

Igor Pereira Gomes

**Previsão de Eventos e Localização  
Não-Supervisionada de Pessoas em Ambientes  
Inteligentes**

Belo Horizonte

2018



Igor Pereira Gomes

# **Previsão de Eventos e Localização Não-Supervisionada de Pessoas em Ambientes Inteligentes**

Rascunho inicial de Dissertação. Ainda não  
será apresentado à banca.

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Prof. Antônio de Pádua Braga

Belo Horizonte

2018



*Este trabalho é dedicado à Escola de Engenharia da UFMG  
e a todos que lutam por sua excelência.*



# Agradecimentos

Agradeço enormemente ao professor Antônio de Pádua Braga pelo apoio, orientação e auxílio que me foram preciosos.

Também essencial foi apoio dos professores Cristiano Leite de Castro e Hani Camille Yehia, da UFMG, e de Marco Túlio Sousa e Jullierme Dias, da empresa Neocontrol.





*“Casa não é o lugar onde você vive,  
e sim, o lugar onde te entendem.  
(Christian Morgenstern)*



# Resumo

Este trabalho apresenta informações da implementação realizada para um sistema de coleta e armazenamento de dados de utilização de uma residência utilizando o sistema de automação residencial *Minibox*, da empresa brasileira Neocontrol. Também apresenta e testa métodos para a análise e modelagem de eventos a partir destes dados, com foco nas informações sobre entrada, saída e ocupação da casa. Isto é feito de forma não-supervisionada através de modelos estatísticos (Modelo Oculto de Markov), que gera dados para alimentar um classificador (N-Gramas + SVM), que trabalha com um espaço amostral completamente binário, viabilizando sua implementação na eletrônica embarcada nos sistemas Neocontrol.

**Palavras-chaves:** Inteligência Artificial. Hidden Markov Model. Reconhecimento de Padrões. Smart Home. Automação Residencial. Sistemas Inteligentes.



# Lista de ilustrações

Figura 1 – Visualização dos sinais temporais correspondentes a uma mesma rota percorrida duas vezes. . . . .	24
Figura 2 – Visualização dos sinais da figura 1 convoluídos com função linear. . . .	25
Figura 3 – Visualização de dois pontos $a$ e $b$ vizinhos em um Grafo de Gabriel. . .	25
Figura 4 – Visualização de dois pontos $a$ e $b$ não-vizinhos em um Grafo de Gabriel devido à presença do ponto $c$ . . . . .	26
Figura 5 – Planta da residência e localização dos sensores para o banco de testes. .	31
Figura 6 – Visualização para o Teste de Friedman com teste post-hoc de Bonferroni	38
Figura 7 – Visualização da superfície de separação para o CHIP-CLAS original . .	39
Figura 8 – Visualização da superfície de separação para o AM-CHIP-CLAS . . . .	39
Figura 9 – Visualização para o teste <i>post-hoc</i> de Bonferroni-Dunn . . . . .	41



# Lista de abreviaturas e siglas

TCP	Transfer Control Protocol
IP	Internet Protocol
IoT	Internet of Things
SVM	Máquina de Vetor de Suporte
RF	Random Forest
MQTT	Message Queue Telemetry Transport
SQL	Standard Query Language
RAM	Memória de Acesso Aleatório
HMM	Modelo Oculto de Markov





# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>17</b>
1.1	Motivação	17
1.2	Objetivos	17
1.3	Contribuições	18
1.4	Estrutura do Trabalho	18
<b>2</b>	<b>CONTEXTUALIZAÇÃO</b>	<b>21</b>
2.1	Bases de Dados de Smart Homes	21
2.2	Reconhecimento de Atividades em Smart Homes	22
2.3	Previsão de Eventos em Smart Homes	22
2.4	Localização de Usuários	22
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>23</b>
3.1	Classificação Supervisionada de Sequências Temporais	23
3.2	Extração de Características de Sequências Temporais Categóricas	23
3.2.1	Contagem de Elementos	23
3.2.2	N-Gramas	24
3.2.3	Convolução com Função Linear	24
3.3	Métodos para Classificação Supervisionada	25
3.3.1	Máquina de Vetores de Suporte	25
3.3.2	Classificadores por Arestas de Suporte (CLAS)	25
3.4	Aprendizado de Métrica para problemas de Classificação	26
3.4.1	Distância de Mahalanobis	26
3.4.2	Large Margin Nearest Neighbors (LMNN)	27
3.5	Modelo Oculto de Markov	28
3.5.1	Problema da Decodificação	29
3.5.2	Problema do Aprendizado	29
<b>4</b>	<b>METODOLOGIA</b>	<b>31</b>
4.1	Banco de Testes	31
4.2	Coleta, Armazenamento e Apresentação dos Dados	31
4.3	Previsão de Eventos	32
4.4	Extração de Características	32
4.4.1	Contagem de Acionamentos	32
4.4.2	N-Gramas	32
4.4.3	Convolução com Função Linear	33

<b>4.5</b>	<b>Redução de Dimensionalidade</b>	<b>33</b>
<b>4.6</b>	<b>Classificação</b>	<b>34</b>
4.6.1	AM-CHIP-CLAS	34
<b>4.7</b>	<b>Modelo Não-Supervisionado para Tracking na Smart Home</b>	<b>35</b>
4.7.1	Modelo Oculto de Markov	35
4.7.2	Extensão do modelo para Múltiplos Residentes	36
<b>5</b>	<b>EXPERIMENTOS E RESULTADOS</b>	<b>37</b>
<b>5.1</b>	<b>Recursos Computacionais Utilizados</b>	<b>37</b>
<b>5.2</b>	<b>Validação da Extração de Características para Previsão de Eventos</b>	<b>37</b>
5.2.1	Resultados	37
5.2.2	Discussão	38
<b>5.3</b>	<b>Validação da abordagem AM-CHIP-CLAS para Classificação</b>	<b>38</b>
5.3.1	Visualização da Superfície de Separação	39
5.3.2	Utilização dos Dados	39
5.3.3	Desempenho	39
5.3.4	Discussão	41
<b>5.4</b>	<b>Arquitetura Convolução + PCA + AM-CHIP-CLAS para os dados da Neocontrol</b>	<b>41</b>
<b>5.5</b>	<b>Validação dos Modelos para múltiplos residentes</b>	<b>41</b>
5.5.1	Dados para Validação	41
5.5.2	Resultados	41
<b>5.6</b>	<b>Validação da Extração de Características</b>	<b>42</b>
5.6.1	Dados para Validação	42
5.6.2	N-Gramas	42
5.6.3	Convolução	42
<b>5.7</b>	<b>Validação do Algoritmo de Classificação AM-CHIP-CLAS</b>	<b>42</b>
<b>5.8</b>	<b>Trabalhos Futuros</b>	<b>42</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>43</b>
	<b>REFERÊNCIAS</b>	<b>45</b>

# 1 Introdução

## 1.1 Motivação

Em 1998, já eram delimitados os problemas a serem resolvidos para o desenvolvimento de um sistema inteligente e adaptativo integrado com residências, com previsões de que não demorariam a surgir utensílios domésticos equipados com processadores que se comunicariam entre si e tomariam decisões, como por exemplo, uma lavadora de louças que se comunica com o aquecedor de água ou aparelhos de entretenimento que reagissem à presença do morador (MOZER, 1998). Mesmo antes, uma proto-inteligência já estava sendo concebida, com sistemas de controle de energia para utensílios domésticos já sendo desenvolvidos por pesquisadores (HUNT; HOLMES, 1986) e até mesmo presentes em patentes (CARR et al., 1987).

