



Universidade Católica Dom Bosco
Curso de Bacharelado em Engenharia de Computação

**Modelos Ocultos de Markov Aplicados na
Identificação de Comportamento de Animais**

Jonathan de Andrade Silva
Wesley Nunes Gonçalves

Prof. Orientador: Dr. Hemerson Pistori

Relatório final apresentado à Coordenação do Curso de Bacharelado em Engenharia de Computação da Universidade Católica Dom Bosco como parte dos requisitos para a obtenção do título de Bacharel em Engenharia de Computação.

UCDB - Campo Grande - MS - Novembro/2007

às nossas famílias.

Agradecimentos

Primeiramente agradecemos a Deus pela oportunidade e saúde que nos deu durante nossas vidas. As nossas famílias pela compreensão, incentivo e amor que manifestaram ao longo destes 5 anos de curso. Obrigado por sempre confiarem em nós e principalmente pelos valores e conselhos que nos passaram. Agradecer a chellis por me liberar para fazer os trabalhos e estudar para as provas nos nossos dias. Gostaria também de agradecer ao meu pai, Zilto Salviano da Silva, por me educar e por passar pra mim tudo o que conhecia sobre a vida e por ter existido em minha vida.

E nossos agradecimentos especiais:

Ao nosso orientador Prof. Hemerson Pistori que participou ativamente na nossa formação acadêmica, por nos ensinar a encarar os problemas da vida de uma outra maneira, por nos apoiar na criação da nossa empresa, pelo incentivo a pesquisa e nos mostrar uma outra visão a respeito do curso, aplicando os conceitos aprendidos dentro da sala de aula.

Aos professores Albert Schiaveto de Souza e Paula Helena Santa Rita que nos ajudaram no projeto através da preparação do ambiente com estruturas para a captura de imagens dos animais e pelas conversas para tirar diversas dúvidas a respeito dos projetos. Ao Joe Colombo pelo apoio no desenvolvimento do nosso projeto e confiança depositada na empresa Tec-Vision. A Valguima Odakura pela ajuda na publicação de artigos através de correções, traduções e por participar ativamente, com idéias, na implementação de alguns algoritmos, além das conversas sobre diversos algoritmos de segmentação e rastreamento.

Aos professores do curso de engenharia de computação da universidade católica Dom Bosco pelo convívio e ensinamento durante todo o curso. Ao professor Gonda pela ajuda em questões relacionadas ao futuro da nossa profissão e pela eficiência durante a coordenação de curso, sempre respondendo as questões com grande agilidade.

Ao menino Abela (vulgo Bruno Brandoli Machado) pelas musics sagradas, pelas "malas" do projeto que sempre tinha saco e criatividade pra fazer, nesse mundo só menino Abela pode fazer! Pelos sucos na faixaaaa! Por fazer

as perguntas, em vários embromention, lá na Europa, apesar de ficar chorando o dia inteiro! Pelos vários rangos que comemos juntos lá no gaúcho. Além de tudo pelos estudos que realizamos juntos, pelas "conferências", pelos softwares comerciais pra tirar aquela graninha pra viajar. É isso aí parga Abela, Valeu!! Abraços!!! Mais Nada!!!

A galera do GPEC, Breno com o projeto Larvic, Toledo com o k-curvatura, Lia por sempre estar pronta pra ajudar quando pedimos ajuda, Jéssica, por cobrar as documentações (ainda tenho q fazer algumas, foi mal), Renato, "o puts" pelos Futs, Kleber, por abandonar os pargas na viagem, pelas caronas no calor e ainda fechar as janelas e largar a gente sempre mais longe do lugar que íamos parar. Principalmente pelas conversas sobre HMM e pelas suas risadinhas discretas e zoeiras no GPEC. Paragua, pela agilidade nas explicações no PCA (no máximo 5 minutos), atropelando as palavras. E pelas zoeiras descontraídas junto com seu parceiro Kleber (Batman & Robin). Vinimax, pela ajuda em algumas implementações de alguns segmentadores e por sempre estar antenado no mundo dos games e acessórios para jogos. A Galera da nossa turma pelas confraternizações e zoeiras dentro da sala. E por sempre arrumar um jeito de adiar a prova ou trabalho no último dia de desespero, ligando para os professores, dando um jeito é o que importa. Enfim, a todos que nos ajudaram durante esse projeto e os 5 anos de curso.

Resumo

Neste trabalho foi avaliada a aplicação dos modelos ocultos de Markov (*hidden Markov models* - HMM) no reconhecimento dos comportamentos de camundongos e serpentes. O estudo do comportamento animal é importante em várias situações, como a avaliação de novos fármacos e a estimativa da degradação do meio ambiente. Com a análise do comportamento animal é possível avaliar diferentes fármacos antes da utilização em seres humanos. A identificação comportamental de animais é realizada na maioria das vezes de forma manual, com o pesquisador observando o animal e registrando a ocorrência de algum padrão de comportamento considerado importante.

A modelagem dos comportamentos para a automatização dos experimentos científicos é baseada em algoritmos de visão computacional. Para a representação dos comportamentos foram aplicados diferentes algoritmos de extração de atributos, como os momentos de imagens, k-curvatura, cadeia de códigos e descritores de Fourier. Devido à alta dimensão do vetor de atributos extraídos, foi aplicado o algoritmo de análise de componentes principais (PCA) para a seleção de atributos. Em seguida, para os HMMs discretos, foi aplicado um algoritmo de discretização baseado em quantização vetorial (*Vector Quantization-LBG*).

Os experimentos foram realizados sobre a comparação entre diferentes configurações dos HMMs que representam os comportamentos de camundongos e serpentes. Para treinar os modelos, um conjunto de 326 vídeos referentes aos comportamentos dos camundongos (exploração vertical, horizontal e auto-limpeza) e das serpentes (ataque e não ataque) foi utilizado. Além disso, a distribuição dos atributos ou observações foi modelada através de uma distribuição discreta, contínua com uma Gaussiana e com mistura de Gaussianas. Os resultados mostraram que os HMMs com uma distribuição de mistura de Gaussianas obtêm a maior taxa de classificação correta de 96%. O HMM com distribuição modelada através de uma Gaussiana obtêm uma taxa de classificação correta de 94% e os HMMs discretos uma taxa de 87%. O tempo de execução mostrou que os modelos com mistura de Gaussianas possuem um alto custo de treinamento, enquanto que, os modelos

com distribuição Gaussiana e discreta apresentam um tempo de treinamento menor, com uma grande vantagem para os modelos discretos. Com relação ao tempo de classificação, todos os modelos são possivelmente aplicáveis em tempo real, pois nesse trabalho não foi computado o tempo da segmentação, extração e seleção de atributos das seqüências de imagens.

Abstract

In this work the application of the hidden Markov models - HMM in the recognition of the behaviors of mice and snakes was evaluated. The study of the animal behavior is important in several situations, as the evaluation of new drugs and the estimation of the degradation of the environment. With the analysis of the animal behavior is possible evaluate different drugs before use in humans. The identification of animals behavior is carried out usually manual form, with the researcher observing the animal and recording the occurrence of some important pattern of behavior.

The behaviors modeling for the experiments scientific automatization is based in computer vision algorithms. Feature extraction algorithms, as the moments of images, k-curvature, chain codes and Fourier descriptors, were applied for the behaviors representation. The principal components analysis (PCA) algorithm for the features selection was applied due to the high dimension of the feature vector extracted. A discretization algorithm based in Vector Quantization - LBG was applied for the discrete HMMs.

The experiments were carried on the comparison between different configurations of the HMMs that represent the behaviors of mice and snakes. To train the models, a collection of 326 videos regarding the behaviors of the mice (horizontal, vertical exploitation and self-cleaning) and snakes (attack and non-attack) was used. Moreover, the feature or observation distribution was modeled through discrete, continuous with a Gaussian and with Gaussian mixture. The results showed that the HMMs with a distribution of Gaussian mixture get the highest correct classification rate of 96%. The HMM with distribution modeled through a Gaussian get a correct classification rate of 94% and the discrete HMMs a rate of 87%. The execution time showed that the models with Gaussian mixture possess a high cost of training, whereas, the models with a Gaussian and discrete present a smaller training time, with a great advantage for the discrete models. Regarding the classification time, all of the models are potentially applicable in real time, because the segmentation, extraction and feature selection time in the images sequences was not computed in this work.

Conteúdo

| | | |
|----------|--|-----------|
| 1 | Introdução | 13 |
| 2 | Trabalhos Correlatos | 17 |
| 3 | Extração de Atributos | 19 |
| 3.1 | Momentos de Imagens | 19 |
| 3.2 | K-Curvatura | 20 |
| 3.3 | Cadeias de Códigos | 22 |
| 3.4 | Descritores de Fourier | 24 |
| 4 | Modelos Ocultos de Markov | 25 |
| 4.1 | Os problemas básicos dos Modelos Ocultos de Markov | 28 |
| 4.1.1 | Problema de Avaliação | 28 |
| 4.1.2 | Problema de Decodificação | 32 |
| 4.1.3 | Problema da Aprendizagem | 35 |
| 4.2 | Modelos Ocultos de Markov com Observação Contínua | 36 |
| 5 | Banco de Imagens | 38 |
| 6 | Desenvolvimento | 41 |
| 6.1 | Segmentação dos Vídeos | 41 |
| 6.2 | Algoritmo para a inicialização dos HMMs | 42 |
| 6.3 | Algoritmo de Discretização | 44 |
| 7 | Experimentos e Resultados | 47 |
| 8 | Conclusão e Trabalhos Futuros | 56 |
| A | Fundamentação Teórica | 58 |
| A.1 | Variáveis Aleatórias | 58 |
| A.2 | Probabilidade Conjunta e Condicional | 59 |
| A.3 | Independência Estatística | 59 |

| | |
|---------------------------------------|-----------|
| A.4 Teorema de Bayes | 60 |
| A.5 Distribuição Gaussiana | 61 |
| A.5.1 Mistura de Gaussianas | 61 |
| B Artigos Publicados | 63 |
| Referências Bibliográficas | 65 |

Lista de Figuras

| | | |
|-----|---|----|
| 3.1 | Aplicação dos momentos de imagem. | 21 |
| 3.2 | Passos realizados para a extração de atributos com o algoritmo de k-curvatura. | 22 |
| 3.3 | Histograma de ângulos extraídos pelo algoritmo de k-curvatura. | 22 |
| 3.4 | Primitivas utilizadas nas cadeias de códigos. | 23 |
| 3.5 | Exemplo da aplicação da cadeia de códigos em um objeto. | 23 |
| 3.6 | Aplicação dos descritores de Fourier para diferentes valores de K. | 24 |
| 4.1 | Estados do exemplo de reconhecimento do comportamento de elevação vertical. | 26 |
| 4.2 | Símbolos do exemplo de reconhecimento do comportamento de elevação vertical. | 28 |
| 4.3 | Autômato representando o HMM do exemplo. | 29 |
| 4.4 | Execução da variável α para uma observação $\{O_1 = v_1, O_2 = v_1, O_3 = v_2\}$ através de uma treliça. | 32 |
| 4.5 | Execução da variável α para uma observação $\{O_1 = v_1, O_2 = v_1, O_3 = v_1\}$ através de uma treliça. | 33 |
| 5.1 | Comportamento de bote realizado por uma serpente. | 39 |
| 5.2 | Comportamento de exploração vertical. | 39 |
| 5.3 | Comportamento de exploração horizontal. | 40 |
| 5.4 | Comportamento de auto-limpeza. | 40 |
| 6.1 | Exemplo de segmentação de imagens com camundongos. | 42 |
| 6.2 | Processo de segmentação da serpente. | 43 |
| 6.3 | Exemplo de HMM inicializado uniformemente. | 44 |
| 6.4 | Iteração do algoritmo LBG. | 46 |
| 7.1 | Etapas do processo de execução dos experimentos. | 50 |
| 7.2 | Resultados para os HMMs discretos em relação ao número de estados internos. | 50 |

| | | |
|-----|--|----|
| 7.3 | Resultados para os HMMs discretos em relação ao número de núcleos na discretização. | 51 |
| 7.4 | Resultados para os HMMs com uma Gaussiana em relação ao número de estados internos. | 52 |
| 7.5 | Resultados para os HMMs com mistura de Gaussiana em relação ao número de estados internos. | 52 |
| 7.6 | Resultados para os HMMs com mistura de Gaussiana em relação ao número de Gaussianas. | 53 |
| 7.7 | Resultados de comparação entre os tipos de HMMs. | 54 |

Lista de Tabelas

| | | |
|-----|---|----|
| 4.1 | Matriz com as probabilidades de transições entre os estados. . . | 27 |
| 4.2 | Matriz com as probabilidades de emissão de símbolos de cada estado. | 28 |
| 7.1 | Classificação dos comportamentos individuais dos modelos . . | 54 |
| 7.2 | Matriz de Confusão dos comportamentos de camundongos. . . | 55 |
| 7.3 | Matriz de Confusão dos comportamentos de serpentes. | 55 |
| A.1 | Tabela de alunos matriculados em dois cursos. | 59 |

Capítulo 1

Introdução

O estudo científico do comportamento animal é extremamente importante em diversas áreas de conhecimento. Esses estudos são realizados com diferentes animais para modelar o comportamento humano [FT04, CL06] em diversas situações, como a avaliação de novos fármacos e a estimativa da degradação do meio ambiente. Com esses estudos, é possível avaliar diferentes fármacos antes de sua utilização em seres humanos através da associação de comportamentos correspondentes aos efeitos causados pelo fármaco. Por exemplo, um comportamento caracterizado através da elevação das patas dianteiras de um camundongo pode corresponder à ansiedade do animal naquele instante. Para diminuir a ansiedade, o fármaco deve, conseqüentemente, diminuir a ocorrência do comportamento de elevação das patas de um camundongo após a sua aplicação neste animal.

Além disso, o estudo do comportamento animal pode fornecer indícios da degradação do meio ambiente, visto que, os animais respondem adequadamente a estímulos proporcionados pelo habitat natural, buscando aumentar suas chances de sobrevivência e reprodução. Através desses indícios é possível proporcionar apoio para monitoramentos ambientais, responsáveis por grande parte da conservação de espécies ameaçadas de extinção. Devido a esses motivos, houve um aumento significativo em pesquisas relacionadas com o comportamento animal, com muitos grupos de pesquisa buscando métodos mais ágeis e consistentes para avaliar o comportamento de diferentes animais.

A identificação comportamental de animais é realizada na maioria das vezes de forma manual, com o pesquisador observando o animal e registrando a ocorrência de algum padrão de comportamento considerado importante. Essa observação pode ser realizada com um investimento relativamente baixo, mas pode ter sua precisão comprometida com a fadiga ou distração do observador. Quando há mais de um observador, pode ocorrer diferença entre classificações

para o mesmo comportamento apresentado pelo animal. Seguindo o exemplo citado acima do comportamento da elevação das patas dianteiras, um observador pode registrar esse comportamento quando apenas uma das patas esteja levantada, enquanto outro observador de outro instituto pode considerar esse comportamento apenas quando as duas patas estejam levantadas, havendo assim uma diferença entre os resultados do mesmo efeito apresentado pelo fármaco devido à subjetividade de observadores. Outro problema referente à observação manual, em alguns casos, é a demanda de um grande período de tempo, exigindo um trabalho exaustivo do observador, aumentando as chances de erros.

Em contrapartida, a automatização da observação do comportamento animal é particularmente apropriada quando há necessidade de medidas exatas, como a distância e a velocidade, em que o observador humano é incapaz de estimar adequadamente [STBN01]. Além disso, é possível registrar os comportamentos de forma confiável e consistente durante longos períodos de tempo, pois as técnicas aplicadas no reconhecimento trabalham sempre da mesma maneira. Com a automatização é possível armazenar imagens dos comportamentos, permitindo a reprodução e a análise de experimentos em diversos dias. Os comportamentos, na observação automatizada, consistem em posturas e mudança de posturas no tempo. Devido a essas mudanças, identificar os comportamentos é considerado uma tarefa complexa e um dos fatores que limitam o desenvolvimento de ferramentas automáticas.

Os modelos ocultos de Markov buscam modelar a mudança entre estados de um sistema. Os estados, neste trabalho, representam as posturas apresentadas pelos animais e os comportamentos são as mudanças entre esses estados modelados através de probabilidades. Esses estados não são diretamente observáveis através das imagens capturadas, por exemplo, não é possível saber diretamente em uma imagem se o camundongo está ou não com a pata dianteira direita elevada. Com isso, é necessário um segundo processo que modela as observações ou atributos extraídos das imagens para cada estado. Através desses modelos é possível reconhecer automaticamente os comportamentos relevantes para a análise em determinados experimentos. Assim, a partir de um conjunto de treinamento etiquetado manualmente, de forma supervisionada, o sistema reconhece comportamentos complexos e com pequenos ajustes, pode ser utilizado para a identificação de comportamentos de diferentes animais. Inicialmente, o sistema foi desenvolvido e avaliado em camundongos e serpentes.

