

**CLASSIFICAÇÃO DISTRIBUÍDA DE ANUROS
USANDO REDE DE SENSORES SEM FIO**

AFONSO DEGMAR RIBAS

**CLASSIFICAÇÃO DISTRIBUÍDA DE ANUROS
USANDO REDE DE SENSORES SEM FIO**

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Ciências Exatas da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Mestre em Informática.

ORIENTADOR: EDUARDO FREIRE NAKAMURA

Manaus - Amazonas

Março de 2013

© 2013, Afonso Degmar Ribas.
Todos os direitos reservados.

Ribas, Afonso Degmar

Classificação Distribuída de Anuros usando Rede de Sensores Sem Fio / Afonso Degmar Ribas. — Manaus - Amazonas, 2013

xi, 51 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal do Amazonas

Orientador: Eduardo Freire Nakamura

1. classificação de múltiplos alvos. 2. aprendizagem de máquina. 3. algoritmos distribuídos. I. Título.

[Folha de Aprovação]

Quando a secretaria do Curso fornecer esta folha, ela deve ser digitalizada e armazenada no disco em formato gráfico.

Se você estiver usando o `pdflatex`, armazene o arquivo preferencialmente em formato PNG (o formato JPEG é pior neste caso).

Se você estiver usando o `latex` (não o `pdflatex`), terá que converter o arquivo gráfico para o formato EPS.

Em seguida, acrescente a opção `approval={nome do arquivo}` ao comando `\ppgccufmg`.

Se a imagem da folha de aprovação precisar ser ajustada, use:
`approval=[ajuste] [escala] {nome do arquivo}`
onde *ajuste* é uma distância para deslocar a imagem para baixo e *escala* é um fator de escala para a imagem. Por exemplo:
`approval=[-2cm] [0.9] {nome do arquivo}`
desloca a imagem 2cm para cima e a escala em 90%.

Resumo

As Redes de Sensores Sem Fios (RSSFs) podem ser utilizadas em aplicações de conservação e estudo ambiental devido à sua capacidade de sensoriamento, monitoramento e comunicação sem fio. Dentro do contexto da Ecologia, os anfíbios são utilizados como bioindicadores de mudanças no ecossistema de uma região e podem precocemente indicar problemas ambientais. Desta forma, os biólogos monitoram a população de anuros (sapos e rãs) a fim de estabelecer estratégias de conservação do meio ambiente. Os anuros são escolhidos por causa sons que emitem (coaxar), que permitem a identificação dessas espécies por meio de microfones e processamento do sinal. Portanto, neste trabalho propomos e avaliamos alguns algoritmos distribuídos para classificação de anuros baseados em suas vocalizações em seu habitat usando RSSF. Este método é interessante pois não é intrusivo e permite o monitoramento remoto. Nossa solução cria grupos de nós sensores cujas medidas acústicas coletadas estão correlacionadas. Os dados dos nós de um mesmo grupo são combinados para gerar decisões de classificação locais. Essas decisões são então combinadas para formar uma decisão global. Para agrupar os nós com medidas correlacionadas, utilizamos o algoritmo k -means, que agrupa instâncias similares. Os experimentos mostram que, em comparação com outros algoritmos da literatura, a taxa de erro da nossa solução chegou ser até 26 pp (pontos percentuais) menor.

Palavras-chave: Algoritmos Distribuídos, Rede de Sensores Sem Fio, Classificação Distribuída, Fusão de Dados.

Abstract

Wireless Sensor Networks (WSNs) can be used in environmental conservation applications and studies due to its wireless communication, sensing, and monitoring capabilities. In the Ecology context, amphibians are used as bioindicators of ecosystemic changes of a region and can early indicate environmental problems. Thus, biologists monitor the anuran (frogs and toads) population in order to establish environmental conservational strategies. Anuran were chosen because the sounds they emit allow classification by using microphones and signal processing. In this work we propose and evaluate some distributed algorithms for anuran classification based on their calls (vocalizations) in the habit using WSNs. This method is interesting because it is not intrusive and it allows remote monitoring. Our solution builds cluster of nodes whose acoustic collected measurements are correlated. The nodes of the same group are combined to generate local classification decisions. Then, these decisions are combined to generate a global decision. We use *k*-means algorithm for clustering nodes with correlated measurements, which groups instances by similarity. Experiments show that, in comparison with other literature algorithms, the error rate of our solution were 26 pp (percentage points) lower.

Palavras-chave: distributed algorithms, wireless sensor networks, distributed classification, data fusion.

Lista de Figuras

1.1	Cenário comum em ambiente florestal.	3
2.1	Modelo de aprendizagem supervisionado.	8
2.2	Três tipos de fusão de dados.	9
3.1	Metodologia de classificação distribuída.	16
3.2	Exemplo da partição em grupos de nós.	17
3.3	Proposta para particionar a região de interesse.	18
3.4	Mapeamento do espaço físico para espaço de dados e em seguida para espaço físico. Somente os nós ativos estão representados.	19
3.5	Exemplo do esquema de SCRs para particionamento da região.	21
3.6	Avaliação da sensibilidade ao ruído branco.	22
3.7	Avaliação de sensibilidade ao ruído dos anuros.	23
3.8	Impacto do número de grupos K	24
3.9	Impacto do tamanho da entrada do k -means ϕ	25
4.1	Cenário com dois anuros vocalizando ao mesmo tempo em uma região de interesse coberta por um RSSF.	27
4.2	Classificação usando um único sensor.	29
4.3	Classificação baseada em fusão de dados.	30
4.4	Exemplo de classificação usando agrupamento.	31
4.5	Representação detalhada do desempenho de um algoritmo qualquer para classificar dois anuros.	36
4.6	Taxa de erro quando ruído é baixo.	38
4.7	Representação detalhada do desempenho dos algoritmos quando $\sigma_{wn} = 0,15$	39
4.8	Taxa de erro para nível de ruído médio.	39
4.9	Representação detalhada do desempenho dos algoritmos quando $\sigma_{wn} = 0,45$	40

4.10 Taxa de erro para nível de ruído elevado ($\sigma_{wn} = 0,60$).	41
---	----

Lista de Tabelas

3.1	Distribuição de amostras entre as classes.	20
3.2	Parâmetros de configuração padrões.	20
3.3	Resultados do teste-t para medida F1 — Impacto do ruído branco.	22
3.4	Resultados do teste-t para medida F1 — Impacto do <i>ruído dos anuros</i>	23
3.5	Resultados do teste-t para medida F1 — Impacto do número de grupos.	24
3.6	Resultados do teste-t para medida F1 — Impacto do tamanho da entrada do <i>k</i> -means.	24
4.1	Comparação entre os algoritmos propostos.	34
4.2	Distribuição de amostras entre as classes para classificação de dois anuros.	34
4.3	Parâmetros de configuração padrões para classificação de dois anuros.	35
4.4	Resultado do teste-t pareado para nível de ruído baixo.	38
4.5	Resultado do teste-t pareado para nível de ruído médio.	40
4.6	Resultado do teste-t pareado para nível de ruído elevado.	40

Sumário

Resumo	v
Abstract	vi
Lista de Figuras	vii
Lista de Tabelas	ix
1 Introdução	1
1.1 Visão geral	1
1.2 Motivação	2
1.3 Definição do problema	3
1.4 Objetivos	4
1.4.1 Objetivo geral	4
1.4.2 Objetivos específicos	4
1.5 Contribuições	4
1.6 Organização do documento	5
2 Fundamentos e trabalhos relacionados	6
2.1 Redes de sensores sem fio	6
2.2 Classificação distribuída em RSSF	7
2.3 Algoritmos de agrupamento distribuídos	11
3 Classificação de um anuro usando agrupamento e fusão de dados	13
3.1 Visão geral	13
3.2 Modelagem do sinal	13
3.3 Metodologia	14
3.3.1 Particionando a região com k -means	17
3.3.2 Complexidade	18

3.4	Avaliação e resultados	19
3.4.1	Sensibilidade ao ruído	21
3.4.2	Impacto do número de grupos	23
3.4.3	Impacto do tamanho da entrada do k -means	24
3.5	Considerações parciais	25
4	Classificação de dois anuros usando RSSF	26
4.1	Modelagem do sinal	27
4.2	Metodologia	28
4.2.1	Classificação usando um único sensor	28
4.2.2	Classificação baseada em fusão de dados	29
4.2.3	Classificação usando agrupamento e fusão de dados	30
4.2.4	Complexidade	32
4.3	Avaliação e resultados	34
4.3.1	Métricas de avaliação	35
4.3.2	Sensibilidade ao ruído	37
4.4	Considerações parciais	41
5	Conclusões	42
5.1	Considerações finais	42
5.2	Limitações	43
5.3	Trabalhos futuros	44
5.4	Publicações	45
	Referências Bibliográficas	46

Capítulo 1

Introdução

1.1 Visão geral

As Redes de Sensores Sem Fio (RSSFs) são redes sem fio *ad hoc* formadas por vários nós de baixo custo que podem fazer sensoriamento de ambientes [Akyildiz et al., 2002]. As RSSFs podem ser usadas para monitoramento, detecção e rastreamento de alvos [Li et al., 2002; Arora et al., 2004]. Particularmente, a classificação de eventos ou alvos é uma classe relevante de aplicações [Duarte & Hu, 2004; Katenka et al., 2008; Wang et al., 2006; Akyildiz et al., 2002].

Neste trabalho, tratamos o problema da classificação automática de anuros em uma RSSF usando sinais acústicos. Estamos interessados em identificar as várias espécies de anuros (sapos ou rãs) que estão presentes em uma região de interesse. Esta aplicação é relevante do ponto de vista da Ecologia, visto que os anfíbios são muito sensíveis a variações no ecossistema e sua população é rapidamente alterada quando há um problema ambiental. Deste modo, podemos detectar de forma precoce o estresse ecológico de uma região. Escolhemos os anuros em particular por causa do som emitido (coaxar) que pode ser identificado usando processamento do sinal captado por microfones.

Este trabalho estende a pesquisa iniciada por Colonna et al. [2012]. Apesar de seus avanços, consideramos que, em aplicações reais, a presença de ruídos pode impactar fortemente no desempenho dos classificadores. Ainda que os dados fossem enviados para uma base ou servidor central que fizesse todo o processamento para reduzir os efeitos do ruído, essa solução consumiria muita energia da rede.

Desta forma, soluções distribuídas baseadas em fusão de dados podem minimizar o efeito dos ruídos, como será mostrado no decorrer deste trabalho, a um baixo custo de energia, comparadas às soluções centralizadas. A ideia é aproveitar

a redundância inerente ao processo de coleta de dados em uma RSSF para recuperar a informação corrompida pelo ruído.

Dentre os diversos tipos de algoritmos distribuídos para classificação em RSSF, destacam-se os baseados em fusão de dados [Nakamura et al., 2007] e os baseados em *consensus* [Olfati-Saber et al., 2007; Kokiopoulou & Frossard, 2011] ou algoritmos *gossiping* [Dimakis et al., 2010; Chiuso et al., 2011]. Os algoritmos baseados em *gossiping* ou *consensus* preveem trocas de mensagens a cada nova medida coletada, e todos os nós da rede vão obtendo a solução de forma incremental e convergente. Como no problema estudado as medidas coletadas são amostras de áudio de alta dimensionalidade, soluções baseadas nesses algoritmos apresentam alto custo de comunicação e, conseqüentemente, alto consumo de energia.

Portanto, propomos utilizar algoritmos distribuídos baseados em fusão de dados. Nossa solução divide a RSSF em grupos de nós com medidas acústicas correlacionadas. Em seguida aplicamos técnicas de fusão de valores e fusão de decisões até obter uma decisão global. Para dividir a rede em grupos de nós, utilizamos o algoritmo *k*-means que agrupa instâncias de dados similares.

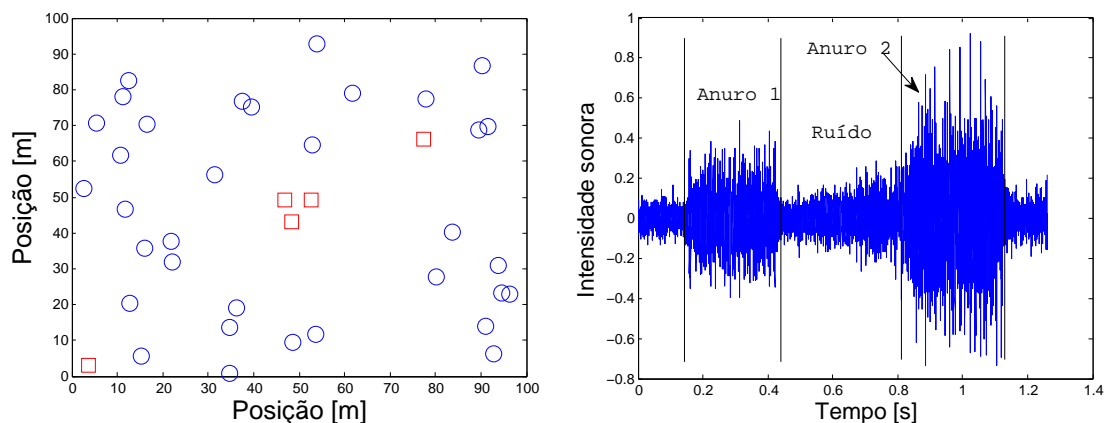
Para o problema de classificar apenas o anuro com vocalização mais predominante, o algoritmo proposto apresentou desempenho superior ao desempenho dos algoritmos da literatura em pelo menos 10 pp (pontos percentuais) para as métricas avaliadas. Para o problema de classificar dois anuros, a taxa de erro da nossa solução chegou ser até 26 pp menor.

1.2 Motivação

Este trabalho está inserido no contexto do projeto ANURA onde se deseja estimar a população de anuros (sapos e rãs) de uma determinada região de floresta para detectar precocemente stress no ecossistema. Em uma época de acentuado interesse por questões ambientais, projetos desta natureza apontam para novas formas e ferramentas para se compreender o impacto da atividade humana no meio ambiente.

Uma das causas mais predominantes para o declínio na população de anfíbios de uma determinada região está relacionada às mudanças ambientais [Carey et al., 2001; Collins & Storfer, 2003]. Desta forma, estudar as populações de anfíbios em seu habitat auxilia na detecção precoce de problemas ecológicos [Williams, 2001].

Dentre os anfíbios, monitorar os anuros é interessante, pois eles vocalizam (coaxam). Desta forma, podemos capturar o áudio emitido por meio de microfones e processá-los automaticamente em computadores. Neste trabalho, tratamos apenas



(a) Cenário com vários animais, incluindo anuros, vocalizando na floresta. Anuros são representados pelos quadrados, enquanto os círculos representam outras fontes sonoras. (b) Sinal coletado por um nó sensor caracterizado pela presença da vocalização de dois anuros e de elevado ruído.

Figura 1.1. Cenário comum em ambiente florestal.

da tarefa de reconhecer as espécies que estão vocalizando, deixando para o futuro a tarefa de estimar a população em uma determinada região.

1.3 Definição do problema

Em cenários reais como num ambiente florestal, normalmente temos diversos animais espalhados pela região, incluindo algumas espécies de anuros, vocalizando e emitindo sons ao mesmo tempo. Desta forma, é comum termos, em um ponto qualquer da região, a presença predominante do som de alguns animais e um ruído ao fundo composto pelos sons dos vários outros animais que se encontram mais distantes do ponto observado.

Uma ilustração deste cenário é apresentada na Figura 1.1. Os círculos da Figura 1.1(a) representam fontes sonoras quaisquer (animais ou sons naturais, como galhos de árvores caindo), enquanto que os quadrados representam anuros. Desta forma, sensores próximos a região central captarão com alta intensidade os sinais emitidos pelos anuros e, em menor intensidade, um ruído que é composto pela soma dos sinais emitidos por todos as outras fontes sonoras. Um exemplo deste sinal captado é apresentado na Figura 1.1(b). Pode-se observar a presença da vocalização de dois anuros, uma com maior e outra com menor intensidade, e de ruído na região de fundo.

Apesar do exemplo da Figura 1.1(b) representar as vocalizações dos anuros

ocorrendo em instantes de tempo diferentes, eventualmente ocorre dos anuros vocalizarem ao mesmo tempo. Neste caso, a classificação dos anuros se torna mais difícil.

Portanto, neste trabalho definimos dois cenários-problema a serem atacados:

1. Quando alguns anuros vocalizam em uma região de interesse coberta por uma RSSF, porém apenas o anuro com maior intensidade sonora é considerado como anuro principal e deve ser classificado. As vocalizações dos outros anuros são consideradas ruídos neste cenário;
2. Dois anuros vocalizam com predominância ao mesmo tempo na região de interesse. Ambos são considerados importantes e devem ser classificados.

1.4 Objetivos

1.4.1 Objetivo geral

Desenvolver e avaliar algoritmos distribuídos para classificação automática de anuros baseada no sinal acústico coletado em RSSF.

1.4.2 Objetivos específicos

Os objetivos específicos incluem:

- Desenvolver, treinar e avaliar um algoritmo para classificar de forma distribuída apenas um anuro, que seja adequado às características da RSSF (apresente baixa complexidade de tempo e de espaço computacional);
- Propor uma métrica de avaliação de sistemas de classificação de dois anuros;
- Desenvolver, treinar e avaliar um algoritmo para classificar dois alvos, que seja adequado às características da RSSF, considerando um modelo de propagação simplificado (apenas atenuação, sem atraso temporal);
- Desenvolver, treinar e avaliar um algoritmo para classificar dois alvos de forma distribuída, que seja adequado às características de RSSF.

