Igor Pereira Gomes

Previsão de Eventos e Localização de Pessoas não-supervisionada em Ambientes Inteligentes com Rede de Sensores de Tamanho Reduzido

Belo Horizonte

2018

Igor Pereira Gomes

Previsão de Eventos e Localização de Pessoas não-supervisionada em Ambientes Inteligentes com Rede de Sensores de Tamanho Reduzido

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

Universidade Federal de Minas Gerais – UFMG

Escola de Engenharia

Curso de 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 infomações 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. Automacão Residencial. Sistemas Inteligentes.

Lista de ilustrações

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

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 (**??**). Mesmo antes, uma proto-inteligência já estava sendo concebida, com sistemas de controle de energia para utensílios domésticos já sendo desenvolvidos por pesquisadores (**??**) e até mesmo presentes em patentes (**??**).

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.

O presente trabalho visa explorar formas de acrescentar inteligência a tecnologias de sensoreamento e automação residencial 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 dificuldades cruciais na implementação de ambientes inteligentes considerando a realidade da companhia:

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

Dados tais problemas, o trabalho visa explorar métodos encontrados na literatura e desenvolver novos métodos para

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

2 Revisão da Literatura

2.1 Prev

A maioria dos trabalhos na área lida com aprendizado supervisionado (**??**), ou seja, quando se tem conhecidos os resultados esperados para o conjunto de treino, tendo-se anotações quanto às atividades desempenhadas junto aos dados dos sensores. Os métodos utilizados são, para estes, métodos convencionais de seleção de características e classificação. Este não é o caso deste trabalho, visto que o objetivo é implementar o sistema em residências reais, onde vivem famílias que normalmente não possuem o tempo necessário para anotar o início e o término de suas atividades cotidianas. Ainda assim, existem métodos que apresentam bons resultados para o aprendizado não-supervisionado utilizando técnicas de mineração de dados (**??**). A utilização destas técnicas é, porém, custosa, sendo de difícil implementação em sistemas embarcados. Além disso, elas são aplicadas para resolver problemas mais complexos que o escopo deste trabalho. Busca-se, então, um compromisso entre métodos poderosos para aprendizado não-supervisionado e simplicidade/custo computacional.

2.2 Aprendizado Não-Supervisionado em Ambientes Inteligentes

2.3 Aprendizado Ativo em Ambientes Inteligentes

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 (**??**):

- 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).

3.1.1 Extração de Características de Sequências Temporais: Contagem de Elementos

3.1.2 Extração de Características de Sequências Temporais: N-Gramas

N-gramas (**??**) são sequências de N símbolos contidas dentro da sequência a ser classificada. A selecão de características consiste em contar quantas vezes cada possível sequência de 3 acionamentos possível no espaço de símbolos ocorre em uma determinada amostra. As características são, então, essa contagem para cada N-grama ou a presença de cada N-grama na amostra. No caso, foi utilizado a presença do N-grama na amostra como característica, de forma a se obter um espaço amostral para as características constituído apenas de valores binários. O espaço de características resultante tem, então, S^N dimensões, sendo S a o número de diferentes símbolos possíveis.

3.1.3 Extração de Características de Sequências Temporais: Autoencoders e Sequence-Embeddings

. .

3.1.4 Extração de Características de Sequências Temporais: Algoritmos de Comprensão

. .

3.1.5 Extração de Características de Sequências Temporais: Convolução

. .

3.2 Métodos para Classificação Supervisionada

3.2.1 Máquina de Vetores de Suporte

3.2.2 Classificadores por Arestas de Suporte (CLAS)

Os Classificadores por Arestas de Suporte (**??**) constituem uma família de algoritmos de classificação de margem larga com métodos de aprendizado baseados em Grafos de Gabriel (**??**). Os Grafos de Gabriel são grafos não-orientados onde dois pontos são interconectados se e somente se não existe um terceiro ponto no interior da hiperesfera cujo diâmetro é definido por estes dois pontos. 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 (**??**) (**??**) (**??**), além de um decisor (**??**) utilizado para o método de treinamento multiobjetivo de redes neurais (**??**).

Como base para este trabalho, utiliza-se o classificador CHIP-CLAS (**??**). 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 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.

3.2.3 Aprendizado de Métrica para problemas de Classificação

3.2.4 Large Margin Nearest Neighbors (LMNN)

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 maior que a distância entre pontos de classes distintas. Para solucionar este problema, pode-se usar métricas parametrizadas de distância, sendo os melhores parâmetros para cada problema obtidos através de um processo de otimização. O LMNN (**??**) é um processo criado para aprendi-

zado 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 3.1, onde ji significa que j está entre os K vizinhos mais próximos de i.

$$\varepsilon_{pull}(M) = \sum_{ji} D_M^2(\mathbf{x}_i, \mathbf{x}_j) \tag{3.1}$$

O segundo termo penaliza curtas distâncias entre cada ponto e pontos de classes distintas entre seus vizinhos mais próximos (impostores). É definido pela Equação 3.2.