As serpentes foram sugeridas como um dos principais grupos para avaliar hipóteses ecológicas e evolucionárias [RB05], além disso, esse grupo é um interessante formador de venenos. O veneno produzido é um importante componente no desenvolvimento de novos fármacos para seres humanos, tais

como, fármacos para controle da hipertensão, analgésicos, anticoagulantes, entre outros. A identificação de comportamentos em experimentos científicos pode auxiliar a criação de cativeiros mais adequados ao habitat das serpentes, pois na natureza a identificação de comportamentos e a manipulação de serpentes podem ser muito complexas.

Outros importantes animais utilizados em pesquisas na área da saúde são os camundongos. Esses animais foram adotados em muitas linhas de pesquisas por seu mecanismo fisiológico possuir diversas semelhanças aos dos seres humanos [CL06]. A contribuição desses animais é extremamente importante na avaliação de novos fármacos. Atualmente, diversos experimentos, como o campo aberto [Eil03], labirinto em cruz [BS02] e labirinto aquático de Morris [Mor84], são realizados com diversas raças de camundongos com o intuito de disponibilizar a sociedade científica da área da saúde métodos-padrão de avaliação de comportamentos e avaliação de drogas.

O objetivo principal desse projeto é modelar alguns comportamentos de camundongos e serpentes. Cada comportamento é representado por um modelo oculto de Markov e utilizado para classificar um conjunto de quadros consecutivos capturados por uma *webcam* em um dos possíveis comportamentos. Para a identificação dos comportamentos foram avaliados três tipos de modelos ocultos de Markov caracterizados pelas distribuições utilizadas na modelagem das observações. O primeiro tipo considera uma distribuição discreta para modelar as observações. Em casos onde essa observação é contínua, é necessária a utilização de algoritmos para realizar a discretização. Uma distribuição Gaussiana é utilizada para modelar as observações do segundo tipo de modelo oculto de Markov. É interessante a utilização de uma modelagem contínua, pois em algumas situações, no processo de discretização pode ocorrer perda de informações. O último tipo utiliza uma mistura de Gaussianas para a modelagem das observações contínuas. Uma biblioteca chamada Jahmm¹ foi utilizada para avaliar os modelos ocultos de Markov. Nesta biblioteca estão implementados os principais algoritmos para utilização dos modelos ocultos de Markov.

Para a utilização dos modelos ocultos de Markov é necessária à extração de atributos das imagens, assim, foram utilizados alguns extratores de atributos baseados no contorno, como os descritores de Fourier e as cadeias de códigos. Além disso, um banco de comportamentos constituído de 326 vídeos de comportamentos de camundongos e serpentes foi desenvolvido e disponibilizado. Esse banco é utilizado como base de treinamento para os modelos. Por fim, todos os módulos desenvolvidos foram integrados ao projeto Topolino [MdASG⁺06], que é um sistema de apoio à análise automática

¹<http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/>

de comportamento animal em laboratório com o objetivo de desenvolver um software livre capaz de processar as imagens adquiridas através de um dispositivo de captura de baixo custo e extrair automaticamente as informações relevantes para a análise de determinados comportamentos. Os resultados dos experimentos apresentam uma taxa de classificação correta máxima de 96% através dos HMMs com modelos de mistura de Gaussianas.

Este trabalho está organizado em oito capítulos e dois apêndices. O Capítulo 2 apresenta os principais trabalhos correlatos relacionando os modelos ocultos de Markov e reconhecimento de padrões. Os atributos extraídos de cada imagem que alimentarão os modelos ocultos de Markov são demonstrados no Capítulo 3. No próximo capítulo, é descrita toda a teoria relacionada com os modelos ocultos de Markov, desde os seus componentes à resolução de alguns problemas relacionados. No Capítulo 5 é demonstrado com detalhes como foi realizada a construção de um banco de imagens de comportamentos de serpentes e camundongos. No Capítulo 6 são discutidos os procedimentos necessários para o treinamento e a classificação utilizando os modelos ocultos de Markov. No Capítulo 7 são apresentados os experimentos realizados e os resultados obtidos na avaliação dos modelos ocultos de Markov na identificação do comportamento de camundongos e serpentes. Por fim, no Capítulo 8 são apresentados a conclusão obtida desse projeto e os trabalhos futuros. No apêndice A é apresentada uma fundamentação teórica de conceitos para o entendimento desse trabalho. O apêndice B apresenta os artigos publicados durante o desenvolvimento desse trabalho.

Capítulo 2

Trabalhos Correlatos

Os modelos ocultos de Markov (*hidden Markov models* - HMM) têm sido amplamente utilizados em diversas áreas, principalmente em sistemas para o reconhecimento de voz [Rab90] e textos manuscritos [HBT96]. Em [HNPS94] o HMM é aplicado ao reconhecimento de objetos em imagens. O HMM, em conjunto com as características do contorno adquiridas dos objetos, foram utilizados na classificação de quatro diferentes tipos de objetos. Para cada tipo de objeto, um HMM é treinado com um conjunto de cinquenta imagens. O sistema foi avaliado com dez imagens para cada tipo de objeto, a classificação resultou em uma taxa de acerto de 75%.

Starner e Pentland [SP95] descrevem um sistema para reconhecimento de sentenças da língua americana de sinais utilizando HMM. A taxa de reconhecimento é de 99.2% para diferentes palavras. No entanto, o conjunto de característica mostrou-se limitado, pois o sistema é treinado para esperar certos gestos em certas posições espaciais na imagem devido à utilização das posições não normalizadas das mãos. Para corrigir esse problema Starner e Pentland [SP95] sugerem a utilização de posições normalizadas no vetor de características, o que pode tornar o sistema mais robusto.

Uma nova técnica para o reconhecimento de textos é apresentada em [ALT95]. As características são extraídas de uma imagem em tons de cinza e um HMM é utilizado para representar cada caractere. Durante o reconhecimento, a mais provável combinação de modelos é encontrada para cada palavra através da técnica de programação dinâmica.

Em [NH98] é descrito um HMM para reconhecimento e detecção de faces. A imagem contendo a face é dividida em cinco blocos (cabelo, testa, olhos, nariz e boca), onde cada bloco é representado como um estado no HMM. Os vetores de características são obtidos de cada bloco utilizando os coeficientes da transformada de Karhunen-Loeve. A detecção é feita calculando-se a probabilidade dos dados de entrada com um dado modelo de face. Para

detectar falsos alarmes, apenas probabilidades maiores que um limiar são reconhecidos como faces.

Os modelos ocultos de Markov são bastante utilizados para descrever uma seqüência de padrões, como os comportamentos. Em [FB03] um sistema para classificação de comportamentos de animais é apresentado. Esse sistema usa uma combinação de HMM e *K-Nearest Neighbors* (k-NN) para o treinamento dos movimentos. O sistema foi avaliado em diversas trajetórias de abelhas extraídas de uma seqüência de vídeo de 15 minutos. Os movimentos foram classificados manualmente e pelo sistema. O sistema desenvolvido foi capaz de classificar movimentos com uma exatidão de 81,5%.

O reconhecimento de movimentos complexos de objetos utilizando HMM é apresentado em [BKS05]. O modelo é construído com base no algoritmo de análise de componentes principais (PCA). Os estados do HMM são representados por misturas de Gaussianas. Os experimentos são executados em duas séries de dados. Uma série é composta de imagens contendo movimentos da língua australiana de sinais (ASL) com 207 trajetórias que descrevem três palavras. A outra série contém 108 trajetórias de atletas executando esportes. Os resultados chegaram a uma taxa de acerto de 90%.

Os diversos sistemas que fazem o uso de modelos ocultos de Markov para o reconhecimento de comportamentos apresentam bons resultados comparados com outras técnicas que não modelam a seqüência de posturas. Entretanto, a taxa de reconhecimento depende muito do conjunto de características visuais utilizadas e do número de estados do modelo. Em [MS04] é realizada a análise de diferentes características no reconhecimento de gestos visuais produzidos pelas mãos. Os resultados obtidos mostram uma alta variação na taxa de reconhecimento, devido à combinação de diferentes características. As taxas de reconhecimento obtiveram uma variação de 50% até 97%. O melhor desempenho foi obtido pelo uso da magnitude e orientação em coordenadas polares e dez estados no modelo.

Além do HMM, outros métodos foram utilizados para o reconhecimento de comportamentos. Em [RvLM⁺98] é realizada a classificação do comportamento de um camundongo por uma rede neural. Foram classificados sete comportamentos como por exemplo: caminhar, sentar e elevar-se. Após o treinamento, a rede classificou corretamente 84,85% dos comportamentos. Segundo Rousseau [RvLM⁺98] essa porcentagem de acerto é comparável à de um observador humano.

Capítulo 3

Extração de Atributos

As imagens são constituídas de diversos objetos pertencentes ao plano de fundo e ao plano de interesse. Os objetos do plano de interesse são, geralmente, os que possuem maior importância para os sistemas de visão computacional. Esses objetos são constituídos de um conjunto de *pixels* representando a sua forma, cor e textura. Os extratores de atributos representam o objeto através de métodos ou modelos matemáticos aplicados, por exemplo, nos *pixels* internos da sua área, na sua textura e nos *pixels* que delimitam o contorno.

Os extratores utilizados neste trabalho foram divididos em duas categorias, baseados no contorno e na região interna do objeto. Os extratores baseados em contorno descrevem a forma do objeto considerando o limite da região após a aplicação de um detector de bordas, como os descritores de Fourier, k-curvatura e cadeias de códigos. Os extratores baseados em região descrevem a área do objeto considerando seu conteúdo interno, como os momentos de imagens.

3.1 Momentos de Imagens

Os momentos [dSP05] de ordem $p + q$ sobre uma função contínua 2D são calculados de acordo com a Equação 3.1.

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) dx dy \quad (3.1)$$

Uma imagem pode ser modelada como uma função 2D discreta I , onde a intensidade de cada *pixel* é indexada como $I(x, y)$. A Equação 3.2 é a derivação da Equação 3.1 de momentos aplicados em imagens.

$$M_{pq} = \sum_{x=1}^{width} \sum_{y=1}^{height} x^p y^q I(x, y) \quad (3.2)$$

Para a representação de um objeto através de momentos são considerados apenas *pixels* pertencentes ao objeto. Através dos momentos de imagens, aplicados a um objeto, é possível derivar algumas propriedades, como por exemplo, a área do objeto é representada pelo momento M_{00} e o centro de massa \bar{x} e \bar{y} é representado pelos momentos $\frac{M_{10}}{M_{00}}$, $\frac{M_{01}}{M_{00}}$, respectivamente.

Os momentos centrais de imagens, definidos na Equação 3.3, podem também ser utilizados para calcular outras interessantes propriedades dos objetos, como a sua variância nos eixos X e Y (Equação 3.4), direção (Equação 3.5) e excentricidade (Equação 3.6).

$$u_{pq} = \sum_{x=1}^{width} \sum_{y=1}^{height} (x - \bar{x})^p (y - \bar{y})^q I(x, y) \quad (3.3)$$

$$\sigma_x^2 = \frac{u_{20}}{m_{00}}, \quad \sigma_y^2 = \frac{u_{02}}{m_{00}} \quad (3.4)$$

$$\theta = \frac{u_{02} - u_{20} - 2u_{11} + \lambda}{u_{02} - u_{20} + 2u_{11} - \lambda} \quad (3.5)$$

$$e^2 = \frac{u_{20} + u_{02} + \lambda}{u_{20} + u_{02} - \lambda} \quad (3.6)$$

$$\lambda = \sqrt{(u_{20} - u_{02})^2 + 4u_{11}^2} \quad (3.7)$$

Além das propriedades citadas acima, calculadas para todos os *pixels* pertencentes ao objeto, neste trabalho, seguindo uma metodologia sugerida por Freeman [FTO96], o objeto é dividido em 4 regiões iguais, e para cada uma dessas regiões, as mesmas propriedades dos momentos de imagens são calculadas. Desta maneira, o sistema pode combinar informações globais e locais durante a fase de classificação. A Figura 3.1 ilustrada, usando uma ferramenta de visualização dos momentos, as propriedades extraídas de uma imagem contendo uma serpente previamente segmentada e binarizada.

3.2 K-Curvatura

O algoritmo k-curvatura representa o contorno de um objeto através do ângulo formado entre dois vetores. Com a aplicação de um detector de bordas na imagem segmentada é possível adquirir um conjunto de pontos (Equação

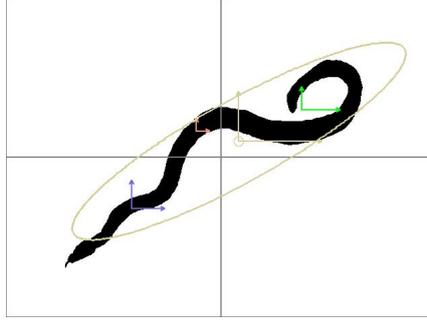


Figura 3.1: Aplicação dos momentos de imagem.

3.8) que representam o contorno (veja Figura 3.2(b)). Esse conjunto é adquirido percorrendo os pontos em um sentido (e.g. horário e anti-horário) a partir de um ponto inicial qualquer.

$$P = \{p_0, p_1, \dots, p_n\} \quad \text{com} \quad p_i = (x_i, y_i) \quad (3.8)$$

Para eliminação de ruídos, por exemplo no processo de segmentação, não são considerados todos os pontos do conjunto P [Tru05]. O cálculo do ângulo é realizado entre três pontos p_{i-k} , p_i e p_{i+k} espaçados de k pontos, como mostra a Figura 3.2(b). O vetor v é formado pelos pontos p_{i-k} e p_i , enquanto o vetor w é formado pelos pontos p_i e p_{i+k} . A Equação 3.9 mostra o cálculo do ângulo entre dois vetores.

$$\theta = \cos^{-1} \frac{v \cdot w}{|v||w|} \quad (3.9)$$

onde $v \cdot w$ é o produto escalar entre dois vetores (Equação 3.10) e $|v|$ é a norma de um vetor (Equação 3.11).

$$v \cdot w = v_1 w_1 + v_2 w_2 + \dots + v_n w_n \quad (3.10)$$

$$|v| = \sqrt{v \cdot v} \quad (3.11)$$

Após os cálculos de todos os ângulos do contorno é construído um histograma, onde cada posição i desse histograma corresponde à frequência de uma faixa de ângulos encontrado no contorno. Através dessa representação por histograma, o extrator k-curvatura é invariante à rotação e translação. Para que o k-curvatura seja invariante à escala é necessário que o k seja atualizado de acordo com a escala aplicada.

Como exemplo, considere o contorno de um objeto constituído pelos ângulos $0^\circ, 82^\circ, 33^\circ, 160^\circ, 102^\circ, 170^\circ, 179^\circ, 131^\circ, 20^\circ, 49^\circ, 51^\circ$ entre os pontos e

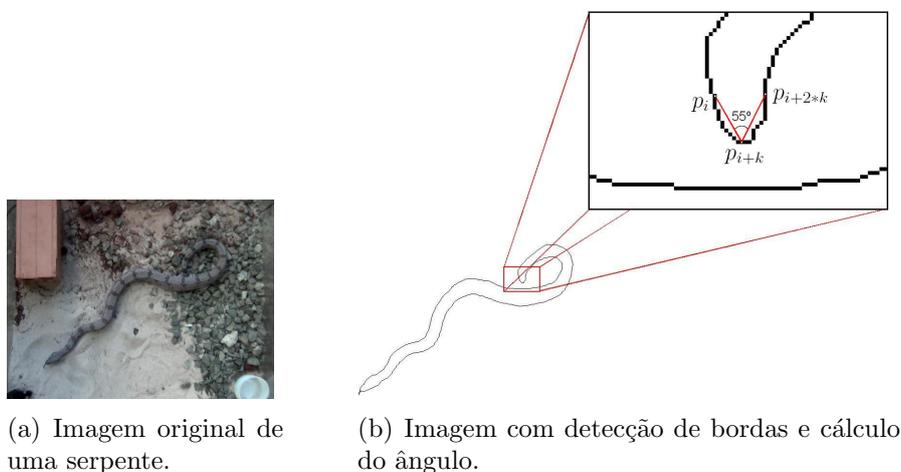


Figura 3.2: Passos realizados para a extração de atributos com o algoritmo de k-curvatura.

uma faixa de 50 ângulos para o histograma. O histograma que representa o exemplo acima é ilustrado na Figura 3.3.

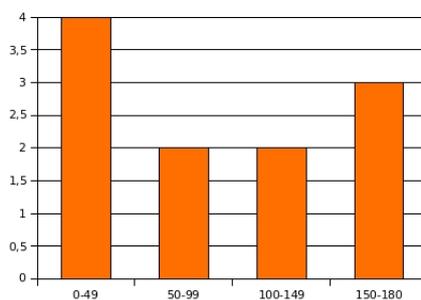


Figura 3.3: Histograma de ângulos extraídos pelo algoritmo de k-curvatura.

3.3 Cadeias de Códigos

As Cadeias de códigos (*chain codes*) foram inicialmente introduzidas por Freeman [Fre74, SSE05]. Este método de extração de atributos é utilizado para representar o contorno de objetos em uma imagem através de uma seqüência de símbolos com determinado tamanho e direção [DB05]. As primitivas de conectividade-de-4 ou de-8 segmentos são, geralmente, utilizadas como símbolos e direção. As primitivas são ilustradas na Figura 3.4.