1.5 Contribuições

As contribuições do trabalho são:

1. Um novo algoritmo de classificação distribuída do anuro com vocalização predominante em meio a ruído;
2. Análise de complexidade de processamento e de comunicação para o algoritmo proposto;
3. Análise comparativa de desempenho entre o algoritmo proposto e outros algoritmos da literatura;
4. Três novos algoritmos para classificar dois anuros em meio a ruído;
5. Análise do custo de processamento, de memória e de comunicação para os algoritmos propostos para classificar dois anuros;
6. Uma nova abordagem para avaliar o desempenho de classificadores para duas fontes sonoras com intensidades diferentes;
7. Avaliação do impacto do ruído no desempenho dos algoritmos propostos para classificar dois anuros.

1.6 Organização do documento

Este trabalho está organizado na seguinte forma: o Capítulo 2 apresenta os fundamentos e trabalhos relacionados. O Capítulo 3 apresenta a solução proposta para o problema de classificar um anuro. O problema de classificar dois alvos foi explorado no Capítulo 4. Por fim, temos as considerações finais apresentadas no Capítulo 5.

Capítulo 2

Fundamentos e trabalhos relacionados

2.1 Redes de sensores sem fio

As Redes de Sensores Sem Fio (RSSFs) são um tipo especial de rede *ad hoc* formadas por vários nós de baixo custo que podem realizar o sensoriamento de um ambiente e transmitir as informações pelo meio sem fio [Akyildiz et al., 2002]. Os nós sensores possuem restrições computacionais de memória e processamento, já que normalmente são construídos com microcontroladores de 8 ou 16 bits, com alguns kB para memória de programa e de dados [Mainwaring et al., 2002]. Nos sistemas e aplicações mais comuns, os nós são alimentados por baterias, o que adiciona restrições de consumo de energia. Desta forma, os algoritmos implementados para RSSFs precisam atender a essas restrições [Akyildiz et al., 2002].

RSSFs podem ser usadas para monitoramento, detecção e rastreamento de alvos [Li et al., 2002; Arora et al., 2004]. Particularmente, a classificação de eventos ou alvos é uma classe relevante de aplicações [Duarte & Hu, 2004; Katenka et al., 2008; Wang et al., 2006]. Neste tipo de aplicação, os nós sensores trabalham de forma colaborativa para identificar um evento ou alvo de interesse.

Colonna [2012] levantou três cenários para o monitoramento de anuros em relação a onde os dados seriam processados. No primeiro cenário, todos os dados coletados pelos nós sensores são transmitidos a uma estação base, que realiza todo o processamento. No segundo cenário, parte do processamento é realizado pelos nós sensores e é finalizado na estação base. O último considera todo o processamento na rede e apenas as decisões são enviadas à base. Neste trabalho, exploramos o

terceiro cenário, onde os nós processam de forma distribuída dos dados coletados e transmitem apenas a decisão global para base.

2.2 Classificação distribuída em RSSF

O problema de classificação é um problema clássico na área de aprendizagem de máquina supervisionado [Mitchell, 1997] onde queremos identificar a classe de uma determinada amostra desconhecida. Formalmente pode ser definido como segue: seja $z \in \mathcal{Z}$ a entrada ou dado medido, onde $\mathcal{Z} \subseteq \mathbb{R}^k$, $k \in \mathbb{N}$; e $x \in \mathcal{X}$ denota o vetor de características, ou seja, algumas informações relevantes extraídas do dado medido que melhor o caracterize, onde $\mathcal{X} \subseteq \mathbb{R}^d$, $d \in \mathbb{N}$. A função $f : \mathcal{Z} \rightarrow \mathcal{X}$ extrai o vetor de características do dado coletado. Esperamos que x discrimine melhor uma entrada do que o próprio z , ainda que $d < k$ para maioria das aplicações. Além disso, a função $h : \mathcal{X} \rightarrow \mathcal{C}$, chamada de hipótese, mapeia o vetor de características para uma saída, alvo ou classe $c \in \mathcal{C}$, onde \mathcal{C} é o conjunto das possíveis classes em um sistema de classificação.

O processo de aprendizado é dividido em duas etapas (Figura 2.1): (1) *treinamento*, onde se “ensina” a uma máquina a realizar a tarefa de classificação, e (2) *inferência*, onde se utiliza o “conhecimento” aprendido para classificar novas amostras. Formalmente temos que, dada uma função de extração de características f , uma hipótese h e uma entrada ou dado coletado z_i , uma predição ou inferência ocorre quando estimamos a saída

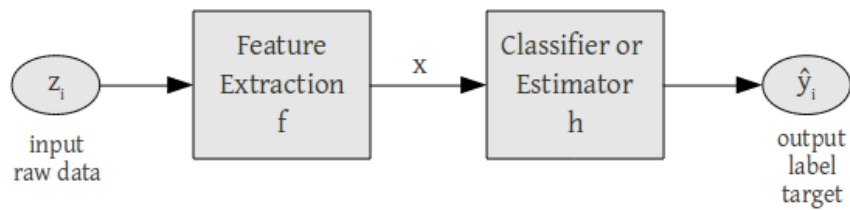
$$\hat{c}_i = h(f(z_i)) \quad (2.1)$$

onde $\hat{c}_i \in \mathcal{C}$. O algoritmo que calcula \hat{c}_i é chamado de *classificador*.

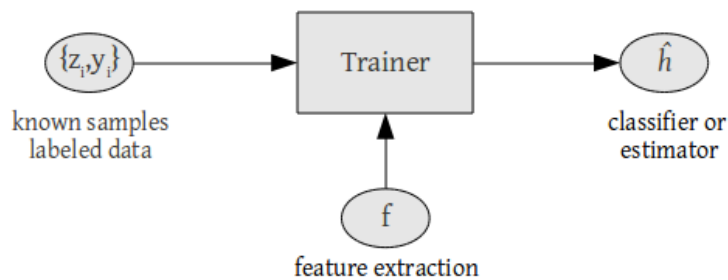
O treinamento é descrito como segue. Seja $S = \{(z_i, c_i)\}$ um conjunto de exemplos, onde $i \in \{1, \dots, m\}$, também chamado de conjunto de amostras conhecidas ou dados rotulados. Seja $J : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{R}$ uma função de custo que computa o desvio entre uma inferência \hat{c}_i e a saída conhecida c_i para um dado $f(z_i)$. O treinador (Figura 2.1(b)) é o algoritmo que encontra a hipótese \hat{h} , o que pode ser feito, por exemplo, resolvendo o problema de otimização

$$\arg \min_h \left[\frac{1}{m} \sum_{i=1}^m J(h(f(z_i)), y_i) \right]. \quad (2.2)$$

Muitos trabalhos na literatura implementam classificadores em RSSF de forma centralizada [Li et al., 2002; Chun-Ting et al., 2006; Aljaafreh & Al-Fuqaha, 2010; Sun



(a) Etapa de Inferência.



(b) Etapa de Treino.

Figura 2.1. Modelo de aprendizagem supervisionado.

& Qi, 2008; Arora et al., 2004]. Esses trabalhos implementam os classificadores nos nós sensores e realizam o processamento de forma local ou enviam todo o sinal coletado para uma estação base para processá-lo em uma máquina computacionalmente mais potente.

Fusão de dados é uma das principais técnicas de classificação distribuída para RSSF, pois pode melhorar o desempenho de classificadores em esquemas colaborativos e diminuir o consumo de energia da rede, se comparado à estratégia de enviar todos os dados coletados a uma base para processamento centralizado.

É comum dividir fusão de dados em três tipos, dependendo do nível de abstração do dado combinado [Chair & Varshney, 1986; Duarte & Hu, 2004; Brooks et al., 2003; Clouqueur et al., 2004; D'Costa & Sayeed, 2003], como mostra a Figura 2.2.

Na fusão de valores ou de medidas (FV - Figura 2.2(a)), um nó mestre combina os dados preprocessados s'_i (normalizados ou filtrados para reduzir ruído, por exemplo) e produz uma decisão ou classificação global. A fusão de características (FC - Figura 2.2(b)) ocorre quando cada nó envia para o nó mestre o vetor de características x_i extraído do dado coletado s_i . Na fusão de decisões (FD - Figura 2.2(c)), cada

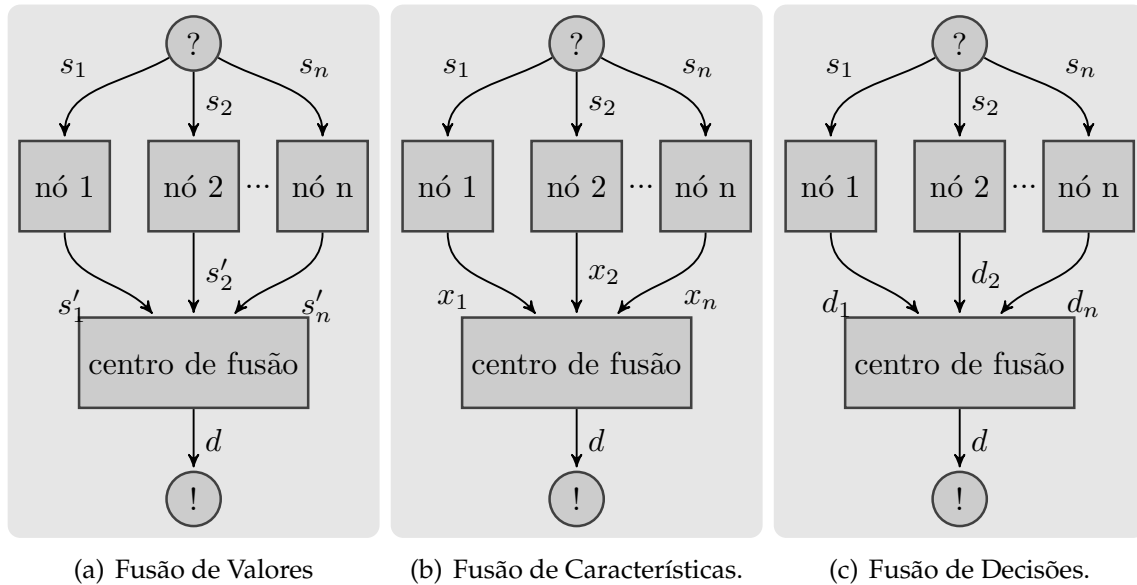


Figura 2.2. Três tipos de fusão de dados.

nó obtém uma decisão d_i independente dos demais e a envia para o nó mestre, que combina todas as decisões por meio de uma regra de fusão. No geral, as regras de fusão para FV e FC usam médias simples, enquanto a regra para FD é aplicar votação majoritária ou outra que considere as estatísticas dos dados [Chair & Varshney, 1986].

Chun-Ting et al. [2006] combinam decisões locais usando a teoria de Dempster-Shafer [Dempster, 1967; Shafer, 1976]. Decisões locais são produzidas por um classificador sobre os k vizinhos mais próximos (*k-nearest neighbor - k-NN*) ponderado, cuja saída é usada para construir as atribuições de probabilidade básica (*basic probability assignment - BPA*) da teoria de evidências. A regra de decisão é escolher a classe que tem a maior probabilidade.

Investigando o problema de classificação de alvos, Duarte & Hu [2003, 2004] propuseram um esquema de fusão de decisões baseada na distância entre alvo e os nós sensores. A ideia é que sensores próximos ao alvo têm mais condições de inferir a classe com mais precisão do que sensores distantes.

Considerando a possibilidade dos sensores apresentarem medidas com falhas ou corrompidas por ruído, Krishnamachari & Iyengar [2004] e Katenka et al. [2008] propuseram uma regra de fusão de decisão em que os sensores primeiro corrigem suas decisões locais usando as decisões de seus vizinhos. Em seguida, os nós transmitem suas decisões locais para o nó mestre. Esses algoritmos exploram o fato de que medidas são espacialmente correlacionadas, enquanto falhas são estatisticamente

não correlacionadas.

Luo et al. [2006] estenderam o trabalho de Krishnamachari & Iyengar [2004] propondo um método para escolher o tamanho apropriado da vizinhança (número de vizinhos) a fim de aumentar a eficiência em termos de energia. Bahrepour et al. [2010] propuseram um esquema de votação baseado em reputação onde cada nó, assumindo que sua decisão está correta, julga quão bem outros nós podem obter a decisão correta. Então, o nó mestre considera apenas o voto de nós com alta reputação.

O algoritmo MRI (*Multi-Resolution Integration*) [Qi et al., 2001; Prasad et al., 1994] foi usado como regra de decisão por Wang et al. [2002] e Wang & Qi [2002]. Este algoritmo é muito adequado quando decisões locais são representados por saídas com intervalos de confiança para cada classe. Ele utiliza um esquema que prioriza a saída que mais esteve coberta pelos intervalos de confiança.

Todos os trabalhos citados anteriormente não consideram ou ignoram a presença de ruído no ambiente (ruído Gaussiano). Desta forma, todos podem ter seu desempenho afetado caso sejam empregados em ambientes como o de floresta, onde a relação sinal ruído é baixa, devido ao alto ruído gerado pela soma dos sons emitidos pelos vários animais comumente presentes.

Um grande número de trabalhos está preocupado com o canal de transmissão entre sensores e o nó mestre. Os chamados algoritmos sensíveis ao canal (*channel-aware algorithms*) são implementados de duas formas diferentes: usando códigos corretores de erros [Wang et al., 2005, 2006; Pai, 2010] ou usando regras de decisão que consideram algum tipo de conhecimento das estatísticas do canal [Lin et al., 2005; Wang et al., 2009; Chen et al., 2009, 2002]. Esses trabalhos consideram que os dados enviados podem sofrer alterações devido a presença de ruídos no canal. No entanto, em termos práticos, os nós sensores atuais utilizam transceptores que trabalham com esquemas de modulação digital [Mainwaring et al., 2002]. Desta forma, é muito pouco provável que um dado seja alterado dentro de uma estrutura de pacote sem que o próprio transceptor utilizado detecte o erro.

Uma preocupação de projeto em sistemas de fusão de dados é decidir quando usar fusão de valores, de características ou de decisões. Brooks et al. [2003] apresentam um *framework* para processamento de sinais colaborativos para classificação e rastreamento de alvos em rede de sensores. Os autores consideram vários cenários de fusão de valor e fusão de decisão usando um ou múltiplos sensores e modalidades (exemplo: acústico e sísmico). Eles sugerem que “FD é preferível devido à baixa carga computacional e de comunicação”, e necessita de menos amostras de treino enquanto que FV pode potencialmente produzir melhor performance na

classificação.

Clouqueur et al. [2004] apresentaram dois algoritmos colaborativos de detecção de alvos que são tolerantes a falhas nos sensores, baseados em FV e FD. Os autores mostraram que FD pode ter melhor desempenho do que FV quando o número de falhas nos sensores aumenta. Malhotra et al. [2008b] realizaram experimentos com medidas acústicas reais e encontraram desempenho similar para FD e FC usando os classificadores k -NN e Maximum Likelihood.

D’Costa & Sayeed [2003] mostraram que, quando as medidas dos nós são perfeitamente correlacionadas, tirar a média das medidas produz o resultado ótimo. Também mostraram que FD é preferida quando sensores coletam medidas independentes. No geral, medidas de diferentes nós exibem uma mistura de medidas correlacionadas e independentes. Desta forma, um classificador ótimo deve combinar FC ou FV com FD. Baseado nesses resultados, D’Costa et al. [2004] modelaram eventos como fontes gaussianas no tempo e espaço e propuseram dividir a região em uniformes *regiões de coerência espacial* (*spatial coherence regions* — SCRs) onde as medidas são fortemente correlacionadas. Fusão de características deve ser utilizada entre nós dentro de um SCR para reduzir os efeitos do ruído, e FD entre diferentes SCRs para reduzir as variabilidades inerentes ao sinal.

2.3 Algoritmos de agrupamento distribuídos

No problema de agrupamento, uma coleção de dados não rotulados é subdividida em grupos de amostras similares segundo algum critério predefinido. explorando somente a similaridade entre os dados. O resultado são grupos de amostras semelhantes segundo algum critério [Duda et al., 2001]. Exemplos de algoritmos são k -means [MacQueen, 1967], Esperança-Maximização (EM) [Dempster et al., 1977] e *Maximum Likelihood Estimator* (ML) [Edgeworth, 1908].

Duas versões distribuídas do k -means (DKM) baseada em “consensus” [Lynch, 1997; Olfati-Saber et al., 2007] foram propostos por Forero et al. [2008b]. O algoritmo usa os multiplicadores de Lagrange para assegurar que os centroides locais são iguais para todos os nós. Para ambos algoritmos, cada sensor realiza os seguintes passos iterativamente:

1. Atribui observações ao cluster;
2. Atualiza seus centroides;
3. Divulga seus novos centroides para os sensores vizinhos;

4. Atualiza seus multiplicares de Lagrange; e
5. Divulga seus novos multiplicadores de Lagrange para os sensores vizinhos;

O problema de agrupamento pode ser visto como uma mistura de modelos de densidade de probabilidade. Neste sentido, o problema torna-se estimar os parâmetros das funções densidade de probabilidade (PDFs), e o algoritmo EM normalmente é utilizado para este fim [Duda et al., 2001].

