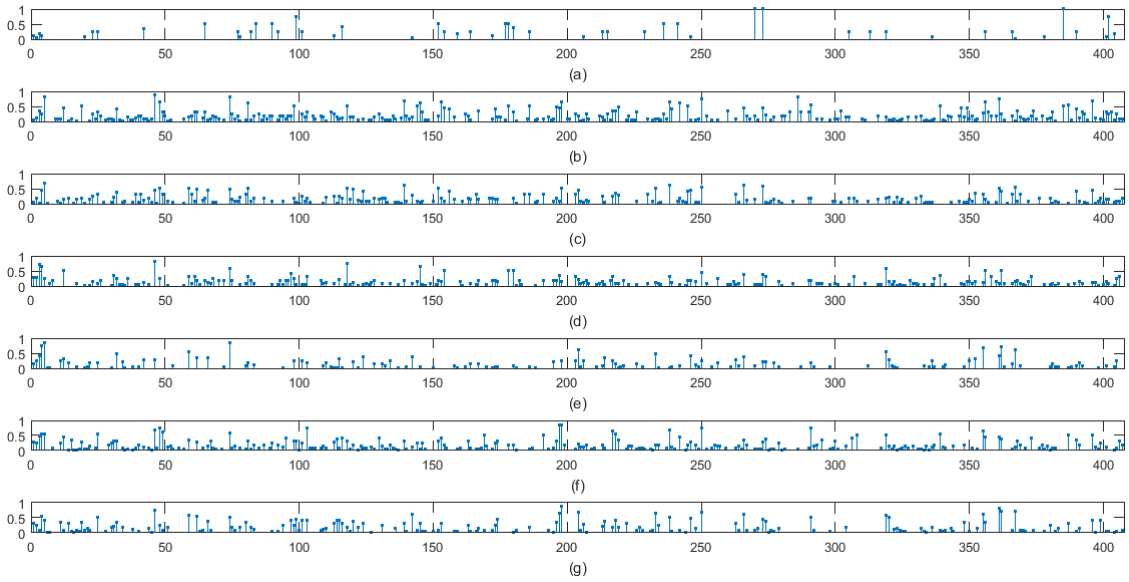

Figura 5.12: Gráfico de taxa de acerto, no intervalo de [0, 1], para cada categoria na SUN407: (a) Filtro Bayesiano; (b) Agrupamento Fixo + PbSOM; (c) Agrupamento Fixo + TSOM; (c) Agrupamento Fixo + TSOM; (d) SOM Raso + PbSOM; (e) SOM Raso + TSOM; (f) SOM Profundo Compartimentado + PbSOM; (g) SOM Profundo Compartimentado + TSOM.

Filtro Bayesiano se degradam rapidamente. Os métodos propostos dissertação comparativamente tiveram uma queda menos acentuada que o Filtro Bayesiano. O problema de Categorização de Lugares baseada em Objetos no mundo real tende a ter muitas variações de tipos de objetos e uma quantidade elevada de categorias. Os modelos propostos buscaram solucionar esse contexto do problema. Os modelos baseados em SOM Profundo Compartimentado tiveram em geral um taxa de acerto um pouco maior do que os métodos do SOM Raso. Isso indica que o processamento em várias camadas é justificável pela melhoria da acurácia. O SOM Profundo Compartimentado se mostra interessante no processo de redução de dimensionalidade das características de foram a contribuir com a melhoria da acurácia pelo categorizador.