A principal vantagem da cadeia de códigos é a flexibilidade em descrever

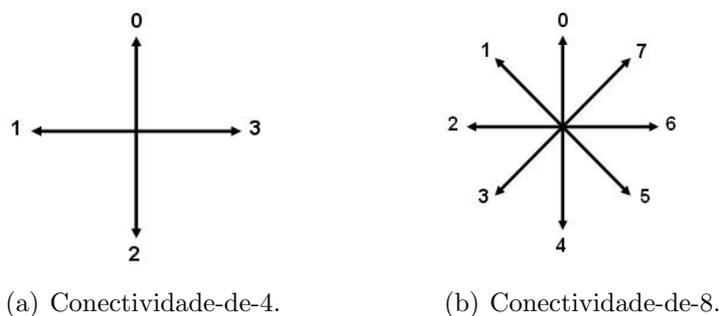


Figura 3.4: Primitivas utilizadas nas cadeias de códigos.

um conjunto amplo de objetos. Em contrapartida, qualquer pequena perturbação ao longo do contorno, devido a ruído ou mudanças causadas pela segmentação defeituosa, pode não ser corretamente relacionada à forma do objeto.

A partir da cadeia de códigos é possível construir um histograma de cadeia de códigos (*Chain Code Histogram - CCH*) [IV96]. O CHH é descrito pela Função discreta 3.12.

$$h(k) = \frac{n_k}{n} \quad (3.12)$$

onde n_k é o número de códigos de direção k e n é o tamanho da cadeia de códigos. Com isso, o CCH mostra a porcentagem das diferentes direções presentes no contorno do objeto. O exemplo de um objeto, cadeia de códigos e CHH são ilustrados na Figura 3.5.

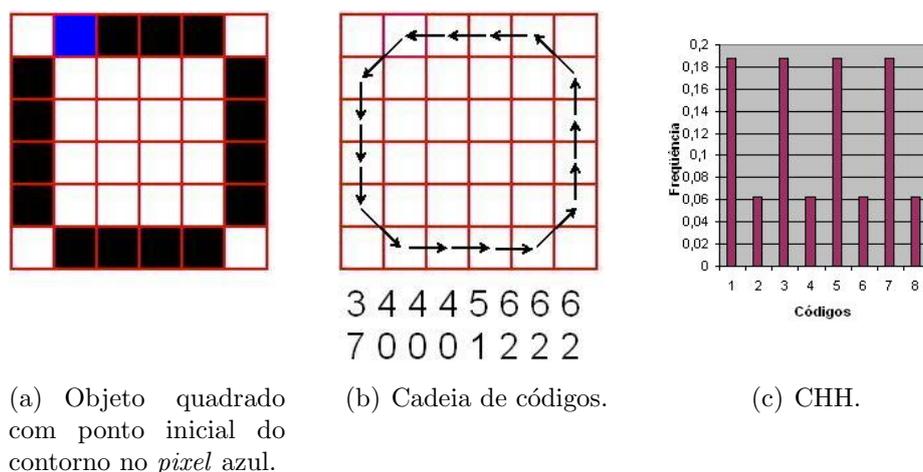


Figura 3.5: Exemplo da aplicação da cadeia de códigos em um objeto.

3.4 Descritores de Fourier

Os descritores de Fourier são muito utilizados para descrever os objetos através do contorno. Os descritores de Fourier aproximam os pontos dos contornos P (Equação 3.8) através da soma de K termos senos e cossenos [GH95] através da Equação 3.13, onde $K \leq \frac{n}{2}$.

$$p_l = \bar{p} + \sum_{k=1}^K a_k \cos\left(\frac{2\pi kl}{n}\right) + b_k \sin\left(\frac{2\pi kl}{n}\right), \quad l = 1, \dots, n \quad (3.13)$$

onde \bar{p} é o centro de massa do objeto, a_k e b_k são descritos nas Equações 3.14 e 3.15.

$$a_k = \frac{2}{n} \sum_{l=1}^K p_l \cos\left(\frac{2\pi kl}{n}\right) \quad (3.14)$$

$$b_k = \frac{2}{n} \sum_{l=1}^K p_l \sin\left(\frac{2\pi kl}{n}\right) \quad (3.15)$$

Para valores de K iguais à $\frac{n}{2}$, a curva interpolada do contorno passa por todos os pontos originais. Para valores de K menores que $\frac{n}{2}$ a curva interpolada é uma aproximação do contorno original. As aproximações podem ser visualizadas na Figura 3.6 para o contorno de uma serpente para diferentes valores de K .

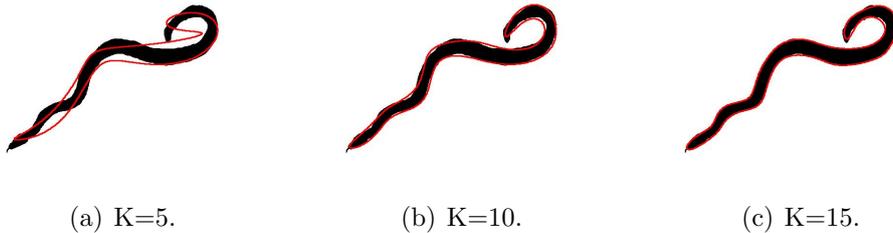


Figura 3.6: Aplicação dos descritores de Fourier para diferentes valores de K .

Essas aproximações são importantes para eliminar pequenos ruídos fornecidos, por exemplo, pelo processo de segmentação. Além disso, através desses descritores é possível reconstruir o contorno original e manter a invariância com respeito à translação [GH95].

Capítulo 4

Modelos Ocultos de Markov

Os modelos ocultos de Markov, ou HMM do inglês *hidden Markov models*, são compostos por um processo estocástico duplo [Rab90]. O primeiro processo é descrito por um conjunto de estados não diretamente observáveis, por isso, o nome desse modelo recebe a palavra oculto. A transição entre os estados segue a condição do processo de Markov, onde em um processo de ordem n , a probabilidade de ocorrência de um evento no tempo t depende apenas dos últimos n eventos, visualizada na Equação 4.1.

$$P(S_t | S_{t-1}, S_{t-2}, \dots, S_1) = P(S_t | S_{t-1}, S_{t-2}, \dots, S_{t-n}) \quad (4.1)$$

Para os modelos ocultos de Markov de primeira ordem, a probabilidade de transição entre os estados depende apenas do estado anterior. Assim, a probabilidade de transição entre os estados desse modelo é representada através de uma matriz de transição $A = \{a_{ij}\}$ descrita na Equação 4.2.

$$a_{ij} = P(q_{t+1} = S_j | q_t = S_i) \quad 1 \leq i, j \leq N \quad (4.2)$$

sendo N o número de estados, q_t o estado atual e $S = \{S_1, S_2, S_3, \dots, S_N\}$ o conjunto de estados do modelo.

Cada elemento a_{ij} da matriz de transição representa a probabilidade de mudança para um estado S_j no tempo $t+1$ sabendo que no tempo t o estado era S_i . As seguintes restrições são requeridas para a matriz A :

$$a_{ij} \geq 0 \quad e \quad \sum_{j=1}^N a_{ij} = 1 \quad (4.3)$$

Em algumas aplicações, os estados não são diretamente observados, com isso, cada estado é observado através de um segundo processo estocástico. A seqüência de observações é representada por um conjunto

$O = O_1, O_2, O_3, \dots, O_T$, onde cada elemento O_t representa um símbolo do conjunto finito de símbolos $V = v_1, v_2, \dots, v_M$ sendo M o número de símbolos. A probabilidade de emissão de símbolos em um estado j é definida na matriz $B = \{b_j(k)\}$ da Equação 4.4.

$$b_j(k) = P(O_t = v_k | q_t = S_j) \quad 1 \leq j \leq N, \quad 1 \leq k \leq M \quad (4.4)$$

Cada elemento $b_j(k)$ representa a probabilidade de observar um símbolo v_k em um estado S_j no tempo t . Essas probabilidades são extremamente importantes, já que, o estado não é diretamente observável.

A probabilidade inicial de cada estado é representada por um conjunto $\pi = \{\pi_i\}$ mostrada na Equação 4.5.

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N, \quad \text{com} \quad \sum_{i=1}^N \pi_i = 1 \quad (4.5)$$

Considere o exemplo de uma aplicação para o reconhecimento em imagens do comportamento de elevação vertical de um camundongo. Esse comportamento será modelado por um HMM composto de três estados, representados por $S = \{S_1 = \text{camundongo_abaixado}, S_2 = \text{camundongo_semi-elevado}, S_3 = \text{camundongo_elevado}\}$. Cada estado é ilustrado nas Figuras 6.1(a), 6.1(b) e 6.1(c), respectivamente.



(a) Estado 1 do exemplo.

(b) Estado 2 do exemplo.

(c) Estado 3 do exemplo.

Figura 4.1: Estados do exemplo de reconhecimento do comportamento de elevação vertical.

A probabilidade de transição entre os estados citados acima é representada pela matriz A da Tabela 4.1. Essas probabilidades atribuídas a esse exemplo descrevem aproximadamente a situação real do comportamento de exploração vertical, sendo este comportamento composto de uma seqüência de movimentos no tempo. Cada movimento é descrito por um estado do

| | Abaixado | Semi-Elevado | Elevado |
|--------------|----------|--------------|---------|
| Abaixado | 0.4 | 0.6 | 0 |
| Semi-Elevado | 0.1 | 0.2 | 0.7 |
| Elevado | 0 | 0 | 1 |

Tabela 4.1: Matriz com as probabilidades de transições entre os estados.

modelo. A probabilidade de o camundongo manter-se com as quatro patas sobre o piso da arena (estado 1) são de aproximadamente 40%. Enquanto, a probabilidade do camundongo mudar desse estado para um movimento intermediário (estado 2) de elevação é de 60%. Como a taxa de captura do dispositivo é de aproximadamente 30 *frames*/segundo, não é possível que o camundongo transite diretamente do estado 1 para um movimento de elevação completo (estado 3), com isso, a probabilidade é de 0%. Além disso, cada estado possui uma probabilidade inicial representada pelo conjunto $\pi = \{0.7, 0.15, 0.15\}$. A probabilidade de maior ocorrência é atribuída ao estado 1, pois os camundongos permanecem na maior parte do tempo abaixados.

Existe a necessidade de utilizar o HMM pois os estados não são diretamente observáveis através das imagens. Com isso, a observação é realizada através de um conjunto de símbolos. Para esse exemplo simples, os símbolos são representados pelo conjunto $V = \{v_1 = \textit{circularidade_baixa}, v_2 = \textit{circularidade_alta}\}$, visualizada nas Figuras 4.2(a) e 4.2(b). A circularidade é calculada através da razão entre o eixo menor e o eixo maior da elipse que representa o objeto.

A probabilidade de emissão de símbolos de cada estado é mostrada na Tabela 4.2. Essas probabilidades estão de acordo com observações realizadas sobre imagens de cada estado. Para o estado 1, existe uma probabilidade de 90% de se observar uma circularidade baixa pois na maioria das vezes o camundongo está abaixado. Em relação ao estado 2, existem probabilidades aproximadamente iguais pois as observações de circularidade alta e baixa são proporcionalmente distribuídas. Para o estado 3, ocorre a situação inversa ao do estado 1, mas com probabilidade de 85%.

O autômato que representa o HMM com os parâmetros $\lambda = (A, B, \pi)$ definidos acima é ilustrado na Figura 4.3. Nesse autômato, os vértices representam os estados e as arestas possuem valores correspondentes às transições entre os vértices. A probabilidade inicial de cada estado é representada através de um retângulo abaixo de cada vértice. A tabela de cada vértice representa a probabilidade de emissão de símbolos desse estado.

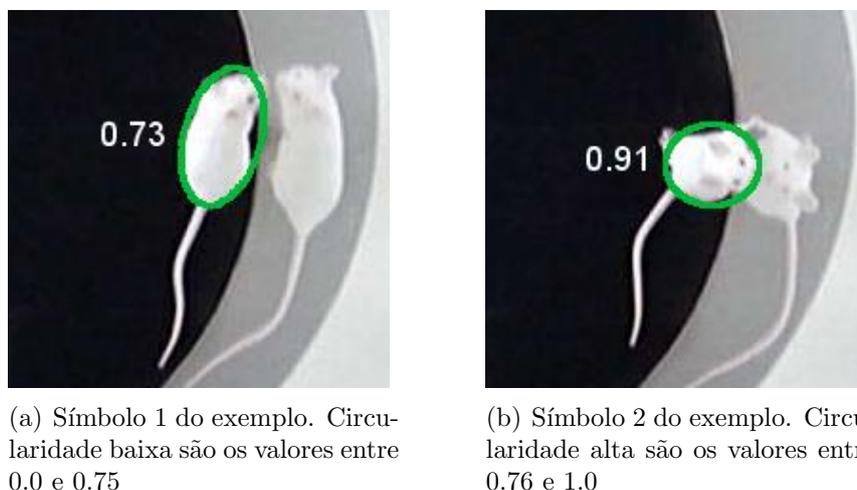


Figura 4.2: Símbolos do exemplo de reconhecimento do comportamento de elevação vertical.

| | Circularidade Baixa | Circularidade Alta |
|--------------|---------------------|--------------------|
| Abaixado | 0.9 | 0.1 |
| Semi-Elevado | 0.4 | 0.6 |
| Elevado | 0.15 | 0.85 |

Tabela 4.2: Matriz com as probabilidades de emissão de símbolos de cada estado.

4.1 Os problemas básicos dos Modelos Ocultos de Markov

Para a utilização dos modelos ocultos de Markov em aplicações do mundo real, é necessária a resolução de três problemas básicos. A resolução desses problemas, conhecidos como problema de avaliação, decodificação e aprendizagem, foi extensivamente descritas em diversos trabalhos na literatura [Rab90, MS04, SP95, ALT95].

4.1.1 Problema de Avaliação

O primeiro problema envolve o cálculo da probabilidade de uma seqüência observada $O = O_1, O_2, \dots, O_T$ ser produzida por um modelo $\lambda = (A, B, \pi)$. Esse problema é conhecido como um problema de avaliação e podemos visualizar a solução desse problema como uma probabilidade que mensura a relação

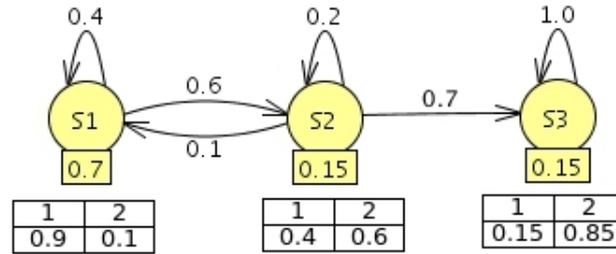


Figura 4.3: Autômato representando o HMM do exemplo.

entre a seqüência observada e um modelo. A solução desse problema é extremamente útil quando é preciso decidir entre vários modelos competitivos. Por exemplo, supondo que existe um HMM modelando cada comportamento dos animais, é possível decidir o comportamento apresentado pelo animal com base nesses modelos. Para isso, a observação do comportamento atual, descrita pelos atributos, é comparada com os diversos modelos através da solução do primeiro problema. Em seguida, o modelo com maior probabilidade de ocorrência é selecionado e, conseqüentemente, o comportamento apresentado pelo animal.

A probabilidade da seqüência observada O dado o modelo λ é definida como $P(O|\lambda)$. Uma maneira simples e objetiva de calcular essa probabilidade é enumerar todas as possíveis seqüências de estados $Q = q_1, q_2, \dots, q_T$ com o tamanho T igual ao tamanho da observação. Com isso, é possível calcular a probabilidade dessas seqüências, assumindo independência estatística entre as observações, através da Equação 4.6.

$$P(O|Q, \lambda) = \prod_{t=1}^T P(O_t|q_t, \lambda) = b_{q_1}(O_1)b_{q_2}(O_2)b_{q_3}(O_3)\dots b_{q_T}(O_T) \quad (4.6)$$

Além da probabilidade das observações, é necessário calcular a probabilidade da seqüência de estado atual Q , mostrada na Equação 4.7.

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} a_{q_3 q_4} \dots a_{q_{T-1} q_T} \quad (4.7)$$

Com isso, a probabilidade de ocorrência da seqüência de estados e observações ao mesmo tempo é a probabilidade conjunta de O e Q , definida na Equação 4.8.

$$P(O, Q|\lambda) = P(O|Q, \lambda)P(Q|\lambda) \quad (4.8)$$

Por fim, a probabilidade $P(O|\lambda)$ é calculada sobre todas as possíveis seqüências de estados através da Equação 4.9.

$$P(Q|\lambda) = \sum_{Q_1, Q_2, \dots, Q_{NT}} P(O|Q, \lambda)P(Q|\lambda) \quad (4.9)$$

O cálculo dessa probabilidade através dessa definição direta é computacionalmente inviável, pois requer cálculos na ordem de $2TN^T$ operações. Devido a esse problema, tornou-se necessário a realização de um procedimento mais eficiente. Um dos procedimentos que resolve esse problema eficientemente, baseado na técnica de programação dinâmica, é conhecido como *Forward-Backward*. Esse procedimento define uma variável $\alpha_t(j)$, mostrada na Equação 4.10, utilizando-a para o cálculo da probabilidade $P(O|\lambda)$.

$$\alpha_t(j) = P(O_1, O_2, O_3, \dots, O_t, q_t = S_j|\lambda) \quad (4.10)$$

A variável $\alpha_t(j)$ define a probabilidade das observações parciais até um tempo t e estado S_j dado um modelo λ . Essa variável é calculada recursivamente através das Equações 4.11, 4.12 e 4.13.

$$\alpha_1(j) = \pi_j b_j(O_1), \quad 1 \leq j \leq N. \quad (4.11)$$

A Equação 4.11 inicializa a variável $\alpha_t(j)$ com a probabilidade inicial de cada estado e a probabilidade desse estado gerar a primeira observação O_1 .

