Caracterização de Algoritmos de Redução de Dados Baseados em Stream de Dados em Redes de Sensores Sem Fio

Saulo H. C. Silva
Departamento de Computação – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Bauxita, Ouro Preto, MG
saulo.henrique.cabral@gmail.com

Suellen S. Almeida
Departamento de Computação – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Bauxita, Ouro Preto, MG
susilvaalmeida@gmail.com

Andre L. L. Aquino
Instituto de Computaçã – Universidade Federal de Alagoas
Campus A.C. Simões, sn, Tabuleiro do Martins, Maceió, AL
alla@ic.ufal.br

Ricardo A. R. Oliveira
Departamento de Computação – Universidade Federal de Ouro Preto
Campus Morro do Cruzeiro, Bauxita, Ouro Preto, MG
rrabelo@gmail.com

RESUMO

Um dos principais problemas em Rede de Sensores Sem Fio (RSSFs) é a energia restrita de cada sensor. Esse artigo analisa e compara três algoritmos de redução de dados baseados em *stream* de dados. A comparação é baseada no teste Kolmogorov-Smirnov, um teste estatístico robusto que permite a comparação entre dois conjuntos de dados em termos de suas distribuições. Além disso, apresentamos uma importante forma para a geração de instâncias de dados permitindo o teste dos algoritmos considerando diferentes distribuições de dados. Esta caracterização ajudará no projeto de algoritmos mais sofisticados para a redução de dados baseados em *stream* de dados em RSSFs.

PALAVRAS CHAVE. Caracterização de algoritmos, redes de sensores, algoritmos baseado em stream de dados. Área de classificação principal.
EST - Estatística

ABSTRACT

An important issue in Wireless Sensor Networks (WSNs) is the limited energy available for the sensors. This paper examines and compare three important sensor stream reduction algorithms. The comparison is based on Kolmogorov-Smirnov test a robust statistic method used to compare two data set considering their distribution similarities. Therefore, we present a important ways to generate the data instances allowing the test of algorithms considering different data distributions. This characterization will help in design of more sofisticaste sensor stream reduction algorithms in WSNs.

KEYWORDS. Algorithm characterization, sensor networks, data stream based algorithms. Main area.
EST - Statistic

1. Introdução

Nos últimos anos, houve um grande avanço tecnológico nas áreas de sensores, circuitos integrados e comunicação sem fio, que levou a criação de redes de sensores sem fio (RSSFs) (Akyildiz et al., 2002). Este tipo de rede pode ser aplicado no monitoramento, rastreamento, coordenação e processamento em diferentes contextos. Por exemplo, pode-se interconectar sensores para fazer o monitoramento e controle das condições ambientais numa floresta, oceano ou um planeta. A interconexão de sensores através de redes sem fio, com a finalidade de executar uma tarefa de sensoriamento maior, esta revolucionando a coleta e processamento de informações.

Normalmente as RSSFs possuem um grande número de nós distribuídos, que possuem graves restrições de energia, processamento e largura de banda limitada. Estes nós podem ser equipados com uma variedade de sensores, tais como acústico, sísmico, infravermelho, calor e temperatura. Além disso, esses nós podem se organizar em grupos onde pelo menos um dos sensores deve ser capaz de detectar um evento na região, processá-lo e tomar uma decisão.

Uma RSSF possui, além dos nós sensores, um ou mais nós sorvedouros que recebem os dados e os processam, além de um *gateway* para efetuar a comunicação das RSSFs com outras redes. A transmissão *ad-hoc* é um tipo de comunicação em que os próprios nós sensores funcionam como roteadores, encaminhando de forma cooperativa dados provenientes dos nós vizinhos.

Como apresentado anteriormente os nós sensores têm várias restrições, sendo a mais crítica a energia restrita. Uma vez que estes sensores são alimentados por baterias, fica inviável recarregar uma rede inteira, por exemplo, recarregar uma rede com 1.000.000 de nós, ou uma rede disposta numa região de difícil acesso como a boca de um vulcão ou uma floresta fechada. Dessa forma, ao projetarmos a rede, é necessário tentar diminuir o consumo de energia dos sensores, a fim de maximizar o tempo de vida dos sensores. Uma maneira de minimizar esse excessivo consumo de energia é reduzir a quantidade de dados enviados, uma vez que a atividade que mais consome energia em RSSFs é o envio dos dados sensoriados (Estrin et al., 2001).