Na atualidade, as barreiras tecnológicas para a concepção e implementação *Smart Homes* já tem sido rompidas por diversas universidades, com diversos experimentos de sucesso. No contexto brasileiro, porém, ainda resta para ser rompida a barreira que separa academia e indústria, impedindo a adoção de sistemas inteligentes em larga escala nas residências do país.

Companhias de Automação Residencial já possuem tecnologias integradas a residências e outros ambientes que geram *streaming* de dados oriundos de dispositivos automatizados e sensores. O presente trabalho visa explorar formas de acrescentar inteligência a tais tecnologias já existentes na indústria nacional. Para isso, efetuou-se a parceria com a companhia *Neocontrol*, de automação residencial, para delimitar problemas e criar soluções para implementação de ambientes adaptativos utilizando as tecnologias da companhia, de forma a viabilizar, em pouco tempo, a criação de um produto acessível para este propósito.

## 1.2 Objetivos

Endereça-se neste trabalho os seguintes problemas, discutidos e validados com a companhia *Neocontrol* como algumas das principais dificuldades na implementação de ambientes inteligentes, considerando a realidade da indústria:

- Alto custo de implementação para uma malha de sensores de tamanho considerável.
- Dificuldade na obtenção de dados anotados para reconhecimento de atividades e eventos.

Este trabalho visa explorar métodos encontrados na literatura e desenvolver novos métodos para realizar, a partir dos sistemas já existentes e considerando as dificuldades listadas, a previsão de eventos no ambiente e a modelagem do comportamento de seus usuários.

### 1.3 Contribuições

Este trabalho apresenta as seguintes contribuições ao campo de pesquisa e à indústria:

- Um estudo exploratório de viabilidade e desempenho de diversos métodos presentes na literatura para extração de características de sequências temporais de dados categóricos aplicada na previsão de eventos com *streaming* de dados de ambientes inteligentes da literatura e da companhia *Neocontrol*.
- Abordagens para tais métodos de seleção de características que levam em conta o custo computacional dos mesmos.
- Uma nova abordagem envolvendo aprendizado de métrica para os classificadores CHIP-CLAS visando classificação supervisionada sem hiperparâmetros de conjuntos com pequeno tamanho amostral, com estudos sobre seu desempenho em dados da literatura e para o problema tratado neste trabalho.
- Uma proposta de arquitetura para sistema de ambiente inteligente que funciona utilizando a infra-estrutura já existente em escala industrial dos produtos da companhia *Neocontrol*.
- Um estudo de viabilidade da modelagem não-supervisionada do comportamento de usuários de ambientes inteligentes utilizando *streaming* de dados de sensores dispositivos de automação residencial no ambiente através de Modelos Ocultos de Markov (HMM).
- O artigo: Gomes, I.P.; Bambirra, L.C.; Braga, A.P.; Aprendizado de Métrica Supervisionado para Classificador por Arestas de Suporte. Publicado no *XIII Congresso Brasileiro de Inteligência Computacional*.

### 1.4 Estrutura do Trabalho

Este trabalho é organizado em sete seções, descritas a seguir:

- Introdução: Explicita a motivação, os objetivos e a contribuição que este trabalho apresenta à área da domótica e da inteligência computacional.

- 
- Contextualização: Mostra o estado da arte na área, enumerando trabalhos anteriores, suas contribuições e como estes se relacionam com o presente trabalho.
  - Fundamentação Teórica: Mostra os fundamentos teóricos sobre o qual este trabalho é construído e que são necessários para sua compreensão.
  - Metodologia: Descreve os métodos desenvolvidos, as abordagens adotadas para solução dos problemas apresentados.
  - Experimentos e Resultados: Descreve



## 2 Contextualização

Este trabalho se insere no contexto de reconhecimento de atividades, previsão de atividades e localização (*tracking*) de usuários em ambientes inteligentes.

Como são utilizados apenas sensores simples, comuns e não-invasivos para o sistema criado pela *Neocontrol*, considera-se para o contexto apenas projetos onde são empregados sensores com essas características. Não se insere no contexto, portanto, o reconhecimento de atividades utilizando aparatos como câmeras, *wereables* e informações de sensores inerciais de *smartphones*.

### 2.1 Bases de Dados de *Smart Homes*

Conjuntos de dados originadas de sensores simples e amplamente disponíveis em *Smart Homes*, incluindo detalhes sobre os dispositivos utilizados na coleta dos dados, estão amplamente disponíveis na literatura, sendo citados exemplos a seguir.

Datasets contendo longos períodos da vida cotidiana em residências com apenas uma pessoa podem ser encontrados, com anotações (especificação de início ou fim de atividades específicas vinculado às ativações de sensores), em [CASAS ARUBA 17] e [CASAS KYOTO 8]. Em [CASAS ARUBA 19], encontramos dados sem anotações.

Tratando de residências com vários residentes, encontramos conjuntos de dados com anotações, como os presentes em [KYOTO] e [OUTRO N CASAS].

Alguns datasets focam no reconhecimento de atividades, com determinadas atividades sendo repetidas e registradas, fora do cotidiano. Exemplos são o CASAS [CASAS KYOTO] e [Universidade de Amsterdam].

Para ambientes de trabalho, exemplos notáveis são o dataset MavLab [MAVLAB] e o Tokyo 13 [CASAS TOKYO 13].

Os dados mencionados são apresentados como sequências de acionamentos de sensores ou dispositivos, sendo cada acionamento descrito por uma *timestamp* com data e hora, o código do sensor, o novo estado do sensor e, para os dados com anotações, uma descrição da atividade à qual a ativação pertence, inicia ou termina.

## 2.2 Reconhecimento de Atividades em *Smart Homes*

## 2.3 Previsão de Eventos em *Smart Homes*

Em (??), é discutida a importância do problema da previsão de atividades em Smart Homes e são descritos os algoritmos *LeZi Update*, baseado no algoritmo de compreensão LZ78 (Lempel-Ziv), e SHIP (*Smart Home Inhabitant Prediction*). É discutido no artigo que a previsão de eventos é necessária para automação de rotinas e tarefas repetitivas para o usuário e que a previsão de sua próxima localização auxilia o ambiente a localizá-lo quando o sistema necessita entrar em contato com o mesmo.

Em (??), são comparados diversas técnicas para diversos problemas relacionadas à previsão de atividades em Smart Homes. Técnicas baseadas em Redes Neurais, como Redes Neurais Recorrentes (RNNs), Long Short-Term Memory (LSTM) e Gated Recurrent Units (GRUs), são comparadas com outras técnicas para predição de sequências discretas. As técnicas *Prediction by Partial Matching* (PPM), *Dependency Graph* (DG) e *All k-order Markov Chains* (AKOM) baseiam-se em Modelos de Markov, enquanto o algoritmo LZ78 é baseado em compreensão de dados. Testa-se os algoritmos no problema da previsão da próxima atividade a ser executada, tempo até a próxima atividade e sequência de próximas atividades. Os resultados indicaram desempenho superior para as LSTMs no problema de previsão da próxima atividade e do tempo até a próxima atividade, porém, não apresenta desempenho consistentemente superior ao se tentar prever uma sequência de atividades, sendo inferior a uma ou mais técnicas baseadas em modelos de Markov em parte dos casos.