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \leq t \leq T-1. \quad (4.12)$$

A Equação 4.12 é o principal passo do procedimento. Esse passo calcula, recursivamente, a probabilidade de um estado S_j gerar uma observação O_{t+1} e ser alcançado no tempo $t+1$ pelos N possíveis estados no tempo t .

$$P(O|\lambda) = \sum_{j=1}^N \alpha_T(j) \quad (4.13)$$

O último passo realizado pelo procedimento *Forward-Backward* é dado na Equação 4.13. Esse passo calcula a probabilidade $P(O|\lambda)$ através da soma das variáveis $\alpha_T(i)$ finais. Esse procedimento, comparado com a solução acima, é mais eficiente, envolvendo cálculos na ordem de N^2T .

A variável $\beta_t(i)$ é calculada de forma similar a variável $\alpha_t(i)$. Essa variável é definida na Equação 4.14 calculando a probabilidade de observações parciais de $t+1$ até T dado o estado S_i no tempo t e um modelo λ .

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, O_{t+3}, \dots, O_T | q_t = S_i, \lambda) \quad (4.14)$$

A variável $\beta_t(i)$ poder ser calculada recursivamente através das Equações 4.15, 4.16 e 4.17. A Equação 4.15 realiza a inicialização dos parâmetros do procedimento, a Equação 4.16 realiza o passo da indução do algoritmo e finalmente a Equação 4.17 realiza o cálculo de $P(O|\lambda)$.

$$\beta_T(i) = 1, \quad 1 \leq i \leq N. \quad (4.15)$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad t = T-1, T-2, T-3, \dots, 1 \quad 1 \leq i \leq N. \quad (4.16)$$

$$P(O|\lambda) = \sum_{i=1}^N \pi_i b_i(O_1) \beta_1(i) \quad (4.17)$$

Para o problema de avaliação, não é necessário que as duas variáveis α e β sejam calculadas. Entretanto, para solucionar os outros problemas, como o problema de treinamento, as duas variáveis são necessárias.

Seguindo o exemplo citado anteriormente, com conjunto de estados $S = \{S_1 = \text{camundongo_abaixado}, S_2 = \text{camundongo_semi - elevado}, S_3 = \text{camundongo_elevado}\}$ e conjunto de símbolos $V = \{v_1 = \text{circularidade_baixa}, v_2 = \text{circularidade_alta}\}$, o cálculo do procedimento *forward-backward* é visualizado através da treliça da Figura 4.4 para uma seqüência de observação $O^1 = \{O_1 = v_1, O_2 = v_1, O_3 = v_2\}$, adquiridas das Figuras 4.2(a) e 4.2(b). Utilizando apenas a variável *forward* a probabilidade dessa seqüência de observação pertencer ao modelo é de 21,4%. Na treliça, as colunas de vértices representam o tempo e as linhas representam os estados. O conteúdo de cada vértice representa a probabilidade de emissão do símbolo da seqüência observada, enquanto as arestas entre dois vértices representam as probabilidades de transição entre os mesmos. As arestas que não possuem dois vértices representam as probabilidades iniciais de cada estado.

Para outra seqüência de observação $O^2 = \{O_1 = v_1, O_2 = v_1, O_3 = v_1\}$, adquiridas da Figura 4.2(a). O mesmo cálculo pode ser visualizado na treliça da Figura 4.5 com probabilidade de 18,3%.

Com isso, existe uma probabilidade maior da seqüência O^1 pertencer ao modelo λ em relação à seqüência O^2 . Essa probabilidade pode ser verificada através das imagens da seqüência O^1 , visto que, essas imagens representam melhor o comportamento de elevação vertical modelado pelo HMM.

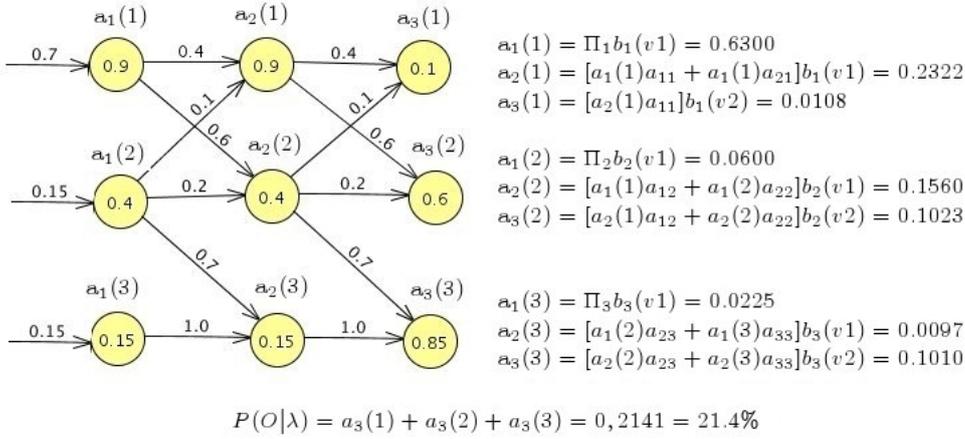


Figura 4.4: Execução da variável α para uma observação $\{O_1 = v_1, O_2 = v_1, O_3 = v_2\}$ através de uma treliça.

4.1.2 Problema de Decodificação

O segundo problema, conhecido como problema de decodificação, busca a melhor seqüência de estados $Q = q_1, q_2, \dots, q_T$ que representa a seqüência de observações $O = O_1, O_2, \dots, O_T$ baseado em um critério ótimo. Existem diversas formas de definir um critério ótimo, uma forma seria analisar todos os estados e encontrar os mais prováveis para a seqüência de observação. Este critério ótimo maximiza a taxa de acerto de cada estado representar a seqüência de observações. Este problema está representado através da Equação 4.18, onde a variável γ_i define a probabilidade do estado S_i no tempo t produzir a observação O_t .

$$\gamma_i = P(q_t = S_i | O_t, \lambda) \quad (4.18)$$

Pode-se representar a Equação 4.18 em termos de α e β dos respectivos algoritmos *forward* e *backward*, conforme a Equação 4.19.

$$\gamma_i = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (4.19)$$

O $\alpha_t(i)$ representa a probabilidade do caminho de um estado inicial até o estado S_i para a seqüência de observação O_1, O_2, \dots, O_t e $\beta_t(i)$ representa a probabilidade do caminho de um estado final até um estado S_i para a seqüência de observação $O_{t+1}, O_{t+2}, \dots, O_T$, dada uma seqüência de observação de $O_1, O_2, \dots, O_{t+1}, O_{t+2}, \dots, O_T$, conforme descrito anteriormente.

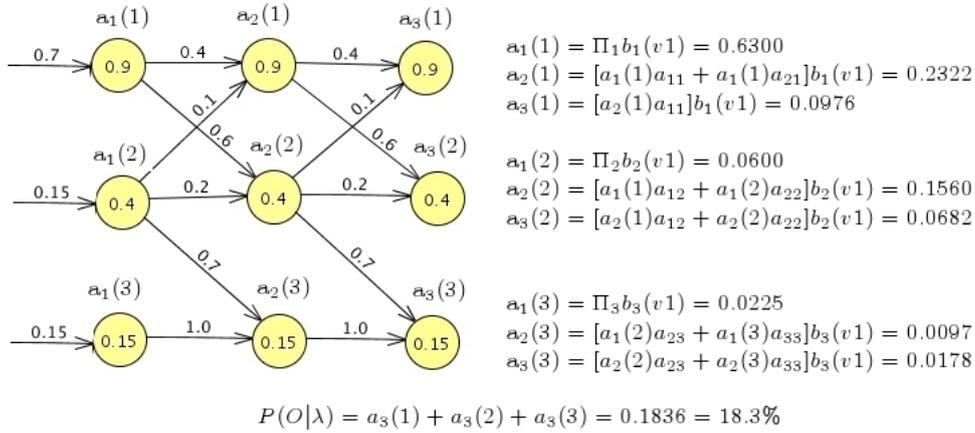


Figura 4.5: Execução da variável α para uma observação $\{O_1 = v_1, O_2 = v_1, O_3 = v_1\}$ através de uma treliça.

Para efeitos de normalização utilizou-se o fator $P(O|\lambda) = \sum_{i=1}^N \alpha_t(i)\beta_t(i)$ para que a soma das probabilidades de γ_t seja máxima, conforme a Equação 4.20.

$$\sum_{i=1}^N \gamma_t(i) = 1 \quad (4.20)$$

Com isso, encontrar o estado $q_t = S_i$ mais provável para a seqüência de observação no tempo t maximizando a probabilidade de $\gamma_t(i)$ é possível através da Equação 4.21.

$$q_t = \arg \max_{1 \leq i \leq N} [\gamma_t(i)], \quad 1 \leq t \leq T. \quad (4.21)$$

Embora a Equação 4.21 encontre um caminho ótimo para um estado $q_t = S_i$ no tempo t , podemos encontrar alguns problemas durante as transições dos estados, como por exemplo quando a probabilidade de transição de um estado for $a_{ij} = 0$, podendo assim encontrar a melhor solução, mas não sendo válida para a seqüência de estados. Este problema deve-se ao fato que a Equação 4.21 considera somente o estado mais provável a cada instante de tempo t , sem considerar a seqüência de estados.

Para solucionar o problema, é necessário modificar as propriedades do critério ótimo. Uma solução possível seria considerar não somente um estado mas, par, triplo ou um conjunto n de estados válidos para a seqüência de observação. A solução existente para resolver esse problema é através do algoritmo baseado em programação dinâmica, conhecido como algoritmo de

Viterbi. O algoritmo de Viterbi escolhe a melhor seqüência de estados dado as observações e pode ser definido através da Equação 4.22.

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_t} P(q_1, q_2, \dots, q_t = i, O_1, O_2, \dots, O_t | \lambda). \quad (4.22)$$

A variável $\delta_t(i)$ define a maior probabilidade da seqüência de estados q_1, q_2, \dots, q_t gerar cada elemento da seqüência de observação O_1, O_2, \dots, O_t percorrendo um caminho ótimo ao longo do modelo até um estado S_i no tempo t , considerando as probabilidades de transições entre os estados e as observações. Dessa forma, por indução a solução do problema pode ser descrita através da Equação 4.23.

$$\delta_{t+1}(j) = [\max_i \delta_t(i) a_{ij}] b_j(O_{t+1}). \quad (4.23)$$

Para maximizar a Equação 4.23, é necessário armazenar a seqüência percorrida em um caminho no modelo no tempo t em um estado j . Esse percurso ao longo do modelo é armazenado em um vetor $\psi_t(j)$. Assim definidas as variáveis, o processo de execução do algoritmo é definido em três passos. A Equação 4.24 realiza o passo de inicialização das variáveis, em seguida, na Equação 4.26 é realizado o passo de recursão do algoritmo, no qual é calculado e armazenado o melhor caminho pelo modelo. Finalmente a Equação 4.28 realiza a finalização do algoritmo recuperando o caminho pelo modelo por *backtracking*.

$$\delta_1(j) = \pi_j b_j(O_1), \quad 1 \leq j \leq N. \quad (4.24)$$

$$\psi_1(j) = 0. \quad (4.25)$$

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] b_j(O_t), \quad 2 \leq t \leq T \quad 1 \leq j \leq N. \quad (4.26)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}], \quad 2 \leq t \leq T \quad 1 \leq j \leq N. \quad (4.27)$$

$$P^* = \max_{1 \leq j \leq N} [\delta_T(j)]. \quad (4.28)$$

$$q_T^* = \arg \max_{1 \leq j \leq N} [\delta_T(j)]. \quad (4.29)$$

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \quad t = T - 1, T - 2, \dots, 1. \quad (4.30)$$

onde P^* é a probabilidade da melhor seqüência de estados, q_T^* e q_t^* são os melhores estados em cada instante de tempo t .

4.1.3 Problema da Aprendizagem

O terceiro problema, considerado um dos mais difíceis do HMM, consiste em determinar um método para ajustar os parâmetros do modelo $\lambda = (A, B, \pi)$, que maximiza localmente $P(O|\lambda)$. Nos modelos ocultos de Markov, os estados não são diretamente observáveis, com isso, não é possível identificar o qual estado pertence certa observação. Devido a esse problema, não é possível utilizar uma fórmula ou algoritmo analítico para realizar a estimação dos parâmetros. Neste caso é necessário utilizar métodos iterativos, como o algoritmo de *Baum-Welch* (especialização do algoritmo *expectation-maximization* (EM)).

O algoritmo de *Baum-Welch* possui um método iterativo para a re-estimação dos parâmetros baseado no método de estimação por máxima verossimilhança. A idéia básica desse algoritmo é realizar a estimação através do número esperado de transições do estado S_i para o estado S_j e o número de emissão do símbolo O_t no estado S_i . A estimação desses números é baseada nos valores dos parâmetros atuais do modelo e na seqüência de observação do treinamento.

Para iniciar a estimação dos parâmetros do modelo é necessário definir a probabilidade de transitar de um estado S_i no tempo t para outro estado S_j no tempo $t + 1$ dado a seqüência de observação de treinamento para o modelo λ , definido pela Equação 4.31.

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda). \quad (4.31)$$

As probabilidades de transição entre os estados S_i e S_j são calculadas através dos métodos *Forward-Backward*, discutidos na seção 4.1.1. A Equação 4.31 pode ser descrita através das variáveis do algoritmo *Forward-Backward*, na forma da Equação 4.32.

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)}. \quad (4.32)$$

$$\xi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}. \quad (4.33)$$

A variável $\gamma_t(i)$ já foi definida na Seção 4.1.2 como a probabilidade de ocorrência do estado S_i . Essa variável pode ser reescrita em função de $\xi_t(i, j)$ através da Equação 4.34.

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j). \quad (4.34)$$

Ao realizar a soma de $\gamma_t(i)$ se obtém uma medida esperada do número de vezes que o estado S_i foi visitado no tempo t conforme a Equação 4.35, e ao somar $\xi_t(i, j)$ se obtém o número de transições ocorridas do estado S_i para o estado S_j , conforme a Equação 4.36.

$$\sum_{t=1}^{T-1} \gamma_t(i). \quad (4.35)$$

$$\sum_{t=1}^{T-1} \xi_t(i, j). \quad (4.36)$$

Com base nas equações acima, que realizam a contagem de eventos ocorridos, pode-se obter um método de estimar os parâmetros do modelo $\lambda = (A, B, \pi)$. Dessa forma, as equações 4.37, 4.38 e 4.39, representam respectivamente a frequência de eventos ocorridos no estado S_i no tempo 1, a média do número de transições ocorridas do estado S_i para o estado S_j e finalmente a média do número de símbolos observados no estado S_j .

$$\bar{\pi}_i = \gamma_1(i). \quad (4.37)$$

$$\bar{a}_{i,j} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \quad (4.38)$$

$$\bar{b}_j(O_t) = \frac{\sum_{t=1, o_t=v_k}^{T-1} \gamma_t(i)}{\sum_{t=1}^{T-1} \gamma_t(i)}. \quad (4.39)$$

Os novos parâmetros do modelo $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$ são re-estimados através das equações 4.37, 4.38 e 4.39. O processo de re-estimação é realizado até que sejam atendidos os critérios de parada, como por exemplo, um número fixo de iterações.

4.2 Modelos Ocultos de Markov com Observação Contínua

Até este momento, foram considerados apenas os modelos ocultos de Markov com observação discreta. Esses modelos utilizam uma função de densidade de probabilidade discreta para modelar as observações de cada estado. Essa modelagem é possível quando às observações são de natureza discreta, assumindo valores de um conjunto finito. Em casos onde a observação é contínua, é necessário a discretização através de algoritmos de quantização

de vetores (*vector quantization* - VQ). O problema desta utilização é que essa discretização pode não ser realizada da melhor forma possível, podendo haver confusão ou perda de informações no processo. Com isso, é interessante considerar as observações contínuas nos modelos ocultos de Markov.

As observações contínuas podem ser modeladas através de uma mistura de distribuições, através da Equação 4.40 [Rab90].

$$b_j(O) = \sum_{m=1}^M c_{jm} \eta(O; \mu_{jm}, \Sigma_{jm}), \quad 1 \leq j \leq N \quad (4.40)$$

onde O é um vetor de observações, M é a quantidade de misturas, c_{jm} é a probabilidade da m -ésima distribuição do estado j , η é geralmente uma distribuição Gaussiana descrita na Equação A.13, onde μ_{jm} e Σ_{jm} são os parâmetros da m -ésima distribuição do estado j .