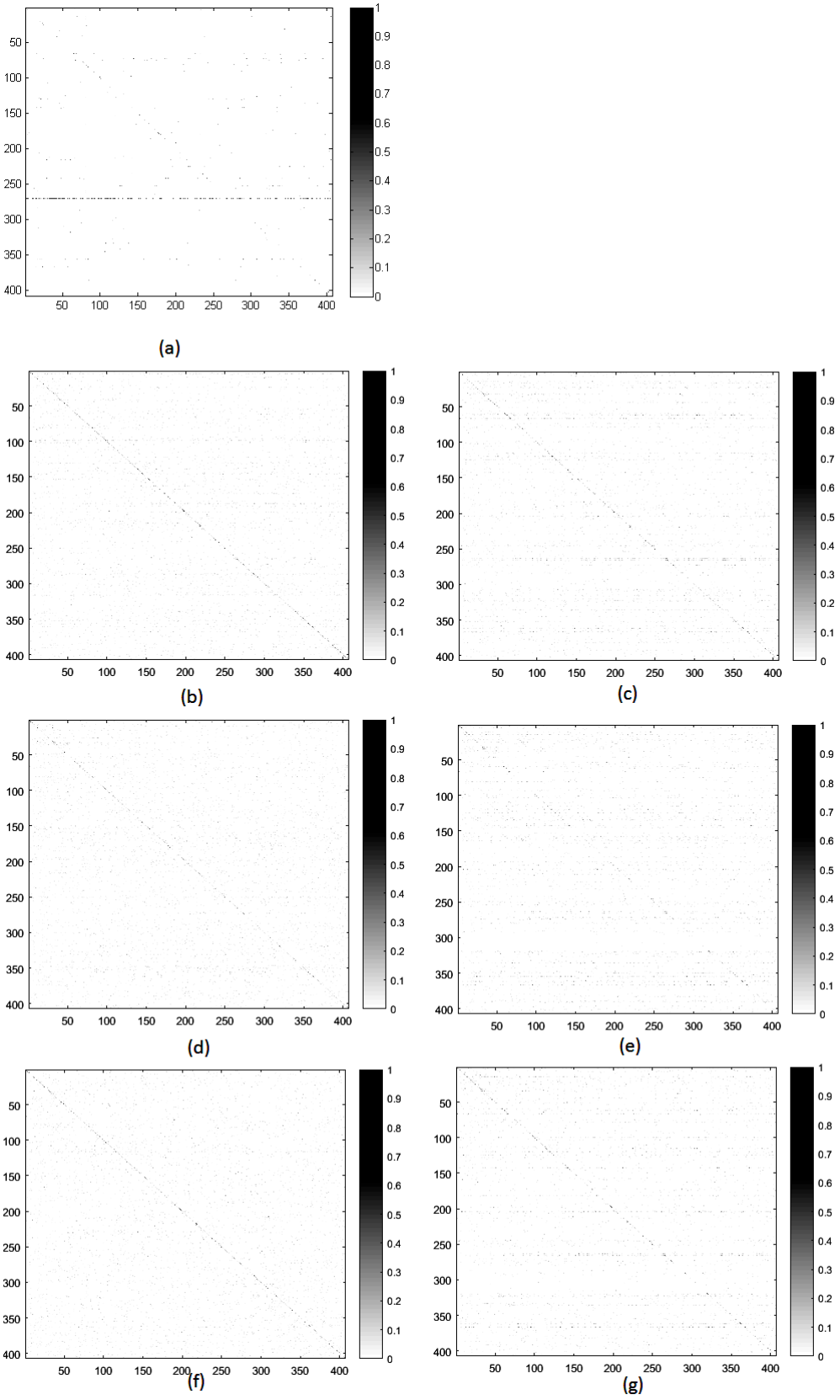

Figura 5.13: Matriz de confusão para SUN407: (a) Filtro Bayesiano; (b) Agrupamento Fixo + PbSOM; (c) Agrupamento Fixo + TSOM; (e) SOM Raso + PbSOM; (e) SOM Raso + TSOM; (f) Profundo Compartimentado + PbSOM; (g) Profundo Compartimentado + TSOM.

6

Conclusão e Perspectivas de Investigação

A pesquisa relatada nesta dissertação apresentou um abordagem para a Categorização de Lugares baseada em Objetos. O foco deste trabalho está nos processos de Redução de Dimensionalidade dos dados representados por vetores de frequência de objetos, e no processo de Categorização dos Lugares propriamente dito. Por meio de experimentos com 4 bases de dados, avaliou-se o desempenho dos métodos propostos neste trabalho com o Filtro Bayesiano: algoritmo existente na literatura para solução do mesmo problema.

Os resultados mostraram que os métodos propostos em comparação ao método da literatura tiveram resultados na mesma ordem de grandeza para a base de dados com 4 categorias. O crescimento da quantidade de categorias nas bases de dados leva a uma mudança, isto é, os resultados para os métodos propostos passam a superar o método da literatura, a taxa de acerto atinge 8 vezes a taxa dos métodos anteriores. O problema de Categorização de Lugares no mundo real possui uma grande quantidade de categorias e uma infinidade de tipos de objetos. Assim, as bases de dados com maiores quantidade de categorias constroem um cenário mais adequado com a realidade do problema. Este trabalho obteve melhorias nos resultados neste contexto, baseado na abordagem em objetos.