Nowak [2003] propôs um algoritmo incremental (I-DEM) que constrói um caminho através de todos os nós da rede para estimar os parâmetros da distribuição gaussiana. Nesta abordagem, cada nó computa suas estatísticas locais no passo Expectation e, no passo Maximization, as estatísticas globais suficientes são acumuladas considerando as estatísticas locais dos nós no caminho pre selecionado. Um passo adicional é necessário para atualizar as estatísticas globais no caminho. Esta abordagem tem muitos problemas, tais como: necessidade de construir um caminho com todos os nós, precisa atualizar nós em uma ordem inversa e pode se tornar lento quando se aumenta o número de nós.

Outras variações foram propostas por Gu [2008] e Pereira et al. [2010] que usam o filtro de consenso médio [Lynch, 1997] para misturas de Gaussianas, e Forero et al. [2008a, 2011] para misturas não-Gaussianas.

Capítulo 3

Classificação de um anuro usando agrupamento e fusão de dados

3.1 Visão geral

Neste capítulo consideramos o problema de classificar de forma distribuída um único alvo baseado na sua vocalização (sinal acústico emitido).

Modelamos o ambiente como um *grid* de $N \times N$ nós separados l metros um do outro. Foi acrescentado um desvio aleatório na posição dos nós que segue a distribuição normal com desvio padrão σ_s . Um evento é caracterizado por um sinal acústico s (um vetor de amplitudes de energia) emitido de uma posição aleatória próximo ao centro do *grid*. Cada nó recebe o sinal s' que é o sinal s alterado devido à propagação. Em seguida, descrevemos o modelo do sinal acústico para medidas dos sensores.

3.2 Modelagem do sinal

O sinal capturado pelos nós, s' , é diferente de s devido à atenuação, atraso no tempo e efeitos de ruído. Consideramos o mesmo modelo de atenuação utilizado por Malhotra et al. [2008a]. O sinal atenuado e atrasado s' em um sensor com distância d da fonte é calculado como segue:

$$s'(s, d) = \text{aten}(\text{atraso}(s, d), d), \quad (3.1)$$

onde s é o vetor correspondente ao sinal de fonte e as funções $\text{aten}(\cdot)$ e $\text{atraso}(\cdot)$ atenuam e atrasam o sinal quando propagado por uma distância d . Essas funções

são definidas como:

$$\text{atraso}(\mathbf{s}(t), d) = \mathbf{s}(t - d/v) \quad (3.2)$$

e

$$\text{aten}(\mathbf{s}, d) = \mathbf{s} \cdot 10^{-\frac{\alpha d}{20}}, \quad (3.3)$$

onde $\mathbf{s}(t)$ denota o vetor de audio original no instante t , v é a velocidade do som no meio da propagação e α é o fator de atenuação [Malhotra et al., 2008a].

Definimos duas fontes de ruído: *ruídos de anuros* e *ruído branco*. Os *ruídos de anuros* são três vocalizações de outros anuros posicionados aleatoriamente no grid. Essas vocalizações representam o caso comum onde vários sinais de diferentes indivíduos podem interferir uma vocalização principal. Limitamos a amplitude dessas vocalizações adicionais a

$$\mathbf{s}'' = \delta \frac{\mathbf{s}_j}{\max(\text{abs}(\mathbf{s}_j))}, \quad (3.4)$$

onde \mathbf{s}_j é o sinal diferente do evento principal. Neste caso, δ representa a intensidade das vocalizações adicionais. O *ruído branco* é um ruído branco Gaussiano aditivo $\mathbf{r}_i \sim \mathcal{N}(0, \sigma)$.

Portanto, o sinal capturado por cada nó i é

$$\mathbf{s}_i = \mathbf{s}'(\mathbf{s}, d) + \sum_{j=1}^3 \mathbf{s}'(\mathbf{s}_j'', d_j) + \mathbf{r}_i. \quad (3.5)$$

3.3 Metodologia

Nesta seção será descrita a solução proposta. Como o foco do trabalho está na tarefa de classificação, consideramos que um algoritmo de detecção de eventos, baseado em limiar de energia, nos indica toda vez que uma vocalização de um anuro desconhecido for captada na região de interesse. Esse algoritmo fica monitorando o áudio coletado e, quando a energia do sinal for maior que um limiar, indica para o algoritmo de classificação que uma nova amostra de sinal precisa ser classificada. Algoritmos de detecção sofisticados foram propostos por Ding et al. [2004] e Li et al. [2002] e podem ser usados para este propósito. Esses algoritmos são mais precisos para identificar os eventos, pois trabalham analisando o nível de ruído do ambiente.

O algoritmo de detecção indica aqueles m nós com maiores níveis de potência do sinal acústico, os chamados *nós ativos*. Como o sinal propagado passa por uma

degradação que aumenta com a distância, os nós sensores mais distantes do evento captam um sinal que pode ser predominantemente ruído. Se esses nós participarem do processo de classificação, eles podem contribuir negativamente, já que suas classificações apresentarão resultados com menor confiabilidade em relação aos nós mais próximos das fontes. Por esse motivo, consideramos que apenas os *nós ativos* participam do processo de classificação. O nó com maior nível de potência do sinal é eleito como *nó mestre* e coordena os *nós ativos*.

O pseudocódigo do algoritmo proposto é apresentado no Algoritmo 3.1 e ilustrado na Figura 3.1. Propomos dividir a região em subregiões não regulares e aplicar FV em nós dentro de uma subregião e FD para as decisões de diferentes subregiões. Mas, ao invés de usar as SCRs de D'Costa et al. [2004] propomos usar o algoritmo de particionamento baseado em agrupamento descrito em detalhe na Seção 3.3.1.

Algoritmo 3.1 Classificação de um alvo

Entrada: conjunto de vetores de sinal medidos pelos nós ativos S_a .

Saída: uma decisão global $d \in C$.

- 1: $G \leftarrow \text{PARTITIONREGION}(S_a)$
 - 2: **para todo** $g \in G$ **faça**
 - 3: $d_g \leftarrow \text{VALUEFUSION}(S_g)$, onde $S_g = \{s_i \mid \text{nó } i \in g\}$
 - 4: **fim para**
 - 5: $d \leftarrow \text{MAJORITYVOTE}(\{d_g \mid g \in G\})$
 - 6: **retorne** d
-

O Algoritmo 3.1 é usado pelo algoritmo de detecção. O nó mestre encontra os grupos de nós com medidas correlacionadas executando o Algoritmo 3.3, descrito mais tarde. O nó mestre então reporta aos nós ativos o grupo ao qual pertencem. Em seguida, cada grupo elege um líder, conhecido como *cluster head*, para coordenar as tarefas do grupo. Assim, cada grupo g obtém uma decisão d_g aplicando FV (Algoritmo 3.2) sobre o conjunto dos sinais medidos pelos nós do grupo g (linha 3). Por fim, os líderes enviam as decisões locais para o nó mestre, que as combina, aplicando a função `MAJORITYVOTE`, para obter a decisão global. O mestre, então, reporta o resultado para estação base.

O algoritmo de fusão de valor é descrito no Algoritmo 3.2. As características selecionadas são os doze primeiros coeficientes cespetais na frequência Mel (MFCCs) [Davis & Mermelstein, 1980] de um banco de filtros de canais. O classificador é uma árvore de decisão [Mitchell, 1997] com podagem que é muito adequado para RSSFs, devido aos requisitos de economia de energia. Este mesmo classificador foi utilizado em um sistema real de monitoramento da espécie *Bufo marinus* na

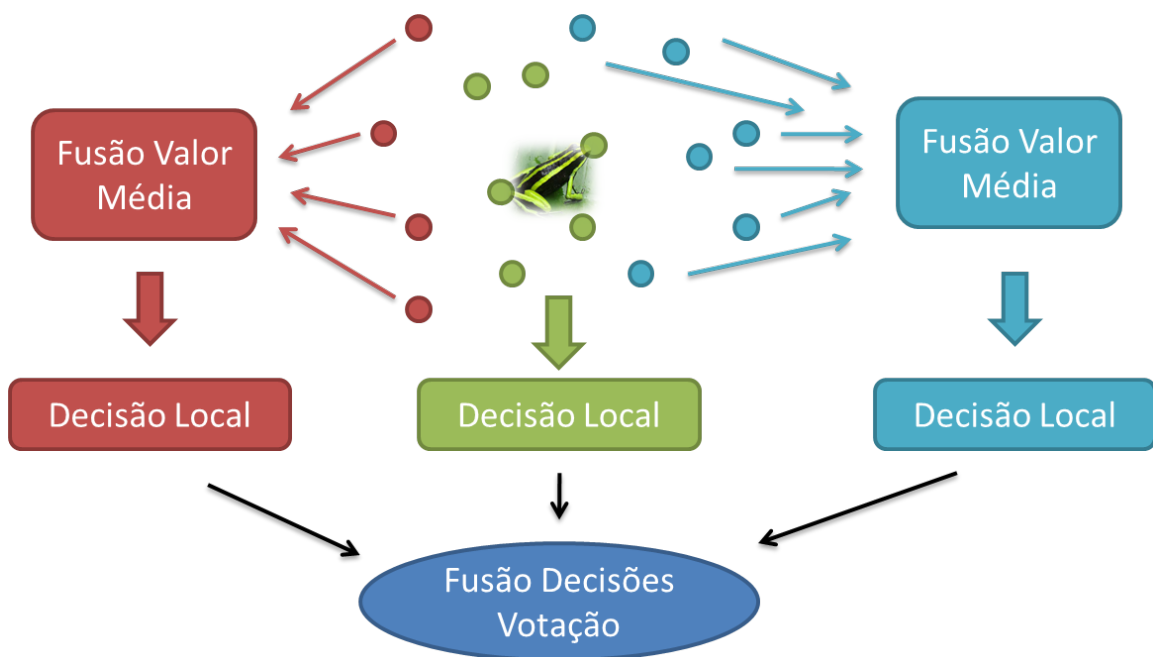


Figura 3.1. Metodologia de classificação distribuída. Os nós próximos ao evento são agrupados utilizando o algoritmo k -means. Cada grupo realiza FV (Algoritmo 3.2) sobre o conjunto de sinais medidos por cada nó sensor do grupo gerando decisões locais. As decisões são combinadas por meio de FD para gerar a decisão global.

Austrália [Hu et al., 2009] pelos mesmos motivos. Com este classificador, os nós necessitam armazenar apenas as regras obtidas que podem ser implementadas com blocos de *ifs*.

Algoritmo 3.2 Fusão de Valores.

Entrada: conjunto de vetores de sinal S .

Saída: uma decisão de classificação (a classe identificada).

- 1: **função** VALUEFUSION(S)
 - 2: $\mathbf{s} \leftarrow \text{MEAN}(S)$
 - 3: $\mathbf{x} \leftarrow \text{EXTRACTFEATURES}(\mathbf{s})$
 - 4: **retorne** CLASSIFY(\mathbf{x})
 - 5: **fim função**
-

A Figura 3.2 ilustra o cenário com um *grid* de 10×10 nós. Os nós mais próximos ao evento recebem o sinal de áudio com maior intensidade e, portanto, são considerados os nós ativos. Os nós sensores com medidas correlacionadas ficam no mesmo grupo.

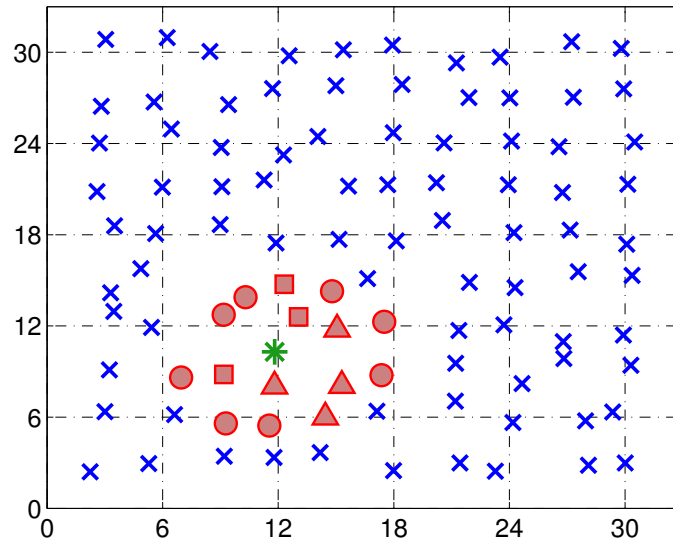


Figura 3.2. Exemplo da partição em grupos de nós. O * representa um anuro vocalizando. Os outros símbolos representam nós sensores. Os elementos representados por \times são nós que não participam da tarefa de classificação. Os demais símbolos (Δ , \square , \circ) formam os diferentes grupos de nós com medidas correlacionadas.

3.3.1 Particionando a região com k -means

Nesta seção, apresentamos o algoritmo baseado em k -means para particionar a região de interesse em subregiões, onde as medidas coletadas são correlacionadas. O k -means é um algoritmo de aprendizado não supervisionado que agrupa instâncias não rotuladas por similaridade [Mitchell, 1997]. O k indica o número de grupos. Propomos formar grupos de nós baseados nos grupos de instâncias retornadas pelo k -means, conforme mostrado na Figura 3.3. O pseudocódigo é apresentado no Algoritmo 3.3.

Os nós ativos enviam uma fração ϕ de suas medidas s para o nó mestre para formar o conjunto de entrada do algoritmo k -means. Devido às capacidades reduzidas dos nós sensores e para economia de energia, preferimos usar uma fração de s no lugar do vetor inteiro. A saída são K grupos de instâncias similares que são mapeados para formar grupos de rede física (*clusters*).

Este mapeamento é ilustrado na Figura 3.4, onde cada nó ativo n_i coleta seu sinal s_i no espaço físico. No espaço de dados, cada ponto corresponde a um sinal s_i . Para simplificar, ilustramos somente duas dimensões, mas em aplicações reais pode-se considerar qualquer espaço N_s -dimensional, onde N_s é o tamanho do vetor s_i . Em seguida, o k -means é executado sobre as amostras do espaço de dados produzindo



Figura 3.3. Proposta para particionar a região de interesse. O k -means será aplicado sobre os vetores de áudio coletados pelos nós sensores e retornará os grupos de nós com medidas correlacionadas.

Algoritmo 3.3 Particionamento Baseado em Agrupamento

Entrada: conjunto de vetores do sinal S , um valor de porcentagem ϕ do sinal capturado e o número de grupos K .

Saída: conjunto de K grupos $G = \{g_1, \dots, g_K\}$ onde g é um conjunto de nós ativos.

```

1: função PARTITIONREGION( $S, \phi, K$ )
2:    $Z \leftarrow \emptyset$ 
3:   para todo  $s \in S$  faça
4:      $Z \leftarrow Z \cup \{ \text{FRACTION}(s, \phi) \}$ 
5:   fim para
6:    $G \leftarrow k\text{-MEANS}(Z, K)$ 
7:   retorne  $G$ 
8: fim função

```

grupos de dados com amostras similares. Neste momento, os grupos de dados no espaço de dados são mapeados para *clusters* de rede no espaço físico através do índice i . Desta forma, nós com medidas similares (no espaço de dados) ficarão no mesmo grupo (agora no espaço físico), e nós com medidas não correlacionadas ficarão separados. Por exemplo, na Figura 3.4 pode-se observar que os sinais s_1 e s_2 , coletados respectivamente pelos nós n_1 e n_2 , pertencem ao mesmo grupo no espaço de dados. Por isso, os nós n_1 e n_2 aparecem no mesmo grupo no espaço físico.

3.3.2 Complexidade

A complexidade de tempo do Algoritmo 3.3 é dada pela complexidade do k -means, da ordem de $O(mN_sKT)$ [Duda et al., 2001], onde m é o número de instâncias que é igual ao número de nós ativos, N_s é a dimensão da instância de entrada, K é o número de grupos e T é o número de iterações para o algoritmo convergir em uma solução.

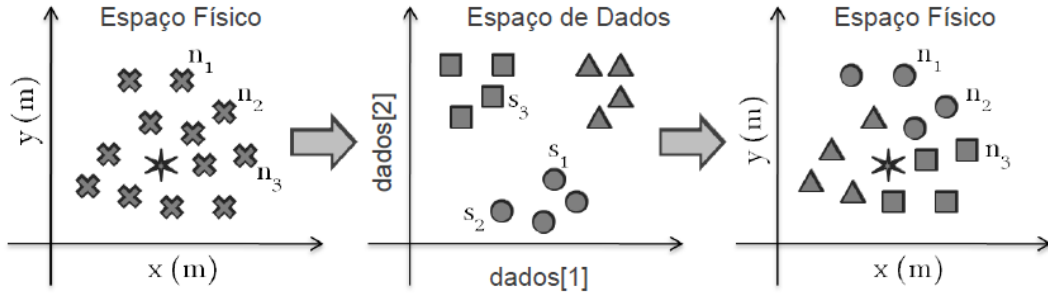


Figura 3.4. Mapeamento do espaço físico para espaço de dados e em seguida para espaço físico. Somente os nós ativos estão representados.

A operação mais complexa no Algoritmo 3.2 é o cálculo da transformada rápida de Fourier (*Fast Fourier Transform* — FFT), aplicado para extrair os coeficientes MFCCs dos sinais medidos, e é executado em $O(N_s \log N_s)$. Esta operação é realizada para cada grupo, o que resulta em $O(KN_s \log N_s)$. A complexidade do voto majoritário pode ser desprezada. Portanto, a complexidade de tempo do algoritmo proposto é $O(mN_sKT) + O(KN_s \log N_s)$.