Em (??), são identificadas relações temporais entre eventos, ou seja, quanto eventos sempre acontecem em sequência, um evento acontece sempre durante outro, um evento acontece sempre ao início ou ao fim de outro, dois eventos acontecem sempre simultaneamente, entre outras. Tais relações podem ser, então, utilizadas para previsão.

## 2.4 Localização de Usuários



## 3 Fundamentação Teórica

### 3.1 Classificação Supervisionada de Sequências Temporais

Informações baseadas em *streaming* de dados de sensores podem ser interpretadas como sequências temporais discretas. Métodos para classificação destas sequências podem ser divididos em três grandes famílias (XING; PEI; KEOGH, 2010):

- Métodos de classificação baseados em extração de características da sequência, classificadas com métodos convencionais.
- Métodos de classificação baseados em distância, onde são obtidas medidas de similaridade entre sequências que determinam a classificação.
- Métodos baseados em modelos inerentemente sequenciais, como o Modelo Oculto de Markov (HMM).

### 3.2 Extração de Características de Sequências Temporais Categóricas

#### 3.2.1 Contagem de Elementos

Como mostram os trabalhos de [COOK] e [], uma característica simples e frequentemente utilizada na classificação de sequências de acionamentos de sensores ou dispositivos em *Smart Homes* é a contagem das ocorrências de cada símbolo na janela a ser classificada.

	E	N	G	E	N	H	A	R	I	A
<b>E:</b>	2									
<b>N:</b>		2								
<b>G:</b>			1							
<b>H:</b>				1						
<b>A:</b>					2					
<b>R:</b>						1				
<b>I:</b>								1		

Tabela 1 – Contagem dos símbolos presentes na palavra **ENGENHARIA**.

	E	N	G	E	N	H	A	R	I	A
<b>ENG:</b>	1									
<b>NGE:</b>		1								
<b>GEN:</b>			1							
<b>ENH:</b>				1						
<b>NHA:</b>					1					
<b>HAR:</b>						1				
<b>ARI:</b>							1			
<b>RIA:</b>								1		

Tabela 2 – Contagem dos *3-Gramas* (N-Gramas com  $N = 3$ ) contidos na palavra ENGENHARIA.

### 3.2.2 N-Gramas

O método das N-gramas (XING; PEI; KEOGH, 2010) é utilizado para extração de características de sequências de símbolos discretos principalmente para processamento de linguagem natural (NLP).

*N-Gramas* são sequências de  $N$  símbolos consecutivos contidas dentro da sequência a ser classificada. Dado o valor de  $N$ , sua utilização como método de extração de características consiste na contagem do número de ocorrências de cada possível *N-Grama* para uma determinada amostra. O espaço de características resultante tem, então,  $S^N$  dimensões, sendo cada uma associada a um dos possíveis *N-Gramas*, sendo  $S$  o número de símbolos distintos existentes na entrada.

### 3.2.3 Convolução com Função Linear

Em (LUNDSTRÖM; JÄRPE; VERIKAS, 2016), realiza-se classificação de sequências temporais de acionamentos de sensores em uma *Smart Home* utilizando representações discretas de duração fixa dos sinais temporais correspondentes aos sensores.

O estado de cada sensor é representado por um sinal temporal discreto  $S_i(t)$ , sendo cada amostra dada por um segmento de duração fixa de todos os sinais dos sensores de entrada. Na figura 1, temos representações de sinais correspondentes a uma janela de 10 segundos de atividade, obtidos por 4 diferentes sensores.

Figura 1 – Visualização dos sinais temporais correspondentes a uma mesma rota percorrida duas vezes.

Fonte: (LUNDSTRÖM; JÄRPE; VERIKAS, 2016)

Considerando-se que podem existir segmentos correspondentes à mesma atividade com pequenas translações, deslocamentos, distorções e fatores de escala, como mostrado na figura 1, é necessário que estes fenômenos não provoquem grandes alterações na série. Com este propósito, efetua-se a convolução do sinal temporal correspondente a cada sensor com

uma função linear de inclinação negativa, conforme a equação 3.1. Dessa forma, conforme visualizado na figura 2, poucas amostras de cada sinal sofrerão grandes alterações com translações e mudanças de escala.

$$St_i(t) = S_t(t) * [-at + 1] \quad (3.1)$$

Figura 2 – Visualização dos sinais da figura 1 convoluídos com função linear.

Fonte: (LUNDSTRÖM; JÄRPE; VERIKAS, 2016)

### 3.3 Métodos para Classificação Supervisionada

#### 3.3.1 Máquina de Vetores de Suporte

Propostas em (??), Máquinas de Vetores de Suporte (SVMs) são algoritmos de classificação que se baseiam nos princípios de Maximização de Margem

#### 3.3.2 Classificadores por Arestas de Suporte (CLAS)

Os Classificadores por Arestas de Suporte (TORRES, 2016) constituem uma família de algoritmos de classificação de margem larga com métodos de aprendizado baseados em Grafos de Gabriel (GABRIEL; SOKAL, 1969).

Os Grafos de Gabriel são grafos não-orientados onde dois pontos  $a$  e  $b$  são vizinhos se e somente se não existe nenhum outro ponto no interior da hipersfera cujo diâmetro é definido por estes dois pontos. Nas figuras 3 e 4, são exemplificadas as condições respectivamente para a conexão e não-conexão de dois pontos  $a$  e  $b$  no grafo.

Nos classificadores CLAS, é construído o Grafo de Gabriel correspondente ao conjunto de dados e são então definidas as Arestas de Suporte, que são arestas que separam pontos de classes distintas. Através delas e de seus pontos médios, são extraídos parâmetros para configuração e construção de classificadores de margem larga (TORRES et al., 2014) (TORRES; CASTRO; BRAGA, 2015) (TORRES et al., 2015), além de um decisor (TORRES; CASTRO; BRAGA, 2012) utilizado para o método de treinamento multiobjetivo de redes neurais (Albuquerque Teixeira et al., 2000).

Figura 3 – Visualização de dois pontos  $a$  e  $b$  vizinhos em um Grafo de Gabriel.

Fonte: (??)

Como base para este trabalho, utiliza-se o classificador CHIP-CLAS (TORRES et al., 2015). Este cria para cada aresta de suporte um hiperplano de separação que passa pelo ponto médio da mesma e maximiza a margem de separação. A classificação é feita

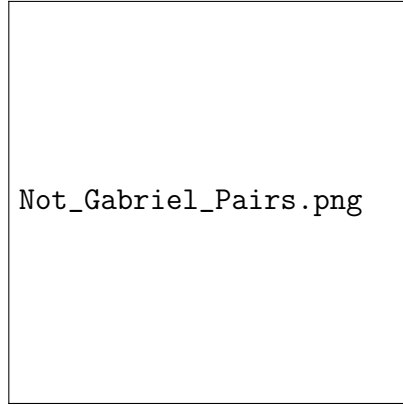


Figura 4 – Visualização de dois pontos  $a$  e  $b$  não-vizinhos em um Grafo de Gabriel devido à presença do ponto  $c$ .

Fonte: (??)

através de votação deste conjunto de hiperplanos. O voto de cada hiperplano é ponderado pela distância dos pontos médios das arestas de suporte ao ponto a ser classificado.

### 3.4 Aprendizado de Métrica para problemas de Classificação

A maioria dos métodos baseados em distâncias, como o SVM, o KNN e o próprio CLAS, foram descritos utilizando-se a distância Euclidiana. Para alguns problemas a distância Euclidiana entre alguns pontos de mesma classe pode ser sistematicamente maior que a distância entre pontos de classes distintas. Para solucionar este problema, pode-se usar métricas parametrizadas de distância, sendo obtidos os melhores parâmetros da métrica para cada problema através de um processo de otimização.

