

NN-clas: classificador geométrico de margem larga baseado na regra do vizinho mais próximo

Liliane dos Reis Gade¹, Cristiano Leite de Castro²,
Luiz Carlos Bambirra Torres³ *,
Frederico Gualberto Ferreira Coelho⁴,
Antônio Pádua Braga⁵,
Janier Arias García⁶, and
Frank Sill Torres⁷

Programa de Pós-Graduação em Engenharia Elétrica
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brasil

¹ liliane-gade@ufmg.br, ² crislcastro@ufmg.br, ³ luizlitc@gmail.com, ⁴ fredgfc@gmail.com, ⁵ apbraga@ufmg.br, ⁶ janier.arias@gmail.com, ⁷ franksill@ufmg.br

Resumo O presente artigo apresenta novas formas de implementar a regra de decisão de um classificador de margem larga. O objetivo é reduzir a complexidade das operações durante a etapa de classificação dos novos padrões. A abordagem proposta utiliza o método de vizinhos mais próximos como regra de decisão. Também são realizados estudos com uma formulação reduzida do classificador de margem larga. Resultados iniciais com 15 bases de dados reais utilizadas na literatura mostraram que a abordagem proposta se mostrou estatisticamente igual ao classificador de margem larga, porém menos complexo de se implementar em sistemas embarcados.

1 Introdução

Os métodos de aprendizagem de máquina têm sido aplicados em diversos setores da indústria, principalmente, quando implementados em computadores de propósito geral. Algumas dessas aplicações, contudo, demandam baixo consumo de energia e tamanho reduzido, o que torna a tarefa mais desafiadora diante da impossibilidade da utilização de tais métodos em computadores convencionais. Nessas situações, evidencia-se a necessidade de algoritmos de aprendizagem de máquina que possam ser implementados em sistemas embarcados, como em ASIC's (*Application Specific Integrated Circuits*) ou em FPGA's (*Field Programmable Gate Arrays*).

O uso dos algoritmos de aprendizagem de máquina em sistemas embarcados surgiu nos anos de 1990 e início dos anos 2000 com as placas de circuitos integrados que implementavam Redes Neurais Artificiais (RNA) [5]. O chip neural

* Bolsista do CNPq-Brasil (N°150254/2016-4)

da Intel ETANN [8], por exemplo, lançado em 1989, já possuía sinapses adaptativas, apesar de ainda ter a fase de treinamento realizada externamente devido aos complexos algoritmos de otimização, e da necessidade de uma interface com o usuário para configuração de seus parâmetros.

Atualmente outros estudos ([11], [12] e [7]) têm sido desenvolvidos no campo da implementação de redes neurais em FPGA. Neles, a fase de treinamento das RNAs também é realizada em um computador externo devido ao custo dos métodos de otimização utilizados para o aprendizado dos pesos das rede. Todavia, é importante ressaltar que essa prática diminui a flexibilidade do sistema pelo fato de não poderem ser treinados durante seu uso, impossibilitando assim a aplicação desses algoritmos em sistemas adaptativos *on-line*.

Para aplicações que exigem uma maior flexibilidade ou para o uso de métodos de aprendizagem de máquina em sistemas *on-line*, é necessário o uso de algoritmos que não exigem a resolução de problemas de otimização complexos. Encontra-se na literatura um método desenvolvido em [14], que se mostrou estatisticamente equivalente aos métodos SVM-RBF e SVM-Poly, e adequado para o uso nas aplicações descritas acima. Sem depender de um método formal de otimização, o classificador proposto em [14] leva em consideração apenas a estrutura dos dados representados pelo grafo de Gabriel [2] para minimizar o erro do conjunto de treinamento e maximizar a margem de separação entre as classes. Nesse método, a superfície de decisão é um resultado de um modelo de mistura de hiperplanos cujos parâmetros são extraídos dos padrões próximos a margem. A decisão sobre a classificação de um padrão arbitrário x depende da avaliação de todos os elementos da mistura. Este procedimento, embora factível de ser implementado em hardware, tem custos envolvidos no procedimento de estimação da superfície de decisão.