As arquiteturas para solução do problema abordado por esta dissertação são baseados em um redução de dimensionalidade através de conceitos de subamostragem e de SOM. A categorização é realizada por PSOM. As arquiteturas de Agrupamento Fixo fazem uma redução de dimensionalidade dos dados de uma forma simples e direta, assim viabilizam que categorizador, pois os modelos de PSOM utilizados para essas bases de dados obtêm matrizes intermediárias no processo de difícil inversão, sendo assim necessário a redução de dimensionalidade dos dados. A redução na quantidade de variáveis também beneficia o tempo de processamento do processo de categorização. As arquiteturas do SOM Raso e SOM Profundo Compartimentado fazem uma Engenharia Automática de Características, selecionando e combinando estas de uma forma não-linear aos atributos dos vetores dos dados. As arquiteturas propostas baseadas em SOM nesta pesquisa demonstraram ser eficazes em realizar uma redução da dimensão dos dados de uma forma autônoma, de maneira a favorecer os melhores resultados do categorizador.

6.1 Principais Contribuições

As principais contribuições desta pesquisa são:

- Solução simples e eficaz para Agrupamento Fixo de Objetos que melhoram as taxas de acerto e o desempenho de processamento para Categorização de Lugares;
- Um novo processo de redução de dimensionalidade de dados para a saída do SOM Raso;
- Um nova arquitetura de Rede Neural Profunda utilizando múltiplas camadas SOM para Engenharia Automática de Características;
- Aplicação de modelos PSOMs, como Categorizadores de Lugares baseado em objetos.

6.2 Limitações do Trabalho

Apesar dos modelos propostos por este trabalho apresentarem resultados promissores para o problema em questão, algumas limitações nos métodos propostos e no modelo de dados podem ser listados:

- Os dados atuais são somente as anotações dos tipos de objetos e sua frequência nas imagens. As categorias que sejam uma mistura de conceitos (como Sala de Jantar são uma categoria que mistura objetos de Cozinha e Sala de Estar). Assim, esses tipos de categorias podem ter dificuldade para agrupadas corretamente. Logo, os dados sobre frequência de objetos existentes nas bases de dados e quantidade de amostras podem não ser discriminantes para que os métodos façam a separação de categorias;
- O processo de Agrupamento Fixo pode distorcer atributos que sejam importantes para a determinação das categorias, fazendo com que o categorizador tenha mais dificuldade em seu treinamento;
- O SOM Profundo compartimentado possui um *pipeline* de SOMs para cada categoria. Para bases de dados com muitas categorias, esse fator pode ser um inconveniente. Esse processo é muito custoso e é possível que algumas das saídas (atributos) não sejam muito relevantes para o categorizador;
- A dimensão do vetor de saída é no mínimo a quantidade de categorias da base de dados, com muitas categorias, esse fato pode limitar o poder da redução de dados. Assim, a dimensão dos dados fica limitada pela quantidade de categorias.

- A quantidade de parâmetros para os modelos SOM Profundo Compartimentado é alta, pois cada uma das camadas tem seus próprios parâmetros. Desta forma, o ajuste de quantidade enorme de parâmetros pode ser custoso.

6.3 Trabalhos Futuros

O campo de pesquisa em Categorização de Lugares é uma área promissora e com muitos desafios interessantes. Esse trabalho em particular deixa algumas oportunidades de trabalhos futuros, como:

- Comparação de modelos de redução com dimensionalidade com outros métodos para o mesmo fim;
- Proposta de métricas de avaliação para as camadas do SOM Profundo Compartimentado;
- Ampliação do domínio dos dados para obtenção de mais informações dos objetos como: textura, dimensão e localização na imagem;
- Desenvolvimento de um SOM Profundo com somente um único *pipeline* do início ao fim para uma redução maior de atributos para bases com muitas categorias;
- Implementação de um Detector de Objetos, terceira etapa da Figura 4.1, para experimentos em um robô navegando em ambiente real;
- Implementação de um algoritmo para construção de uma Mapa Semântico através da Categorização de Lugares para testes em robôs móveis.

Referências

ARAUJO, F. R.; BASSANI, H. F.; ARAUJO, A. F. Learning vector quantization with local adaptive weighting for relevance determination in genome-wide association studies. **International Joint Conference on Neural Networks**, [S.l.], p.1–8, 2013.