Se q é o custo de enviar um escalar através do rádio, então o custo de comunicação do algoritmo de particionamento da região Q_1 é

$$Q_1 = \phi N_s(m - 1)q + (m - 1)q, \quad (3.6)$$

isto é, cada nó ativo envia uma fração de seu sinal medido, e o nó mestre reporta aos $m - 1$ nós o grupo ao qual pertencem. O custo das linhas 2 a 4 do Algoritmo 3.1 é

$$Q_{24} = (m - K)N_sq, \quad (3.7)$$

isto é, os nós ativos enviam seus valores medidos s_i para o *cluster head* local para realizar a fusão de valores. Por fim, cada *cluster head* envia para o nó mestre sua decisão local, resultando em $Q_5 = Kq$. O custo total de comunicação é

$$Q = Q_1 + Q_{24} + Q_5 = N_s(\phi m - \phi + m - K)q + (m + K - 1)q. \quad (3.8)$$

3.4 Avaliação e resultados

Nesta seção, apresentamos a avaliação da solução proposta através de experimentos de simulação implementados em Matlab, e posteriormente, comparamos com outros

Tabela 3.1. Distribuição de amostras entre as classes.

Classe	Nome da Espécie	# sílabas	%
1	<i>Adenomera andre</i>	313	16,03
2	<i>Ameerega trivittata</i>	160	8,19
3	<i>Leptodactylus hylaedactylus</i>	47	2,41
4	<i>Hyla minuta</i>	227	11,62
5	<i>Hypsiboas cinerascens</i>	223	11,42
6	<i>Leptodactylus cuscus</i>	52	2,66
7	<i>Osteocephalus cophagus</i>	420	21,51
8	<i>Rhinella cranulosa</i>	61	3,12
9	<i>Scinax ruber</i>	450	23,04
Total	—	1953	100,0

Tabela 3.2. Parâmetros de configuração padrões.

Parâmetro	Símbolo	Valor
Número de nós em uma dimensão do <i>grid</i>	N	10
Distância entre nós no <i>grid</i>	l	3 m
Desvio padrão de posicionamento dos nós	σ	0,5 m
Tamanho do vetor do sinal acústico	N_s	1147
Fator de atenuação acústica	α	0,1053 dB/m
Número de nós ativos	m	25
Tamanho da entrada do <i>k</i> -means (percentual do sinal)	ϕ	1
Desvio padrão do ruído branco	σ_{wn}	0,15
Número de grupos	K	3
Número máximo de interações do <i>k</i> -means	T_{max}	30

dois esquemas. Estamos utilizando o conjunto de amostras coletadas por Colonna et al. [2012]. O áudio original foi gravado em 8820 Hz e dividido em segmentos menores de vocalizações, chamados sílabas, de tamanho fixo de 1147 amostragens de energia. No total temos 1953 sílabas distribuídas entre nove espécies de anuros, de acordo com a Tabela 3.1.

A análise foi conduzida seguindo a metodologia de validação cruzada estratificada de 10-*folds* e as métricas consideradas foram: acurácia (taxa de classificação) e macro F1 [Mitchell, 1997]. Para cada nova instância a ser classificada, repetimos o processo 30 vezes com alvos em diferentes posições. Somente os valores médios para os dez *folds* são plotados nos gráficos para cada análise. Além disso, realizamos o teste-t pareado com nível de confiança de 95% para comparar os resultados, mas somente os resultados do teste-t para a medida F1 foram apresentados.

A Tabela 3.2 apresenta os parâmetros de configuração padrões utilizados nos experimentos. O fator de atenuação¹ α da Equação 3.3 é o mesmo valor utilizado por Malhotra et al. [2008a] e indica uma severa taxa de decaimento do sinal propagado.

Para efeitos de comparação, implementamos um esquema que usa apenas fusão

¹O fator de atenuação pode ser calculado usando o *Atmospheric Sound Absorption Calculator* (<http://www.csgnetwork.com/atmossndabsorbcalc.html>).

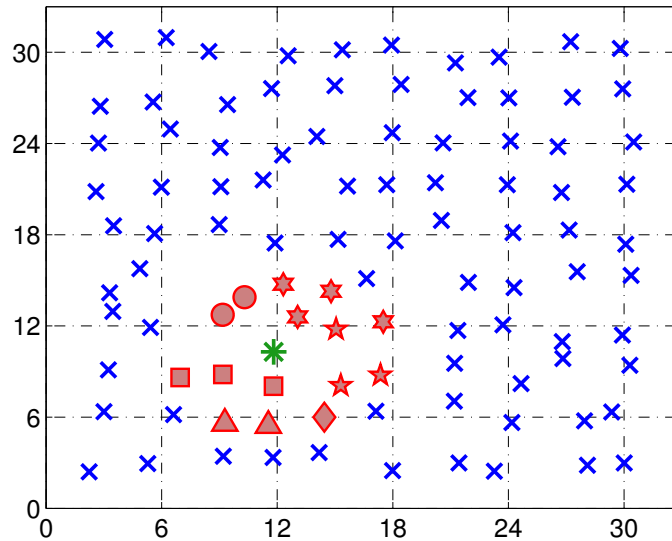


Figura 3.5. Exemplo do esquema de SCRs para particionamento da região.

de decisões e uma adaptação do esquema de SCRs de D’Costa et al. [2004]. Para o esquema *somente fusão de decisões* (para abreviar, *somente FD*), consideramos o voto majoritário dos 25 nós ativos sem nenhum agrupamento ou particionamento. Esta é uma abordagem frequentemente adotada em sistemas de classificação distribuída.

No esquema SCR, a região é particionada em subregiões disjuntas de tamanho (área) uniforme. A Figura 3.5 mostra este esquema aplicado na mesma instância apresentada na Figura 3.2. As subregiões são quadrados demarcados pelas linhas tracejadas. Estamos considerando SCRs de 6×6 m contendo por volta de quatro nós sensores. Um SCR com pelo menos um nó ativo é um SCR ativo. No exemplo, o evento produz seis SCRs ativos. Em seguida, o esquema executa os passos das linhas 2 a 5 do Algoritmo 3.1 para os SCRs ativos. O algoritmo original propõe o uso de fusão de características ao invés de fusão de valores, mas nossos testes preliminares revelaram baixo desempenho de classificação, apesar do custo de comunicação requerido ser muito menor.

3.4.1 Sensibilidade ao ruído

Primeiro analisamos o impacto do ruído em nosso algoritmo. Como descrito na Seção 3.2, temos dois tipos de ruído: branco e dos anuros parametrizados por σ e δ , respectivamente.

A simulação foi configurada com os parâmetros da Tabela 3.2, exceto pelo

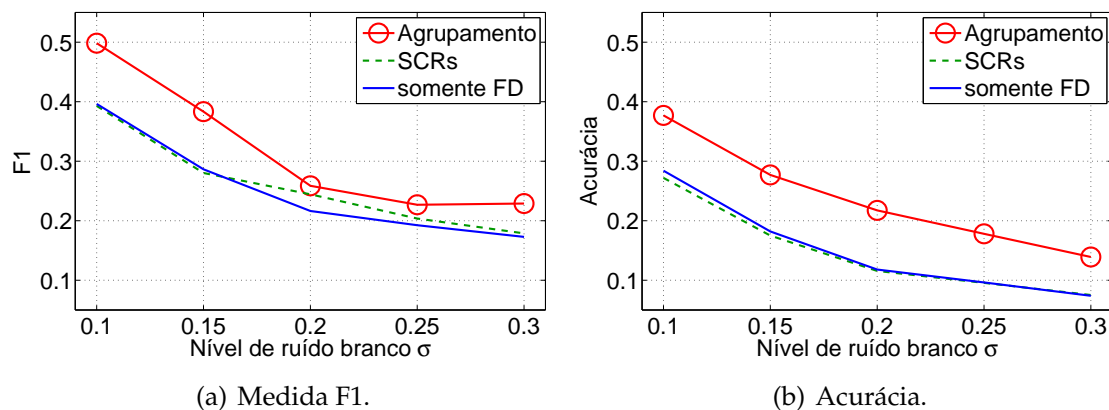


Figura 3.6. Avaliação da sensibilidade ao ruído branco.

Tabela 3.3. Resultados do teste-t para medida F1 — Impacto do ruído branco.

σ	0,10	0,15	0,20	0,25	0,30
min	3,2	5,9	-2,0	-1,8	-0,9
max	17,9	14,7	4,9	6,4	10,9
médio	10,5	10,3	1,4	2,3	5,0
%	26,8	36,8	5,8	11,4	28,0

parâmetro a ser analisado. A Figura 3.6 mostra o F1 e a acurácia quando o ruído branco varia. O desempenho do esquema somente FD e SCRs foi muito similar, mas inferior à solução proposta chamada, deste ponto em diante, de *Agrupamento*. A diferença entre as curvas é maior para valores iniciais e decresce no final. Isto ocorre porque quando o nível de ruído aumenta, todos os algoritmos convergem para um ponto onde todas as instâncias desconhecidas são classificadas como a classe que apresenta maior ruído no áudio original.

A Tabela 3.3 apresenta os resultados do teste-t para medida F1 que confirmam os resultados acima. A segunda e terceira linhas representam o intervalo de confiança retornado pelo teste. Ele indica, por exemplo, que com $\sigma = 0,15$, a medida F1 do esquema usando agrupamento é de 5,9 a 14,7 pp (pontos percentuais) melhor que o esquema dos SCRs. O “médio” é o ponto central do intervalo de confiança em pp. A quarta linha representa a diferença percentual (em %) entre os valores médios dos esquema Agrupamento e SCRs. Exceto para $\sigma = 0,20$ e $0,25$, nossa solução foi pelo menos 26% melhor que esquema dos SCRs. Considerando somente a acurácia, esta diferença é ainda maior (Figura 3.6(b)).

Seguindo o mesmo procedimento usado para analisar o ruído branco, a Figura 3.7 mostra o impacto do “ruído dos anuros” no sistema de classificação. O esquema por Agrupamento apresenta um desempenho significativamente superior até $\delta = 0,6$, com F1 pelo menos 36% maior do que os SCRs. Altos valores de δ

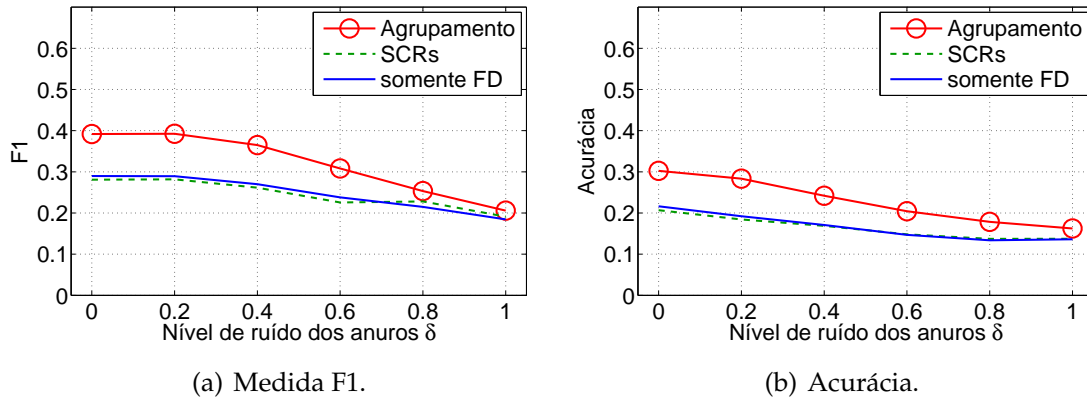


Figura 3.7. Avaliação de sensibilidade ao ruído dos anuros.

Tabela 3.4. Resultados do teste-t para medida F1 — Impacto do ruído dos anuros.

δ	0,0	0,2	0,4	0,6	0,8	1,0
min	4,8	6,3	5,6	4,8	-0,1	-0,2
max	17,4	15,9	15,0	11,8	5,1	3,1
médio	11,1	11,1	10,3	8,3	2,5	1,5
%	39,5	39,4	39,5	36,6	11,0	7,7

(próximos a 1) indicam que há outros anuros na região vocalizando quase na mesma intensidade da vocalização principal. Nesses casos, nenhum dos esquemas apresenta bons resultados, porque os ruídos se tornam tão predominantes que se confundem com os eventos a serem classificados. Este se torna então um problema de classificação de múltiplos alvos e será abordado no próximo capítulo deste trabalho.

Mais uma vez, os resultados dos esquemas de referência foram muito similares, indicando que o SCRs não captura o comportamento de eventos com fontes não gaussianas do sinal.

3.4.2 Impacto do número de grupos

Em seguida, investigamos o impacto do número de grupos K . A Figura 3.8 mostra os resultados para diferentes valores de K . Plotamos os dois esquemas de referência para comparação, mas eles não possuem o parâmetro “número de grupos” a ser variado. No caso do esquema usando SCRs, mesmo que o número de clusters ativos pudesse mudar para cada novo evento, não há nenhum parâmetro a ser configurado.

Nosso esquema apresenta desempenho superior quando $K = 3$, e gradualmente decresce para mais grupos. Assim, $k = 3$ é a melhor escolha. Maiores valores de K levam a poucos nós por grupo, o que impacta na acurácia da classificação local devido à baixa relação sinal ruído. Em outras palavras, a média dos sinais de

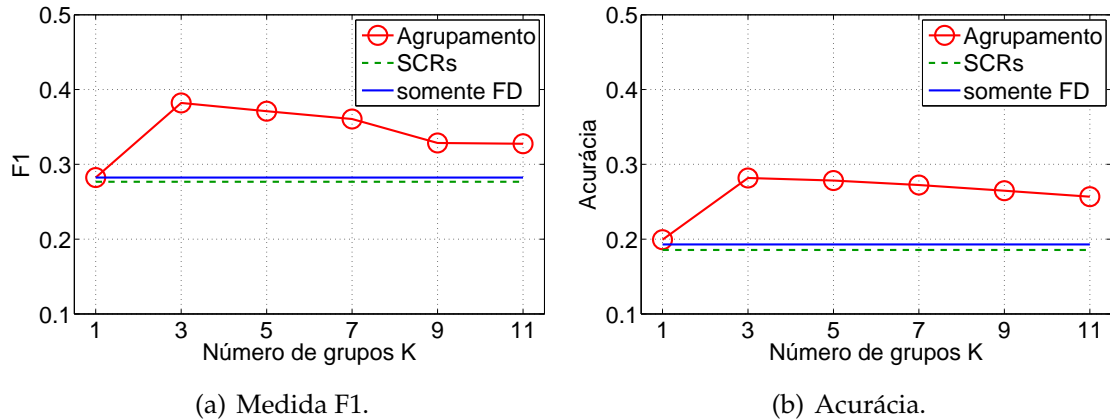


Figura 3.8. Impacto do número de grupos K .

Tabela 3.5. Resultados do teste-t para medida F1 — Impacto do número de grupos.

K	1	3	5	7	9	11
min	-2,9	4,1	4,1	2,6	-1,4	-1,7
max	4,1	17,0	14,8	14,2	11,8	11,94
médio	0,6	10,5	9,4	8,4	5,2	5,1
%	2,1	38,1	34,1	30,4	18,8	18,4

Tabela 3.6. Resultados do teste-t para medida F1 — Impacto do tamanho da entrada do k -means.

ϕ	0,2	0,4	0,6	0,8	1,0
min	0,1	0,5	3,4	2,3	6,3
max	8,9	9,0	15,7	15,3	19,7
médio	4,5	4,3	9,5	8,8	13
%	16,1	15,3	34,2	31,4	46,5

mais nós produzem uma maior amortização do ruído. O número de grupos é importante porque impacta na complexidade de tempo e no custo de comunicação, como discutido na Seção 3.3.2.

3.4.3 Impacto do tamanho da entrada do k -means

Também avaliamos o tamanho da entrada do k -means, que é uma fração ou porcentagem ϕ do sinal medido s_i . Os resultados são apresentados na Figura 3.9. Mais uma vez apresentamos os resultados dos algoritmos de referência para comparação, apesar deste parâmetro não afetá-los.

Este é o parâmetro que tem maior impacto para o sistema, incluindo complexidade de tempo, custo de comunicação e taxas de classificação. Baixos valores levam a baixa complexidade de tempo e baixo custo de comunicação, mas também baixo

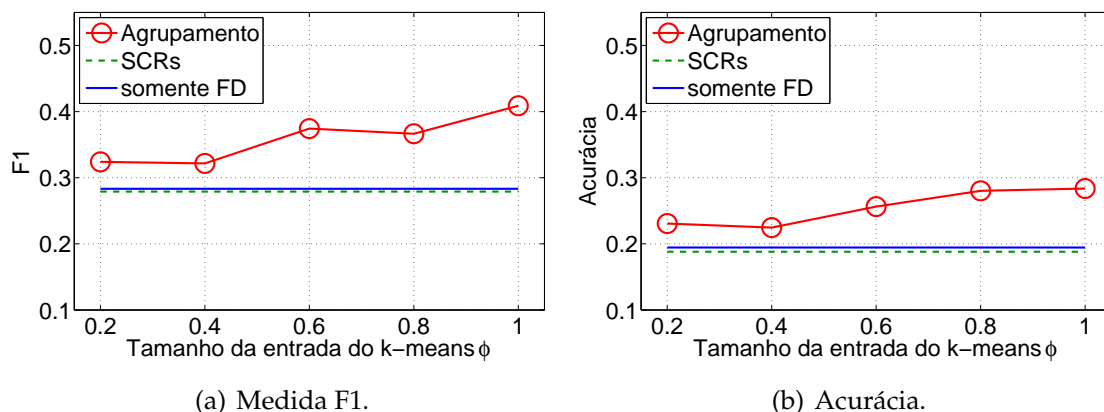


Figura 3.9. Impacto do tamanho da entrada do k -means ϕ .

desempenho de classificação. Por outro lado, valores altos produzem um classificador caro, mas eficiente. Assim, uma boa escolha é o valor de $\phi = 0,6$ que produz um classificador com 34 % de melhoria em relação ao esquema SCRs. Com valores maiores de ϕ , temos melhoras menos significativas na acurácia e F1, mas estes casos levam a maiores custos computacionais.