$$\varepsilon_{push}(M) = \sum_{i,ji} \sum_{l} (1 - y_{il})[1 + D_M^2(\mathbf{x}_i, \mathbf{x}_j) - D_M^2(\mathbf{x}_i, \mathbf{x}_l)] \tag{3.2}$$

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 3.4.

$$M = LL \tag{3.3}$$

$$D_M = dist(L\mathbf{X}, L\mathbf{Y}) \tag{3.4}$$

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

$$L* = Largmin \sum_{ji} d(L\mathbf{x}_i, L\mathbf{x}_j) + \sum_{i,ji} (1 - y_{il}) \xi_{ijl} sujeitoad(L\mathbf{x}_i, L\mathbf{x}_l) - d(L\mathbf{x}_i, L\mathbf{x}_j) \geq 1 - \xi_{ijl}, \xi_{ijl} \geq 0, LL \geq \tag{3.5}$$

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.

3.3 Aprendizado Não-Supervisionado

3.4 Sistemas Baseados em Regras

O mais intuitivo modelo de inteligência não-supervisionada baseada em Um determinado comportamento dispara um gatilho, que executa uma dada mudança de estado. Sistemas baseados em regras são largamente utilizados para Automação Residencial.

3.5 Modelo Oculto de Markov

. .

3.5.1 Ajuste de Modelo: Algoritmo de Baum-Welch

3.5.2 Expectation-Maximization e Algoritmo de Viterbi

3.6 Métodos para Extração de Características de Sequências

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.

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 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 (**??**).

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 de classificação supervisionado. Para tal, é necessária a extração de características sequenciais

4.4 Extração de Características

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 Sensores

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. Esta abordagem, bastante simples, não leva em conta o momento do acionamento dos dispositivos, a ordem ou o intervalo entre acionamentos.

4.4.2 N-Gramas

Parametrizado por um valor inteiro N_g, este método consiste na contagem de ocorrências na janela S_t de cada uma das possíveis combinações de N_g acionamentos consecutivos, produz um número de características $N_X = N_D^{N_g}$.

Dado o grande número de combinações possíveis, especialmente para valores de N_g maiores que 3, as características se tornam esparsas, com poucas tendo valores diferentes de 0. Para limitar o consumo desnecessário de memória com um conjunto esparso 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.

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 convoluído 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 $N_X = 60MN_X$.

4.5 Análise de Componentes Principais

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

4.6 Classificação

Após a etapa de extração de características, pode ser formulado como um problema de classificação supervisionada. 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 possuirem 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 3.5, com ji 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 do Comportamento de uma pessoa em 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

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. É possível, dado um sistema modelado pelo HMM, descobrir a sequência mais provável de estados para uma dada sequência de símbolos emitidos através do Algoritmo de Viterbi, que possui complexide linear com o número de símbolos (**??**).

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 $12 * 12$, 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 treinamento deste modelo de forma a ajustar os parâmetros para melhor se adequar à realidade observada é possível pelo algoritmo de Baum-Welch, mas não foi necessário visto que o modelo apresentou comportamento satisfatório utilizando-se os parâmetros estimados *a priori*, exibindo resultados coerentes com a realidade. Tal treinamento pode ser feito, porém, para identificação dos padrões de comportamento de cada residente, diferenciando-se então o Modelo de Markov utilizado para cada um e possibilitando a

identificação dos mesmos pelo padrão de comportamento dos sensores.
Foram obtidas, desta forma, os estados para todos os símbolos de entrada, utilizando-se o algoritmo de Viterbi, presente na biblioteca utilizada (**??**), com 10.000 símbolos por batelada, utilizando o último estado do último treino como estado inicial para o próximo. Tendo-se o estado atual, é possível se descobrir o posterior a partir do símbolo calculando-se a probabilidade de cada símbolo utilizando o Teorema de Bayes (eq. 4.1), através da função "posterior"presente na biblioteca utilizada (**??**).

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{4.1}$$

4.7.2 Extensão do modelo para Múltiplos Residentes

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 classificação de dados presentes na literatura [ARUBA] e nos dados no banco de testes da companhia *Neocontrol* 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).

5.2.1 Resultados

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* (**??**). 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 (**??**) e 2 problemas de expressão gênica: *Golub* (**??**) e *BcrHess* (**??**).

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 1. O método AM-CHIP-CLAS não desconsiderou nenhuma amostra no processo de classificação, gerando a superfície da Figura 2.