#### 3.4.1 Distância de Mahalanobis

A distância de Mahalanobis (DUDA; HART; STORK, 2000) foi inicialmente criada como uma medida de distância entre um ponto e uma distribuição de probabilidade multivariada. Ela é definida pela Equação 3.2, onde  $\mathbf{X}$  é o vetor que indica a localização do ponto,  $\mathbf{Y}$  é a média e  $M$  é o inverso da matriz covariância da distribuição de probabilidade.

$$D_M(\mathbf{X}, \mathbf{Y}) = \sqrt{(\mathbf{X} - \mathbf{Y})^T M (\mathbf{X} - \mathbf{Y})} \quad (3.2)$$

Pode-se generalizar este conceito e considerar a distância de Mahalanobis entre dois pontos  $\mathbf{X}$  e  $\mathbf{Y}$  quaisquer, sendo a matriz  $M$  um parâmetro para a métrica. Chamamos esta de matriz de Mahalanobis, com a restrição que  $M$  deve ser uma matriz semidefinida positiva  $d$  por  $d$ , onde  $d$  é a dimensão do espaço em que os pontos estão definidos.

A distância de Mahalanobis pode ser considerada uma generalização da distância Euclidiana, sendo esta última equivalente à distância de Mahalanobis com a matriz  $M$

igual à identidade. O conjunto dos pontos equidistantes a um centro utilizando distância de Mahalanobis gera uma superfície elipsoidal, enquanto para a distância Euclidiana, esta superfície é circular.

### 3.4.2 Large Margin Nearest Neighbors (LMNN)

O LMNN (WEINBERGER; SAUL, 2009) é um processo criado para aprendizado de métrica para classificadores KNN. A melhor matriz de Mahalanobis é encontrada através da minimização de uma função convexa baseada no erro Leave-One-Out (LOO) deste classificador.

O método recebe o número de vizinhos mais próximos do KNN como parâmetro ( $K$ ). Pode-se definir a função objetivo para o LMNN como composta de dois termos. O primeiro penaliza a soma das distâncias de cada ponto a seus vizinhos mais próximos, tendo efeito de aproximá-los, sendo dado pela Equação 3.3, onde  $j \rightsquigarrow i$  significa que  $j$  está entre os  $K$  vizinhos mais próximos de  $i$ .

$$\varepsilon_{pull}(M) = \sum_{j \rightsquigarrow i} D_M^2(\mathbf{x}_i, \mathbf{x}_j) \quad (3.3)$$

O segundo termo penaliza curtas distâncias entre cada ponto e pontos de classes distintas entre seus vizinhos mais próximos (impostores). É definido pela Equação 3.4.

$$\varepsilon_{push}(M) = \sum_{i, j \rightsquigarrow i} \sum_l (1 - y_{il}) [1 + D_M^2(\mathbf{x}_i, \mathbf{x}_j) - D_M^2(\mathbf{x}_i, \mathbf{x}_l)] \quad (3.4)$$

Para implementação direta no CLAS, que trabalha com distância Euclidiana, a matriz  $M$  pode ser decomposta como o quadrado de uma matriz simétrica. A distância de Mahalanobis pode então ser obtida através da distância Euclidiana das transformações lineares dos pontos por esta matriz simétrica, como visto na Equação 3.6.

$$M = LL^T \quad (3.5)$$

$$D_M = dist(LX, LY) \quad (3.6)$$

Assim, a classificação utilizando a distância Euclidiana, utilizando-se esta transformação linear  $L$  nos dados de entrada, é equivalente à utilização da distância de Mahalanobis. Somando-se as penalidades, adicionando a restrição Semidefinida-Positiva para a matriz  $M$  e modificando o segundo termo da função objetivo de forma a adicionar variá-

veis de folga e colocá-la numa forma mais adequada para a solução, temos a formulação final do problema de otimização:

$$\begin{aligned}
 L^* = \arg \min_L \quad & \sum_{j \rightsquigarrow i} d(L\mathbf{x}_i, L\mathbf{x}_j) + \sum_{i, j \rightsquigarrow i} (1 - y_{il}) \xi_{ijl} \\
 \text{sujeito a} \quad & d(L\mathbf{x}_i, L\mathbf{x}_l) - d(L\mathbf{x}_i, L\mathbf{x}_j) \geq 1 - \xi_{ijl}, \\
 & \xi_{ijl} \geq 0, \\
 & LL \succeq 0.
 \end{aligned} \tag{3.7}$$

Após o aprendizado de métricas, o algoritmo LMNN toma a decisão utilizando o classificador KNN com a métrica de distância aprendida. Assim, cada amostra do conjunto de testes é classificada de acordo com seus  $K$  vizinhos mais próximos segundo a métrica de Mahalanobis obtida.

### 3.5 Modelo Oculto de Markov

Um HMM é um modelo estatístico onde assume-se que o sistema pode ser modelado por uma Cadeia de Markov. A Cadeia de Markov é um processo estocástico que consiste em um conjunto de estados, cada um possuindo um conjunto de probabilidades de transição, que indica a probabilidade de se encontrar em cada outro estado no próximo instante de tempo, e uma probabilidade de emissão de símbolos, que indica quais símbolos podem ser emitidos pelo sistema naquele estado e com qual probabilidade.

No HMM, o estado do sistema não é diretamente visível ao observador, apenas a saída que o sistema emite. Tem-se, então, três problemas canônicos para HMMs:

- **Avaliação:** Dado um modelo  $\lambda$  e uma sequência de observações  $\mathbf{O}$ , encontrar a probabilidade  $p(\mathbf{O}|\lambda)$  das observações serem geradas pelo modelo.
- **Decodificação:** Dado um modelo  $\lambda$  e uma sequência de observações  $\mathbf{O}$ , encontrar sequência de estados  $\mathbf{S}$  mais provável para produzir a sequência de observações.
- **Aprendizado:** Dado um modelo  $\lambda$  e uma sequência de observações  $\mathbf{O}$ , qual o melhor ajuste dos parâmetros  $\{\mathbf{A}, \mathbf{B}, \pi\}$  de forma a maximizar a probabilidade  $p(\mathbf{O}|\lambda)$ .

Para a aplicação neste trabalho, são importantes a solução dos problemas de Decodificação e Aprendizado.

### 3.5.1 Problema da Decodificação

Um algoritmo de decodificação baseado no princípio de Máxima Verossimilhança e em programação dinâmica foi proposto em ([VITERBI, 1967](#)) e é de uso canônico para encontrar a sequência de estados mais provável em modelos como HMMs.

Também existem algoritmos aproximados, baseados em técnicas de amostragem, úteis principalmente para problemas de grandes dimensões, computacionalmente intratáveis utilizando o algoritmo de Viterbi.

### 3.5.2 Problema do Aprendizado

Também baseado no princípio de Máxima Verossimilhança, o algoritmo de Baum-Welch é utilizado para encontrar mínimos locais para o melhor ajuste dos parâmetros de um HMM, dado um ponto de partida e uma sequência de símbolos observados.