A re-estimação dos parâmetros é realizada através das Equações 4.41, 4.42 e 4.43.

$$c_{jm} = \frac{\sum_{t=1}^T \gamma_t(j, m)}{\sum_{t=1}^T \sum_{k=1}^M \gamma_t(j, k)} \quad (4.41)$$

A re-estimação da Equação 4.41 denota a taxa entre o número esperado no estado j para a m -ésima distribuição e o número esperado no estado j .

$$\mu_{jm} = \frac{\sum_{t=1}^T \gamma_t(j, m) * O_t}{\sum_{t=1}^T \gamma_t(j, m)} \quad (4.42)$$

A Equação 4.42 resulta no valor esperado e a Equação 4.43 na matriz de covariância das observações para a m -ésima distribuição.

$$\Sigma_{jm} = \frac{\sum_{t=1}^T \gamma_t(j, m) (O_t - \mu_{jm})(O_t - \mu_{jm})'}{\sum_{t=1}^T \gamma_t(j, m)} \quad (4.43)$$

A variável $\gamma_t(j, m)$ é uma generalização da variável $\gamma_t(j)$, no caso de uma única mistura ou observações discretas. A variável $\gamma_t(j, m)$ denota a probabilidade de estar no estado j no tempo t com a m -ésima mistura. Essa variável é calculada através da Equação 4.44.

$$\gamma_t(j, m) = \frac{\alpha_t(j) \beta_t(j)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)} \frac{c_{jm} \eta(O_t; \mu_{jm}, \Sigma_{jm})}{\sum_{k=1}^M c_{jk} \eta(O_t; \mu_{jk}, \Sigma_{jk})} \quad (4.44)$$

Capítulo 5

Banco de Imagens

Os métodos de aprendizagem supervisionada, como é o caso dos modelos ocultos de Markov, fazem o uso de uma base de dados para construir o modelo. A base de dados, neste trabalho, é chamada de banco de imagens, constituído de vídeos com diversos comportamentos de animais. Esse banco foi desenvolvido com base em opiniões de especialistas na área da saúde em relação à seleção dos comportamentos. Os comportamentos foram selecionados de acordo com sua importância apresentada para os estudos científicos realizados no Centro de Ciências Biológicas e da Saúde - UCDB ¹.

Os vídeos foram capturados por uma câmera TRENDNET IP301W em laboratórios científicos, totalizando 326 vídeos separados por comportamentos e por animais. Atualmente este banco contém 32 vídeos de comportamentos de serpentes com iluminação natural e 294 vídeos de comportamentos de camundongos com iluminação artificial.

Para as serpentes foram armazenados vídeos do comportamento de bote realizado por diferentes espécies de serpentes (jararaca, boca de sapo e jibóia), visualizada na seqüência da Figura 5.1. O comportamento de bote foi selecionado, por tornar possível a avaliação de indícios sobre o comportamento alimentar, além de medir o nível de socialização entre serpentes através da latência do bote de uma serpente quando existe uma presa em potencial.

Para os camundongos os comportamentos de exploração vertical, exploração horizontal e auto-limpeza foram selecionados. O primeiro comportamento, visualizado na seqüência da Figura 5.2, indica a elevação das patas dianteiras do camundongo. Através desse comportamento é possível medir a ansiedade do camundongo, exposto a um novo ambiente, após a aplicação de algum fármaco.

¹Os experimentos avaliam e comparam fármacos com diferentes composições através da mensuração de funções motoras (e.g. distância e velocidade) e mentais (e.g. ansiedade e grau de memorização).



Figura 5.1: Comportamento de bote realizado por uma serpente.

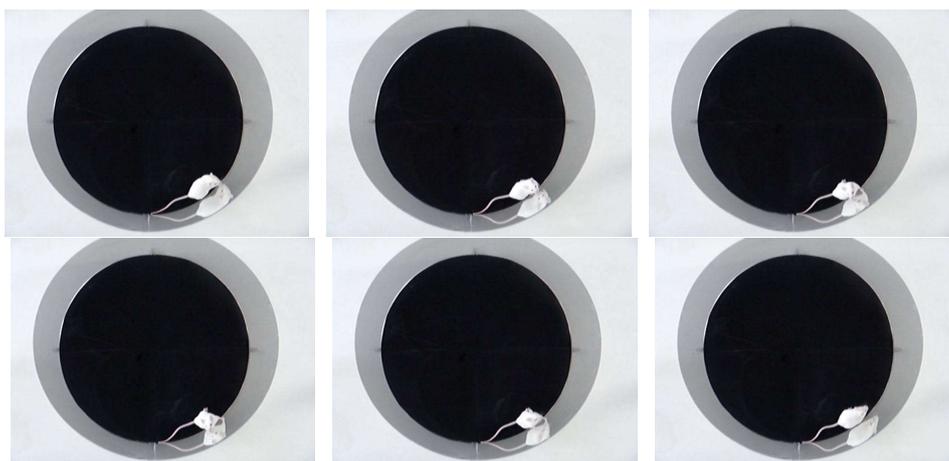


Figura 5.2: Comportamento de exploração vertical.

O comportamento de exploração horizontal, visualizado na seqüência da Figura 5.3, é caracterizado quando o camundongo está com as quatro patas sobre o chão. Esse comportamento é importante, por exemplo, na mensuração da distância percorrida quando existe alguma lesão em seus membros, antes e após a aplicação do fármaco.

O último comportamento, ilustrado na seqüência da Figura 5.4, é distinto para cada espécie de roedor e usado para limpeza e regularização da temperatura do animal [WHK99]. Nesse comportamento, o animal utiliza as patas dianteiras para limpar partes do corpo.

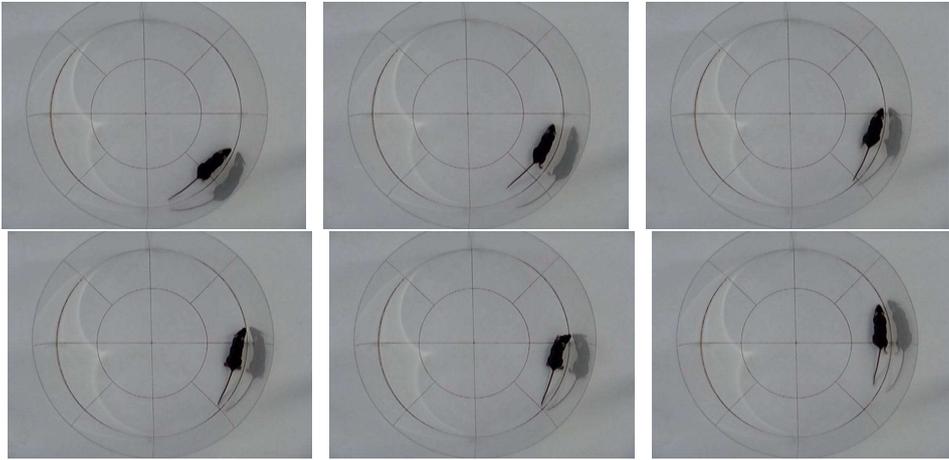


Figura 5.3: Comportamento de exploração horizontal.

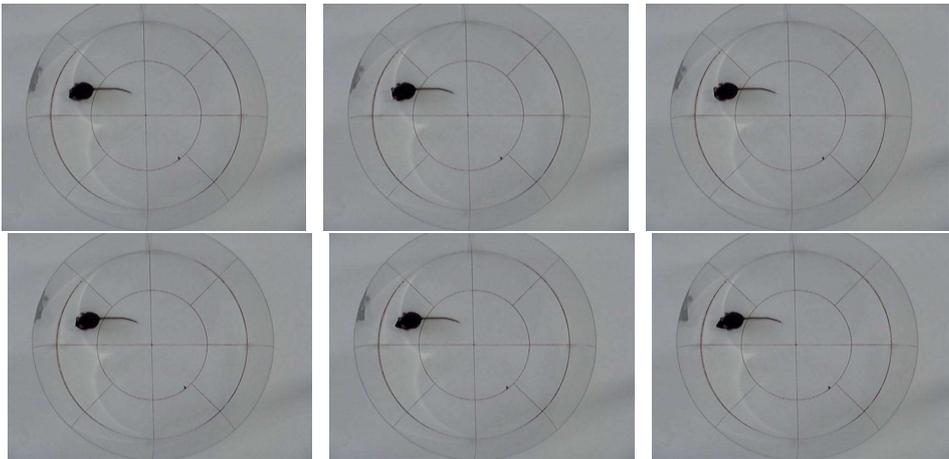


Figura 5.4: Comportamento de auto-limpeza.

Capítulo 6

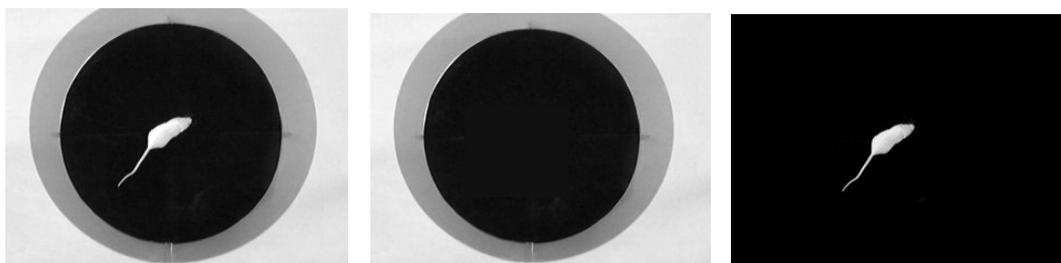
Desenvolvimento

Neste capítulo são apresentados os procedimentos adotados para a utilização dos modelos ocultos de Markov na identificação dos comportamentos de animais. Entre os procedimentos estão a segmentação dos vídeos, inicialização dos HMMs e a discretização das observações para os HMMs discretos.

6.1 Segmentação dos Vídeos

A segmentação é uma importante etapa para a análise de imagens em sistemas de visão computacional. O objetivo da segmentação é o particionamento de imagens em objetos de interesse e objetos irrelevantes. Este processo simplifica a representação da imagem de forma mais significativa para análise do problema e a extração de atributos. No processo de segmentação de imagens não existe uma técnica geral e robusta que obtenha um resultado aceitável para todas as aplicações [dA04]. Por isso, a escolha da técnica é influenciada pela aplicação e os requerimentos do problema.

Os vídeos contendo comportamentos de camundongos foram segmentados utilizando a técnica de subtração de fundo. A subtração de fundo [McI00] é uma das técnicas mais utilizadas devido ao seu baixo custo de processamento. O princípio básico é subtrair cada nova imagem de uma imagem de referência, obtida a partir de um fundo estático e sem o objeto de interesse. A grande dificuldade apresentada pela técnica de subtração de fundo é não se adaptar a modificações no fundo da cena, como mudança de iluminação ou objetos que entram em cena e em seguida ficam estáticos e sem relevância. Entretanto, para as imagens da análise de comportamentos de camundongos a técnica obteve bons resultados pois, o fundo é estático e a iluminação é controlada, com isso, o fundo se mantém praticamente o mesmo para todo o experimento [GSM⁺07]. Um exemplo da técnica é ilustrado na Figura 6.1.



(a) Image original.

(b) Imagem de referência.

(c) Resultado da subtração.

Figura 6.1: Exemplo de segmentação de imagens com camundongos.

Para os vídeos contendo comportamentos de serpentes uma estratégia baseada em aprendizagem de máquina foi adotada. As serpentes usam a camuflagem para se esconder dos predadores e capturar com mais facilidade suas presas, o qual torna o problema de segmentação, neste contexto, mais complexo. A estratégia de aprendizagem supervisionada, baseada em máquinas de vetores de suporte (*Support Vector Machine* - SVM) [Bur98], foi utilizada para separar a serpente do plano de fundo. Para cada imagem, uma região ao redor da serpente foi manualmente determinada para tornar a fase de segmentação mais precisa. Atributos baseados em cores e informações espaciais dessas cores na imagem, extraídos de amostras da serpente e das regiões do fundo, foram usados para alimentar o processo de aprendizagem das máquinas de vetores de suporte. O tamanho dessas amostras foi definido em 20×20 *pixels*. Após o treinamento do modelo, a classificação de um *pixel* na imagem foi realizada da seguinte maneira. Todos os *pixels* da imagem são utilizados como o *pixel* central de uma amostra e classificado através do modelo de máquinas de vetores de suporte treinado. Esse procedimento pode ser ilustrado na Figura 6.2(a) para um *pixel* x, y .

O resultado do procedimento de segmentação para duas espécies de serpentes é ilustrado na Figura 6.2(b). Para cada espécie, foi necessária a coleta de diferentes amostras da pele das serpentes.

6.2 Algoritmo para a inicialização dos HMMs

O processo de aprendizagem dos modelos ocultos de Markov é baseado em re-estimação de parâmetros. Com isso, a escolha dos parâmetros iniciais é uma importante etapa para que o algoritmo de treinamento alcance a estabilização e evite os máximos locais. Os parâmetros iniciais das variáveis π e A podem ser inicializados com uma distribuição uniforme ou aleatória [Rab90]. Porém, para o parâmetro B é necessário que os valores sejam inicializados de maneira



(a) Classificação de um *pixel* central de uma amostra. (b) Exemplos de segmentação de imagens com serpentes.

Figura 6.2: Processo de segmentação da serpente.

mais precisa. Neste trabalho, seguindo [Rab90], a inicialização das variáveis π e A foram realizadas uniformemente de acordo com a equação 6.1.

$$\pi_i = \frac{1}{N}, \quad a_{ij} = \frac{1}{N}, \quad 1 \leq i, j \leq N \quad (6.1)$$

onde N é número de estados internos do HMM. Um exemplo desses parâmetros inicializados uniformemente é ilustrado na Figura 6.3 para um HMM com três estados.

Para a variável B , os valores são mais difíceis de serem inicializados corretamente. A inicialização dos valores dessa variável foi baseada no algoritmo k-means [MW07]. Cada componente das seqüências de observações de treinamento $O^i = O^1, O^2, \dots, O^{NO}$ é associada a um dos k núcleos através do algoritmo k-means. Cada núcleo k representa um estado interno do HMM, com isso, cada estado possui um conjunto de observações mais prováveis. A partir do conjunto de cada estado é realizada uma técnica simples de contagem, com o valor inicial da variável $b_j(k)$ para um HMM discreto ilustrado na Equação 6.2.

$$b_j(k) = \frac{N_{jk}}{N_j} \quad (6.2)$$

onde N_{jk} é o número de observações com valores k associadas ao estado j e N_j é o número de observações associadas ao estado j .

Para observações modeladas através de distribuições contínuas, a variável $b_j(O_t)$ representa a probabilidade da observação O_t pertencer a Gaussiana

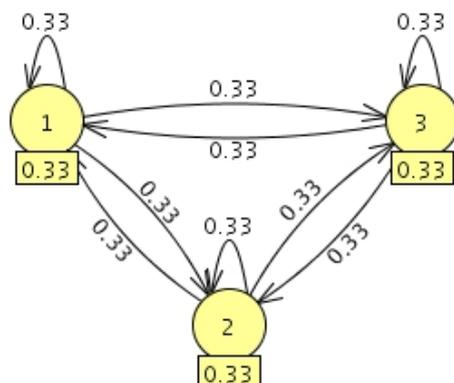


Figura 6.3: Exemplo de HMM inicializado uniformemente.

do estado j . Essa Gaussiana é inicializada como no Apêndice A, onde o conjunto de treinamento é o conjunto de observações associadas ao estado j . Da mesma forma, para as misturas de Gaussianas, o algoritmo EM é aplicado no conjunto associado ao estado j , com isso, as Gaussianas são inicializadas.

6.3 Algoritmo de Discretização

O princípio do algoritmo LBG (Linde, Buzo e Gray) é mapear os vetores t_m de um conjunto de treinamento $T = \{t_1, t_2, \dots, t_M\}$ em um dos k vetores base ou *codebooks* pertencentes ao conjunto $C = \{c_1, c_2, \dots, c_K\}$. O algoritmo inicia com a atribuição de um *codebook* para representar o conjunto de treinamento através da Equação 6.3 e com a variável $N = 1$, sendo N o número de *codebooks*.

$$c_N = \frac{1}{M} \sum_{m=1}^M t_m \quad (6.3)$$

Em seguida é calculada a dispersão dos vetores t_m com o *codebook* através da Equação 6.4. Essa dispersão é utilizada para mensurar a distância entre os dados e o *codebook* aos quais os dados foram associados.

$$D_{ave} = \frac{1}{Md} \sum_{m=1}^M \|t_m - c_N\|^2 \quad (6.4)$$

Na Figura 6.4(a), pode ser visualizado a inicialização do algoritmo. Os dados de treinamento, neste exemplo em duas dimensões, são os pontos verdes e o *codebook* inicial está em vermelho.