Entretanto, ao efetuarmos a redução dos dados a representatividade dos mesmos pode ser comprometida. Nesse caso, efetuar grandes reduções pode descaracterizá-los e assim os invalidando para a aplicação. Portanto, é importante a aplicação de técnicas de redução "inteligentes". Para podermos comparar a "inteligência" das diferentes técnicas é necessário um projeto que permita não só a utilização de um conjunto de dados de entrada padrão para os testes dos algoritmos, mas também a verificação da qualidade dos dados de forma homogênea. Nesse trabalho, a partir da caracterização da redução de dados em RSSFs, apresentamos: diferentes distribuições de dados para gerar dados de entrada; utilizamos diferentes algoritmos baseados em *stream* de dados (Muthukrishnan, 2005; Aquino et al., 2007) para redução de dados; e discutimos uma estratégia genérica para verificar a qualidade dos dados após redução.

Este artigo esta organizado da seguinte maneira. A seção 2 apresenta a caracterização da redução de dados em RSSFs. A seção 3 mostra a representação genérica para os dados de entrada a serem reduzidos. A seção 4 apresenta os algoritmos de redução considerados. A seção 5 discute uma regra de decisão que pode ser utilizada para aferir a qualidade dos algoritmos de redução. A seção 6 detalha as simulações realizadas. Por fim, a seção 7 apresenta a conclusão e futuras direções.

2. Caracterização da redução de dados em redes de sensores sem fio

Para caracterizar o *stream* de sensoriamento temos que as RRSFs consistem de dispositivos de sensoriamento autônomos que trabalham de forma distribuída e cooperativa com o objetivo de monitorar condições físicas ou ambientais, tais como temperatura, som, vibrações, pressão, movimento ou poluição (Romer and Mattern, 2004). Tais sistemas podem ser representados pelo diagrama mostrado na figura 1 (Aquino et al., 2008) onde $\mathscr{N}$ denota o ambiente e o processo a ser medido, F é o fenômeno de interesse, com V^* seu domínio espaço-temporal. Se uma observação foi completada sem problemas, teremos um conjunto de regras (R^*) ideais para tomada de decisões ideais (D^*).

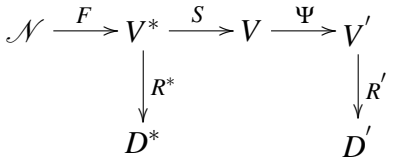

Figura 1: Representação de um sistema de uma RSSFs.

Ao invés de uma situação ideal, temos um conjunto de s sensores, $S = (S_1, \ldots, S_s)$, monitorando um fenômeno e produzindo conjuntos de amostras no domínio V_i, com $1 \leq i \leq s$; todos os possíveis conjuntos do domínio são denotados por $V = (V_1, \ldots, V_s)$.

Porém, utilizar todo o conjunto V pode ser muito oneroso em termos de, potência, largura de banda, recursos computacionais e, especialmente, no tempo de entrega das mensagens gerando atrasos para as aplicações. Uma vez que existe redundância nos dados da maioria das aplicações, a redução dos dados pode não degradar a informação sensoriada. As técnicas de redução de dados nos sensores são denotadas por, ψ, e a redução dos dados do domínio V é denotada por V'. Com isso, as novas regras que usam V' são denotadas por R', e elas conduzem a um conjunto de decisões D^*.

Neste trabalho estamos tratando da seguinte sequência $V \rightarrow V' \rightarrow D^*$. Com isso, mostraremos uma forma padronizada para gerar o conjunto de dados V, apresentaremos um conjunto de algoritmos que representam a transformação Ψ e discutiremos uma regra genérica R' para ser utilizada pelos projetistas das RSSFs.

3. Geração dos dados de entrada

Considerando que o conjunto de dados V, apresentado anteriormente, será a entrada dos dados de sensoriamento, temos que para avaliar os algoritmos de redução de dados devemos considerar um "gerador de dados". O gerador de dados, desenvolvido na linguagem *R* (R Development Core Team, 2010), foi concebido para permitir a análise do comportamento dos algoritmos considerando como entrada conjuntos de dados que seguem diferentes distribuições estatísticas apresentadas nas próximas subseções.

3.1. Distribuição de Poisson

A característica principal da Distribuição de Poisson que nos leva a aplicá-la em RSSFs é o fato de podermos modelar os eventos monitorados V considerando que a probabilidade de que um evento ocorra seja a mesma para cada intervalo de tempo e que o número de ocorrência em um intervalo independe do número de ocorrência em outros intervalos. Uma maneira de se modelar esse tipo de evento é

$$R_k(t) = e^{-\lambda_k t} \tag{1}$$

onde λ_k é a constante de tempo de ocorrência do evento k e t é o tempo decorrido.