3.5 Considerações parciais

Neste capítulo, apresentamos um algoritmo para melhorar o desempenho de classificadores utilizando o esquema de fusão de dados. Consideramos um cenário ruidoso no qual um evento observado é a combinação de medidas correlacionadas e não correlacionadas. Nesses casos, a combinação de valores e a fusão das decisões é requerida. Nossa solução é baseada no algoritmo de agrupamento de dados conhecida como k -means, utilizando medidas correlacionadas. Aplicamos fusão de valores para nós do mesmo grupo e as decisões locais são combinadas por votação para os diferentes grupos.

As simulações mostram a eficiência de nossa solução para a aplicação de classificação de anuros baseada em medidas acústicas. Nosso algoritmo apresentou desempenho aproximadamente 30% superior ao do algoritmo que usa os SCRs.

Capítulo 4

Classificação de dois anuros usando RSSF

Neste capítulo exploramos o problema de classificar dois anuros em uma região de interesse coberta por uma RSSF. Este cenário é importante por conta da motivação deste trabalho onde se quer estimar a população de anuros em uma região. Em casos práticos, como o de monitoramento em ambiente florestal, é comum a presença de mais de um anuro vocalizando ao mesmo tempo. No Capítulo 3, a modelagem do problema previa que os anuros que vocalizavam com menor intensidade eram tratados como ruído e não precisavam ser classificados. Logo, estes anuros poderiam não ser considerados na contagem populacional. Portanto, neste capítulo consideramos que dois anuros estão vocalizando ao mesmo tempo e pretendemos classificá-los.

A Figura 4.1 ilustra o problema. Consideramos que os dois anuros podem vocalizar com intensidades diferentes. Para isso, definimos o anuro cujo som emitido é o mais intenso (mais alto) como *anuro principal*, e o outro anuro é chamado de *anuro secundário*.

Um algoritmo de classificação para este problema deve retornar duas decisões: decisão primária d_{pri} e secundária d_{sec} . Esta nomenclatura indica uma hierarquia de nível de certeza entre as decisões, isto é, um dado algoritmo tem maior crença que o anuro obtido em d_{pri} está presente na região de interesse em relação ao anuro d_{sec} . Desta forma, podemos associar a decisão primária com a classificação do anuro principal e a decisão secundária com a classificação do anuro secundário.

Iniciamos apresentando uma solução baseada apenas no dado coletado de um único nó sensor. Posteriormente, consideramos os dados coletados por vários nós sensores num esquema baseado em fusão de decisões. Por último, apresentamos outra solução semelhante à proposta no capítulo anterior, baseada em agrupamento

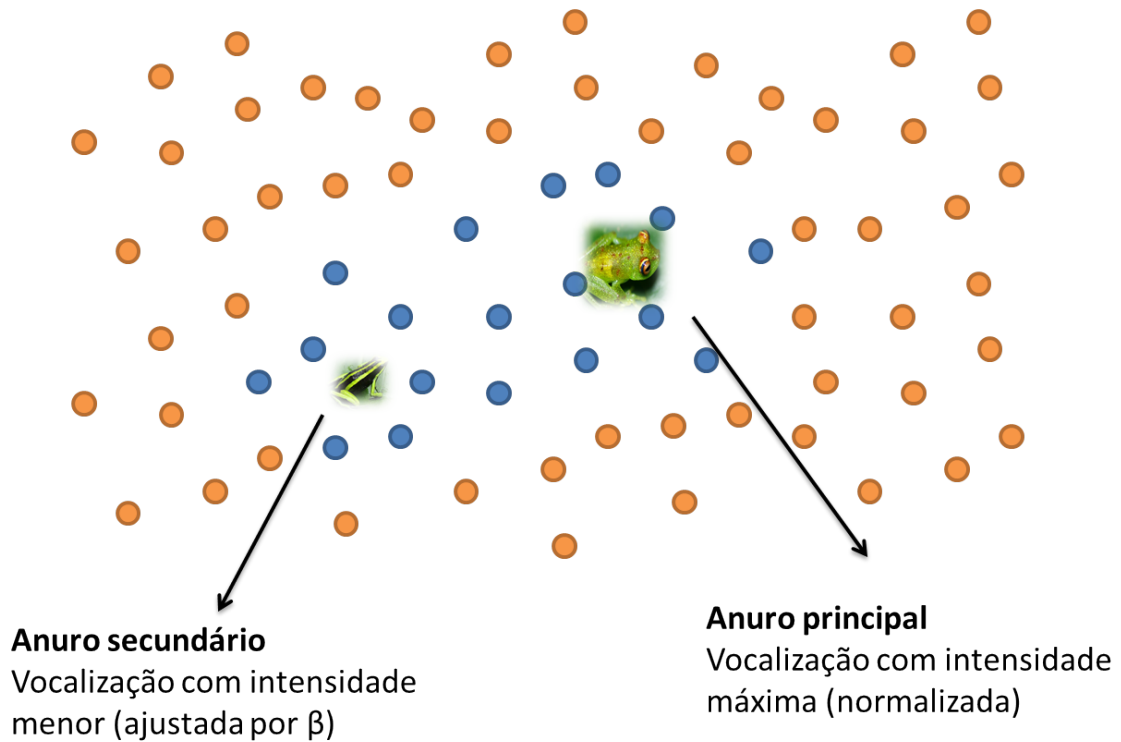


Figura 4.1. Consideramos que um nó vocaliza com maior intensidade que o outro.

seguido de fusão de valores e decisão. Nos experimentos realizados, a solução que utiliza agrupamento apresentou melhor desempenho principalmente em cenários com ruído elevado.

4.1 Modelagem do sinal

Para este problema, consideramos que o sinal captado pelo sensor i no *grid* é

$$\mathbf{s}_i = \mathbf{r}_i + \sum_{j=1}^2 \text{aten}(\mathbf{s}_j, d_{ij}), \quad (4.1)$$

onde \mathbf{r}_i é o ruído branco Gaussiano aditivo tal que $\mathbf{r}_i \sim \mathcal{N}(0, \sigma_{wn})$, \mathbf{s}_j são os anuros a serem classificados e d_{ij} é a distância do anuro j para o sensor i . A função $\text{aten}(\cdot)$ já foi definida na Equação 3.3.

O modelo utilizado para o problema anterior já incluía mais de um anuro na região de interesse. No entanto, os anuros adicionais eram considerados como ruído para o evento principal. Neste capítulo, tratamos os dois anuros presentes na região de interesse como importantes e que devem ser classificados. A seguir descrevemos

a metodologia para classificação.

4.2 Metodologia

Para classificar dois alvos baseados no sinal acústico emitido, seguimos a estratégia apresentada no capítulo anterior, onde um algoritmo de detecção nos entrega um conjunto de sinais S_a coletados pelos m nós ativos.

Todos os esquemas de classificação que serão apresentados a seguir são baseados na distância de Mahalanobis [Mahalanobis, 1936]. Na etapa de treino, utilizamos as amostras de treino normalizadas¹ para modelar cada classe como uma variável aleatória seguindo uma distribuição normal multivariada. Na etapa de inferência ou classificação, calculamos a distância de mahalanobis, $\text{dist}(\mathbf{x}, c)$, de uma amostra desconhecida \mathbf{x} para cada classe c , como segue:

$$\text{dist}(\mathbf{x}, c) = \sqrt{(\mathbf{x} - \mu_c)^T \mathbf{S}_c^{-1} (\mathbf{x} - \mu_c)}, \quad (4.2)$$

onde \mathbf{x} é o vetor de características extraído do sinal \mathbf{s} normalizado, medido por um nó sensor, μ_c e \mathbf{S}_c são, respectivamente, o vetor de médias e a matriz de covariâncias para classe c , calculados na etapa de treino.

Optamos por um classificador baseado em distância de Mahalanobis por prover, como solução parcial, um número real que tem relação com a qualidade da decisão. Neste caso, menores distâncias podem indicar alta probabilidade de acerto ou alta certeza na decisão. Esta informação local pode ser utilizada por algoritmos de fusão de dados em etapas de refinamento da decisão global.

Ainda utilizamos os MFCCs como características a serem extraídas do sinal, conforme apresentado na Seção 3.3. A seguir descreveremos as técnicas de classificação propostas para este problema.

4.2.1 Classificação usando um único sensor

No caso de se utilizar apenas um nó sensor na região de interesse, podemos seguir os passos descritos no Algoritmo 4.1 e ilustrados na Figura 4.2. Inicialmente, calculamos a distância de Mahalanobis do sinal coletado para cada classe. O algoritmo retorna como decisão primária a classe mais próxima da amostra coletada e, como decisão

¹Neste contexto, “normalizar” significa dividir a amplitude das amostragens pela potência de cada sinal. Assim, a potência do sinal normalizado será igual a 1.

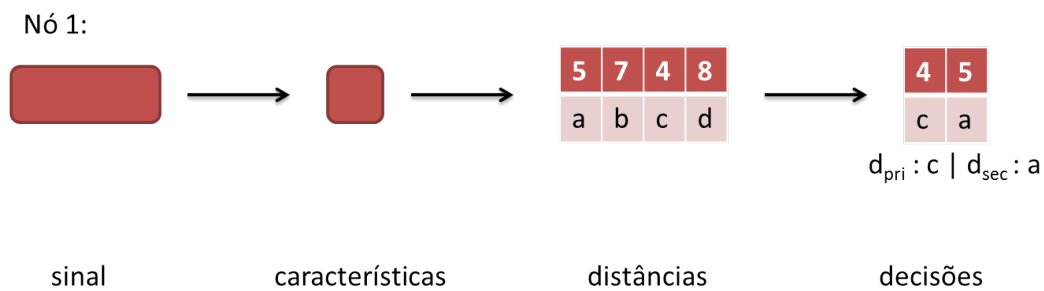


Figura 4.2. Classificação usando um único sensor. As características são obtidas do sinal coletado e se calcula a distância de Mahalanobis para cada classe. Neste exemplo, as classes c e a são as classes mais próximas da classe da amostra coletada e, por isso, foram escolhidas como decisão primária e secundária, respectivamente. Provavelmente, a amostra coletada não pertence as classes b e d , já que apresentaram maior distância.

secundária, a segunda classe mais próxima. Definimos a função $\arg \min_2(X)$ (linha 4) como o argumento do segundo menor elemento do conjunto X .

Algoritmo 4.1 Classificação de dois alvos usando um sensor.

Entrada: sinal coletado pelo sensor s .

Saída: as decisões primária e secundária.

- 1: $\mathbf{x} \leftarrow \text{EXTRACTFEATURES}(s)$
 - 2: $\text{distSet} \leftarrow \{\text{dist}(\mathbf{x}, c) \mid c \in C\}$
 - 3: $d_{\text{pri}} \leftarrow \arg \min_c \{\text{distSet}\}$
 - 4: $d_{\text{sec}} \leftarrow \arg \min_2_c \{\text{distSet}\}$
 - 5: **retorne** $(d_{\text{pri}}, d_{\text{sec}})$
-

Esta técnica será utilizada como referência para os experimentos.

4.2.2 Classificação baseada em fusão de dados

Para o caso da classificação usando uma RSSF, podemos trabalhar com fusão de dados para melhorar as estimativas. Desta forma, propomos o Algoritmo 4.2 baseado em fusão de decisões.

Inicialmente, cada nó ativo obtém uma única decisão local, como se estivesse classificando apenas um anuro (linhas 1-4). Essas decisões são enviadas ao nó mestre que aplica a função $\text{TWO MAJORITY VOTE}(\cdot)$ que retorna as duas classes mais votadas. A Figura 4.3 exemplifica o algoritmo proposto.

Algoritmo 4.2 Classificação de dois alvos usando fusão de decisões.

Entrada: conjunto Sa de sinais coletados pelos nós ativos.

Saída: as decisões primária e secundária.

```

1: decSet  $\leftarrow \emptyset$ 
2: para todo  $s \in Sa$  faça
3:    $\mathbf{x} \leftarrow \text{EXTRACTFEATURES}(s)$ 
4:   distSet  $\leftarrow \{\text{dist}(\mathbf{x}, c) \mid c \in C\}$ 
5:   decSet  $\leftarrow \text{decSet} \cup \{\arg \min_c(\text{distSet})\}$ 
6: fim para
7:  $(d_{\text{pri}}, d_{\text{sec}}) \leftarrow \text{TWO MAJORITY VOTE}(\text{decSet})$ 
8: retorne  $(d_{\text{pri}}, d_{\text{sec}})$ 

```

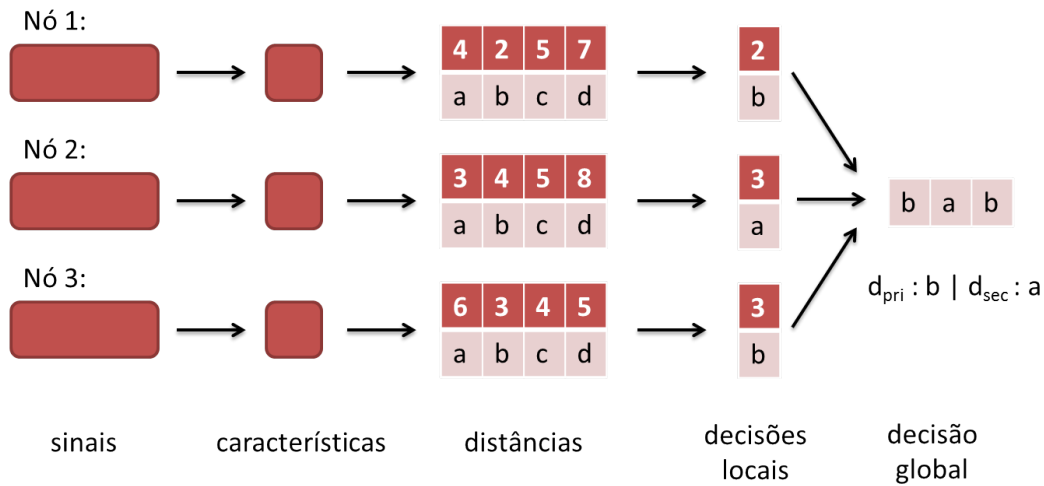


Figura 4.3. Classificação baseada em fusão de dados. Neste caso, temos três nós participando da tarefa de classificação, onde cada um obtém a sua decisão local. As decisões finais serão as duas classes mais votadas.

4.2.3 Classificação usando agrupamento e fusão de dados

Também propomos um esquema de classificação muito similar ao proposto no capítulo anterior, baseado em agrupamento de medidas por similaridade, fusão de valores, seguido de fusão de decisões, descrito no Algoritmo 4.3. As funções `PARTITIONREGION` e `VALUEFUSION` foram descritas nos Algoritmos 3.3 e 3.2, respectivamente.

Inicialmente, cada nó ativo envia uma fração do seu sinal medido para o nó mestre para obter, por meio do k -means, o conjunto G composto pelos K grupos de nós com medidas correlacionadas (linha 1). Em seguida, cada grupo g combina os sinais coletados pelos nós do grupo para obter uma única decisão local d_g . As decisões locais são enviadas ao nó mestre, que retorna as duas classes mais votadas (função `TWO MAJORITY VOTE`). Um exemplo é apresentado na Figura 4.4.

Algoritmo 4.3 Classificação usando agrupamento e fusão de dados

Entrada: conjunto Sa de sinais coletados pelos nós ativos, um percentual ϕ e o número de grupos K .

Saída: as decisões primária e secundária.