Após a inicialização dessas variáveis é realizado o passo de divisão ou *splitting*. Neste passo, cada *codebook* i é dividido em dois novos *codebooks* com base em um limiar E próximo a 0, conforme as Equações 6.5 e 6.6. Dessa forma o valor de N é atualizado para $N = 2N$. Um exemplo da divisão é ilustrado nas Figuras 6.4(a) e 6.4(b).

$$c_i = (1 + E) * c_i, \quad (6.5)$$

$$c_{N+i} = (1 - E) * c_i, \quad \text{para } i = 1, 2, \dots, N \quad 0 < E \ll 1 \quad (6.6)$$

Após a divisão dos *codebooks* o algoritmo realiza o mapeamento de cada t_m com um *codebook*, sendo realizado pela função Q através da Equação 6.7.

$$Q(t_m) = \min_{k=1}^K \|t_m - c_k\|^2 \quad \text{para } m = 1, 2, \dots, M \quad (6.7)$$

onde $\|\cdot\|$ representa a distância euclidiana.

Em seguida são atualizados os valores dos *codebooks* através da Equação 6.8.

$$c_n = \frac{\sum_{Q(t_m)=c_n} t_m}{\sum_{Q(t_m)=c_n} 1} \quad (6.8)$$

Após a atualização dos *codebooks* é recalculada a distorção entre os vetores e os *codebooks* associados. A distorção é calculada através da Equação 6.9.

$$D_{ave} = \frac{1}{Md} \sum_{m=1}^M \|t_m - Q(t_m)\|^2 \quad (6.9)$$

sendo d a dimensão dos vetores.

Esse processo é realizado iterativamente a partir da Equação 6.7 a 6.9 até que a diferença da distorção atual e anterior esteja dentro do valor do limiar E definido. Caso contrário o procedimento continua desde as equações 6.5 e 6.6 até 6.9 enquanto o valor de N seja o número de discretizações desejadas. Os passos do algoritmo para a discretização com 4 valores são apresentados na Figura 6.4.

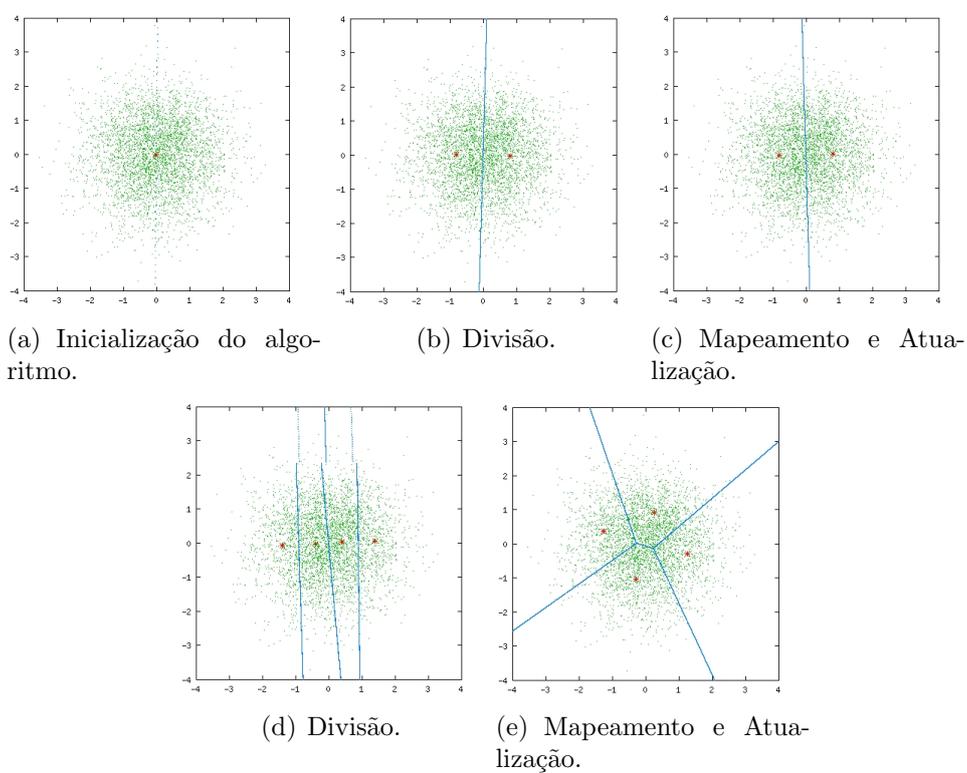


Figura 6.4: Iteração do algoritmo LBG.

Capítulo 7

Experimentos e Resultados

Todos os experimentos foram realizados em três computadores P4 2.8 GHz com 512 MB de memória RAM e sistema operacional Fedora Core 5. O primeiro passo realizado nos experimentos foi a segmentação de todos os vídeos do banco de imagens através das técnicas explicadas no Capítulo 6. Para a segmentação dos vídeos com os camundongos foi necessária a captura de uma imagem de fundo sem o camundongo. Para as serpentes, a SVM foi treinada com 50 amostras para cada espécie de serpente (jararaca, boca de sapo e jibóia) e 100 amostras do plano de fundo. Para a segmentação, uma região em torno da serpente foi manualmente determinada. Essa região foi setada para melhorar o desempenho de processamento do algoritmo. A determinação manual da região não é um grande problema, pois essa determinação pode ser automatizada através dos filtros preditivos. Estes filtros usam um modelo de dinâmica e incerteza para propagar os estados do sistema (e.g. região ao redor da serpente), em seguida o estado predito é corrigido através de um modelo de observação. Os filtros preditivos apresentam muitas vantagens como eficiência em aplicações em tempo real e habilidade de tratar a sobreposição de objetos.

Após a etapa da segmentação, foi realizada a extração de atributos nas imagens. Nesse passo foram utilizados os quatro extratores de atributos, descritos no Capítulo 3: descritores de Fourier, k-curvatura, cadeia de códigos e momentos de imagens. Os parâmetros dos extratores de atributos foram selecionados por experimentos iniciais conforme [dASGM⁺07a]. Para os descritores de Fourier foram utilizados 10 descritores para representar os camundongos ($K = 10$), totalizando 40 atributos. Para o algoritmo de k-curvatura foram configurados os parâmetros da variação dos ângulos de 0 a 180 graus, intervalo entre os ângulos de 10 e k igual a 3, sendo extraídos 18 atributos. Para o algoritmo de cadeia de códigos foram utilizadas as primitivas da conectividade-de-8 para descrever o contorno do camundongo, sendo ex-

traídos 8 atributos. Por fim, o algoritmo de momentos de imagens foi configurado com 4 quadrantes, onde para cada quadrante foram extraídos o centro de massa em x e y , desvio padrão em x e em y e o total de *pixels* pertencentes ao objeto. Além disso, para toda a região do objeto foi extraída, além dos atributos extraídos de cada quadrante, a orientação da elipse que representa o objeto, excentricidade, regressão em A e B , resultando em 29 atributos. Devido à alta dimensão dos atributos, totalizando 95, foi utilizado o algoritmo de análise de componentes principais (PCA) [SIR95] da ferramenta Weka¹ para a seleção dos atributos mais relevantes, resultando em 8 atributos.

Os experimentos com os HMMs foram realizados na comparação entre três tipos de densidade das observações. O primeiro modelo avaliado é o HMM com observações discretas. Esse modelo foi avaliado sobre a taxa de acerto e tempo de execução de acordo com a variação do número de núcleos (2 a 128) utilizados na discretização das observações. A discretização foi realizada através do algoritmo LBG descrito no Capítulo 6 com constante E setada em 0.001 de acordo com experimentos iniciais com a dispersão entre os dados e os *codebooks*. O segundo modelo avaliado foi o HMM com observações contínuas modeladas com uma Gaussiana com parâmetros estabelecidos no processo de aprendizagem. Essa comparação é interessante para avaliar se com apenas uma Gaussiana, a taxa de acerto é aceitável com um tempo de execução teoricamente menor que o tempo de execução com uma mistura de Gaussianas. A taxa de acerto aceitável é considerada como comparável a um observador humano. Segundo Rousseau [RvLM⁺98] essa porcentagem de acerto é de aproximadamente 85% para a classificação de comportamentos de camundongos utilizando redes neurais. O último modelo avaliado foi o HMM com uma mistura de Gaussianas para modelar as observações. Os parâmetros das Gaussianas são estabelecidos de acordo com o Capítulo 4 e avaliado através da variação do número de Gaussianas (2 a 5) pela taxa de acerto e tempo de execução. Dessa forma é possível obter o número de Gaussianas que atenda os requisitos de taxa de acerto em um tempo de execução aceitável. Além disso, foram realizados experimentos sobre o número de estados internos dos HMMs (2 a 5) e o tempo de execução para os três tipos de HMMs. A necessidade da variação desses parâmetros é para se obter os parâmetros que maximize a porcentagem de classificação correta de cada modelo.

Todos os modelos foram treinados utilizando o banco de imagens desenvolvido no Capítulo 5. A separação entre vídeos para o treinamento e para a classificação foi baseada no algoritmo de validação cruzada, onde os dados são divididos em K conjuntos ou pastas. Com isso, um conjunto é utilizado

¹<http://www.cs.waikato.ac.nz/ml/weka/>

para classificação e $K - 1$ conjuntos são utilizados como base de treinamento. O processo é repetido K vezes para que todos os K conjuntos sejam utilizados para classificação. Nesse trabalho, 10 conjuntos foram utilizados no algoritmo de validação cruzada. Após as iterações, é realizada uma média sobre os resultados obtidos.

A inicialização aleatória dos modelos contínuos gerou alguns erros no processo de aprendizagem. Em algumas situações, a probabilidade $b_j(O_{t+1}) = 0$ pois, os modelos das observações de todos os estados foram inicializados muito distante do ideal (aleatório). Com $b_j(O_{t+1}) = 0$, para todos j , o denominador da Equação 4.33 é igual a 0, gerando erro na divisão. Os modelos discretos podem ser inicializados aleatoriamente, pois a probabilidade das observações será sempre diferente de 0 para algum estado. Com isso, para o treinamento dos HMMs foi necessário inicializar seus parâmetros através do algoritmo de k-means. Esta etapa é fundamental nos modelos ocultos de Markov para inicializar os parâmetros dos modelos próximos ao máximo global para posteriormente, na etapa de treinamento seus parâmetros sejam ajustados para maximizar a probabilidade do modelo. O processo de treinamento dos modelos, representando os comportamentos dos camundongos e das serpentes, foi realizado pelo algoritmo *Baum-Welch*, conforme descrito no Capítulo 4. De acordo com os experimentos realizados em [GdASM⁺07, dASGM⁺07b], o número de iterações não influenciaram significativamente para a porcentagem de classificação correta. Com isso, foram utilizadas 1000 iterações no algoritmo de treinamento. Após o processo de treinamento dos modelos, a partir das observações extraídas, foi realizada a etapa de classificação dos comportamentos com base nos algoritmos *Forward-Backward* e validação cruzada. As etapas para a realização dos experimentos podem ser visualizadas na Figura 7.1.

Os resultados para os HMMs discretos são apresentados nos gráficos das Figuras 7.2 e 7.3 para uma média entre o resultado máximo da classificação dos comportamentos de camundongos e serpentes. O gráfico da Figura 7.2(a) apresenta a porcentagem de classificação correta para os números de estados internos. O gráfico mostra uma porcentagem de classificação máxima de 87% com 4 estados. Os resultados mostraram pouca variação na porcentagem de classificação correta, com uma ligeira queda a partir de 5 estados. O tempo de execução, em milisegundos, é apresentado no gráfico da Figura 7.2(b). De acordo com o gráfico, o tempo de execução total (tempo de treinamento e classificação) apresenta uma curva polinomial, isso confirma a teoria com relação à complexidade na ordem de N^2T .

O gráfico da Figura 7.3 apresenta os resultados para o número de núcleos na discretização das observações. De acordo com gráfico 7.3(a), a porcentagem de classificação correta se estabiliza a partir de 64 núcleos de dis-

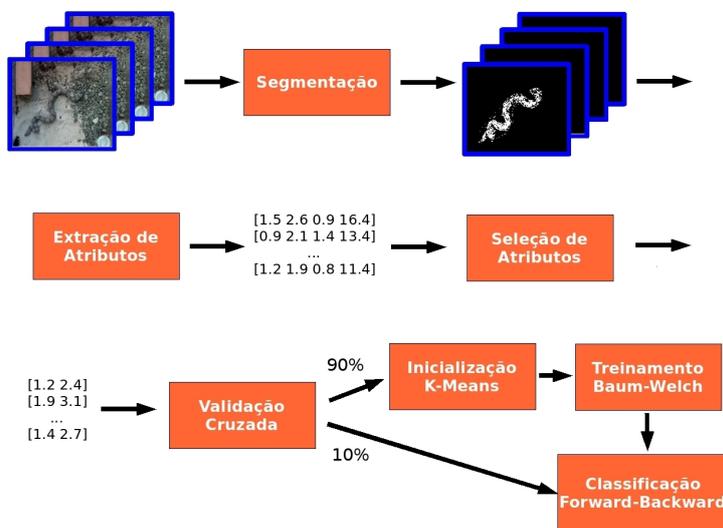
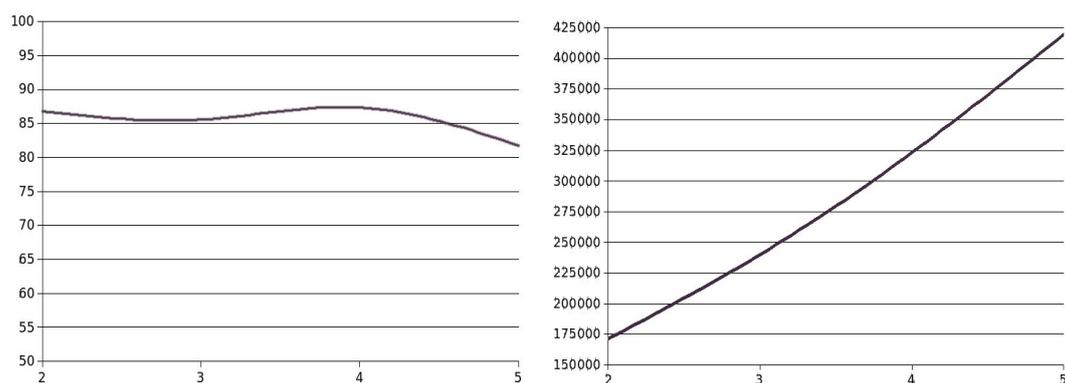


Figura 7.1: Etapas do processo de execução dos experimentos.

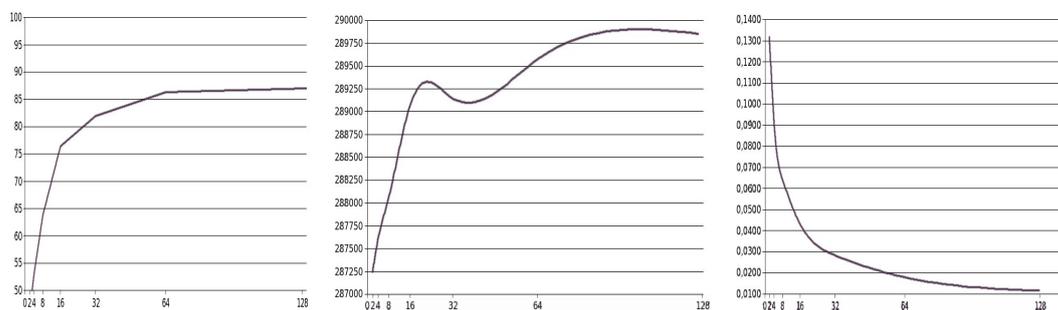


(a) Número de estados internos x Porcentagem de classificação correta. (b) Número de estados internos x Tempo de execução (ms).

Figura 7.2: Resultados para os HMMs discretos em relação ao número de estados internos.

cretização. Isso pode ser verificado através do gráfico da dispersão entre os dados de treinamento e os núcleos, apresentado na Figura 7.3(c). Esse gráfico mostra uma queda exponencial da dispersão, sendo estabilizada a partir de 64 núcleos. O tempo de execução para os núcleos pode ser visualizado no gráfico da Figura 7.3(b).

Através desses experimentos, foi alcançado o número ideal de 4 estados internos e 64 núcleos de discretização para os HMMs discretos. A dificuldade do HMM discreto foi classificar o comportamento de auto-limpeza, visto que,



(a) Número de núcleos x Porcentagem de classificação correta. (b) Número de núcleos x Tempo de execução (ms). (c) Número de núcleos x Dispersão.

Figura 7.3: Resultados para os HMMs discretos em relação ao número de núcleos na discretização.

esses comportamentos são próximos ao comportamento de exploração horizontal. Mesmo assim, o HMM alcançou uma porcentagem de classificação correta de 74%, conforme a Tabela 7.1. O tempo médio de classificação de uma seqüência de observações foi de 275 milisegundos e o tempo médio de treinamento do modelo discreto para 326 seqüências e 1000 iterações foi de 4,67 minutos. Esses dados do tempo de execução foram computados apenas na execução dos algoritmos de treinamento *Baum-Welch* e classificação *Forward-Backward*.