## 4 Metodologia

### 4.1 Banco de Testes

O Banco de Testes que fornece os dados para este trabalho consiste em um apartamento de 7 ambientes (2 quartos de solteiro, quarto de casal, sala de estar, sala de jantar, cozinha e escritório). Por estes cômodos, tem-se o histórico em tempo real dos dados de 28 atuadores (como relés ou *dimmers* para lâmpadas e motores de cortina), 13 interfaces com o usuário (interruptores e outros comandos), 6 sensores infravermelhos de presença (sala de estar, sala de jantar, cozinha, quarto de casal e escritório) e 2 sensores de abertura de porta nas entradas do apartamento (entrada principal e entrada pela cozinha). Também são registrados comandos enviados ao sistema por dispositivos móveis, como celulares e tablets.

Figura 5 – Planta da residência e localização dos sensores para o banco de testes.

### 4.2 Coleta, Armazenamento e Apresentação dos Dados

O sistema inteligente foi criado com base no sistema de automação residencial Minibox, da companhia nacional Neocontrol. Este sistema se baseia na comunicação de todos os sensores, interfaces e demais dispositivos de uma determinada residência ou imóvel com uma central, que transmite e recebe os dados para um servidor na nuvem através do protocolo MQTT para comunicação com dispositivos móveis.

O protocolo MQTT é um protocolo de transmissão e recepção de mensagens muito utilizado em contextos de *Internet of Things* (IoT) que roda sobre TCP/IP. O protocolo é aberto e foi projetado para ser simples, leve e de fácil implementação, necessidades para tipo de aplicação (OASIS, 2014).

Foi criado um cliente em linguagem C com o trabalho de ouvir e interpretar todas as comunicações do servidor MQTT com os diversos dispositivos do banco de testes e registrá-las em um banco de dados PostgreSQL, acessível remotamente. As informações apresentadas são uma sequência temporal de eventos de dispositivos, contendo para cada amostra uma *timestamp*, o dispositivo ativado (identificador do dispositivo e canal) e um valor descrevendo o estado do sensor.

Os possíveis valores para cada tipo de dispositivo presente são descritos a seguir:

Os sensores de movimento e de porta possuem estados binários, "ON/OFF" para movimento e "OPEN/CLOSE" para porta. Observa-se, portanto, que cada amostra carrega

muito pouca informação, sendo necessário um pré-processamento dos dados de forma a condiciona-los para exibir informações significativas sobre o evento corrente.

### 4.3 Previsão de Eventos

Certos eventos dos quais são obtidas informações são disparados pelos usuários da casa, como a ativação de cenas, o acionamento de interruptores e o abrir de portas. Espera-se que as informações contidas na *timestamp* destes eventos, tais como hora do dia, dia da semana, aliadas às informações dos dispositivos acionados nos minutos anteriores ao evento de interesse possam servir como preditores para um possível acionamento do evento em um futuro próximo, mostrando a capacidade do sistema de encontrar padrões no comportamento de seus usuários.

A previsão de eventos pode ser formulada como um problema supervisionado de classificação, onde

### 4.4 Extração de Características

É necessária para a classificação a extração de características das sequências temporais de acionamentos de dispositivos. Os métodos para Seleção de Características visam extrair da sequência temporal correspondente a uma janela  $S_t$  de  $M$  minutos. São explorados os seguintes métodos encontrados na literatura:

#### 4.4.1 Contagem de Acionamentos

Este método produz um número  $N_X$  de características  $X$  igual ao número  $N_D$  de dispositivos  $D$  presentes. Para cada dispositivo  $D_i$ , é adicionada uma característica contendo o número de vezes que o mesmo é presente na janela  $S_t$ , como descrito na subseção 3.2.1.

Esta abordagem, bastante simples, não leva em conta o momento do acionamento dos dispositivos, a ordem ou o intervalo entre acionamentos. Propõe-se uma modificação neste método para caracterizar o intervalo entre acionamentos através da inserção de um símbolo indicador de inatividade na sequência a ser classificada em momentos onde o intervalo entre dois acionamentos for maior que um *threshold*.

#### 4.4.2 N-Gramas

Parametrizado por um valor inteiro  $N$ , este método consiste na contagem de ocorrências na janela  $S_t$  de cada uma das possíveis combinações de  $N$  acionamentos consecutivos.

tivos, produz um número de características  $N_X = N_D^N$ , assim como descrito na subseção 3.2.2.

Dado o grande número de combinações possíveis, especialmente para valores de  $N$  maiores que 3, as características se tornam esparsas, com poucas tendo valores diferentes de 0. Para limitar o consumo desnecessário de memória e tempo de processamento gasto com um conjunto grande e esparsa de características, utiliza-se neste trabalho apenas as combinações presentes nos conjuntos utilizados para treinamento. As demais possíveis combinações, caso surjam para avaliação, são consideradas raras e não farão parte do conjunto.

Este método não leva em conta o momento do acionamento dos dispositivos, ou o intervalo entre acionamentos, mas leva em conta a ordem destes. A inserção de um símbolo indicador de inatividade na sequência a ser classificada em momentos onde o intervalo entre dois acionamentos for maior que um *threshold*, proposta para o método de Contagem de Símbolos, também pode ser utilizada para caracterizar momentos de inatividade neste método.

#### 4.4.3 Convolução com Função Linear

Este método, assim como a contagem de sensores, produz um número  $N_X$  de características  $X$  igual ao número  $N_D$  de dispositivos  $D$  presentes. Neste método, considera-se o *streaming* de dados de cada dispositivo como um sinal temporal discreto.

Para que todo momento no tempo possua, o sinal correspondente a cada dispositivo é convoluido com uma função linear  $x = -at + 1$  de inclinação negativa  $a = inv(M)$ , dependente da largura  $M$ , em minutos, da janela  $S_t$ .

Por razões de custo computacional, não integra o conjunto de características o sinal temporal correspondente a todo o intervalo da janela  $S_t$  discretizado em segundos, como feito em [Lundström], mas apenas para o instante final. Desta forma, evita-se um espaço de características de dimensão  $N_X = 60MN_X$ .

Este método leva em conta o momento do acionamento dos dispositivos, não sendo necessário portanto a modificação da sequência a ser classificada com a inserção de símbolos em momentos de inatividade.

### 4.5 Redução de Dimensionalidade

Para redução do tempo de treinamento para a classificação, o espaço de características é transformado através de Análise de Componentes Principais (PCA).

No PCA, aplica-se uma transformação linear de forma a maximizar a variância

## 4.6 Classificação

Após a etapa de extração de características e as devidas transformações no espaço de entrada para redução de. Utiliza-se como *baseline* para classificação o algoritmo SVM, treinado através de 10-fold Cross Validation, e propõe-se a utilização do algoritmo CHIP-CLAS.

O ajuste de hiperparâmetros para o SVM é custoso computacionalmente, devendo ser feito através de *Grid Search* para obtenção de melhores resultados, inviabilizando sua utilização em produção. Classificadores da família CLAS demonstraram empiricamente ter desempenho estatisticamente equivalente ao SVM com *kernels* RBF e Polinomial, com a vantagem de não possuírem hiperparâmetros a serem ajustados, evitando assim o processo de *Grid Search*.

Classificadores da família CLAS, porém, descartam amostras durante o processo de treinamento para eliminar a sobreposição entre classes, processo análogo ao relaxamento da restrição de margem máxima do SVM. Nas análises preliminares dos dados, encontram-se eventos cuja ocorrência é rara, como o acionamento das luzes indiretas da sala de estar. Para estes, o descarte amostras pode levar a perda de informações importantes devido à menor redundância dos dados. Para eventos raros, então, propõe-se a abordagem AM-CHIP-CLAS, que inclui uma etapa de aprendizado de métrica anterior à filtragem de amostras para minimizar o descarte de dados sem prejuízo ao desempenho.