ASTUDILLO, C. A.; OOMMEN, B. J. Topology-oriented self-organizing maps: a survey. **Pattern Analysis and Applications**, [S.l.], p.223–248, 2014.

BARRETO, G. A.; ARAUJO, A. F. R. A self-organizing NARX network and its application to prediction of chaotic time series. **Proceedings of the International Joint Conference on Neural Networks**, [S.l.], p.2144–2149, 2001.

BAY, H. et al. Speeded-Up Robust Features (SURF). **Computer Vision and Image Understanding**, [S.l.], p.346–359, 2008.

BELLMAN, R. Adaptive control processes: a guided tour. , [S.l.], 1960.

BENGIO, Y. Learning deep architectures for AI. **Foundations and trends® in Machine Learning**, [S.l.], p.1–127, 2009.

BENGIO, Y. et al. Greedy layer-wise training of deep networks. **Advances in neural information processing systems**, [S.l.], p.153–160, 2007.

BEYER, K. et al. When is “nearest neighbor” meaningful? **International conference on database theory**, [S.l.], p.217–235, 1999.

BILMES, J. A. et al. **A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models**. [S.l.: s.n.], 1998.

BISHOP, C. M.; SVENSéN, M.; WILLIAMS, C. K. I. GTM: the generative topographic mapping. **Neural Computation**, [S.l.], p.215–234, 1998.

BLAKE, C.; MERZ, C. J. **UCI Repository of machine learning databases [http**://www. ics. uci. edu/˜ mlearn/mlrepository. html]. 1998.

BONIN-FONT, F.; ORTIZ, A.; OLIVER, G. Visual Navigation for Mobile Robots: a survey. **Journal of Intelligent and Robotic Systems**, [S.l.], p.263–296, 2008.

CADENA, C. et al. Robust place recognition with stereo sequences. **IEEE Transactions on Robotics**, [S.l.], p.871–885, 2012.

CARPINTEIRO, O. A. S. A hierarchical self-organizing map model for sequence recognition. **Neural Processing Letters**, [S.l.], v.9, p.209–220, 1999.

CELEUX, G.; GOVAERT, G. A classification EM algorithm for clustering and two stochastic versions. **Computational Statistics & Data analysis**, [S.l.], p.315–332, 1992.

CHARALAMPOUS, K. et al. Place categorization through object classification. **IEEE International Conference on Imaging Systems and Techniques**, [S.l.], p.320–324, 2014.

CHARALAMPOUS, K.; GASTERATOS, A. Bio-inspired deep learning model for object recognition. **IEEE International Conference on Imaging Systems and Techniques**, [S.l.], p.51–55, 2013.

CHENG, S.-S.; FU, H.-C.; WANG, H.-M. Model-based clustering by probabilistic self-organizing maps. **IEEE Transactions on Neural Networks**, [S.l.], p.805–826, 2009.

CHOSET, H. et al. **Principles of Robot Motion**: theory, algorithms, and implementations. [S.l.]: MIT Press, 2005.

DEBOECK, G.; KOHONEN, T. **Visual explorations in finance**: with self-organizing maps. [S.l.]: Springer Science & Business Media, 2013.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. **Journal of the Royal Statistical Society: Series B**, [S.l.], p.1–38, 1977.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. **Numerische Mathematik**, [S.l.], p.269–271, 1959.

DITTENBACH, M.; MERKL, D.; RAUBER, A. The Growing Hierarchical Self-Organizing Map. , [S.l.], p.15–19, 2000.

DOERSCH, C.; GUPTA, A.; EFROS, A. A. Mid-level visual element discovery as discriminative mode seeking. **Advances in Neural Information Processing Systems**, [S.l.], p.494–502, 2013.

ERHAN, D. et al. Why does unsupervised pre-training help deep learning? **Journal of Machine Learning Research**, [S.l.], p.625–660, 2010.

EVERINGHAM, M. et al. The Pascal Visual Object Classes Challenge: a retrospective. **International Journal of Computer Vision**, [S.l.], p.98–136, 2015.

FAZL-ERSI, E.; TSOTSOS, J. K. Energy minimization via graph cuts for semantic place labeling. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, [S.l.], p.5947–5952, 2010.

FEI-FEI, L.; FERGUS, R.; PERONA, P. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. **Computer Vision and Image Understanding**, [S.l.], p.59–70, 2007.

FELZENSZWALB, P. et al. Visual object detection with deformable part models. **Communications of the ACM**, [S.l.], p.97–105, 2013.