Em suma na Distribuição de Poisson a probabilidade de que existam exatamente k ocorrências, sendo k um inteiro não negativo, $k = 0, 1, 2, \ldots$, é

$$f(k;y) = \frac{e^{-\gamma}\gamma^k}{k!} \tag{2}$$

onde e é a base do logaritmo natural e γ é um número real, igual ao número esperado de ocorrências em um dado intervalo de tempo.

3.2. Distribuição Binomial

A Distribuição Binomial é a probabilidade discreta do número de sucessos numa seqüência de n tentativas tais que as tentativas são independentes; cada tentativa resulta apenas em duas possibilidades, sucesso ou fracasso (a que se chama de tentativa de Bernoulli); a probabilidade de cada tentativa, p, permanece constante.

A probabilidade de ter exatamente k sucessos é dada pela função de probabilidade

$$f(k;n,p) = \frac{n}{k}p^k(1-p)^{n-k} \tag{3}$$

onde $k = 0, 1, 2, \ldots, n$ e $\frac{n}{k}$ é uma combinação.

Uma variação da Distribuição Binomial é a Binomial Negativa que indica o número de tentativas necessárias para obter k sucessos de igual probabilidade θ ao fim de n experiências, sendo a última tentativa um sucesso. A sua função de probabilidade é dada por

$$b(n;k;\theta) = \frac{n-1}{k-1}\theta^k(1-\theta)^{n-k} \tag{4}$$

3.3. Distribuição Geométrica

A distribuição geométrica é constituída por duas funções de probabilidade discretas: a distribuição de probabilidade do número X de tentativas de Bernoulli necessárias para alcançar um sucesso, suportadas pelo conjunto $1, 2, 3, \ldots$, ou a distribuição de probabilidade do número $Y = X - 1$ de insucessos antes do primeiro sucesso, suportadas pelo conjunto $0, 1, 2, 3, \ldots$.

Se a probabilidade de sucesso de cada tentativa é p, então a probabilidade de n tentativas serem necessárias para ocorrer um sucesso é

$$P(X = n) = (1-p)^{n-1}p, \tag{5}$$

para $n = 1, 2, 3, \ldots$. De forma equivalente, a probabilidade de serem necessários n insucessos antes do primeiro sucesso é

$$P(Y = n) = (1-p)^n p, \tag{6}$$

para $n = 1, 2, 3, \ldots$. Em ambos os casos, a seqüência de probabilidades é uma progressão geométrica.

4. Algoritmos de redução baseados em stream de dados

Seguindo a caracterização apresentada anteriormente (Figura 1) o próximo elemento que deve ser considerado é a transformação Ψ, ou seja, os algoritmos de redução propriamente ditos. Nessa seção, apresentamos dois algoritmos de redução. O primeiro é um algoritmo de amostragem que utiliza a transformada *Wavelet* (Aquino et al., 2010) e o segundo é outro algoritmo de amostragem que utiliza a distribuição de frequência acumulada (Aquino et al., 2007). Ambos os algoritmos são baseados em técnicas de *stream* de dados e foram implementados na linguagem *R* (R Development Core Team, 2010).

4.1. Algoritmo de amostragem baseado em transformada de wavelet

Algoritmo de amostragem baseado em transformada de wavelet aqui apresentado utiliza a transformada *wavelet* com funções de base *coiflets*, permitindo assim uma amostragem dinâmica dos dados (Mallat, 1998). Por intermédio desse tipo de amostragem, é possível uma detecção mais detalhada de eventos, comparando com outras funções de base da *wavelet* (Mallat, 1998), como por exemplo, *Haar* ou *Daubechies*. Isso ocorre, pois a base *coiflets* possui melhores resultados quando os dados podem ser interpolados por uma função polinomial. Com isso, o algoritmo de *wavelet*, ao ser aplicado em RSSFs, reduz os dados com eficiência, sem perder a representatividade dos mesmos.

Seja V'_j uma sequência de subespaços fechados de $L^2(\mathbb{R})$ e $f(x) \in L^2(\mathbb{R})$ seja o sinal observado. Cada V'_j representa aproximações sucessivas do sinal original, considerando a resolução de 2^j. Os detalhes da projeção em 2^j e 2^{j-1}, denotado por W_j, é definido por $W_j \bigoplus V'_j = V'_{j-1}$, onde $\bigoplus$ denota a soma direta de dois espaços vetoriais.