### 4.6.1 AM-CHIP-CLAS

A abordagem AM-CHIP-CLAS foi baseada no método LMNN, que utiliza aprendizado de métrica para maximizar desempenho e diminuir superposição em classificadores KNN.

Adapta-se, então, o aprendizado de métrica do método LMNN para utilização em classificadores CLAS. Espera-se que o processo não possua hiperparâmetros, de forma que continue sendo desnecessário o ajuste de parâmetros através de validação cruzada e busca em *grid*. Para tal, modifica-se a função objetivo do LMNN. Ao invés de considerar os  $K$  vizinhos mais próximos para cada ponto, a função é calculada considerando-se os vizinhos conectados a ele em um Grafo de Gabriel construído utilizando distância Euclidiana. Elimina-se assim a necessidade de um parâmetro  $K$  e leva-se em conta no aprendizado de métrica a estrutura geométrica do problema, também utilizada na classificação.

A formulação do problema de otimização mantém-se na forma vista na Equação 3.7, com  $j \rightsquigarrow i$  significando que  $j$  é conectado a  $i$  no Grafo de Gabriel. Isto mantém as características de convexidade e de restrições esparsamente violadas do problema de otimização do LMNN.

Obtida a matriz  $L$ , são feitas as transformações lineares nos conjuntos de treino e teste e o problema de classificação é solucionado pelo algoritmo CHIP-CLAS.

## 4.7 Modelo Não-Supervisionado para Tracking na Smart Home

Busca-se extrair dos dados, de forma não-supervisionada, o número de pessoas presentes na casa e a localização delas, uma informação simples, porém importante para fatores de segurança.

### 4.7.1 Modelo Oculto de Markov

No modelo construído, os estados correspondem à presença de um residente em cada cômodo da casa ou então fora da casa. Para um modelo de um residente, o espaço possui 12 estados: 11 indicando os cômodos e 1 para fora da casa. O sistema pode ser estendido para mais residentes utilizando-se um modelo idêntico para cada um, e então, realizando-se o produto cartesiano entre todos os modelos. Para dois residentes, isso resultou em um modelo de  $12 \times 12$ , ou seja, 144 estados. Os símbolos possíveis são o nome de cada sensor e mais um símbolo em branco "BLNK" para cada 2m de inatividade de forma a produzir saída quando nenhum dos residentes está em casa.

Os parâmetros de transição entre os estados foram estimados *a priori*:

- 0 entre os estados correspondentes a cômodos que não se conectam (uma pessoa não tem como transitar entre dois cômodos que não se ligam),
- 1 entre os estados correspondentes a cômodos que se conectam (uma pessoa tem probabilidade de mudar de cômodo entre duas ativações de sensores),
- 50 entre os estados e eles próprios. (Uma pessoa tem uma grande probabilidade de continuar no mesmo cômodo entre duas ativações de sensores).

Os parâmetros de emissão, ou seja, as probabilidades da emissão de cada símbolo observado na saída dado um determinado estado, foram estimados como:

- 100, se o sensor correspondente ao símbolo estiver presente em um cômodo correspondente ao estado (o símbolo em branco não pertence a nenhum cômodo.),
- 1 caso não estiver (não-nulo para o sistema tolerar algum nível de ruído sem entrar em estado inconsistente).

Os parâmetros de transição e emissão para cada estado foram então normalizados de forma que a soma resulte sempre em 1.

O aprendizado deste modelo de forma a ajustar os parâmetros para melhor se adequar à realidade observada é, então, possível pelo algoritmo de Baum-Welch e a decodificação pode ser feita pelo algoritmo de Viterbi.

Dado que é possível obter a probabilidade posterior de cada estado a partir do modelo e do estado atual, também torna-se possível a previsão da próxima localização em um modelo treinado.

Para este trabalho, o aprendizado do modelo não foi necessário visto que o modelo apresentou comportamento satisfatório utilizando-se os parâmetros estimados *a priori*, exibindo resultados coerentes com a realidade por inspeção. Para trabalhos futuros, propõe-se o aprendizado do modelo para identificação de padrões de comportamento específicos de cada residente, de forma a permitir a identificação dos mesmos pelo sistema.

#### 4.7.2 Extensão do modelo para Múltiplos Residentes

Quando mais

## 5 Experimentos e Resultados

### 5.1 Recursos Computacionais Utilizados

Os experimentos foram executados em um laptop com processador Intel Core i5, 4 GB de memória RAM, sem placa de vídeo dedicada.

### 5.2 Validação da Extração de Características para Previsão de Eventos

Os experimentos para validação da extração de características na previsão de eventos em *Smart Homes* consistem na avaliação e comparação dos diferentes métodos para para os dados no banco de testes utilizando o algoritmo classificador SVM com *kernel* RBF e ajuste de hiperparâmetros via busca em *grid*. A avaliação é feita via *5-fold Cross Validation*, utilizando como métrica de desempenho a área sob a curva ROC (AUC).

Os seguintes eventos foram considerados para a base Neocontrol:

- COZINHA: Acionamento do interruptor da luz da Cozinha.
- MESA: Acionamento do interruptor da luz sobre a mesa de jantar.
- SOFA: Acionamento do interruptor da luz sobre o sofá.
- VARANDA: Acionamento do interruptor da luz da varanda.
- HALL: Acionamento dos interruptores das luzes do Hall.
- ABAJOUR: Acionamento do interruptor da luz do Abajour.

Os eventos foram avaliados um a um, em uma abordagem um-contratodos, sendo também incluídos na classe negativa para cada evento os períodos que marcam a metade de grandes intervalos entre eventos (maiores de 10 minutos).

#### 5.2.1 Resultados

Os resultados dos experimentos utilizando os métodos propostos para extração de características, junto com o *rank* médio de cada método, são mostrados na Tabela 3. Para verificar a relevância estatística do *rank* médio, foi utilizado o teste de Bonferroni-Dunn, visualizado na figura 6.

Tabela 3 – AUC Média para previsão de eventos para diversos métodos de extração de características.

Evento	Contagem (AUC)	SD	N-Gramas (AUC)	SD	Convolução (AUC)	SD
COZINHA	0.7185	0.0315	0.7208	0.0248	0.7512	0.0250
MESA	0.9869	0.01556	0.9866	0.0170	0.9811	0.0236
SOFA	0.9041	0.0513	0.8918	0.0619	0.9140	0.0621
VARANDA	0.8334	0.1132	0.6462	0.1154	0.7727	0.1575
HALL	0.9670	0.0382	0.9477	0.0307	0.9609	0.0430
ABAJOUR	0.9672	0.0453	0.8963	0.2096	0.9455	0.0747
<b>Rank Médio</b>	1.5		2.67		1.83	

Figura 6 – Visualização para o Teste de Friedman com teste post-hoc de Bonferroni

### 5.2.2 Discussão

Os resultados apresentados validam os métodos utilizados neste trabalho para extração de características. Destaca-se o método da Contagem de Acionamentos de Sensores, descrito na seção 3.2.1, que apresentou o melhor *rank* médio para os eventos considerados mesmo sendo o mais simples dentre os três. Por este motivo, escolhe-se sua utilização no restante dos experimentos deste trabalho.

O teste de Friedman não indicou, porém, que não se pode ter certeza da superioridade de nenhum dos métodos em relação aos demais.