FELZENSZWALB, P.; MCALLESTER, D.; RAMANAN, D. A discriminatively trained, multiscale, deformable part model. **IEEE Conference on Computer Vision and Pattern Recognition**, [S.l.], p.1–8, 2008.

FISCHLER, M.; BOLLES, R. Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography. **Communications of the ACM**, [S.l.], p.381–395, 1981.

FRALEY, C.; RAFTERY, A. E. Bayesian regularization for normal mixture estimation and model-based clustering. **Journal of Classification**, [S.l.], p.155–181, 2007.

FUKUSHIMA, K. Self-organization of shift-invariant receptive fields. **Neural Networks**, [S.l.], p.791–801, 1999.

GOSER, K. Self-organising maps for intelligent process control. **Workshop on Self-Organizing Maps**, [S.l.], 1997.

GUZEL, M. Autonomous vehicle navigation using vision and mapless strategies: a survey. **Advances in Mechanical Engineering**, [S.l.], p.263–296, 2013.

HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A formal basis for the heuristic determination of minimum cost paths. **IEEE Transactions on Systems Science and Cybernetics**, [S.l.], p.100–107, 1968.

HARTLEY, R.; ZISSERMAN, A. **Multiple View Geometry in Computer Vision**. [S.l.]: Cambridge University Press, 2003.

HE, H.; GRACO, W.; YAO, X. Application of genetic algorithm and k-nearest neighbour method in medical fraud detection. **Asia-Pacific Conference on Simulated Evolution and Learning**, [S.l.], p.74–81, 1998.

HESKES, T. Self-organizing maps, vector quantization, and mixture modeling. **IEEE Transactions on Neural Networks**, [S.l.], p.1299–1305, 2001.

HINNEBURG, A.; AGGARWAL, C. C.; KEIM, D. A. What is the nearest neighbor in high dimensional spaces? **International Conference on Very Large Databases**, [S.l.], 2000.

HINTON, G. E.; OSINDERO, S.; TEH, Y.-W. A fast learning algorithm for deep belief nets. **Neural computation**, [S.l.], p.1527–1554, 2006.

JAIN, A. K.; DUBES, R. C. **Algorithms for clustering data**. [S.l.]: Prentice-Hall, Inc., 1988.

JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data Clustering: a review. **ACM computing surveys**, [S.l.], p.264–323, 1999.

KARPATHY, A. **Notas de Aula do curso de Convolutional Neural Networks for Visual Recognotion de Stanford**. Accessed: 2016-09-27, `http://cs231n.github.io/convolutional-networks/`.

KATSURA, H. et al. A view-based outdoor navigation using object recognition robust to changes of weather and seasons. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, [S.l.], p.2974–2979, 2003.

KOHONEN, T. The self-organizing map. **Proceedings of the IEEE**, [S.l.], p.1464–1480, 1990.

KOHONEN, T.; HARI, R. Where the abstract feature maps of the brain might come from. **Trends in neurosciences**, [S.l.], p.135–139, 1999.

KOHONEN, T.; SCHROEDER, M. R.; HUANG, T. S. (Ed.). **Self-Organizing Maps**. [S.l.]: Springer-Verlag New York, Inc., 2001.

KORN, F. et al. Fast nearest neighbor search in medical image databases. **Proceedings of 22th International Conference on Very Large**, [S.l.], 1998.

KOSTAVELIS, I.; GASTERATOS, A. Learning spatially semantic representations for cognitive robot navigation. **Robotics and Autonomous Systems**, [S.l.], p.1460–1475, 2013.

KOSTAVELIS, I.; GASTERATOS, A. Semantic mapping for mobile robotics tasks: a survey. **Robotics and Autonomous Systems**, [S.l.], p.86–103, 2015.

KOSTAVELIS, I.; NALPANTIDIS, L.; GASTERATOS, A. Collision risk assessment for autonomous robots by offline traversability learning. **Robotics and Autonomous Systems**, [S.l.], p.1367–1376, 2012.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, [S.l.], p.1097–1105, 2012.

LAFFERTY, J. D.; MCCALLUM, A.; PEREIRA, F. C. N. Conditional Random Fields: probabilistic models for segmenting and labeling sequence data. **International Conference on Machine Learning**, [S.l.], p.282–289, 2001.

LAZEBNIK, S.; SCHMID, C.; PONCE, J. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, [S.l.], p.2169–2178, 2006.