Filtros discretos são definidos para escolher os níveis de frequência presentes nos dados, que variam em escala temporal. Dois conjuntos de funções são aplicados: funções de escala ($\phi(t)$) e funções de *wavelet* ($\psi(t)$). Dessa forma, pode ser aproximado através da seguinte expansão:

$$f(x) = \sum_n s_{i_0}[n]\, \psi_{i_0,n(x)} + \sum_{i=i_0}^{i_1} w_i[n]\, \phi_{i_0,n(x)} \tag{7}$$

onde $s_{i_0}[n] \in V'_j$ são coeficientes de escala e $w_i[n] \in W_i$ são coeficientes de *wavelet*.

O método em estudo pode ser dividido em alguns passos:

Passo 1 Definir o *stream* original V como o dado sensoriado e gerar as funções $h(i)$ e $g(i)$ da base *coiflets*. Esses filtros h e g são a forma discreta de dois tipos de funções aplicadas à transformada *wavelet*, funções de wavelet ($\psi(t)$) e funções de escala ($\phi(t)$).

Passo 2 Aplicar a transformada de *wavelet* por intermédio de g e h. O resultado será a decomposição do sinal em diferentes subespaços, cada um com diferentes resoluções de tempo e frequência.

Passo 3 Calcular uma taxa aproximada de erro para manter a representatividade dos dados, incluindo quando algum evento externo ocorrer. O erro aproximado é:

$$\left\| f(x) - \sum_\tau g(\tau)\, \phi_{s,\tau(x)} \right\| = O(2^{s.2L}) \tag{8}$$

onde as variáveis s e τ são as novas dimensões obtidas depois da transformada, escala e translação; e $2L$ é o número de momentos *coiflets*.

O algoritmo 1 apresenta o pseudocódigo da redução baseada em transformada *wavelet*.

Seja M o número de níveis decompostos, o número total de operações no vetor de tamanho $|V|$ terá o número de operações na ordem de $O(M|V|\log|V|)$[2] (Aquino et al., 2010).

[2]Em todo o trabalho, ao utilizarmos $\log x$, estaremos sempre nos referindo ao logaritmo de x na base dois.

Algorithm 1 Algoritmo baseada em transformada *wavelet*

Require: V – *stream* original, h, g – Filtros da base Coiflets
Ensure: V'_j – amostra resultante

1: Gera os filtros $h(i)$ e $g(i)$
2: **for** $t \in [0, |V|/2-1]$ **do**
3: $u = 2t+1$
4: $V'_{jt} = g_1 V_{u+1}$
5: **for** $n \in [1, |h|-1]$ **do**
6: $u = u-1$
7: **if** $u \leqslant 0$ **then**
8: $u = |V|-1$
9: **end if**
10: $V'_{jt} = V'_{jt} + g_{n+1} V_{u+1}$
11: **end for**
12: $V'_{j(t+1)} = V'_{jt}$
13: **end for**

4.2. Algoritmo de amostragem baseado na distribuição de frequência acumulada

O algoritmo de amostragem baseado na distribuição de frequência acumulada destina-se a manter a frequência de ocorrências dos dados monitorados. O algoritmo *Sampling* (Aquino et al., 2007) fornece uma solução que permite um melhor equilíbrio, entre os dados e os requisitos da rede. A execução do algoritmo de *Sampling* pode ser dividido nos seguintes passos:

Passo 1 Construir um histograma da freqüência dos dados sensoriados V.

Passo 2 Criar uma amostra com base no histograma obtido no Passo 1. Para criar esta amostra, nós escolhemos aleatoriamente os elementos de cada classe (coluna) do histograma, respeitando o tamanho da amostra e as freqüências de classe do histograma. Assim, o resultado da redução será representado pelo histograma criado.

Passo 3 Organizar os dados da amostra V' de acordo com a ordem dos dados originais, para manter a ordem de chegada em relação aos dados originais.

O algoritmo 2 apresenta o pseudocódigo da redução baseada na distribuição de frequência acumulada.