## 5.3 Validação da abordagem AM-CHIP-CLAS para Classificação

A abordagem AM-CHIP-CLAS, com aprendizado de métrica, foi avaliada através de *10-fold Cross Validation* (KOHAVI, 1995). Mediu-se a porcentagem dos dados desconsiderados no treinamento e o desempenho da classificação através de AUC (área sob a curva ROC). Os experimentos foram realizados com 13 bases de dados reais obtidas através do repositório UCI (BACHE; LICHMAN, 2013) e 2 problemas de expressão gênica: *Golub* (GOLUB, 1999) e *BcrHess* (HESS et al., 2006).

A porcentagem desconsiderada dos dados foi comparada com a obtida para o algoritmo CHIP-CLAS em sua abordagem original, sem aprendizado de métrica. O desempenho foi comparado com o algoritmo CHIP-CLAS sem aprendizado de métrica e com o classificador SVM com *Kernels* RBF e Polinomial. Os melhores parâmetros para o SVM foram encontrados através de *10-fold Cross Validation* e busca em *grid*.

Buscou-se, também, visualizar os efeitos do aprendizado de métrica na superfície de separação.



### 5.3.1 Visualização da Superfície de Separação

Para visualização do efeito do aprendizado de métrica, o método AM-CHIP-CLAS e o CHIP-CLAS original foram utilizados para separação de um conjunto de dados sintético de duas dimensões. Foi criado para tal um conjunto de dados consistindo em fileiras intercaladas de classes distintas alinhadas com o eixo  $X$ , adicionadas de ruído gaussiano em ambas dimensões. Desta forma, algumas amostras significativas para o treinamento ficam próximas de mais pontos da classe oposta que as demais. Assim, induz-se ao erro o método para eliminação de sobreposição do CHIP-CLAS original, destacando assim a diferença entre ambas as metodologias.

O método CHIP-CLAS original desconsiderou 41.67% dos dados no processo de classificação, gerando a superfície de separação da Figura 7. O método AM-CHIP-CLAS não desconsiderou nenhuma amostra no processo de classificação, gerando a superfície da Figura 8.

Figura 7 – Visualização da superfície de separação para o CHIP-CLAS original

Figura 8 – Visualização da superfície de separação para o AM-CHIP-CLAS

### 5.3.2 Utilização dos Dados

A porcentagem dos dados desconsiderados no treinamento para cada execução da validação cruzada foi medida. Calculou-se a razão entre a quantidade de amostras descartadas no processo de filtragem e o total de dados da base, com e sem aprendizado de métrica. Os resultados médios obtidos para as execuções se encontram na Tabela 4.

Para 6 das 15 bases testadas, a porcentagem desconsiderada dos dados diminuiu consideravelmente, sofrendo variação de menos de 1% para cima ou para baixo nas bases restantes. Isto sugere uma maior utilização dos dados para o AM-CHIP-CLAS. A significância estatística desta superioridade pode ser estabelecida através de um teste estatístico de Wilcoxon pareado (DEMŠAR, 2006). O teste unilateral foi utilizado, com nível de confiança de 95% ( $\alpha = 0.05$ ). O Valor- $p$  obtido no teste foi  $p = 0.040$ , de forma que  $p < \alpha$ , confirmando estatisticamente a maior utilização dos dados para a nova abordagem com 95% de confiança.

### 5.3.3 Desempenho

A AUC para cada execução da validação cruzada foi medida e foi extraída a média para cada base de dados. Os resultados obtidos se encontram na Tabela 5, juntamente

Tabela 4 – Porcentagem média desconsiderada dos dados.

	dataset	CHIP-CLAS	AM-CHIP-CLAS
1	sonar	0.00	0.00
2	breastcancer	15.32	10.85
3	australian	37.97	38.89
4	diabetes	44.68	44.73
5	breastHess	38.94	26.16
6	bupa	48.28	48.76
7	haberman	45.97	45.24
8	banknote	0.00	0.00
9	fertility	43.11	24.78
10	parkinsons	10.36	3.24
11	climate	39.55	22.55
12	ILPD	47.55	47.27
13	german	46.86	47.30
14	heart	43.25	43.66
15	golub	37.05	0.00

Tabela 5 – AUC Média das execuções e Rank Médio dos Classificadores.

dataset	AM-CHIP-CLAS	CHIP-CLAS	RBF-SVM	Poly-SVM
sonar	0.84	0.88	0.84	0.87
breastcancer	0.97	0.96	0.97	0.96
australian	0.86	0.85	0.86	0.87
diabetes	0.71	0.72	0.71	0.71
breastHess	0.83	0.81	0.76	0.77
bupa	0.58	0.61	0.67	0.72
haberman	0.54	0.56	0.52	0.50
banknote	1.00	0.99	1.00	1.00
fertility	0.50	0.59	0.50	0.50
parkinsons	0.89	0.90	0.77	0.81
climate	0.85	0.84	0.53	0.72
ILPD	0.57	0.57	0.49	0.50
german	0.70	0.67	0.66	0.68
heart	0.81	0.80	0.83	0.83
golub	0.55	0.77	0.80	0.78
<b>Rank Mean</b>	2.20	2.40	2.93	2.47

com a média da posição de cada classificador num *ranking* de desempenho para cada base de dados.

Para avaliação estatística dos resultados de múltiplos classificadores, é indicado o teste de Friedman (DEMŠAR, 2006). Para um nível de confiança de 95% ( $\alpha = 0.05$ ), foi obtido um Valor-p de  $p = 0.445$ . O resultado obtido não é suficiente para rejeitar a hipótese nula de que nenhum dos classificadores possui desempenho estatisticamente diferente dos demais. Para melhor visualizar o desempenho dos classificadores, foi feito o teste *post-hoc* de Bonferoni-Dunn (DEMŠAR, 2006), obtendo-se o gráfico da Figura 9,

com o eixo horizontal indicando o *rank* (quanto menor, melhor o desempenho).

Figura 9 – Visualização para o teste *post-hoc* de Bonferroni-Dunn

#### 5.3.4 Discussão

Verifica-se que o desempenho da abordagem AM-CHIP-CLAS não difere significativamente do classificador CHIP-CLAS sem aprendizado de métrica, com *rank* médio pouco superior a este. Ambos CHIP-CLAS e AM-CHIP-CLAS se mostram também superiores no *rank* médio aos classificadores SVM testados, para o *benchmark* utilizado.

### 5.4 Arquitetura Convolução + PCA + AM-CHIP-CLAS para os dados da Neocontrol

## 5.5 Validação dos Modelos para múltiplos residentes

#### 5.5.1 Dados para Validação

Falar sobre o dataset TWOR2010 da WSU.

Para validar o método independentemente da qualidade dos dados coletados, buscou-se na literatura dados oriundos de fontes semelhantes e dispostos em modelo parecido com os dados fornecidos pelo sistema de automação residencial da Neocontrol. Foram escolhidos dados divulgados pelo grupo de pesquisa CASAS, da WSU (Washington State University) produzidos ao longo de um ano em uma residência de dois moradores, com anotações do início e do fim de determinadas atividades (COOK; SCHMITTER-EDGEcombe, 2009). Os dados selecionados provêm de um conjunto de 51 sensores de movimento espalhados pela casa e do sensor que detecta abertura da porta principal. Dados presentes na base de dados provenientes de outros sensores foram descartados devido à ausência de correspondência com os dados do sistema da Neocontrol.