LECUN, Y.; BENGIO, Y. Convolutional networks for images, speech, and time series. **The Handbook of Brain Theory and Neural Networks**, [S.l.], p.255–258, 1995.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, [S.l.], p.436–444, 2015.

LI, K.; MENG, M. Q. Indoor scene recognition via probabilistic semantic map. **IEEE International Conference on Automation and Logistics**, [S.l.], p.352–357, 2012.

LINDE, O.; LINDEBERG, T. Object recognition using composed receptive field histograms of higher dimensionality. **Proceedings of International Conference on Pattern Recognition**, [S.l.], p.1–6, 2004.

LÓPEZ-RUBIO, E. Multivariate Student-t self-organizing maps. **Neural Networks**, [S.l.], p.1432–1447, 2009.

LóPEZ-RUBIO, E. Probabilistic Self-organizing Maps for Continuous Data. **IEEE Transactions on Neural Networks**, [S.l.], p.1543–1554, 2010.

LÓPEZ-RUBIO, E. et al. Dynamic topology learning with the probabilistic self-organizing graph. **Neurocomputing**, [S.l.], v.74, n.16, p.2633–2648, 2011.

LÓPEZ-RUBIO, E.; LAZCANO-LOBATO, J. M. Ortiz-de; LÓPEZ-RODRÍGUEZ, D. Probabilistic PCA self-organizing maps. **IEEE Transactions on Neural Networks**, [S.l.], v.20, n.9, p.1474–1489, 2009.

LÓPEZ-RUBIO, E.; LUQUE-BAENA, R. M.; DOMINGUEZ, E. Foreground detection in video sequences with probabilistic self-organizing maps. **International journal of neural systems**, [S.l.], p.225–246, 2011.

LÓPEZ-RUBIO, E.; PALOMO, E. J. Growing hierarchical probabilistic self-organizing graphs. **IEEE Transactions on Neural Networks**, [S.l.], v.22, n.7, p.997–1008, 2011.

LOWE, D. G. Distinctive image features from scale-invariant keypoints.**International journal of computer vision**, [S.l.], p.91–110, 2004.

MADOKORO, H.; UTSUMI, Y.; SATO, K. Scene classification using unsupervised neural networks for mobile robot vision. **Proceedings of Society of Instrument and Control Engineers Annual Conference**, [S.l.], p.1568–1573, 2012.

MAIA, J. E. B.; BARRETO, G. A.; COELHO, A. L. V. Evolving a self-organizing feature map for visual object tracking. **Advances in self-organizing maps**, [S.l.], p.121–130, 2011.

MARC'AURELIO RANZATO, C. P.; CHOPRA, S.; LECUN, Y. Efficient learning of sparse representations with an energy-based model. **Neural Information Processing Systems**, [S.l.], p.1137–1144, 2007.

MARTINETZ, T. M.; BERKOVICH, S. G.; SCHULTEN, K. J. 'Neural-gas' network for vector quantization and its application to time-series prediction. **IEEE Transactions on Neural Networks**, [S.l.], p.558–569, 1993.

MATSUMOTO, Y. et al. View-based approach to robot navigation. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, [S.l.], p.1702–1708, 2000.

MCLACHLAN, G.; KRISHNAN, T. **The EM algorithm and extensions**. [S.l.]: John Wiley & Sons, 2007.

MILIC, L. **Multirate Filtering for Digital Signal Processing**: matlab applications: matlab applications. [S.l.]: IGI Global, 2009.

MORAVEC, H. The Stanford Cart and the CMU Rover. In: **Autonomous Robot Vehicles**. [S.l.]: Springer-Verlag, 1983.

MORENO, S. et al. Robust growing hierarchical self organizing map. , [S.l.], p.341–348, 2005.

MORIOKA, Y.; TOZUKA, T.; YAMAGATA, T. Climate variability in the southern Indian Ocean as revealed by self-organizing maps. **Climate dynamics**, [S.l.], p.1059–1072, 2010.

MOSCHOU, V.; VERVERIDIS, D.; KOTROPOULOS, C. Assessment of self-organizing map variants for clustering with application to redistribution of emotional speech patterns. **Neurocomputing**, [S.l.], p.147–156, 2007.

NATTHARITH, P. **Mobile Robot Navigation Using a Behavioural Control Strategy**. [S.l.]: University of Newcastle upon Tyne, 2011.