Analisando o algoritmo 2 temos que devido a ordenação uma complexidade de tempo de $O(n \log n)$. Apenas destacando, as linhas de 5 a 18 definem o loop externo em que os dados de entrada são lidos e os elementos da amostra são escolhidos. O laço interno é executado somente quando a condição em linha 6 é satisfeita. A complexidade total do laço externo é $O(n)+O(m) = O(n+m)$, uma vez que temos uma execução intercalada. Consideramos que $numClass$ sendo o número de classe dos histogramas, $colOrig_i$ e $colSample_i$ as colunas originais e os histogramas reduzidos respectivamente, onde $0 < i \leq numClass$. Antes da avaliação a condição da linha 6, $colOrig_i$ é contada e $n/numClass$ interações são executados. Sempre que esta condição é satisfeita a $colSample_i$ é construída e $m/numClass$ interações são executadas (laço $8-13$). Para construir um histograma completo, deve-se abranger todas as classes ($numClass$), então nós temos $numClass\,(n+m)/numClass = n+m$. Como $O(n+m) < O(n \log n)$, a complexidade de tempo do algoritmo 2 é de $O(n \log n)$. A complexidade do espaço é $O(n+m) = O(n)$, porque nós guardamos o original fluxo de dados e a amostra resultante (Aquino et al., 2007).

Algorithm 2 Algoritmo de amostragem baseada na distribuição de frequência acumulada

Require: $\_dataIn$ – Stream original, m – Tamanho da redução
Ensure: $_{d}ataOut$ – amostra resultante

1: Ordene $\_dataIn$
2: $histScale \leftarrow$ "Class width"
3: $first \leftarrow \_dataIn[0]$
4: $count \leftarrow 0$, $j \leftarrow 0$
5: **for** $i \leftarrow 0$ até n **do**
6: **if** $(\_dataIn[i] > first + histScale)$ ou $(i = n-1)$ **then**
7: $colFreq \leftarrow m(count/dataInSize)$
8: **while** $colFreq > 0$ **do**
9: $index \leftarrow$ "elemento aleatório da classe histograma"
10: $\_dataOut[j] \leftarrow \_dataIn[index]$
11: $j \leftarrow j+1$
12: $colFreq \leftarrow colFreq - 1$
13: **end while**
14: $count \leftarrow 0$
15: $first \leftarrow \_dataIn[i]$
16: **end if**
17: $count \leftarrow count + 1$
18: **end for**
19: Ordene $\_dataOut$ {Ordenação de acordo com os dados originais}

5. Regras de decisão

Por fim, seguindo a caracterização apresentada anteriormente (Figura 1) o último elemento que deve ser considerado corresponde as regras de decisão R' que nos conduzem à decisões (D) coerentes em relação ao fenômeno observado. Tais regras são de suma importância por auxiliarem na determinação do quanto os algoritmos de redução degradam os dados. Para avaliar a representatividade dos dados, após a redução, consideramos como regra de decisão o teste estatístico *Kolmogorov-Smirnov* (Reschenhofer, 1997) e o implementamos na linguagem *R* (R Development Core Team, 2010).

Em estatística, esse teste é usado para determinar se duas distribuições de probabilidade subjacentes diferem uma da outra, ou se uma das distribuições difere da distribuição em hipótese, em qualquer dos casos com base em amostras finitas. Esse teste avalia se duas amostras V e V' têm distribuições similares não exigindo que as amostras sigam a distribuição normal, ou seja, caso os valores amostrados sigam outra distribuição este teste também pode ser utilizado. Essa é uma das principais razões para a escolha desse teste, pois o mesmo pode ser utilizado satisfatoriamente de forma genérica para a caracterização dos algoritmos. O teste KS é descrito a seguir:

1. Construir a distribuição acumulada Fn dos dois grupos V e V' usando a mesma classe para ambas as distribuições.

2. Determinar as diferenças acumuladas para cada ponto da distribuição e considerar a maior das diferenças (D_{max}).

3. Computar o valor crítico,

$$D_{crit} = y\sqrt{(|V| + |V'|)/|V||V'|} \tag{9}$$

onde y é um valor tabelado e depende do nível de significância do teste.

4. As amostras seguem a mesma distribuição se

$$D_{max} \leq D_{crit}. \tag{10}$$

6. Simulação e resultados

Para o estudo de casos foram utilizados dados empíricos baseados nas distribuições estatísticas apresentadas anteriormente e os dados sensoriados em um ambiente real. Os dados gerados envolvem tamanhos iguais a $256, 512, 1024, 2048$. Para os testes realizados utilizamos uma redução dos dados de $\log n$ e $n/2$. Para cada distribuição estatística, cada algoritmo foi testado com 33 instâncias de dados diferentes. Ao final, é apresentado a média e o intervalo de confiança para cada cenário considerado. Para os resultados apresentados consideramos $|V| = n$.