Para os HMMs contínuos modelados com uma Gaussiana os resultados são apresentados no gráfico da Figura 7.4. Através do gráfico da Figura 7.4(a) percebe-se que com 2 estados foi alcançado a porcentagem de classificação correta máxima de 94%. O tempo médio de classificação de uma seqüência de observações para os HMMs com uma Gaussiana foi de 307 milisegundos e o tempo médio de treinamento do modelo para todas as seqüências foi de 9 horas. O alto custo requerido na fase de treinamento pode ser desconsiderado, pois esse treinamento, na maioria das vezes, é realizado em uma fase *off-line*. O tempo total para o número de estados pode ser visualizado pelo gráfico da Figura 7.4(b), mantendo uma ordem polinomial.

Os resultados para os HMMs com mistura de Gaussianas são apresentados nos gráficos das Figuras 7.5 e 7.6. No gráfico da Figura 7.5(a) é apresentado o número de estados internos com relação à porcentagem de classificação correta. A porcentagem máxima de 96% é alcançada com 4 estados internos, concluindo que, com 4 estados internos é possível representar, com um maior desempenho, a variação dos padrões de cada comportamento. O tempo de execução é mostrado no gráfico da Figura 7.5(b). Esse tempo mantém uma

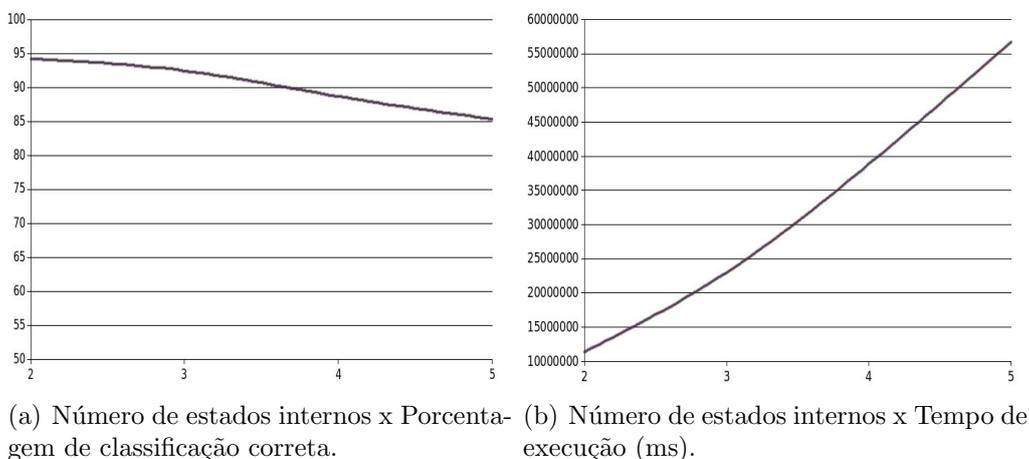


Figura 7.4: Resultados para os HMMs com uma Gaussiana em relação ao número de estados internos.

curva polinomial, assim como nos outros modelos, mas com um tempo de execução muito superior.

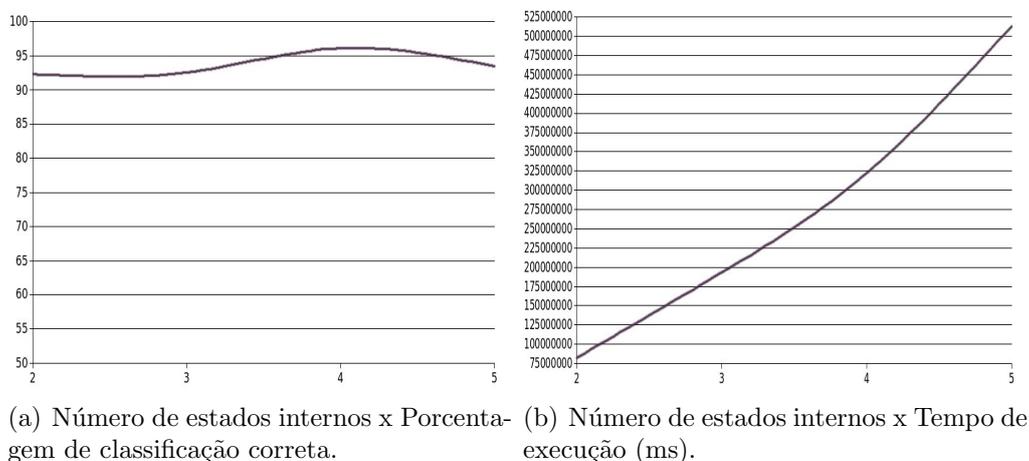
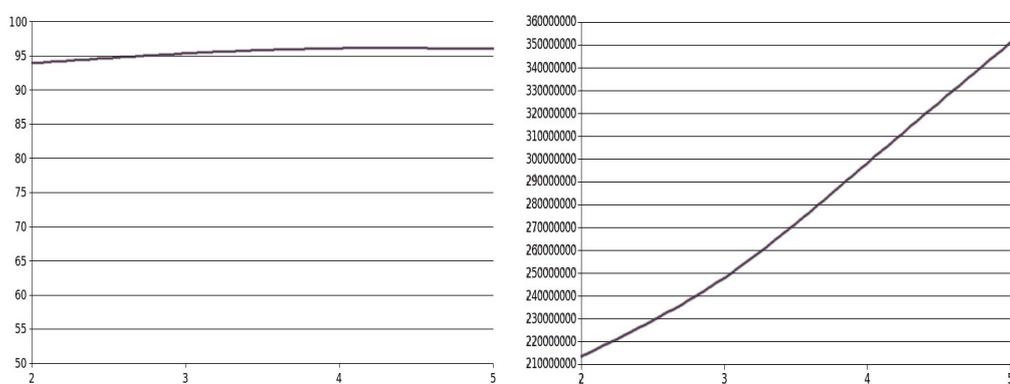


Figura 7.5: Resultados para os HMMs com mistura de Gaussiana em relação ao número de estados internos.

Os resultados com o número de Gaussianas utilizadas na modelagem das observações podem ser visualizados nos gráficos da Figura 7.6. No primeiro gráfico é apresentada a porcentagem de classificação correta, alcançando uma porcentagem máxima de 96% com 4 Gaussianas e mantendo essa porcentagem com 5 Gaussianas. Conclui-se que, a distribuição das observações possivelmente possui 4 modas. A visualização dessas modas é impossível pois

a dimensão dos atributos é superior a 3 dimensões. No segundo gráfico é apresentado o tempo de execução. À medida que aumentamos o número de Gaussianas percebe-se um crescente aumento no tempo de execução de forma polinomial. Isto é evidente devido ao fato de ao adicionarmos mais Gaussianas para modelar as distribuições das observações mais tempo de processamento é requerido na fase de treinamento e classificação do modelo. Para os HMMs com mistura de Gaussianas, o tempo médio de classificação de uma seqüência de observações foi de 704 milissegundos e o tempo médio de treinamento do modelo para todas as seqüências foi de 77 horas.



(a) Número de Gaussianas x Porcentagem de classificação correta. (b) Número de Gaussianas x Tempo de execução (ms).

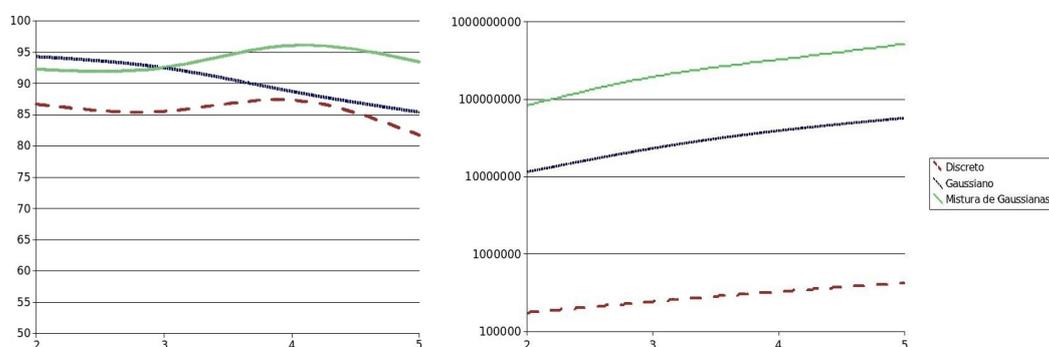
Figura 7.6: Resultados para os HMMs com mistura de Gaussiana em relação ao número de Gaussianas.

Por fim, são apresentados gráficos para a comparação entre os três modelos. No gráfico da Figura 7.7(a) é apresentada uma comparação entre o número de estados pela porcentagem de classificação dos três modelos. Podemos concluir que, o modelo com uma Gaussiana obtém resultados melhores que o modelo discreto. Além disso, o modelo Gaussiano obtém resultados satisfatórios com relação ao modelo com misturas de Gaussianas, principalmente com poucos estados e com relação ao tempo de execução, mostrado no gráfico da Figura 7.7(b). A grande vantagem do modelo discreto é o seu tempo de execução (Figura 7.7(b)) comparado ao alto custo dos outros modelos, principalmente com relação ao tempo de treinamento. Os modelos com mistura de Gaussianas alcançaram as melhores porcentagens de classificação para os comportamentos, entretanto, essas porcentagens foram pouco superiores ao modelo Gaussiano com um tempo de treinamento muito superior. Com isso, os experimentos mostraram que, para esse problema, o modelo Gaussiano é uma opção satisfatória e viável. A tabela 7.1 apresenta a porcen-

| Comportamentos | Discreto | Gaussiano | Mistura de Gaussianas |
|--------------------|----------|-----------|-----------------------|
| Vertical | 100% | 100% | 100% |
| Horizontal | 96,03% | 100% | 100% |
| Auto-limpeza | 74,14% | 82,67% | 84,17% |
| Ataque da Serpente | 84,21% | 90,27% | 91,54% |

Tabela 7.1: Classificação dos comportamentos individuais dos modelos

tagem de classificação correta máxima para os comportamentos individuais de cada modelo.



(a) Tipos de modelos x Porcentagem de classificação correta. (b) Tipo de modelos x Tempo de execução (escala de classificação correta). (escala logarítmica).

Figura 7.7: Resultados de comparação entre os tipos de HMMs.

O comportamento de exploração vertical foi corretamente classificado pelos modelos. O comportamento de auto-limpeza obteve uma porcentagem de classificação menor, devido à semelhança com o comportamento horizontal, como pode ser visto em uma das Tabelas de confusão 7.2 resultante de um experimento. Para as serpentes, a matriz de confusão é apresentada na Tabela 7.3. Nessa matriz, podemos ver que o verdadeiro positivo, representado pelo comportamento de ataque classificado como ataque, possui uma porcentagem máxima. Porém, os experimentos apresentaram 11% para o falso negativo (não-ataque classificado com ataque) e 89% para o verdadeiro negativo (não-ataque classificado como não-ataque).

| | Vertical | Horizontal | Auto-limpeza |
|--------------|-----------------|-------------------|---------------------|
| Vertical | 100% | 0% | 0% |
| Horizontal | 0% | 100% | 0% |
| Auto-limpeza | 0% | 16% | 84% |

Tabela 7.2: Matriz de Confusão dos comportamentos de camundongos.

| | Ataque | Não Ataque |
|------------|---------------|-------------------|
| Ataque | 100% | 0% |
| Não Ataque | 11% | 89% |

Tabela 7.3: Matriz de Confusão dos comportamentos de serpentes.

Capítulo 8

Conclusão e Trabalhos Futuros

A análise do comportamento animal é extremamente importante em diversas pesquisas, como a avaliação de novos fármacos e a estimativa da degradação do meio ambiente. A identificação comportamental de animais é realizada na maioria das vezes de forma manual, enquanto que, os sistemas automáticos apresentam, geralmente, soluções apenas para valores relacionados com medidas espaciais, como a distância percorrida, velocidade e aceleração. Entretanto, existem pesquisas que necessitam mensurar atividades exploratórias, como a exploração vertical.

Neste projeto foi investigada a modelagem dos comportamentos de exploração vertical, horizontal e auto-limpeza para os camundongos e o comportamento de ataque para as serpentes através dos modelos ocultos de Markov. Os modelos foram utilizados para classificar os vídeos capturados por uma *webcam* em um dos possíveis comportamentos com o auxílio da biblioteca Jahmm. Os HMMs foram avaliados sobre três tipos de distribuições para as observações: discretas, contínuas com uma Gaussiana e com mistura de Gaussianas. Para a avaliação dos HMMs foi desenvolvido um banco de imagens composto de diversos comportamentos de camundongos e serpentes. Além disso, para essas imagens, foram extraídos diversos atributos para descrever o contorno e a região de objetos. O algoritmo de análise de componentes principais foi aplicado nos atributos extraídos devido à alta dimensão do vetor.

Os resultados da avaliação mostraram que os HMMs com mistura de Gaussianas obtêm melhores resultados em relação aos outros modelos. Entretanto, o HMM com uma Gaussiana apresenta resultados um pouco inferior ao modelo com mistura de Gaussianas com um tempo de execução relativamente menor. O modelo discreto mostrou resultados satisfatórios, segundo a avaliação humana [RvLM⁺98], com um tempo de execução consideravelmente menor e aplicável em tempo real. Os comportamentos foram classifi-

cados com uma porcentagem de classificação correta de 96% com um HMM com mistura de Gaussianas, 94% para um HMM com uma Gaussiana e 87% para um HMM discreto.

Como trabalhos futuros, seria interessante aplicar os HMMs para mais comportamentos com diferentes animais e seres humanos. Para melhorar os HMMs discretos, é necessário investigar novos algoritmos de discretização, uma linha interessante seria aplicar algoritmos de aprendizagem não supervisionada para discretizar as observações. A implementação e avaliação de novos extratores de atributos para melhorar a discrepância entre o comportamento de auto-limpeza e exploração horizontal são necessárias. Outra sugestão como trabalhos futuros seria a avaliação dos tipos de topologia dos HMMs, como modelo esquerdo-direita, onde a probabilidade de transição para um estado anterior é nula. Por fim, precisam ser realizados alguns procedimentos para a utilização dos HMMs em tempo real, como a determinação automática do quadro de início e fim de cada comportamento. Uma solução existente seria definir um tamanho fixo para cada comportamento determinado pela média das seqüências do conjunto de treinamento.

Apêndice A

Fundamentação Teórica

Neste apêndice são descritos os conceitos básicos para o entendimento deste trabalho [Fer73, Mor99]. Os conceitos de probabilidade e estatística são de fundamental importância para a base dos modelos ocultos de Markov. Entre os principais conceitos está o teorema de Bayes, probabilidade condicional e distribuição Gaussiana.

A.1 Variáveis Aleatórias

Uma variável aleatória discreta x é uma variável que pode assumir um valor referente ao conjunto $X = \{x_1, x_2, \dots, x_n\}$. A probabilidade p_i de uma variável aleatória x assumir um valor x_i é definida na Equação A.1.

$$p_i = p(x = x_i) = \frac{s}{n}, \quad i = 1 \dots n \quad (\text{A.1})$$

onde s é o número de ocorrências de um evento e n é o número total de ocorrências do conjunto de eventos. O conjunto de probabilidades $p_i, i = 1 \dots n$ descreve a distribuição de probabilidades de uma variável aleatória discreta.

Como exemplo, considere a probabilidade de ocorrência dos números através de lançamentos de um dado. O conjunto de valores que a variável aleatória x pode assumir é definido pelo conjunto $X = \{1, 2, 3, 4, 5, 6\}$. A distribuição de probabilidades é definida como $\{p_1 = \frac{1}{6}, p_2 = \frac{1}{6}, p_3 = \frac{1}{6}, p_4 = \frac{1}{6}, p_5 = \frac{1}{6}, p_6 = \frac{1}{6}\}$.

As variáveis aleatórias contínuas podem assumir um valor em um determinado intervalo no domínio contínuo. Como exemplo, considere uma variável aleatória contínua x que represente a área de um objeto. Neste caso, a variável x pode assumir qualquer valor entre o intervalo $[1, \infty]$, podendo assumir uma infinidade de possibilidades.

| | Engenharia | Psicologia |
|--------|------------|------------|
| Homem | 10 | 3 |
| Mulher | 2 | 20 |

Tabela A.1: Tabela de alunos matriculados em dois cursos.

A.2 Probabilidade Conjunta e Condicional

A probabilidade conjunta entre duas variáveis aleatórias é a probabilidade de ocorrência de ambas variáveis ao mesmo tempo. A probabilidade conjunta entre x e y é definida como $p_{ij} = p(x = x_i, y = y_j)$. Desta forma é possível determinar a função de probabilidades conjuntas $p(x, y)$ para cada par (x_i, y_j) , com $i = 1, \dots, n$ e $j = 1, \dots, m$.

Em diversos experimentos aplicáveis a uma variável aleatória é possível obter informações adicionais sobre a sua ocorrência, sendo importante na maioria das vezes, considerarmos essa informação no cálculo da probabilidade. A probabilidade condicional envolve duas variáveis x e y , sendo possível estimar com maior precisão o valor da variável x dado o valor condicionante de uma variável y que já foi observada, expressa na Equação A.2.

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad (\text{A.2})$$

Como exemplo, considere os alunos de ambos os sexos matriculados em dois cursos, descritos na Tabela A.1.