- 1: $G \leftarrow \text{PARTITIONREGION}(Sa, \phi, K)$
- 2: **para todo** $g \in G$ **faça**
- 3: $d_g \leftarrow \text{VALUEFUSION}(S_g)$, onde $S_g = \{s_i \mid \text{nó } i \in g\}$
- 4: **fim para**
- 5: $(d_{\text{pri}}, d_{\text{sec}}) \leftarrow \text{TWO MAJORITY VOTE}(\{d_g \mid g \in G\})$
- 6: **retorne** $(d_{\text{pri}}, d_{\text{sec}})$

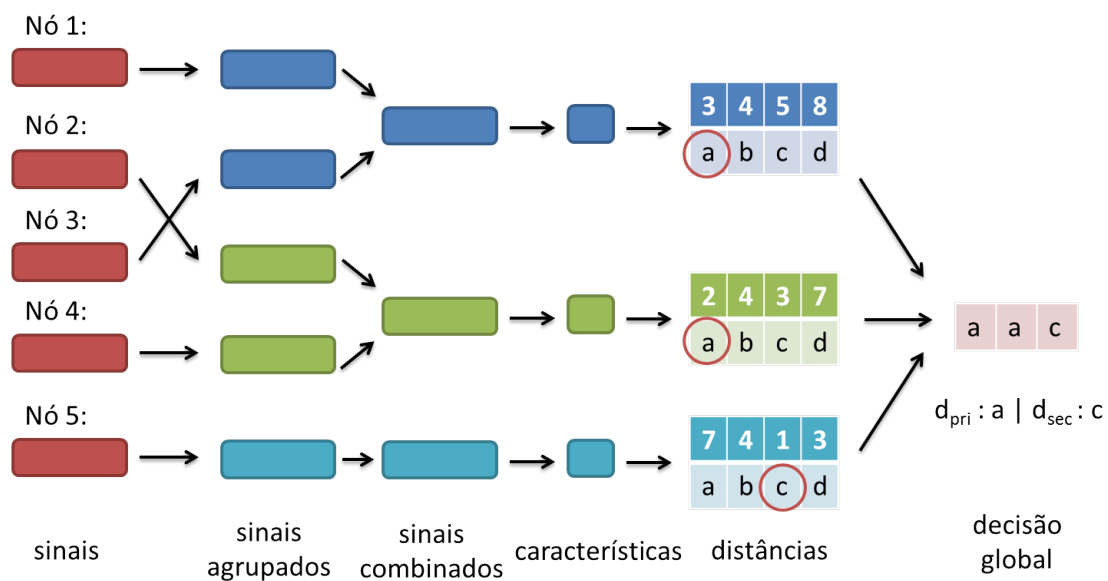


Figura 4.4. Exemplo de classificação usando agrupamento. Os nós 1 e 3 coletaram medidas semelhantes e por isso formam um grupo. Um segundo grupo é formado pelos nós 2 e 4, e o último grupo contém apenas o nó 5. Os sinais coletados pelos nós de um grupo são combinados e cada grupo obtém a sua decisão local. A saída continua sendo as duas classes mais votadas.

Com este algoritmo, teremos grupos de nós formados próximos aos anuros que estão vocalizando (fontes do sinal). Isto é interessante, pois estes grupos coletam sinais cujas componentes mais fortes se referem ao sinal emitido pelo anuro mais próximo, permitindo que esta espécie seja mais facilmente reconhecida. Além disso, o ruído ainda pode ser reduzido dentro de cada grupo por meio da aplicação da média dos sinais medidos.

Portanto, cada grupo fica responsável apenas por classificar a espécie que está vocalizando mais próxima ao grupo. Os nós de grupos posicionados na área de interseção entre as duas fontes tendem a receber uma mistura de sinais cujas compo-

mentos referentes às fontes possuem intensidades muito similares, e, eventualmente, produzir classificações erradas. Na etapa de votação, essas decisões erradas recebem poucos votos e acabam sendo desprezadas.

4.2.4 Complexidade

Nesta seção, fazemos uma análise do custo computacional, de comunicação e de memória de cada algoritmo proposto. No final, apresentaremos um quadro comparativo.

As complexidades computacional e de comunicação são obtidas de modo global, considerando todos os nós ativos. Elas são úteis quando se quer analisar, por exemplo, o tempo de vida da rede como um todo, e não apenas de nós individuais.

Já o custo de memória representa a quantidade de escalares a serem armazenados por um nó sensor. Esse custo é analisado apenas para o nó mestre que, além de realizar o processamento sobre o seu sinal coletado, ainda precisa realizar as fusões sobre os dados dos outros sensores. Desta forma, este é o nó que mais precisará de memória na rede. Assim, apresentamos apenas o pior caso do custo de memória. Uma vez que qualquer nó sensor da rede pode assumir a função de nó mestre, todos os nós precisam ter, pelo menos, esta quantidade de memória disponível.

Para o algoritmo mais básico, que utiliza apenas um sensor (Algoritmo 4.1), o custo de extrair as características é dada pela computação dos coeficientes da FFT, conforme já explicamos na seção 3.3.2. Isto vale $O(N_s \log N_s)$, onde N_s representa a dimensão do sinal de áudio. O cálculo das distâncias de Mahalanobis é obtido em $O(N_x^2)$, devido as multiplicações envolvendo o vetor de características x , extraído do sinal s , e a inversa da matriz de covariâncias² S , realizadas para cada classe, resultando em $O(|C|N_x^2)$, onde $|C|$ denota o número de classes conhecidas. No total, o custo computacional do Algoritmo 4.1 é $O(N_s \log N_s) + O(|C|N_x^2)$.

Como este algoritmo é executado em apenas um nó sensor e este já obtém uma decisão global sem precisar trocar pacotes com nós vizinhos, o custo de comunicação é nulo. Em termos de memória, este nó sensor precisa armazenar as informações obtidas para cada classe na etapa de treino: o vetor de médias μ_c e a matriz de covariâncias S_c (ou sua inversa S_c^{-1}), resultando em $|C|(N_x + N_x^2)$. Também precisa armazenar o vetor referente ao sinal medido de tamanho N_s , o vetor de características de tamanho N_x e as distâncias de Mahalanobis computadas para cada classe. No total, o custo de memória é de $|C|(N_x + N_x^2) + N_s + N_x + |C|$ escalares.

²Em termos de implementação, podemos armazenar apenas a inversa de S para cada classe, economizando processamento e energia em cada nó.

No Algoritmo 4.2, baseado em fusão de decisões, cada um dos m nó ativos obtém uma decisão local que é computada da mesma forma que no Algoritmo 4.1, nas linhas 1 a 3. Logo, o custo computacional é $O(mN_s \log N_s) + O(m|C|N_x^2)$. O custo da linha 7 é da ordem de $O(m)$ e pode ser desprezado.

Em termos de comunicação, cada nó ativo só precisa enviar as suas decisões locais para o nó mestre (que também é um nó ativo). Logo, temos apenas $(m - 1)$ escalares enviados pelo rádio. Em relação a memória, além dos escalares computados para o Algoritmo 4.1, referentes à obtenção da decisão local, o nó mestre ainda precisa armazenar as decisões de todos os m nós ativos, resultando num custo de memória total de $|C|(N_x + N_x^2) + N_s + N_x + |C| + m$ escalares.

Para o Algoritmo 4.3, os custos computacional e de comunicação são os mesmos já calculados na Seção 3.3.2. Quanto a memória, o nó sensor precisa armazenar:

1. as informações referentes ao treino: $|C|(N_x + N_x^2)$ escalares;
2. a parcela do sinal coletado dos m nós ativos para ser utilizado como entrada para o k -means: $\phi m N_s$ escalares;
3. Uma tabela com a distribuição dos nós entre os grupos: $2m$ escalares;
4. o sinal coletado pelos nós de um grupo (considerando que ele também é o *cluster head*) para aplicar fusão de valores: $m N_s / K$ escalares;
5. o vetor de características extraído do sinal gerado a partir da fusão dos sinais dos nós do grupo: N_x ;
6. as distâncias de Mahalanobis para cada classe: $|C|$;
7. uma decisão de cada grupo: K .

No item 4, estamos considerando que, em média, os m nós ativos se dividem igualmente entre os K grupos. No total, o custo de memória para o Algoritmo 4.3 é $|C|(N_x + N_x^2) + \phi m N_s + 2m + m N_s / K + N_x + |C| + K$. Em termos de implementação, algumas otimizações podem ser realizadas para diminuir este custo. Além disso, recomendamos que o cluster head não seja o mesmo nó mestre, para balancear os custos entre esses dois nós.

A Tabela 4.1 apresenta um quadro comparativo. Em todos os itens analisados, o algoritmo que utiliza agrupamento seguido de fusão de dados é o que apresenta maior complexidade. No entanto, conforme será descrito na Seção 4.3, este é o algoritmo que produz melhores resultados conforme as métricas adotadas.

Tabela 4.1. Comparação entre os algoritmos propostos.

Complexidade	Um sensor	Fusão de decisões	Agrupamento e FD
Computacional	$O(N_s \log N_s) + O(C N_x^2)$	$O(mN_s \log N_s) + O(m C N_x^2)$	$O(mN_sKT) + O(KN_s \log N_s)$
Comunicação [escalar. transm.]	0	$(m - 1)$	$N_s(\phi m - \phi + m - K) + m - 1 + K$
Memória [escalar. armazen.]	$ C (N_x + N_x^2) + N_s + N_x + C $	$ C (N_x + N_x^2) + N_s + N_x + C + m$	$ C (N_x + N_x^2) + \phi m N_s + 2m + mN_s/K + N_x + C + K$

Tabela 4.2. Distribuição de amostras entre as classes para classificação de dois anuros.

Classe	Nome da Espécie	# sílabas	%
1	<i>Adenomera andre</i>	297	16,03
2	<i>Ameerega trivittata</i>	160	8,63
3	<i>Leptodactylus hylaedactylus</i>	37	2,00
4	<i>Hyla minuta</i>	217	11,71
5	<i>Hypsiboas cinerascens</i>	222	11,98
6	<i>Osteocephalus oophagus</i>	418	22,56
7	<i>Rhinella granulosa</i>	53	2,86
8	<i>Scinax ruber</i>	449	24,23
Total	—	1953	100,0

4.3 Avaliação e resultados

Nesta seção avaliamos os algoritmos propostos. As amostras utilizadas nos experimentos seguem distribuição de espécies apresentadas na Tabela 4.2. Optamos por retirar as amostras (sílabas) que apresentavam potência normalizada abaixo de 0,1. Neste procedimento, percebemos que muitas amostras pertenciam a espécie *Leptodactylus fuscus* (classe 6), resultando em apenas 13 amostras. Por isso, decidimos retirar todas as amostras desta classe do conjunto de dados. Posteriormente, percebemos que apenas esta classe reduzia o desempenho de todos os algoritmos avaliados em aproximadamente 20 pp (pontos percentuais).

Seguimos utilizando a mesma metodologia de validação cruzada estratificada de 10-*folds*. Em cada rodada, posicionamos os nós e os dois anuros aleatoriamente no *grid*. Normalizamos o sinal emitido pelo anuro principal para simular uma vocalização com intensidade máxima. No caso do anuro secundário, limitamos a intensidade por um fator β sobre a intensidade máxima. Isto simula o caso de termos dois anuros vocalizando com intensidades diferentes.

Tabela 4.3. Parâmetros de configuração padrões para classificação de dois anuros.

Parâmetro	Símbolo	Valor
Número de nós em uma dimensão do <i>grid</i>	N	10
Distância entre nós no <i>grid</i>	l	10 m
Desvio padrão de posicionamento dos nós	σ	1 m
Tamanho do vetor do sinal acústico	N_s	1024
Tamanho do vetor de características sinal	N_x	10
Fator de atenuação acústica	α	0.1053 dB/m
Número de nós ativos	m	25
Tamanho da entrada do <i>k</i> -means (percentual do sinal)	ϕ	1
Desvio padrão do ruído branco	σ_{wn}	0.30
Fator de intensidade do sinal do anuro secundário	β	0.75
Número de grupos	K	5
Número máximo de interações do <i>k</i> -means	T_{max}	30

Treinamos um novo classificador para cada *holdout*. Para cada amostra em um *holdout*, executamos 30 rodadas de teste variando aleatoriamente o anuro secundário e as posições no *grid* de todos os elementos. Os parâmetros padrões utilizados nos experimentos são apresentados na Tabela 4.3.

Deste ponto em diante, chamaremos o Algoritmo 4.1 de *Naïve*, o Algoritmo 4.2 de *Somente FD* e o algoritmo que usa agrupamento e fusão de dados de *Agrupamento*.

4.3.1 Métricas de avaliação

Dada a característica do problema, precisamos definir uma nova métrica para avaliar os resultados, visto que as métricas acurácia e F1 não consideram a classificação de dois ou mais alvos. Como temos dois anuros na região de interesse, e um algoritmo para este problema precisa retornar duas saídas, podemos considerar o gráfico como o apresentado na Figura 4.5.

A Figura 4.5(a) indica resposta do algoritmo na tentativa de classificar o anuro principal. A região mais abaixo representa a taxa com que o anuro principal foi reconhecido na decisão primária d_{pri} , e a região central representa a taxa com que o anuro principal foi reconhecido na decisão secundária d_{sec} . A região mais acima indica a taxa de erro de um algoritmo em classificar o anuro principal.

Da mesma forma, a Figura 4.5(b) representa como um algoritmo reconheceu o anuro secundário. A região acima representa o erro em classificar o anuro secundário. As duas regiões mais abaixo indicam que o anuro secundário foi reconhecido na decisão secundária (região central) ou na decisão primária (região mais abaixo).

Em ambos os gráficos da Figura 4.5, o eixo das abscissas representa a intensidade das vocalizações do sinal do anuro secundário. Conforme já explicamos, os

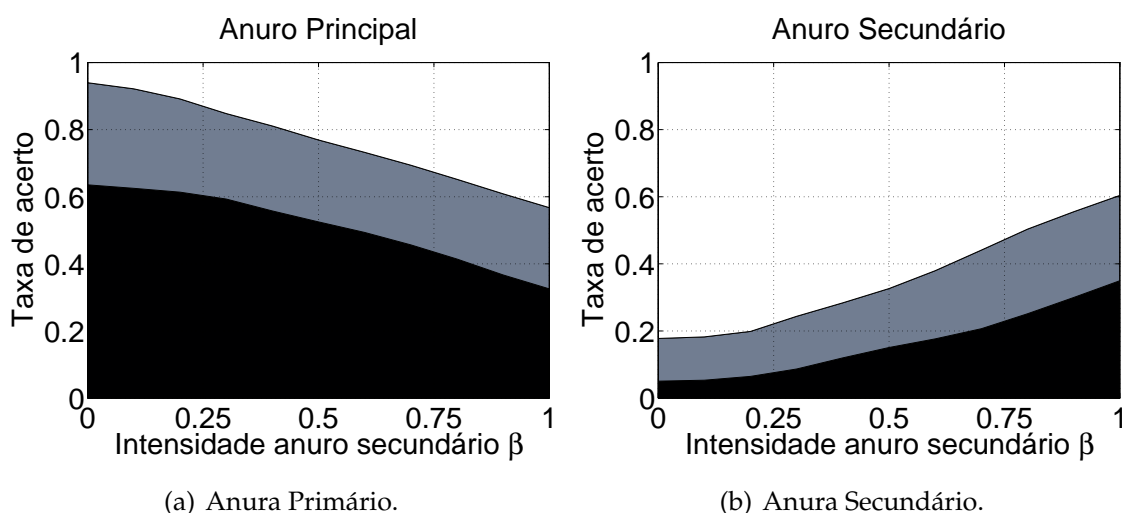


Figura 4.5. Representação detalhada do desempenho de um algoritmo qualquer para classificar dois anuros.

anuros vocalizam em intensidades diferentes. Variamos esta intensidade (parâmetro β) de zero a um, ou seja, quando temos apenas o anuro principal vocalizando seguindo até quando o dois anuros vocalizam com mesma intensidade.

Esta forma de representar os resultados apresenta algumas características gerais, tais como:

1. É esperado que a taxa de acerto para classificar o anuro principal comece com um valor alto quando apenas um anuro está presente no *grid* ($\beta = 0$) e diminua à medida que o anuro secundário vocaliza com mais intensidade;
2. Para o anuro secundário, é esperado que a taxa de acerto comece com um valor baixo, próximo a $\frac{1}{|C|}$, onde o classificador praticamente escolhe aleatoriamente qualquer uma das classes, e vá aumentando a medida que suas vocalizações se tornam mais altas;
3. Quando $\beta = 1$, a taxa de acerto do anuro principal e secundário devem ser iguais. Esta característica é mais facilmente observada quando se analisa o erro, que para este caso, também é igual;
4. A taxa de acerto da decisão primária é sempre maior do que a taxa da decisão secundária. Na realidade, esta é mais uma característica do tipo de classificador utilizado, onde a decisão primária é sempre a melhor escolha (a classe com menor distância, maior probabilidade ou a mais votada) e a decisão secundária

é a segunda melhor escolha. Por isso, temos uma maior probabilidade de errar a decisão secundária ao invés da decisão primária;

5. Bons algoritmos conseguem reduzir a área referente ao erro. Esta característica é importante, pois nos permite comparar dois ou mais algoritmos utilizando apenas o erro, ou seja, o melhor algoritmo é aquele que tiver as menores áreas referentes ao erro.

Baseado nessas características, propomos uma métrica que reúna as informações tanto do anuro principal quanto do secundário em uma única medida. Também temos que levar em conta o parâmetro β , já que o custo de se errar a classificação do anuro secundário deve aumentar à medida que ele vocaliza mais intensamente e deve ser desprezada quando $\beta = 0$.

Desta forma, definimos a taxa de erro Err de um algoritmo como:

$$Err = \frac{ErrP + \beta \cdot ErrS}{1 + \beta}, \quad (4.3)$$

onde $ErrP$ é a taxa de erro ao classificar o anuro principal, representada pela região superior da Figura 4.5(a), e $ErrS$ é a taxa de erro ao classificar o anuro secundário, região superior da Figura 4.5(b). Quando $\beta = 0$, a taxa de erro do algoritmo é igual à taxa de erro para o anuro principal, desprezando as tentativas de classificar o anuro secundário (que de fato não está presente na área de interesse). Quando $\beta = 1$, ambos os erros são levados em conta com o mesmo peso.