Para superar essa dificuldade, o presente artigo tem por objetivo avaliar novas formas de se implementar a regra de decisão do classificador de margem larga visando reduzir os custos computacionais, facilitando assim sua implementação em hardware, sem perder, contudo, o desempenho alcançado em [14] em termos da qualidade da classificação. É apresentada então, uma nova metodologia que se baseia na abordagem de vizinhos mais próximos (kNN) para realizar a classificação de um novo padrão x . Além disso, é feito um estudo que utiliza uma formulação reduzida (baseada no hiperplano mais próximo) da regra de decisão proposta em [14]. Resultados iniciais com 15 bases de dados reais mostram que uma das regras de decisão implementadas é estatisticamente igual ao do método proposto em [14], porém tem um menor custo de implementação em hardware.

O restante do artigo encontra-se organizado da seguinte forma. Na Seção 2 é apresentado, de maneira pormenorizada, o classificador de margem larga baseado em distância original, juntamente com a técnica alternativa para a regra de decisão do método utilizando o hiperplano mais próximo. Em seguida, na Seção 3, descreve-se o algoritmo kNN para uma melhor compreensão da nova abordagem desenvolvida. Na Seção 4, é apresentada a metodologia proposta pelo presente artigo. Superadas essas etapas, os experimentos e resultados são

detalhados na Seção 5. Finalmente, na Seção 6 são apresentadas as considerações finais.

2 Classificador de margem larga baseado em distância

O classificador de margem larga baseado em distância (CHIP-clas)[14] obtém a informação estrutural dos dados através do grafo de Gabriel, que, por sua vez, depende apenas da distância entre as amostras de treinamento. O grafo de Gabriel de um conjunto de pontos consiste em um grafo cujo conjunto de vértices V e arestas A deve obedecer à definição expressa pela Equação 1, onde os pontos (p_i, p_j) constituirão uma aresta se e somente se nenhum outro ponto estiver contido na hipersfera de diâmetro $D(p_i, p_j)$, sendo que tal diâmetro consiste na distância entre os pontos p_i e p_j .

$$(p_i, p_j) \in A \leftrightarrow D^2(p_i, p_j) \leq [D^2(p_i, z) + D^2(p_j, z)] \forall z \in V, p_i, p_j \neq z \quad (1)$$

O grafo $G = \{V, A\}$ de um conjunto de dados de treino $T = \{(x_i, y_i) | i = 1, \dots, N\}$, onde $y_i \in \{+1, -1\}$ e $x_i \in \mathbb{R}^n$ também possui um conjunto de Arestas de Suporte AS , o qual representa todas as arestas de A que possuem um par de vértices (x_i, x_j) de classes distintas. Caso não haja sobreposição entre as amostras, é possível afirmar que os vértices de AS localizam-se na margem de separação entre as classes, como pode ser visto na Figura 1. Dessa forma, o hiperplano H_l que passa pelo ponto médio de um par de vértices x_i e x_j pertencentes a AS , refere-se ao classificador de margem máxima em relação aos ditos vértices [15].

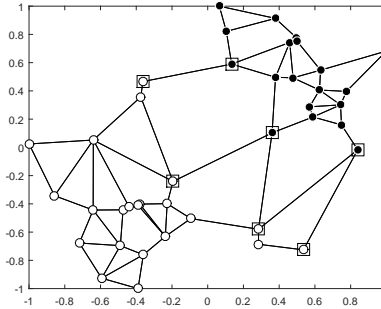


Figura 1: Grafo de Gabriel de duas gaussianas com arestas de suporte indicadas pelos quadrados

Para que a metodologia descrita acima possa ser aplicada em problemas com sobreposição ou ruídos entre as classes de dados, deve-se utilizar no CHIP-clas uma técnica de filtragem, como a de [3], para eliminar as amostras ruidosas antes da detecção do conjunto de Arestas de Suporte AS . Essa técnica de filtragem utiliza operações próprias da teoria dos Grafos, como por exemplo, o grau do vértice, o qual representa o número de arestas conectadas ao referido vértice. Segundo [3], os vértices de grau baixo podem ser rotulados como um possível ruído. Adotou-se assim, uma medida de qualidade para cada vértice do grafo, aqui formalmente definida pela Equação 2, onde $Gr(x_i)$ representa o grau do vértice x_i e $\hat{Gr}(x_i)$ o grau de x_i menos os vértices de classe distinta de x_i [15].