A probabilidade de ocorrer um homem sabendo que ele estuda no curso de engenharia é dada por $p(x = \text{homem} | y = \text{Engenharia})$ descrita na Equação A.3.

$$p(x = \text{homem} | y = \text{Engenharia}) = \frac{\frac{10}{35}}{\frac{12}{35}} = 0,83 = 83\%. \quad (\text{A.3})$$

A.3 Independência Estatística

Calcular a independência estatística entre variáveis é de suma importância na computação, mas demonstrar que duas variáveis são realmente independentes não é uma tarefa simples. Através da independência é possível diminuir os cálculos e a quantidade de memória necessária para armazenar as probabilidades.

Dois variáveis aleatórias x e y , representando o dia do mês e a previsão de chuva por exemplo, são estatisticamente independentes se e somente se

$p(x, y) = p(x)p(y)$. Em outras palavras, se as duas variáveis são independentes, é possível expressar a probabilidade conjunta como o produto da probabilidade individual de cada variável. Da mesma forma, a probabilidade condicional entre essas variáveis é dada por $p(x|y) = p(x)$, visto que, a variável x não depende estatisticamente da variável y .

A.4 Teorema de Bayes

O teorema de Bayes representa a probabilidade condicional e a probabilidade marginal entre duas variáveis aleatórias. A probabilidade de uma variável x condicionada a uma variável y pode ser diferente da probabilidade da variável y condicionada à variável x . Entretanto, existe uma relação entre as duas probabilidades e conseqüentemente entre as variáveis. O teorema de Bayes busca representar essa relação através da Equação A.4.

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (\text{A.4})$$

onde $p(x)$ é a probabilidade a priori da variável x . A priori significa que essa variável não considera nenhuma informação sobre outras variáveis aleatórias. As probabilidades $p(x|y)$ e $p(y|x)$ são as probabilidades condicionais ou posteriores, pois as variáveis estão condicionadas a outra variável aleatória. A probabilidade $p(y)$ é a probabilidade a priori de variável y , atuando como uma constante de normalização.

O teorema é derivado através das probabilidades condicionais $p(x|y)$ e $p(y|x)$ de duas variáveis aleatórias, descritas nas Equações A.5 e A.6.

$$p(x|y) = \frac{p(x, y)}{p(y)} \quad (\text{A.5})$$

$$p(y|x) = \frac{p(y, x)}{p(x)} = \frac{p(x, y)}{p(x)} \quad (\text{A.6})$$

Reorganizando ambas as probabilidades condicionais são obtidas as Equações A.7 e A.8.

$$p(x, y) = p(x|y)p(y) \quad (\text{A.7})$$

$$p(x, y) = p(y|x)p(x) \quad (\text{A.8})$$

Substituindo a Equação A.7 na Equação A.8 é obtido o teorema de Bayes.

$$p(x|y)p(y) = p(y|x)p(x) \quad (\text{A.9})$$

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad (\text{A.10})$$

A.5 Distribuição Gaussiana

A distribuição Gaussiana pode ser aplicada como uma técnica de reconhecimento de padrões paramétrica. As distribuições paramétricas são definidas por um conjunto de parâmetros, sendo estimados para descrever o conjunto de dados. A distribuição Gaussiana é definida como $\eta(\mu, \Sigma)$, modelando as classes de um problema através de dois parâmetros. Os parâmetros da Gaussiana são estimados através de um conjunto de treinamento, calculado na Equação A.11.

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad \Sigma = \frac{1}{n} \sum_{i=1}^n [x_i - \mu][x_i - \mu]^T \quad (\text{A.11})$$

onde μ representa a média, Σ representa a matriz de covariância, x representa um vetor do conjunto de treinamento e n a quantidade de vetores de treinamento.

A probabilidade de um vetor x pertencer a uma Gaussiana é dada pela Equação A.12.

$$\eta(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp^{-\frac{1}{2}(x-\mu)^T \Sigma (x-\mu)} \quad (\text{A.12})$$

sendo d a dimensão do vetor e $|\Sigma|$ o determinante da matriz.

A.5.1 Mistura de Gaussianas

O vetor de atributos extraídos de um objeto pode possuir uma distribuição multimodal. Com isso, a utilização de uma única Gaussiana pode não representar o conjunto de dados de maneira eficiente. Dada esta situação, é desejável a utilização de mistura de Gaussianas para modelar um determinado conjunto de vetores de atributos.

A probabilidade de um vetor x pertencer a um conjunto de mistura de Gaussianas é dada pela Equação A.13.

$$P(x|X) = \sum_{k=1}^K \omega_k * \eta(x; \mu_k, \Sigma_k) \quad (\text{A.13})$$

onde X é o conjunto de treinamento, K é o número de Gaussianas, ω_k é o peso associado a k -ésima distribuição, μ_k é a média, Σ_k é a matriz de covariância da k -ésima distribuição e η é função de densidade de probabilidade, dada pela Equação A.12.

Para calcular os parâmetros de uma mistura de Gaussianas pode-se utilizar o algoritmo de maximização de expectância (EM)[Del02]. Este algoritmo é utilizado para encontrar estimativas máximas da probabilidade de um conjunto de parâmetros em modelos probabilísticos.

Apêndice B

Artigos Publicados

Os artigos abaixo apresentam a aplicação dos modelos ocultos de Markov discretos (HMMs) no reconhecimento de comportamentos de serpentes e camundongos.

- GONÇALVES, W. N.; SILVA, J. A.; MACHADO, B. B.; PISTORI, H.; SOUZA, A. S. **Hidden Markov Models Applied To Snakes Behavior Identification**. Lecture Notes in Computer Science, v. A, p. 1, 2007.
- GONÇALVES, W. N.; SILVA, J. A.; MACHADO, B. B.; PISTORI, H.; SOUZA, A. S. **Modelos Ocultos de Markov Aplicados na Identificação de Comportamento de Serpentes**. In: III WVC - Workshop de Visão Computacional, 2007, São José do Rio Preto. III WVC - Workshop de Visão Computacional, 2007.
- SILVA, J. A.; GONÇALVES, W. N.; MACHADO, B. B.; PISTORI, H.; SOUZA, A. S. **Modelos Ocultos de Markov Aplicados na Identificação de Comportamento de Camundongos**. Hifen (Uruguaiana), v. A, p. 1, 2007.

No trabalho abaixo foram comparados dois grupos de técnicas de segmentação de imagens, o primeiro baseado em subtração de fundo e o segundo em aprendizagem supervisionada de modelos de cores. As comparações foram realizadas com base em imagens reais, segmentadas manualmente, referentes a dois importantes problemas que vêm sendo estudados por diversos grupos de pesquisa em visão computacional: a interpretação de língua gestual de sinais e a análise de comportamentos de camundongos em experimentos na área da saúde. As métricas utilizadas na comparação foram baseadas no trabalho de M. Sezgin e B. SanKur, e que consideram, além da taxa de

acerto na classificação de cada *pixel*, informações relacionadas com as bordas e a forma geral do objeto segmentado em comparação com a imagem segmentada manualmente (ground truth).

- GONÇALVES, W. N.; Silva, V. A. S. da; MACHADO, B. B.; SILVA, J. A.; SOUZA, K. P. DE; PISTORI, H. **Técnicas de Segmentação baseadas em Subtração de Fundo e Modelos de Cores: Um Estudo Comparativo.** In: XXVIII CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering, 2007, Porto. XXVIII CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering, 2007.

O artigo abaixo apresenta uma nova abordagem para o rastreamento de múltiplos objetos que combina filtro de partículas e k-means. A abordagem foi testada no âmbito de uma importante situação do mundo real, a utilização de fármacos em camundongos, que revelou ser um interessante problema para a avaliação dos algoritmos de rastreamento. Os resultados obtidos são comparados com outros modelos.

- GONÇALVES, W. N.; MONTEIRO, J. B. O.; MACHADO, B. B.; SILVA, J. A.; ODAKURA, V. V. V. A.; PISTORI, H. **Multiple Mice Tracking using a Combination of Particle Filter and K-Means.** In: SIBGRAPI - Brazilian Symposium on Computer Graphics and Image Processing, 2007, Belo Horizonte. IEEE Computer Society, 2007.

No artigo abaixo é realizado um estudo comparativo entre os descritores de formas baseados em contorno e região. Na abordagem por contorno, foram avaliadas as cadeias de códigos, k-curvatura e descritores de Fourier, e na abordagem por região utilizou-se momentos de imagem. A classificação é baseada pela extração de características das formas. Estes descritores foram aplicados ao problema de identificação do comportamento de camundongos. A avaliação foi realizada através da variação dos parâmetros de cada técnica e na acurácia dos classificadores, alcançando 92.6% de acerto a técnica momentos de imagem.

- SILVA, J. A.; GONÇALVES, W. N.; MACHADO, B. B.; PISTORI, H.; SOUZA, A. S. **Comparação de Descritores de Formas no Reconhecimento de Objetos.** In: III WVC - Workshop de Visão Computacional, 2007, São José do Rio Preto. III WVC - Workshop de Visão Computacional, 2007.

Referências Bibliográficas

- [ALT95] Kjersti Aas, Eikvil Line, and Andersen Tove. Text recognition from grey level images using hidden Markov models. *Lecture Notes in Computer Science*, 970:503–508, 1995.
- [BKS05] Faisal Bashir, Ashfaq Khokhar, and Dan Schonfeld. Hmm-based motion recognition system using segmented pca. *Proc. IEEE International Conference on Image Processing*, 2005.
- [BS02] Pawel Boguszewski and Jolanta Zagrodzka Szmagalska. Emotional changes related to age in rats a behavioral analysis. *Behavioural Brain Research*, 133:332–332, 2002.
- [Bur98] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [CL06] Tais H. F. Carvalho and Oswaldo U. Lopes. O emprego de camundongo geneticamente modificado como modelo de estudo para doenças cardiovasculares. In *X Simpósio Brasileiro de Fisiologia Cardiovascular*, volume 39, pages 110–116, Ribeirão Preto, Brasil, 2006.
- [dA04] Marcos Carneiro de Andrade. An interactive algorithm for image smoothing and segmentation. *ELCVIA*, 4(1):32–48, October 2004.
- [dASGM⁺07a] Jonathan de Andrade Silva, Wesley Nunes Gonçalves, Bruno Brandoli Machado, Hemerson Pistori, and Albert Schiaveto de Souza. Comparação de descritores de formas no reconhecimento de objetos. *III WVC - Workshop de Visão Computacional*, October 2007.
- [dASGM⁺07b] Jonathan de Andrade Silva, Wesley Nunes Gonçalves, Bruno Brandoli Machado, Hemerson Pistori, and Albert Schi-

- aveto de Souza. Modelos ocultos de markov aplicados na identificação de comportamento de camundongos. *Hífen*, November 2007.
- [DB05] Padmaja Dukkupati and Leonard Brown. Improving the recognition of geometrical shapes in road signs by augmenting the database. *ICCSA International Conference on Computer Science and its Applications*, V2, June 2005.
- [Del02] F. Dellaert. The expectation maximization algorithm. Technical Report GIT-GVU-02-20, GVU, February 2002.
- [dSP05] Kleber Padovani de Souza and Hemerson Pistori. Implementação de um extrator de características baseado em momentos da imagem. In *Proceedings of SIBGRAPI 2005 - XVIII Brazilian Symposium on Computer Graphics and Image Processing*, Natal, RN, Brazil, October 2005. SBC, IEEE Press.
- [Eil03] David Eilam. Open-field behavior withstands drastic changes in arena size. *Behavioural Brain Research*, 142:53–62, 2003.
- [FB03] Adam Feldman and Tucker Balch. Automatic identification of bee movement. Technical report, Georgia Institute of Technology, Atlanta, Georgia 30332, USA, 2003.
- [Fer73] Pedro Jesus Fernandez. *Introdução a teoria das probabilidades*. Rio de Janeiro; LTC, 1973.
- [Fre74] Herbert Freeman. Computer processing of line-drawing images. *ACM Computing Surveys*, 6(1):57–97, March 1974.
- [FT04] Djalma José Fagundes and Murched Omar Taha. Modelo animal de doença: critérios de escolha e espécies de animais de uso corrente. *Acta Cirúrgica Brasileira*, 19:59–65, 2004.
- [FTO96] W. Freeman, K. Tanaka, and J. Ohta. Computer vision for computer games. In *Int'l Workshop on Automatic Face- and Gesture-Recognition*, Killington, Vermont, USA, 1996.
- [GdASM⁺07] Wesley Nunes Gonçalves, Jonathan de Andrade Silva, Bruno Brandoli Machado, Hemerson Pistori, and Albert Schiaveto de Souza. Hidden markov models applied to snakes behavior identification. *Lecture Notes in Computer Science*, December 2007.

- [GH95] C. A. Glasbey and G. W. Horgan. *Image analysis for the biological sciences*. John Wiley & Sons, Inc., New York, NY, USA, 1995.
- [GSM⁺07] Wesley Nunes Gonçalves, Vinícius Assis Saueia, Bruno Brandoli Machado, Jonathan Andrade Silva, Kleber Padovani de Souza, and Hemerson Pistori. Técnicas de segmentação baseadas em subtração de fundo e modelos de cores: Um estudo comparativo. *XXVIII CILAMCE - Iberian Latin American Congress on Computational Methods in Engineering*, pages 13–15, Junho 2007.
- [HBT96] Jianying Hu, Michael K. Brown, and William Turin. Hmm based on-line handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(10):1039–1045, 1996.
- [HNPS94] Joachim Hornegger, Heinrich Niemann, Dietrich Paulus, and Gero Schlottke. Object recognition using Hidden Markov Models. In E. S. Gelsema and L. N. Kanal, editors, *Pattern Recognition in Practice IV: Multiple Paradigms, Comparative Studies and Hybrid Systems*, volume 16, pages 37–44, Amsterdam, 1994. Elsevier.
- [IV96] Jukka Iivarinen and Ari Visa. Shape recognition of irregular objects. In D. P. Casasent, editor, *Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, Proc. SPIE 2904*, pages 25–32, 1996.
- [McI00] Alan McIvor. Background subtraction techniques. In *Proceedings of Image and Vision Computing*, 2000.
- [MdASG⁺06] Bruno Brandoli Machado, Jonathan de Andrade Silva, Wesley Nunes Gonçalves, Hemerson Pistori, and Albert Schiaveto de Souza. Topolino: Software livre para automatização do experimento do campo aberto. *XV Seminário de Computação*, pages 19–28, November 2006.
- [Mor84] R. Morris. Developments of a water-maze procedure for studying spatial learning in the rat. *Journal of Neuroscience Methods*, 11(1):47–60, 1984.
- [Mor99] Luiz Gonzaga Morettin. *Estatística básica: Probabilidade*. Makron Books, 1999.

- [MS04] José Antonio Montero and Luis Enrique Sucar. Feature selection for visual gesture recognition using hidden markov models. In *ENC*, pages 196–203, 2004.
- [MW07] Dariusz Malyszko and Slawomir T. Wierzchon. Standard and genetic k-means clustering techniques in image segmentation. *cisim*, 0:299–304, 2007.
- [NH98] Ara V. Nefian and Monson H. Hayes. Face detection and recognition using hidden markov models. In *ICIP (1)*, pages 141–145, 1998.
- [Rab90] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1990.
- [RB05] Jesús A. Rivas and Gordon M. Burghardt. Snake mating systems, behavior, and evolution: The revisionary implications of recent findings. *Journal of Comparative Psychology*, 119(4):447–454, 2005.
- [RvLM⁺98] J B I Rousseau, P B A van Lochem, W Melder, Christophe Costa Florêncio, and Willem Hendrik Gispen. Classification of rat behavior by a neural network. *2 International Conference on Methods and Techniques in Behavioral Research*, pages 18–21, 1998.
- [SIR95] Heung-Yeung Shum, Katsushi Ikeuchi, and Raj Reddy. Principal component analysis with missing data and its application to polyhedral object modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):854–867, September 1995. note that this paper has been previously published in CVPR’94.
- [SP95] Thad Starner and Alex Pentland. Visual recognition of american sign language using hidden markov models. Technical Report Master’s Thesis, MIT, Program in Media Arts & Sciences, Massachusetts Institute of Technology, Cambridge, USA, Feb 1995.
- [SSE05] Abdel-Badeeh M. Salem, Adel A. Sewisy, and Usama A. Elyan. A vertex chain code approach for image recognition. *ICGST International Journal on Graphics, Vision and Image Processing*, 05, 2005.

- [STBN01] A. J. Spink, R. A. J. Tegelenbosch, M. O. S. Buma, and L. P. J. J. Noldus. The ethovision video tracking system—a tool for behavioral phenotyping of transgenic mice. *Physiology and Behavior*, 73(5):731–744, August 2001.
- [Tru05] Michel Alain Quintana Truyenque. Uma aplicação de visão computacional que utiliza gestos da mão para interagir com o computador. Master’s thesis, Pontifícia Universidade Católica - PUC-Rio, Rio de Janeiro-RJ, Brasil, 2005.
- [WHK99] Ian Q. Wishaw, Forrest Haun, and Bryan Kolb. Analysis of behavior in laboratory rodents. In *Modern techniques in neuroscience*, pages 1243–1275, Germany, 1999. Springer.