NIETO-GRANDA, C. et al. Semantic map partitioning in indoor environments using regional analysis. **IEEE/RSJ International Conference on Intelligent Robots and Systems**, [S.l.], p.1451–1456, 2010.

NILSSON, N. J. **Shakey The Robot**. [S.l.]: AI Center, SRI International, 1984.

NISTÉR, D.; NARODITSKY, O.; BERGEN, J. Visual odometry. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, [S.l.], p.I–652, 2004.

NISTER, D.; STEWENIUS, H. Scalable Recognition with a Vocabulary Tree. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, [S.l.], p.2161–2168, 2006.

OLIVA, A.; TORRALBA, A. Modeling the shape of the scene: a holistic representation of the spatial envelope. **International journal of computer vision**, [S.l.], p.145–175, 2001.

OLIVA, A.; TORRALBA, A. Building the Gist of a Scene: the role of global image features in recognition. **Visual Perception, Progress in Brain Research**, [S.l.], p.73–78, 2006.

OLIVER, A. et al. Using the Kinect As a Navigation Sensor for Mobile Robotics. **Proceedings of Conference on Image and Vision Computing New Zealand**, [S.l.], p.509–514, 2012.

PATTERSON, G.; HAYS, J. Sun attribute database: discovering, annotating, and recognizing scene attributes. **IEEE Conference on Computer Vision and Pattern Recognition**, [S.l.], p.2751–2758, 2012.

PEEL, D.; MCLACHLAN, G. J. Robust mixture modelling using the t distribution. **Statistics and Computing**, [S.l.], p.339–348, 2000.

PRONOBIS, A. **Semantic Mapping with Mobile Robots**. 2011. Tese (Doutorado em Ciência da Computação) — KTH Royal Institute of Technology.

PRONOBIS, A. et al. A Discriminative Approach to Robust Visual Place Recognition. **IEEE International Workshop on Intelligent Robots and Systems**, [S.l.], p.3829–3836, 2006.

PRONOBIS, A.; JENSFELT, P. Large-scale Semantic Mapping and Reasoning with Heterogeneous Modalities. **IEEE International Conference on Robotics and Automation**, [S.l.], p.3515–3522, 2012.

QUATTONI, A.; TORRALBA, A. Recognizing indoor scenes. **IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, [S.l.], p.413–420, 2009.

RABINER, L. R.; JUANG, B.-H. An introduction to hidden Markov models. **IEEE ASSP Magazine**, [S.l.], p.4–16, 1986.

RAMOS, J. Using tf-idf to determine word relevance in document queries. **Proceedings of Instructional Conference on Machine Learning**, [S.l.], p.177–182, 2003.

RANGANATHAN, A.; DELLAERT, F. Semantic Modeling of Places using Objects. **Robotics: Science and Systems Conference**, [S.l.], p.27–30, 2007.

ROBBINS, H.; MONRO, S. A stochastic approximation method. **Annals of Mathematical Statistics**, [S.l.], p.400–407, 1951.

ROGERS III, J. G.; CHRISTENSEN, H. et al. A conditional random field model for place and object classification. **IEEE International Conference on Robotics and Automation**, [S.l.], p.1766–1772, 2012.

RUSSELL, B. C. et al. LabelMe: a database and web-based tool for image annotation. **International journal of computer vision**, [S.l.], p.157–173, 2008.

SÁNCHEZ, J. et al. Image classification with the fisher vector: theory and practice. **International journal of computer vision**, [S.l.], p.222–245, 2013.

SCHMIDT, M. **UGM**: matlab code for undirected graphical models. 2012.

SENGUPTA, S. et al. Urban 3D semantic modelling using stereo vision. **IEEE International Conference on Robotics and Automation**, [S.l.], p.580–585, 2013.

SHI, J.; MALIK, J. Normalized Cuts and Image Segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, [S.l.], p.888–905, 2000.

SHOHAM, S. Robust clustering by deterministic agglomeration EM of mixtures of multivariate t-distributions. **Pattern Recognition**, [S.l.], p.1127–1142, 2002.

SIVIC, J.; ZISSERMAN, A. Efficient visual content retrieval and mining in videos. **Advances in Multimedia Information Processing**, [S.l.], p.471–478, 2005.

SUDDERTH, E. B. et al. Describing Visual Scenes Using Transformed Objects and Parts. **International Journal of Computer Vision**, [S.l.], p.291–330, 2008.