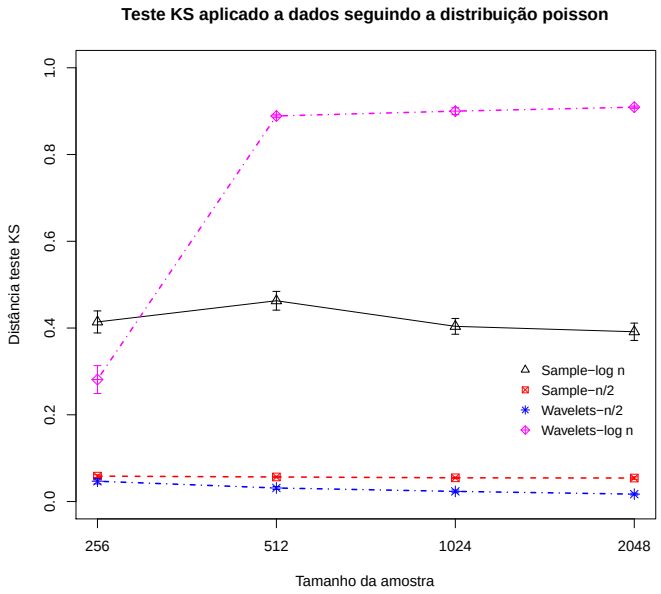

Figura 2: Erro KS com Distribuição Poisson.

Para aplicações onde os dados seguem uma Distribuição de Poisson (Figura 2), podemos notar que para o algoritmo de *Wavelets* utilizando redução de $\log n$ aplicado a dados com tamanho de 256, é relativamente melhor que o algoritmo de *Amostragem*. No entanto para todos os outros casos o algoritmo de *Amostragem* se mantém com um erro constante enquanto o algoritmo de *Wavelets* apresenta um erro significativo.

No entanto, apesar do algoritmo de *Amostragem*, utilizando redução de $n/2$, se apresentar estável para todos os tamanhos dos dados, o algoritmo de *Wavelets* apresenta os melhores resultados para esse tipo de cenário.

A figura 3 apresenta o teste KS para Distribuição Binomial. Podemos observar que tanto o algoritmo de *Amostragem* quanto o de *Wavelets* utilizando redução de tamanho $n/2$, apresentam perda de dados bem menor do que utilizando redução de $\log n$. A distância do teste KS chega quase a ser nula para a redução *Wavelet*, quando os dados sensoriados tem tamanho de 2048.

Além disso, para esta situação mesmo o algoritmo de *Amostragem* tendo um resultado constante para todos os tamanhos de dados sensoriados, é possível observar que o método

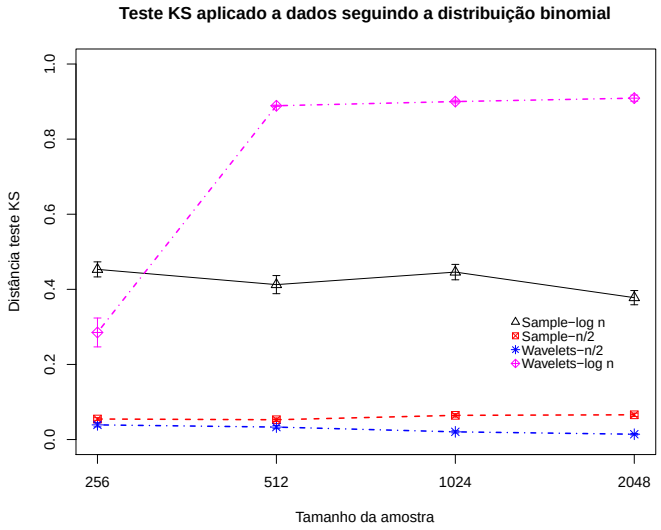

Figura 3: Erro KS com Distribuição Binomial.

de *Wavelets* apresenta o melhor resultado quando a redução é de metade dos dados. Assim como no cenário anterior, quando a redução é de $\log n$, esse algoritmo obteve a distância do teste KS menor que a do algoritmo de *Amostragem* apenas para amostras de tamanho 256, em todos os outros casos ele mostrou-se pior.

Para aplicações onde os dados que seguem uma Distribuição Binomial e não permitido grandes erros nos dados recebidos, por exemplo, erros acima de 10%, tanto o algoritmo de *Amostragem* quanto o de *Wavelet* utilizando redução $n/2$ são os apropriados, no entanto para todos os tamanhos de dados o algoritmo de *Wavelet* apresenta menor discrepância nas suas reduções. Para a situação onde os erros maiores são aceitável, o algoritmo de *Amostragem* utilizando redução de $\log n$, é mais apropriado para a situação do que a redução utilizando *Wavelets* com redução de $\log n$, com exceção de quando temos tamanho de dados igual 256. Vale ressaltar que a diferença de significância dos dados em alguns casos, quando utilizado redução $\log n$, pode chegar a uma distância de até 40% para a *Amostragem* $\log n$ e 80% para *Wavelet* $\log n$.