Mais uma vez, optamos por não mostrar nos gráficos o desvio padrão ou intervalo de confiança no formato de barras de erro. No entanto, executamos o teste-t pareado, com nível de confiança de 95%, sob os resultados obtidos para cada *holdout*, sempre retornando a comparação entre dois algoritmos.

4.3.2 Sensibilidade ao ruído

Primeiro analisamos o impacto do ruído sob os algoritmos propostos. A Figura 4.6 apresenta as variações da taxa de erro em relação ao β , calculada pela Equação 4.3, quando o ruído branco gaussiano aditivo é baixo ($\sigma_{wn} = 0$ e $\sigma_{wn} = 0,15$).

Para $\sigma_{wn} = 0$, (Figura 4.6(a)), podemos observar que todos os algoritmos propostos apresentam desempenho parecido. O teste-t, (Tabela 4.4), confirma indicando que o algoritmo *Agrupamento* teve desempenho inferior em relação ao *Naïve* (de $-0,95$ pp) e ao *Somente FD* (de $-1,0$ pp). Para todos os algoritmos, a taxa de erro foi

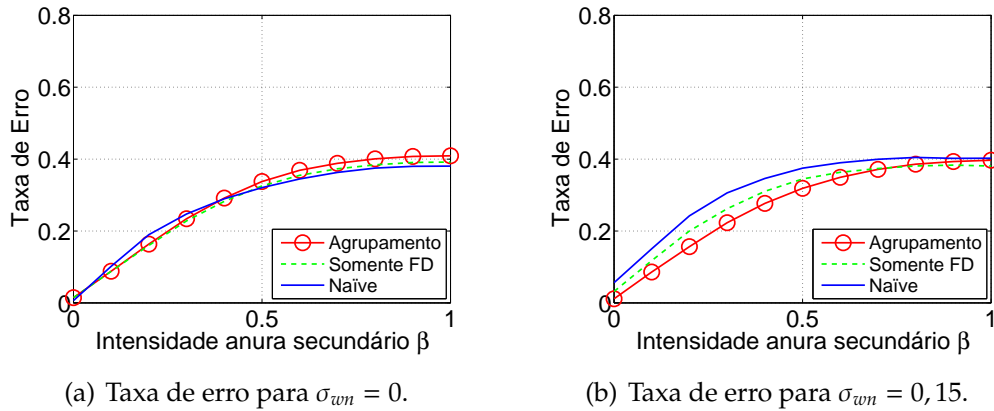


Figura 4.6. Taxa de erro quando ruído é baixo.

Tabela 4.4. Resultado do teste-t pareado para nível de ruído baixo.

	$\sigma_{wn} = 0$	$\sigma_{wn} = 0,15$
<i>Naive</i> [pp]	$-0,95 \pm 1,0$	$4,6 \pm 1,2$
<i>Somente FD</i> [pp]	$-1,0 \pm 0,3$	$1,6 \pm 0,9$

bem próxima de zero quando $\beta = 0$ e foi aumentando até o ponto de convergir um pouco abaixo de 0,40.

Quando o desvio padrão do ruído vale 0,15, (Figura 4.6(b)), podemos observar que o algoritmo *Agrupamento* já apresenta um erro menor que os demais, por volta de 4,6 pp em relação ao *Naive* e de 1,6 pp menor em relação ao *Somente FD*. O desempenho dos algoritmos ainda converge para 40 %, no entanto, a taxa de crescimento é levemente superior, principalmente para o algoritmo *Naive*.

A Figura 4.7 mostra o desempenho detalhado dos algoritmos quando $\sigma_{wn} = 0,15$. As regiões do gráfico apresentam semelhanças, no entanto, percebemos que para o algoritmo *Agrupamento*, a região inferior da Figura 4.7(c) inicia com valor alto, e permanece alto mesmo para valores de β maiores.

Apresentamos na Figura 4.8 a taxa de erro quando temos um ruído médio ($\sigma_{wn} = 0,30$ e $\sigma_{wn} = 0,45$). Podemos notar claramente que os algoritmos *Naive* e *Somente FD* são fortemente afetados pelo ruído. Para $\sigma_{wn} = 0,30$, o algoritmo *Agrupamento* conseguiu manter praticamente os mesmos valores de taxa de erro em relação aos obtidos para $\sigma_{wn} = 0$ e $\sigma_{wn} = 0,15$, convergindo para um erro máximo de 0,4. Para $\sigma_{wn} = 0,45$, os algoritmos *Naive* e *Somente FD* ultrapassaram 50 % de erro, enquanto que o algoritmo *Agrupamento* se manteve por volta de 40 %.

Podemos perceber que à medida que o ruído aumenta, os gráficos dos algoritmos *Naive* e *Somente FD* se distanciam da reta do algoritmo *Agrupamento*. Essa diferença também é refletida no teste-t vista na Tabela 4.5. Para o algoritmo *Naive*,

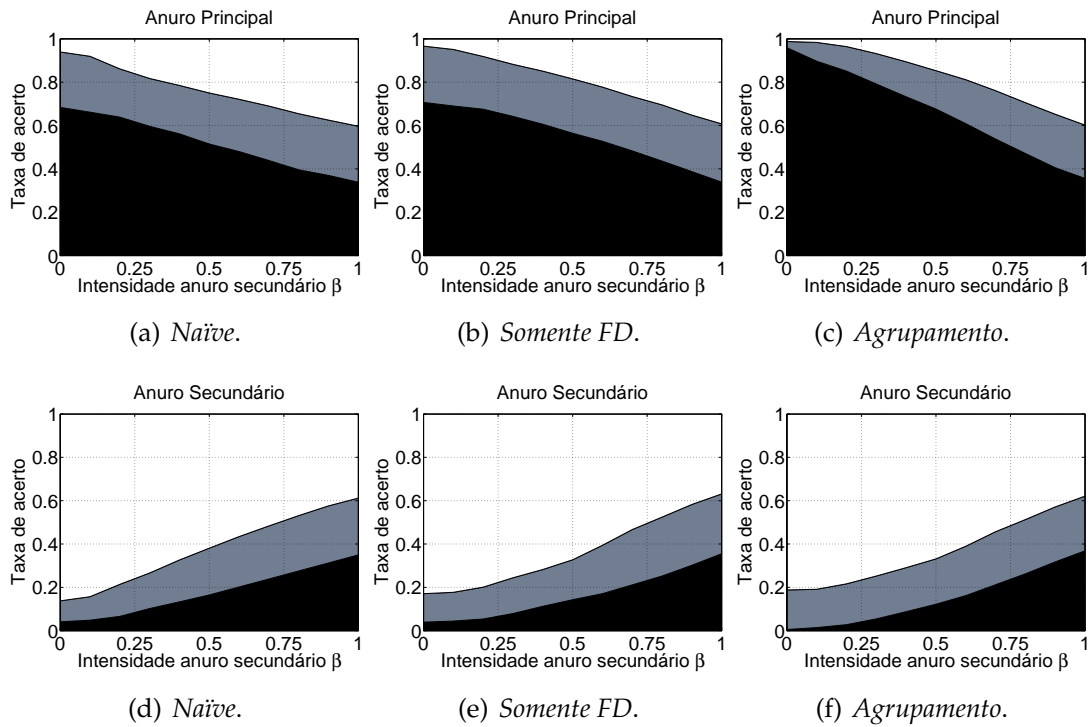


Figura 4.7. Representação detalhada do desempenho dos algoritmos quando $\sigma_{wn} = 0,15$.

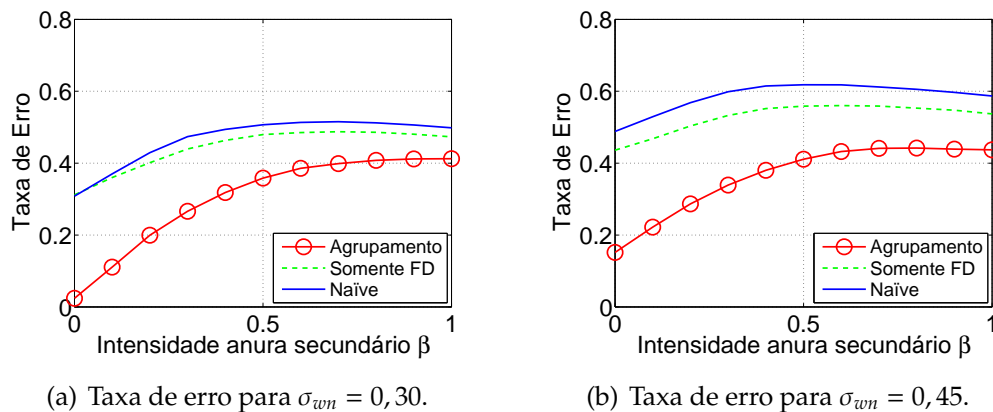


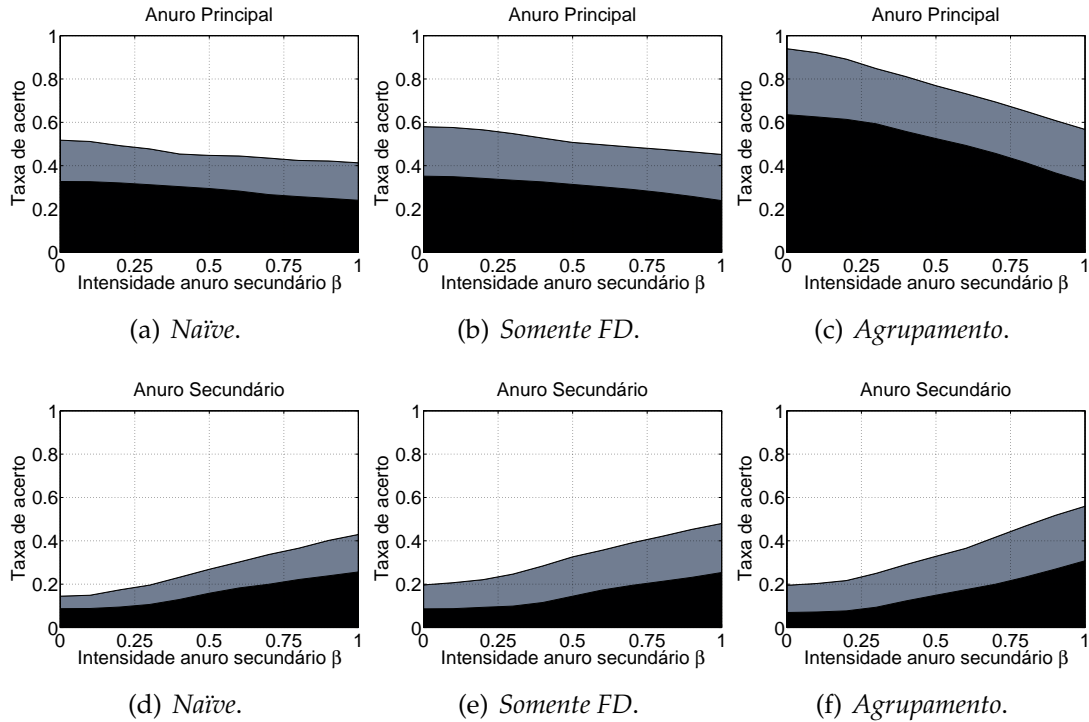
Figura 4.8. Taxa de erro para nível de ruído médio.

o aumento na diferença em relação ao algoritmo *Agrupamento* quando passamos de $\sigma_{wn} = 0,30$ para $\sigma_{wn} = 0,45$ foi de aproximadamente 6 pp. Para o algoritmo *Somente FD*, esse aumento foi de 2 pp sem considerar a margem para cima e para baixo.

Na Figura 4.9 apresentamos o desempenho detalhado dos algoritmos quando $\sigma_{wn} = 0,45$. A região de erro para classificação do anuro principal usando os algoritmos *Naive* e *Somente FD* aumentou significativamente se comparadas com os

Tabela 4.5. Resultado do teste-t pareado para nível de ruído médio.

	$\sigma_{wn} = 0,30$	$\sigma_{wn} = 0,45$
<i>Naïve</i> [pp]	$16,6 \pm 3,1$	$22,3 \pm 3,2$
<i>Somente FD</i> [pp]	$14,3 \pm 2,9$	$16,6 \pm 2,6$

**Figura 4.9.** Representação detalhada do desempenho dos algoritmos quando $\sigma_{wn} = 0,45$.**Tabela 4.6.** Resultado do teste-t pareado para nível de ruído elevado.

	$\sigma_{wn} = 0,60$
<i>Naïve</i> [pp]	$23,1 \pm 2,7$
<i>Somente FD</i> [pp]	$15,6 \pm 2,0$

gráficos apresentados na Figura 4.7.

A Figura 4.10 apresenta o gráfico da variação da taxa de erro quando temos um alto nível de ruído ($\sigma_{wn} = 0,60$). De modo geral, o desempenho de todos os algoritmos diminuiu. No entanto, A taxa de erro de classificação do algoritmo *Agrupamento* não ultrapassou 50 % mesmo em condições de elevado ruído e mistura do sinal. Também observamos, mais uma vez, uma grande diferença entre o desempenho dos algoritmos. O algoritmo *Agrupamento* apresentou superioridade de 23 pp (Tabela 4.6) em relação ao *Naïve* e de 15 pp em relação ao *Somente FD*.

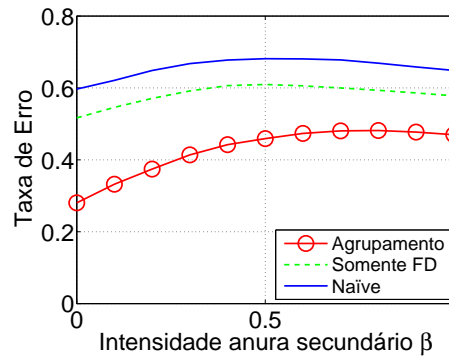


Figura 4.10. Taxa de erro para nível de ruído elevado ($\sigma_{wn} = 0,60$).

4.4 Considerações parciais

Neste Capítulo abordamos o problema de classificar dois anuros em uma região de interesse monitorada por uma RSSF. Neste problema, tratamos todos os anuros presentes na região como alvos importantes e que devem ser classificados. Para isso, propomos três algoritmos: (1) *Naïve*: baseado nas medidas de apenas um nó sensor, necessário quando se tem redes esparsas; (2) *Somente FD*: utilizando vários nós sensores da RSSF em um esquema de fusão de decisões; e (3) *Agrupamento*: baseado em agrupamento por similaridade seguido de fusão de valores e de decisões.

Os três algoritmos apresentam diferentes níveis de complexidade computacional, de comunicação e de utilização de memória. O algoritmo *Naïve* é o mais simples em todos os aspectos; no entanto, é o que apresenta pior desempenho em termos de taxa de classificação. O algoritmo *Somente FD* é uma adaptação do *Naïve* para vários nós, e por isso, carrega parte de sua simplicidade com melhor desempenho. O algoritmo *Agrupamento* pode ter melhor desempenho, no entanto exige maiores recursos do *hardware*.

Nos experimentos avaliamos a taxa de erro de classificação para os algoritmos propostos. Variamos os cenários para considerar ambientes com ruído em diferentes níveis e anuros vocalizando com intensidades diferentes. Na situação ideal, sem ruído, todos os algoritmos apresentaram desempenho muito similar. No entanto, à medida que aumentamos o ruído no ambiente, percebemos um aumento maior na taxa de erro para os algoritmos *Somente FD* e *Naïve* do que para o algoritmo *Agrupamento*. Para níveis de ruído médio e elevado, a diferença de desempenho entre o algoritmo *Agrupamento* e os demais ultrapassa 16 pp (pontos percentuais), chegando até 26 pp.

Capítulo 5

Conclusões

5.1 Considerações finais

Neste trabalho apresentamos soluções distribuídas para o problema de classificação de anuros usando RSSF e medidas acústicas.

Levando em conta o fato de que, num ambiente natural, é comum a presença vários anuros vocalizando e outros sons sendo emitidos ao mesmo tempo, consideramos dois cenários-problema. No primeiro, o objetivo é classificar apenas o anuro com vocalização mais predominante. No segundo cenário-problema, consideramos que dois anuros vocalizam na região de interesse e os dois devem ser classificados.

Nossa solução principal é adequada para ambos os problemas. Ela é baseada em um esquema da literatura que particiona a região de interesse em grupos de nós com medidas correlacionadas. Os sinais coletados pelos nós de um grupo são combinados por meio de fusão de valores para se obter decisões locais para cada grupo. Essas decisões são então combinadas para se obter uma decisão global.

Alguns algoritmos da literatura consideram que os sinais emitidos (fontes) podem ser modelados por variáveis aleatórias Gaussianas no tempo e no espaço. Por esse motivo, esses algoritmos particionam a região de interesse em sub regiões de coerência espacial (SCR), que são regiões no espaço uniformemente espaçadas. Os nós sensores dentro de um SCR formam um grupo de nós com medidas correlacionadas. No entanto, em casos reais, a presença de outros sons e ruídos no ambiente fazem com que os dados não sigam uma gaussiana.

Desta forma, propomos utilizar o algoritmo k -means sobre os dados coletados pelos sensores para particionar a região em grupos. Assim, nós com medidas correlacionadas ficaram no mesmo grupo. Os experimentos mostraram que esta abordagem consegue capturar as irregularidades nos dados, em contraste com os

algoritmos da literatura. Por isso, esta solução apresentou desempenho superior a outros algoritmos em pelo menos 10 pp (pontos percentuais) para o primeiro problema e 16 pp para o segundo problema com nível de ruído médio ou elevado.