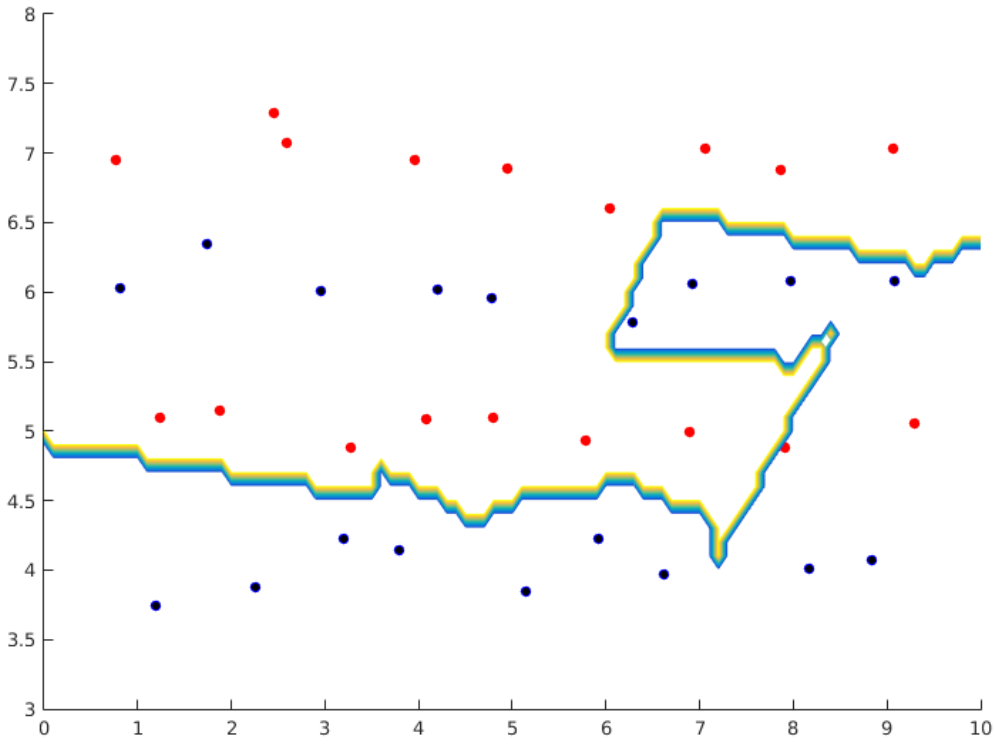

Figura 1 – Visualização da superfície de separação para o CHIP-CLAS original

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

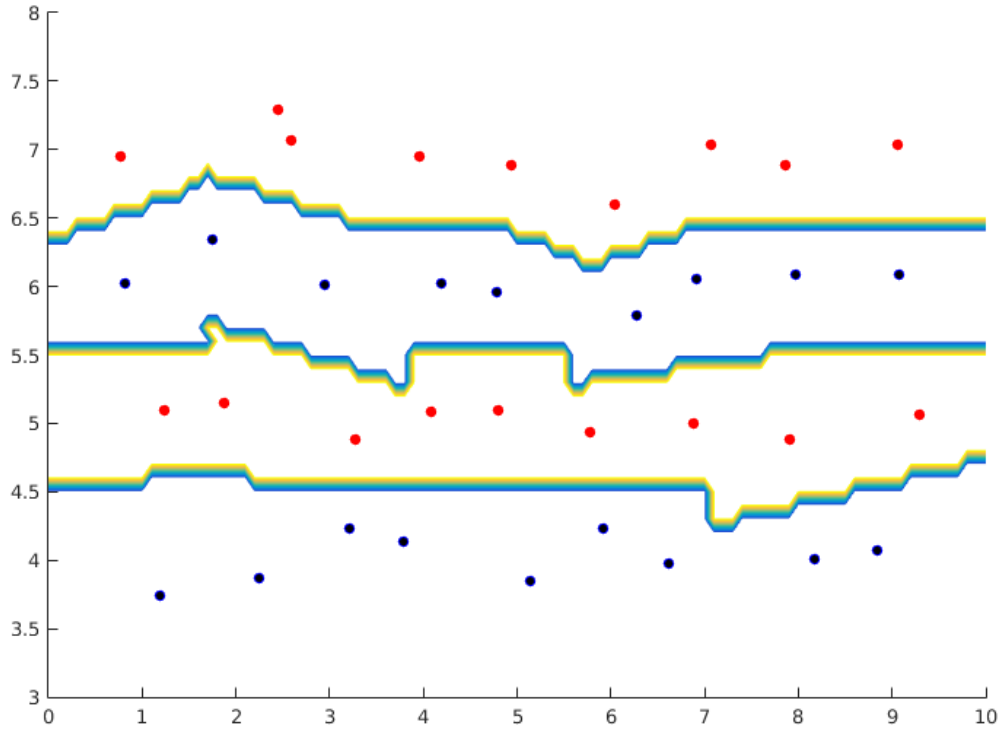

Figura 2 – Visualização da superfície de separação para o AM-CHIP-CLAS

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 1.

Tabela 1 – 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

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 (**??**). 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 2, juntamente com a média da posição de cada classificador num *ranking* de desempenho para cada base de dados.

Tabela 2 – 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
Rank Mean	2.20	2.40	2.93	2.47

Para avaliação estatística dos resultados de múltiplos classificadores, é indicado o teste de Friedman (**??**). 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 (**??**), obtendo-se o gráfico da Figura **??**, com o eixo horizontal indicando o *rank* (quanto menor, melhor o desempenho).

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 Desempenho

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 automaçao 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 (**??**). Os dados selecionados provém 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 transicões 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 3 – 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 assumpção 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