#### 5.5.2 Resultados

Mesmo sem o treinamento dos parâmetros, utilizando apenas os valores estimados, o HMM funcionou bem, com suas transições sendo coerentes com as anotações feitas e com os sensores acionados. Essa coerência foi verificada por inspeção, inspecionando-se 10 sequências de aproximadamente 1000 amostras, 200 antes e 800 depois dos 5 primeiros acionamentos do sensor que detecta abertura da porta principal. Foi verificado visualmente a consistência dos estados com a trajetória plotada na visualização e, no caso de residentes que deixam a casa, as atividades em branco dos sensores.

Porém, devido a ausência de um método para identificação dos residentes, resultados improváveis foram observados em momentos onde os mesmos ocupavam o mesmo cômodo. A correção deste problema será objeto de trabalhos futuros.

## 5.6 Validação da Extração de Características

### 5.6.1 Dados para Validação

Falar sobre o dataset TWOR2010 da WSU.

### 5.6.2 N-Gramas

O classificador SVM, com características selecionadas através de N-gramas, obteve um resultado, como esperado, pior do que a classificação feita através do algoritmo de Viterbi, alcançando uma precisão de 80.9% para um treinamento com 8000 símbolos, porém demorou um tempo consideravelmente menor para obter o resultado, trabalha com um espaço amostral completamente binário e, uma vez treinado, o classificador SVM não precisa dos estados anteriores para realizar as próximas classificações.

Nota-se, porém, que a classificação não ultrapassou essa marca, não importa o *gamma* sendo utilizado para o classificador.

Pred/Obs	0	1	2
0	88%	12%	0.2%
1	6%	69%	25%
2	2%	20%	77%

Tabela 6 – Matriz Confusão para resultados obtidos em classificador com *gamma* igual a 0.3

### 5.6.3 Convolução

## 5.7 Validação do Algoritmo de Classificação AM-CHIP-CLAS

## 5.8 Trabalhos Futuros

## 6 Conclusão

Durante o trabalho, foi desenvolvido um meio de se colher dados de utilização de uma casa utilizando tecnologia nacional, em parceria com a indústria. Foram também desenvolvidas formas viáveis de se criar um modelo estatístico HMM destes dados e, a partir deste modelo, a obtenção do estado do modelo para cada observação, eliminando-se a necessidade de anotações para o treinamento supervisionado de algoritmos de classificação sobre o mesmo. Foi também desenvolvido um classificador (N-Gramas+SVM), trabalhando apenas com dados binários no espaço amostral, para classificação destes dados sem necessidade de estados anteriores ou do modelo estatístico após o treinamento. Apesar do desempenho aquém do desejado, a classificação mostrou-se possível. Foi ainda verificada a importância, ao se trabalhar com reconhecimento de padrões, que se utilize ferramentas que levem em conta todas as informações disponíveis sobre os dados. Foi claro durante a execução do projeto a diferença de desempenho dos algoritmos que não faziam suposição alguma sobre os dados (sem seleção de características) daqueles que levavam em conta o caráter sequencial do mesmo (N-Gramas), e ainda melhor foi o comportamento quando se utilizou um modelo estatístico coerente, criado a partir de informações do sistema monitorado (HMM).



# Referências

- Albuquerque Teixeira, R. de et al. Improving generalization of MLPs with multi-objective optimization. *Neurocomputing*, v. 35, p. 189–194, 2000. ISSN 09252312. Citado na página 25.
- BACHE, K.; LICHMAN, M. *UCI Machine Learning Repository*. 2013. 0 p. Disponível em: <<http://www.ics.uci.edu/~mllearn/MLRepository.ht>>. Citado na página 38.
- CARR, R. S. et al. *Home energy monitoring and control system*. Google Patents, 1987. Disponível em: <<https://www.google.com/patents/US4644320>>. Citado na página 17.
- COOK, D. J.; SCHMITTER-EDGEcombe, M. Assessing the quality of activities in a smart environment. In: *Methods of Information in Medicine*. [S.l.: s.n.], 2009. v. 48, n. 5, p. 480–485. ISBN 1000341380. ISSN 00261270. Citado na página 41.
- DEMŠAR, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006. ISSN 1532-4435. Citado 2 vezes nas páginas 39 e 40.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. [S.l.: s.n.], 2000. 654 p. ISBN 1740634X. ISBN 978-0-471-05669-0. Citado na página 26.
- GABRIEL, K. R.; SOKAL, R. R. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, v. 18, n. 3, p. 259–278, 1969. ISSN 00397989. Disponível em: <<http://sysbio.oxfordjournals.org/cgi/content/abstract/18/3/259>>. Citado na página 25.
- GOLUB, T. R. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, v. 286, n. 5439, p. 531–537, 1999. ISSN 00368075. Disponível em: <<http://www.sciencemag.org/cgi/doi/10.1126/science.286.5439.531>>. Citado na página 38.
- HESS, K. R. et al. Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of Clinical Oncology*, v. 24, n. 26, p. 4236–4244, 2006. ISSN 0732183X. Citado na página 38.
- HUNT, J.; HOLMES, J. Electrical Energy Monitoring and Control System for the Home. *IEEE Transactions on Consumer Electronics*, C, n. 3, p. 578–583, 1986. Disponível em: <<http://ieeexplore.ieee.org/xpls/abs/all.jsp?arnumber=4071>>. Citado na página 17.
- KOHAVI, R. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*, p. 1137–1145, 1995. ISSN 10450823. Citado na página 38.
- LUNDSTRÖM, J.; JÄRPE, E.; VERIKAS, A. Detecting and exploring deviating behaviour of smart home residents. *Expert Systems with Applications*, v. 55, p. 429–440, 2016. ISSN 09574174. Citado 2 vezes nas páginas 24 e 25.

- MOZER, M. C. The neural network house: An environment that adapts to its inhabitants. *American Association for Artificial Intelligence Spring Symposium on Intelligent Environments*, n. December, p. 110–114, 1998. Citado na página 17.
- OASIS. MQTT Version 3.1.1. *OASIS Standard*, n. October, p. 81, 2014. Disponível em: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>. Citado na página 31.
- TORRES, L.; CASTRO, C.; BRAGA, A. A Computational Geometry Approach for Pareto-Optimal Selection of Neural Networks. *International Conference on Artificial Neural Networks*, n. 22, 2012. Citado na página 25.
- TORRES, L.; CASTRO, C.; BRAGA, A. Gabriel Graph for Dataset Structure and Large Margin Classification: A Bayesian Approach. *Proceedings of the European Symposium on Neural Networks 2015*, p. 237–242, 2015. Citado na página 25.
- TORRES, L. et al. Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters*, v. 51, n. 24, p. 1967–1969, 2015. ISSN 0013-5194. Citado na página 25.
- TORRES, L. et al. A geometrical approach for parameter selection of radial basis functions networks. In: *Lecture Notes in Computer Science*. [S.l.: s.n.], 2014. v. 8681 LNCS. ISBN 9783319111780. Citado na página 25.
- TORRES, L. C. B. *Classificador por Arestas de Suporte (CLAS): Métodos de Aprendizado Baseados em Grafos de Gabriel*. Tese (Doutorado) — Universidade Federal de Minas Gerais, 2016. Citado na página 25.
- VITERBI, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, IEEE, v. 13, n. 2, p. 260–269, 1967. Citado na página 29.
- WEINBERGER, K. Q.; SAUL, L. K. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *The Journal of Machine Learning Research*, v. 10, p. 207–244, 2009. ISSN 1532-4435. Citado na página 27.
- XING, Z.; PEI, J.; KEOGH, E. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter*, v. 12, n. 1, p. 40, 2010. ISSN 19310145. Citado 2 vezes nas páginas 23 e 24.