As principais características observadas para este método são:

- A lógica do algoritmo se concentra na técnica de particionamento e de fusão dados, e não no classificador utilizado. Por isso, outras características e classificadores podem ser utilizados sem alterações no esquema proposto;
- Apresenta maior imunidade ao ruído se comparado com algoritmos da literatura evidenciado pelos experimentos realizados;
- Autoadaptação a variações no cenário. Os grupos de nós são formados sob demanda, a cada evento pre detectado (necessitando ainda ser classificado). Portanto, caso haja alguma mudança no cenário, como o deslocamento de um anuro ou vocalização de outros animais, por exemplo, a rede pode se auto-reconfigurar antes da classificação.

5.2 Limitações

Apesar do algoritmo apresentar resultados melhores que os algoritmos da literatura, o custo computacional (de processamento e memória) e de comunicação é maior que os dos algoritmos avaliados, basicamente por conta do algoritmo *k*-means. No entanto, podemos também utilizar algumas implementações distribuídas do *k*-means, como as apresentadas na Seção 2.3 para compartilhar o custo deste processamento (e de consumo de energia) entre os nós da rede. Isto não iria reduzir a complexidade do algoritmo proposto, mas poderia prolongar o tempo de vida de rede.

Atualmente, o processo de extração de sílabas obtém um vetor de amplitudes de tamanhos fixos e iguais para todas as espécies. Por isso, pode ocorrer de nem sempre o vetor coletado corresponder de fato a uma sílaba real, sendo apenas parte dela. Outro problema relacionado é que o algoritmo extrator de sílabas considera que a amplitude máxima dentro de uma sílaba (pico do sinal) ocorre na região central da sílaba. No entanto, algumas espécies apresentam no seu vocabulário sílabas cujo pico se encontra no início da vocalização.

Do ponto de vista da aplicação que motiva este trabalho, ainda precisamos de um módulo que distingue entre sons de anuros e de outros animais. Este trabalho está limitado ao escopo de considerar que todo som apresentado ao classificador

é de um ou mais anuros, e o objetivo é apenas indicar quais são as espécies que vocalizam.

5.3 Trabalhos futuros

Este trabalho pode ser continuado e estendido de diferentes formas. Primeiro, podemos pensar num modelo de classificação incremental utilizando um banco de estimadores cujas saídas indicam probabilidades de determinadas espécies estarem presentes na região de interesse. Com o passar do tempo (ou a cada nova sílaba), as probabilidades seriam atualizadas até convergir para uma decisão global. Esta decisão seria tomada por um observador que pudesse escolher a classe com maior probabilidade.

A frequência ou intervalo de tempo entre vocalizações poderiam ser utilizadas como características adicionais para classificação, uma que carregam muita informação discriminante. Além disso, podemos melhorar os estágios de detecção e extração de sílabas para resolver as limitações do algoritmo apresentadas na Seção 5.2.

Também podemos avaliar o impacto de falhas nos nós sensores e em suas medidas. As falhas são muito comuns em RSSF, pelo fato dos nós sensores utilizarem componentes de baixo custo. Desta forma, é necessário que as aplicações e algoritmos considerem este problema. Também podemos analisar o efeito da vocalização de outros animais como grilos, cigarras e pássaros dentro da região de interesse.

Podemos também estender este trabalho para classificação de múltiplos alvos, iniciando pela avaliação da taxa de erro deste algoritmo para classificar três anuros vocalizando ao mesmo tempo. Num segundo estágio, seria necessário definir uma forma de estimar a quantidade de vocalizações simultâneas. Assim, o algoritmo seria capaz receber um áudio qualquer com vocalizações de um, dois, ou três anuros, e responder indicando quais anuros estão vocalizando.

Para melhorar a classificação, também poderíamos modelar o problema como o problema do *cocktail party* (ou separação de fontes). Neste problema, temos várias fontes sonoras e vários microfones em uma região. O sinal coletado pelos microfones é uma composição do sinal das várias fontes, e o objetivo é separá-los para obter os sinais originais. Numa RSSF, cada nó sensor carrega um microfone que coleta os sinais misturados. A rede poderia obter os sinais originais e com isso a classificação poderia alcançar melhores resultados.

5.4 Publicações

Este trabalho de mestrado resultou na seguinte publicação:

- Ribas, A. D.; Figueiredo, C. M. S. & Nakamura, E. F. (2012). Similarity clustering for data fusion in wireless sensor networks using k -means. Em *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 488–494.

Durante o mestrado, alguns trabalhos e publicações surgiram como fruto de colaboração e esforço dentro do grupo de pesquisa:

- Ribas, A. D.; Carvalho Jr, A. R.; Figueiredo, C. M. S. & Nakamura, E. F. (2013). Characterizing the Communication in the Amazon Rainforest: Towards a Realistic Simulation. *Journal of the Brazilian Computer Society*. Aceito para ser publicado.
- Colonna, J.; Ribas, A.; dos Santos, E. & Nakamura, E. (2012). Feature subset selection for automatically classifying anuran calls using sensor networks. Em *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 3086–3093.
- Carvalho Jr, A. R.; Ribas, A. D.; Camara Neto, V. F.; Nakamura, E. F. & Figueiredo, C. M. S. (2012). An RSSI-based navigation algorithm for a mobile robot in wireless sensor networks. Em *Proceedings of the 37th IEEE Conference on Local Computer Networks (LCN 2012)*, pp. 1–4.
- Carvalho Jr, A. R.; Ribas, A. D. & Figueiredo, C. M. S. (2011). Algoritmo de Navegação Robótica em Redes de Sensores sem Fio baseado no RSSI. Em *III Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP)*.

Referências Bibliográficas

- Akyildiz, I. F.; Su, W.; Sankarasubramaniam, Y. & Cyirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393--422.
- Aljaafreh, A. & Al-Fuqaha, A. (2010). Multi-target classification using acoustic signatures in wireless sensor networks: A survey. *Signal Processing: An International Journal (SPIJ)*, 4(4):175--200.
- Arora, A.; Dutta, P.; Bapat, S.; Kulathumani, V.; Zhang, H.; Naik, V.; Mittal, V.; Cao, H.; Gouda, M.; Choi, Y.; Herman, T.; Kulkarni, S.; Arumugam, U.; Nesterenko, M.; Vora, A. & Miyashita, M. (2004). A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks*, 46(5):605--634.
- Bahrepour, M.; Meratnia, N.; Poel, M.; Taghikhaki, Z. & Havinga, P. J. (2010). Distributed Event Detection in Wireless Sensor Networks for Disaster Management. Em *Proceedings of the 2nd International Conference on Intelligent Networking and Collaborative Systems*, pp. 507--512, Washington, DC, USA.
- Brooks, R. R.; Ramanathan, P. & Sayeed, A. M. (2003). Distributed target classification and tracking in sensor networks. *Proceedings of the IEEE*, 91(8):1163--1171.
- Carey, C.; Heyer, W. R.; Wilkinson, J.; Alford, R.; Arntzen, J. W.; Halliday, T.; Hun Gerford, L.; Lips, K. R.; Middleton, E. M.; Orchard, S. A. & Rand, A. S. (2001). Amphibian declines and environmental change: Use of remote-sensing data to identify environmental correlates. *Conservation Biology*, 15(4):903--913.
- Carvalho Jr, A. R.; Ribas, A. D.; Camara Neto, V. F.; Nakamura, E. F. & Figueiredo, C. M. S. (2012). An RSSI-Based Navigation Algorithm for a Mobile Robot in Wireless Sensor Networks. Em *Proceedings of the 37th IEEE Conference on Local Computer Networks (LCN 2012)*, pp. 308--311, Clearwater, USA.

- Carvalho Jr, A. R.; Ribas, A. D. & Figueiredo, C. M. S. (2011). Algoritmo de Navegação Robótica em Redes de Sensores sem Fio baseado no RSSI. Em *Anais do 34º Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP 2011)*, Natal, Brasil.
- Chair, Z. & Varshney, P. (1986). Optimal data fusion in multiple sensor detection systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(1):98--101.
- Chen, B.; Jiang, R.; Kasetkasem, T. & Varshney, P. (2002). Fusion of decisions transmitted over fading channels in wireless sensor networks. Em *Conference Record of the 36th Asilomar Conference on Signals, Systems and Computers*, volume 2, pp. 1184--1188.
- Chen, W.; Hong, Y.-z. & Wang, P.-p. (2009). Decision fusion with wireless sensor networks under single-sideband gaussian fading channel. Em *Proceedings of the International Conference on Communications, Circuits and Systems (ICCCAS)*, pp. 210--212, San Jose, USA.
- Chiuso, A.; Fagnani, F.; Schenato, L. & Zampieri, S. (2011). Gossip algorithms for simultaneous distributed estimation and classification in sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):691--706.
- Chun-Ting, L.; Hong, H.; Tao, F.; De-Ren, L. & Xiao, S. (2006). Classification fusion in wireless sensor networks. *Acta Automatica Sinica*, 32(6):947--955.
- Clouqueur, T.; Saluja, K. K. & Ramanathan, P. (2004). Fault tolerance in collaborative sensor networks for target detection. *IEEE Transactions on Computers*, 53(3):320--333.
- Collins, J. P. & Storfer, A. (2003). Global amphibian declines: sorting the hypotheses. *Diversity and Distributions*, 9:89--98.
- Colonna, J. (2012). Uma abordagem para classificação de anuros baseada em vocalizações. Dissertação de mestrado, Universidade Federal do Amazonas.
- Colonna, J.; Ribas, A.; dos Santos, E. & Nakamura, E. (2012). Feature subset selection for automatically classifying anuran calls using sensor networks. Em *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2012)*, pp. 3086--3093, Brisbane, Australia.
- Davis, S. & Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 28(4):357--366.

- D'Costa, A.; Ramachandran, V. & Sayeed, A. M. (2004). Distributed classification of Gaussian space-time sources in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1026--1036.
- D'Costa, A. & Sayeed, A. M. (2003). Data versus decision fusion for distributed classification in sensor networks. Em *Proceedings of the 22nd IEEE Military Communications Conference (MILCOM 2003)*, pp. 585--590 Vol.1, Boston, USA.
- Dempster, A.; Laird, N. & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(1):1--38.
- Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The Annals of Mathematical Statistics*, 38(2):325--339.
- Dimakis, A.; Kar, S.; Moura, J.; Rabbat, M. & Scaglione, A. (2010). Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847--1864.
- Ding, J.; Cheung, S.-Y.; Tan, C.-W. & Varaiya, P. (2004). Signal processing of sensor node data for vehicle detection. Em *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems*, pp. 70--75, Shanghai, China.
- Duarte, M. & Hu, Y.-h. (2003). Distance based decision fusion in a distributed wireless sensor network. Em *Proceedings of the 2nd International Workshop on Information Processing in Sensor Networks (IPSN 2003)*, pp. 22--23.
- Duarte, M. & Hu, Y.-h. (2004). Vehicle classification in distributed sensor networks. *Journal of Parallel and Distributed Computing*, 64(7):826--838.
- Duda, R. O.; Hart, P. E. & Stork, D. G. (2001). *Pattern Classification*. Wiley, New York, 2. edição.
- Edgeworth, F. Y. (1908). On the probable errors of frequency-constants. *Journal of the Royal Statistical Society*, 71(3):499--512.
- Forero, P. A.; Cano, A. & Giannakis, G. B. (2008a). Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks. Em *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'08*, pp. 1989--1992, Las Vegas, USA.

- Forero, P. A.; Cano, A. & Giannakis, G. B. (2008b). Consensus-based k-means algorithm for distributed learning using wireless sensor networks. Em *Proceedings of the 1st Workshop on Sensors, Signal and Information Processing*, pp. 1--4, Sedona, USA.
- Forero, P. A.; Cano, A. & Giannakis, G. B. (2011). Distributed clustering using wireless sensor networks. *IEEE Journal of Selected Topics in Signal Processing*, 5(4):707--724.
- Gu, D. (2008). Distributed EM algorithm for gaussian mixtures in sensor networks. *IEEE Transactions on Neural Networks.*, 19 (7):1154 -- 1166.
- Hu, W.; Bulusu, N.; Chou, C. T.; Jha, S.; Taylor, A. & Tran, V. N. (2009). Design and evaluation of a hybrid sensor network for cane toad monitoring. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):4:1--4:28.
- Katenka, N.; Levina, E. & Michailidis, G. (2008). Local vote decision fusion for target detection in wireless sensor networks. *IEEE Transactions on Signal Processing.*, 56(1):329--338.
- Kokiopoulou, E. & Frossard, P. (2011). Distributed classification of multiple observation sets by consensus. *IEEE Transactions on Signal Processing.*, 59(1):104--114.
- Krishnamachari, B. & Iyengar, S. (2004). Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Transactions on Computers.*, 53(3):241--250.
- Li, D.; Wong, K.; Hu, Y. H. & Sayeed, A. (2002). Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine.*, 19 (2):17--29.
- Lin, Y.; Chen, B. & Varshney, P. (2005). Decision fusion rules in multi-hop wireless sensor networks. *IEEE Transactions on Aerospace and Electronic Systems.*, 41(2):475--488.
- Luo, X.; Dong, M. & Huang, Y. (2006). On distributed fault-tolerant detection in wireless sensor networks. *IEEE Transactions on Computers*, 55(1):58--70.
- Lynch, N. A. (1997). *Distributed Algorithms*. Morgan Kaufmann, San Francisco, CA.
- MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. Em *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*. University of California Press., pp. 281--297.

- Mahalanobis, P. C. (1936). On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India* 2, 1:49--55.
- Mainwaring, A.; Culler, D.; Polastre, J.; Szewczyk, R. & Anderson, J. (2002). Wireless sensor networks for habitat monitoring. Em *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, WSNA '02*, pp. 88--97, New York, NY, USA. ACM.
- Malhotra, B.; Nikolaidis, I. & Harms, J. (2008a). Distributed classification of acoustic targets in wireless audio-sensor networks. *Computer Networks*, 52(130):2582--2593.
- Malhotra, B.; Nikolaidis, I. & Nascimento, M. A. (2008b). Distributed and efficient classifiers for wireless audio-sensor networks. Em *Proceedings of the 5th International Conference on Networked Sensing Systems (INSS'08)*, pp. 203--206, Kanazawa, Japan.
- Mitchell, T. (1997). *Machine Learning*. McGraw-Hill Education.
- Nakamura, E. F.; Loureiro, A. A. F. & Frery, A. C. (2007). Information fusion for wireless sensor networks: Methods, models, and classifications. *ACM Computing Surveys*, 39(3):1--55.
- Nowak, R. (2003). Distributed EM algorithms for density estimation and clustering in sensor networks. *IEEE Transactions on Signal Processing*, 51(8):2245--2253.
- Olfati-Saber, R.; Fax, J. & Murray, R. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215--233.
- Pai, H.-T. (2010). Reliability-based adaptive distributed classification in wireless sensor networks. *IEEE Transactions on Vehicular Technology*, 59(9):4543--4552.
- Pereira, S.; Barbarossa, S. & Pagès-Zamora, A. (2010). Consensus for distributed EM-based clustering in WSNs. Em *Proceedings of the 6th IEEE Sensor Array and Multichannel Signal Processing Workshop, SAM 2010*, pp. 45--48, Jerusalem, Israel.
- Prasad, L.; Iyengar, S. S.; Rao, R. L. & Kashyap, R. L. (1994). Fault-tolerant sensor integration using multiresolution decomposition. *Physical Review E*, 49(4):3452--3461.
- Qi, H.; Iyengar, S. S. & Chakrabarty, K. (2001). Multiresolution data integration using mobile agents in distributed sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 31(3):383--391.

- Ribas, A. D.; Carvalho Jr, A. R.; Figueiredo, C. M. S. & Nakamura, E. F. (2013). Characterizing the Communication in the Amazon Rainforest: Towards a Realistic Simulation. *Journal of the Brazilian Computer Society*. Accepted.
- Ribas, A. D.; Figueiredo, C. M. S. & Nakamura, E. F. (2012). Similarity Clustering for Data Fusion in Wireless Sensor Networks Using k-means. Em *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2012)*, pp. 488--494, Brisbane, Australia.
- Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.
- Sun, Y. & Qi, H. (2008). Dynamic target classification in wireless sensor networks. Em *Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*, pp. 1--4.
- Wang, T.-Y.; Han, Y.; Chen, B. & Varshney, P. (2006). A combined decision fusion and channel coding scheme for distributed fault-tolerant classification in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 5(7):1695--1705.
- Wang, T.-Y.; Han, Y. S.; Varshney, P. K. & Chen, P.-N. (2005). Distributed fault-tolerant classification in wireless sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):724--734.
- Wang, X. & Qi, H. (2002). Acoustic target classification using distributed sensor arrays. Em *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4186--4186, Orlando, USA.
- Wang, X.; Qi, H. & Iyengar, S. S. (2002). Collaborative multi-modality target classification in distributed sensor networks. Em *Proceedings of the 5th International Conference on Information Fusion (ISIF)*, pp. 285 -- 290, Colorado, USA.
- Wang, Y.; Xiong, M.; Yue, D.-w. & He, R. (2009). Decision Fusion Rules over Rician Fading Channel for Wireless Sensor Networks. Em *Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom)*, pp. 3197--3200, Beijing, China.
- Williams, S. (2001). Multiple determinants of australian tropical frog biodiversity. *Biological Conservation*, 98(1):1--10.