Para aplicações onde os dados sensoriados seguem uma Distribuição Binomial Negativa (Figura 4), como no caso anterior, os algoritmos com redução de $n/2$ apresentam uma significância dos dados bem melhor que os com redução de $\log n$.

Os algoritmos de *Amostragem* e *Wavelets* utilizando redução de $n/2$ apresentam-se estáveis para todos os tamanhos de dados. Contudo, para essa redução, o método de *Wavelets* continua apresentando a menor distância KS em todos os casos. A *Amostragem* com redução de $n/2$ apresenta seus melhores resultados com tamanho de dados igual a 256 bem próximo do resultado do algoritmo de *Wavelet*. O algoritmo de *Wavelet* como nos casos anteriores se mostra como a melhor opção para redução de dados com tamanho de $n/2$.

Por intermédio da figura 5 é possível observar que quando os dados seguem uma Distribuição Geométrica, em média temos a menor diferença entre as reduções de tamanho $n/2$ e $\log n$ para a *Amostragem*. Em especial podemos observar que para dados com tamanho igual a 256 o erro chega a ser menos que 10%.

Para este cenário o algoritmo de *Amostragem* com redução $\log n$ obteve seus melhores resultados comparado com os cenários anteriores, mesmo em comparação com o algoritmo

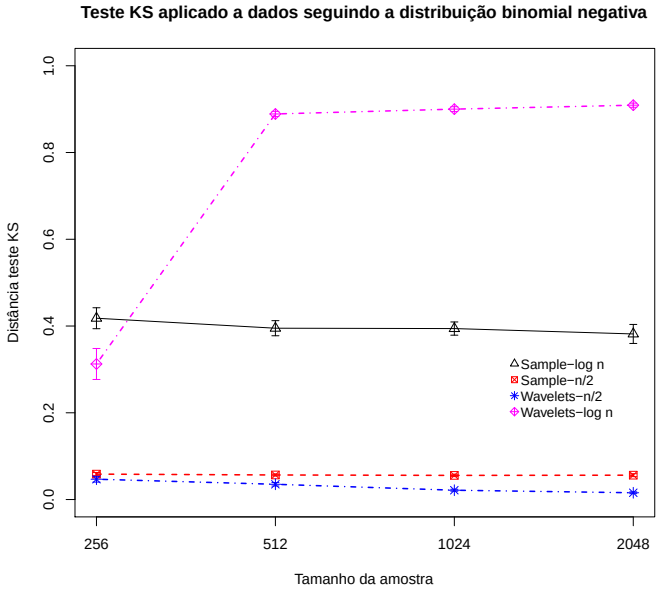

Figura 4: Erro KS com Distribuição Binomial Negativa.

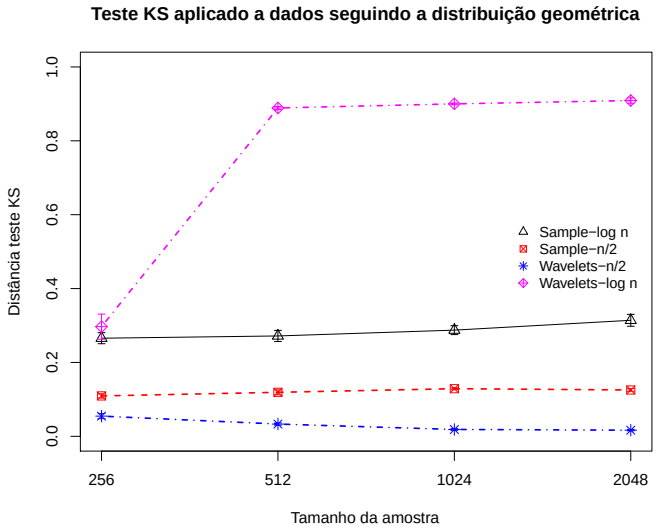

Figura 5: Erro KS com Distribuição Geométrica.

de *Wavelets* com amostras de tamanho 256 que vinha apresentando melhor resultado.