$$q(x_i) = \frac{\hat{Gr}(x_i)}{Gr(x_i)}, \quad (2)$$

$$\hat{Gr}(x_i) = \{\forall x_j \in Gr(x_i) | y_j = y_i\}$$

A filtragem é baseada na medida de qualidade $q(\cdot)$ e pode ser dividida em três fases: calcula-se, inicialmente $q(x_i)$ para todo $x_i \in G$. Em seguida, os $q(x_i)$ são agrupados por classe, de forma que Q^+ e Q^- representam a medida de qualidade entre as classes $+1$ e -1 , respectivamente. Posteriormente, deve se calcular de acordo com a Equação 3 o valor do limiar t^+ e t^- de cada classe como a média da medida de qualidade pertencente à Q^+ e Q^- . Por último, deve se remover todos os vértices de G que possuem $q(x_i)$ menores que t^+ e t^- [15]. No caso específico do CHIP-clas, a filtragem somente é realizada se $q(x) > 0$, $\forall x \in T$.

A Figura 2 exemplifica a importância do processo de filtragem em bases com sobreposição e ruídos. O gráfico mais a esquerda mostra um conjunto de dados que possui amostras ruidosas, como se pode observar os ruídos, indicados pelos quadrados, dificultam o processo de obtenção das arestas de suporte, podendo então definir algumas arestas, como de suporte, que não estejam localizadas na margem de separação entre as classes. Eliminando as amostras ruidosas, como exemplificado pelo gráfico a direita, é possível identificar de maneira mais precisa os vértices de AS localizados na borda de separação entre as classes.

$$t^+ = \frac{\sum_{q(x_i) \in Q^+} q(x_i)}{|Q^+|}, \quad t^- = \frac{\sum_{q(x_i) \in Q^-} q(x_i)}{|Q^-|} \quad (3)$$

Após a eliminação das amostras ruidosas e obtenção dos vértices de AS , será gerado um classificador representado pelo hiperplano H_i que passa pelo ponto médio de tais vértices. É importante ressaltar, que um único hiperplano H_i não separa todas as amostras do conjunto de treino, mas a combinação de todos os hiperplanos produz, segundo [14], um classificador com a informação espacial de todas as amostras em T .

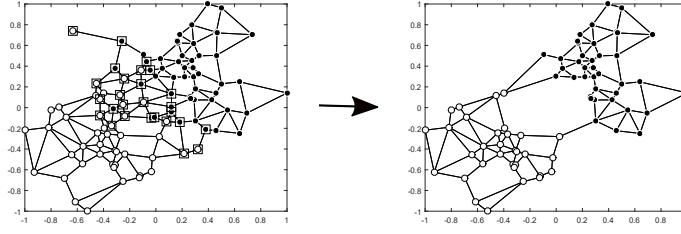


Figura 2: Remoção de amostras ruidosas de uma base de dados com sobreposição entre as classes

2.1 Mistura de hiperplanos

A classificação final resulta de uma Mistura Hierárquica de Especialistas (MHE), na qual cada hiperplano terá um peso diferente diante a um novo padrão de entrada x a ser rotulado. De acordo com [14], a arquitetura da MHE é representada através de uma rede, conforme a Figura 3, onde a primeira camada corresponde aos especialistas locais $\{H_1, \dots, H_m\}$. Já a saída de m especialistas para um novo padrão de entrada x a ser rotulado é representada pelas funções $h_1(x), \dots, h_m(x)$. Cada especialista é ponderado por um módulo *Gating Network*, onde o peso $c_i(x)$ para o i -ésimo especialista é obtido de acordo com a Equação 4, o $p_l = (x_i + x_j)/2$ representa o ponto médio da aresta de suporte formada pelos vértices x_i e x_j e $D(x, p_l)$ é a distância entre a amostra x a ser rotulada e o ponto médio p_l .