SUM, J. et al. Yet another algorithm which can generate topography map. **IEEE Transactions on Neural Networks**, [S.l.], p.1204–1207, 1997.

SYMONS, M. J. Clustering criteria and multivariate normal mixtures. **Biometrics**, [S.l.], p.35–43, 1981.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition, Third Edition**. [S.l.]: Academic Press, Inc., 2006.

TIPPING, M. E. Sparse Bayesian learning and the relevance vector machine. **Journal of machine learning research**, [S.l.], p.211–244, 2001.

TITTERINGTON, D.; SMITH, A.; MAKOV. **Statistical analysis of finite mixture models**. [S.l.]: Wiley, Chichester, UK, 1985.

TORRALBA, A. How many pixels make an image? **Visual neuroscience**, [S.l.], p.123–131, 2009.

TRULLIER, O. et al. Biologically-based Artificial Navigation Systems: review and prospects. **Progress in Neurobiology**, [S.l.], p.483–544, 1997.

UEDA, N.; NAKANO, R. Deterministic annealing EM algorithm. **Neural Networks**, [S.l.], p.271–282, 1998.

ULLAH, M. M. et al. **Towards robust place recognition for robot localization.**, [S.l.], p.530–537, 2008.

VAN HULLE, M. M. Joint entropy maximization in kernel-based topographic maps. **Neural Computation**, [S.l.], v.14, n.8, p.1887–1906, 2002.

VAN HULLE, M. M. Maximum likelihood topographic map formation. **Neural Computation**, [S.l.], v.17, n.3, p.503–513, 2005.

VASUDEVAN, S.; SIEGWART, R. Bayesian Space Conceptualization and Place Classification for Semantic Maps in Mobile Robotics. **Robotics and Autonomous Systems**, [S.l.], p.522–537, 2008.

VELLIDO, A. Missing data imputation through GTM as a mixture of t-distributions. **Neural Networks**, [S.l.], v.19, n.10, p.1624–1635, 2006.

VERBEEK, J. J.; VLASSIS, N.; KRÖSE, B. J. Self-organizing mixture models. **Neurocomputing**, [S.l.], v.63, p.99–123, 2005.

VISWANATHAN, P. et al. Automated Spatial-Semantic Modeling with Applications to Place Labeling and Informed Search. **Proceedings of the Canadian Conference on Computer and Robot Vision**, [S.l.], p.284–291, 2009.

VISWANATHAN, P. et al. Automated place classification using object detection. **Proceedings of the Canadian Conference on Computer and Robot Vision**, [S.l.], p.324–330, 2010.

WALTER, J.; SCHULTEN, K. J. et al. Implementation of self-organizing neural networks for visuo-motor control of an industrial robot. **IEEE Transactions on Neural Networks**, [S.l.], p.86–96, 1993.

WANG, M.-L.; LIN, H.-Y. An extended-HCT semantic description for visual place recognition. **International Journal of Robotics Research**, [S.l.], p.1403–1420, 2011.

WANG, Y.; HAMME, H. V. Gaussian Selection Using Self-Organizing Map for Automatic Speech Recognition. **International Workshop on Self-Organizing Maps**, [S.l.], p.218–227, 2011.

WOŹNIAK, M.; GRAÑA, M.; CORCHADO, E. A survey of multiple classifier systems as hybrid systems. **Information Fusion**, [S.l.], p.3–17, 2014.

XIAO, J. et al. Sun database: large-scale scene recognition from abbey to zoo. **IEEE conference on Computer vision and pattern recognition**, [S.l.], p.3485–3492, 2010.

YANG, J. et al. Evaluating Bag-of-visual-words Representations in Scene Classification. **Proceedings of the International Workshop on Multimedia Information Retrieval**, [S.l.], p.197–206, 2007.

YIN, H.; ALLINSON, N. M. Self-organizing mixture networks for probability density estimation. **IEEE Transactions on Neural Networks**, [S.l.], v.12, n.2, p.405–411, 2001.

ZENDER, H. et al. Conceptual Spatial Representations for Indoor Mobile Robots. **Robotics and Autonomous Systems**, [S.l.], p.493–502, 2008.

ZHANG, J. et al. Local Features and Kernels for Classification of Texture and Object Categories: a comprehensive study. **International Journal of Computer Vision**, [S.l.], p.213–238, 2007.

ZHOU, B. et al. Learning deep features for scene recognition using places database. **Advances in Neural Information Processing Systems**, [S.l.], p.487–495, 2014.