Em casos em que não se podem transmitir grandes quantidades de dados e existe uma garantia que os dados seguem uma Distribuição Geométrica o algoritmo de *Amostragem* utilizando redução de $\log n$ pode ser utilizado. Lembrando que neste caso a aplicação deve permitir que os dados reduzidos tenham erros de aproximadamente 30%. O algoritmo de *Amostragem* com redução $n/2$ apresenta-se estável para todos os tamanhos, é importante observar, no entanto que para esta situação temos os piores resultados obtidos até agora da *Amostragem* com redução de $n/2$. Para essa redução, o algoritmo de *Wavelet* ainda apresenta os melhores resultados para amostras de todos os tamanhos.

7. Conclusão e trabalhos futuros

RSSFs possuem um grande número de nós distribuídos, que possuem graves restrições de energia, de processamento e largura de banda limitada. Neste trabalho, avaliamos dois algoritmos de redução baseados em *stream* de dados. De forma complementar verificamos como os algoritmos se comportam com diferentes distribuições de dados. Esse estudo foi realizado seguindo a proposição de uma caracterização de tais algoritmos num ambiente de RSSFs.

Foi possível observar que em situações em que é preciso manter uma boa representatividade dos dados e que largura de banda proporcione envio para pelo menos metade dos dados, o algoritmo de *Wavelets*-$n/2$ apresentou os melhores resultados, sendo o erro teste KS quase nulo para todos os cenários considerados. Já para situações em que a largura de banda é muito restrita, suporta o envio de no máximo $\log n$ dos dados, o algoritmo de *Amostragem*-$\log n$ é mais eficiente que o de *Wavelet* para a maioria dos cenários. No entanto, estas aplicações devem tolerar erros na representatividade dos dados.

Em suma, as aplicações que aceitam em baixo erro, deve-se utilizar o algoritmo de *Wavelets*, uma vez que seus resultados não apresentam grandes alterações nos cenário considerados. A análise efetuada apresenta a importância para o projetista da rede no que diz respeito à caracterização das aplicações, tal análise é a principal contribuição destacada nesse trabalho.

Como proposta de trabalhos futuros, pretende-se, com base no estudo feito propor um novo algoritmo de redução considerando aspectos positivos de ambas as abordagens aqui apresentadas. Ademais, pretende-se incorporar a caracterização conjuntos de dados reais, com o objetivo de classificar tais algoritmos em ambientes reais.

Agradecimentos

Este trabalho é parcialmente financiado pela FAPEMIG processo CEX-APQ-00577-09 e pelo CNPq processo 477946/2010-0.

Referências

I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, A survey on sensor networks, *IEEE Commun. Mag.*, 40(8):102–114, August 2002.

Andre L. L. Aquino, Carlos M. S. Figueiredo, Eduardo F. Nakamura, Alejandro C. Frery, Antonio A. F. Loureiro, and Antonio Otavio Fernandes, Sensor stream reduction for clustered wireless sensor networks, *23rd ACM Symposium on Applied Computing 2008 (SAC'08)*, pages 2052–2056, Fortaleza, Brazil, March 2008, ACM.

Andre L.L Aquino, Carlos M.S. Figueiredo, Eduardo F. Nakamura, Luciana S. Buriol, Antonio A.F Loureiro, Antonio Otavio Fernandes, and Claudionor J.N.Jr Coelho, Data stream based algorithms for wireless sensor network application, *AINA '07 Proceedings of the 21st International Conference on Advanced Networking and Applications*, pages 869–876, 2007.

Andre L.L. Aquino, Ricardo A.R. Oliveiro, and Elizabeth F. Wanner, A wavelet-based sampling algorithm for wireless sensor networks applications, *SAC'10 Proceedings of the 2010 ACM Symposium on Applied Computing*, pages 1604–1608, 2010.

D. Estrin, L. Girod, G. Pottie, and M. Srivastava, Instrumenting the world with wireless sensor networks, *ICASSP'01 Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 2033–2036, Salt Lake City, UT, USA, 2001.

S. Mallat, *A Wavelet Tour of Signal Processing (Wavelet Analysis and Its Applications)*, Academic Press-Elsevier, Los Angeles, EUA, 1998.

S. Muthukrishnan, *Data Streams: Algorithms and Applications*, Now Publishers Inc, Hanover, MA, USA, January 2005, ISBN 1-933019-14-X.

R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2010, URL `http://www.R-project.org`, ISBN 3-900051-07-0.

Erhard Reschenhofer, Generalization of the kolmogorov-smirnov test, *Computational Statistics & Data Analysis*, 24(4):422–441, June 1997.

Kay Romer and Friedmann Mattern, The design space of wireless sensor networks, *IEEE Wireless Communications*, 11(6):54–61, December 2004.