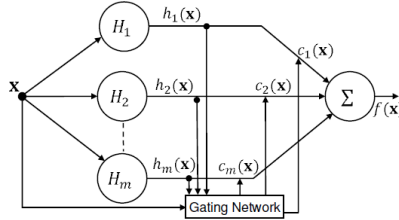


Figura 3: Arquitetura da Mistura Hierárquica de Especialistas

$$c_l(x) = \exp - \left(\frac{(\max(D(x, p_k)))^2}{D(x, p_l)} \right), \forall k = 1, \dots, m \quad (4)$$

Após o cálculo dos pesos pela Equação 4 uma normalização é imposta, tal que $\sum_{l=1}^m c_l(x) = 1$. Em seguida, o resultado final da classificação é obtido por $f(x) = \text{sign}(\sum_{l=1}^m h_l(x)c_l(x))$, sendo que $h_l = \text{sign}(x^T w_l - b_l)$. O $\text{sign}(\cdot)$ representa a

função sinal, $w_l = (x_l - x_j)$ e o termo de polarização $b_l = [(1/2)(x_l + x_j)]w_l^T$. O algoritmo 1 descreve de forma detalhada o processo de classificação realizado por esse método de margem larga.

Algorithm 1 Classificação - CHIP-clas

Entradas : conjunto de pontos médios das arestas de suporte P , conjunto de parâmetros B e W dos hiperplanos e conjunto de teste Xt .

Saída : classes do conjunto de teste CTe

```

for  $j$  in  $Xt$  do
  for  $i$  in  $P$  do
    Calcula a distância entre o novo padrão e o ponto médio da aresta de suporte
     $d(i) = D(Xt(j), P(i))$ 
  end for
  Calcula a distância máxima
   $mD = \max(d)$ 
  for  $i$  in  $W$  do
     $c(i) = \exp - \left( \frac{(mD)^2}{d(i)} \right)$ 
     $h(i) = \text{sign}(Xt(j)^T W(i) - B(i))$ 
  end for
   $c = \frac{1}{c + (1e^{-16})}$ 
   $c = \frac{c}{\text{sum}(c)}$ 
   $CTe(j) = \text{sign}(\text{sum}(h * c))$ 
end for
retorna  $CTe$ 

```

Percebe-se que a mistura de hiperplanos ainda é custosa para implementação em hardware, principalmente por causa das operações que utilizam exponencial, como representado pela Equação 4, e pelos cálculos para se obter a função $f(x) = \text{sign}(\sum_{l=1}^m h_l(x)c_l(x))$.

2.2 CHIP-clas Reduzido

A classificação pela técnica do CHIP-clas reduzido é obtida da seguinte maneira: dado uma amostra x a ser rotulada, encontra-se o ponto médio l mais próximo de x através de:

$$\arg \min_l D(x, P_l), \quad l = 1, \dots, m \quad (5)$$

onde $D(\cdot)$ é a função que retorna o valor da distância entre dois vetores. O hiperplano de separação que passa pelo ponto médio l mais próximo de x é definido como $H_l(x) = (x^T w_l - b_l)$, onde $b_l = [(1/2)(x_i + x_j)]w_l^T$ e $w_l = (x_i - x_j)$, sendo (x_i, x_j) o par de vértices que formam a Aresta de Suporte associada ao ponto médio l . O rótulo de x é dado pela função de classificação $f(x)$, descrita em 6. Esse processo pode ser visto de maneira mais detalhada pelo Algoritmo 2.

$$f(x) = \begin{cases} +1 & \text{se } H_l(x) > 0 \\ -1 & \text{se } H_l(x) \leq 0. \end{cases} \quad (6)$$

Algorithm 2 Classificação - CHIP-clas Reduzido

Entradas : conjunto de parâmetros B e W dos hiperplanos, conjunto de pontos médios das arestas de suporte P e conjunto de teste Xt .

Saída : classes do conjunto de teste CTe

```

for  $j$  in  $Xt$  do
  for  $i$  in  $P$  do
    Calcula a distância entre o novo padrão e os pontos médios
     $d(i) = D(Xt(j), P(i))$ 
  end for
  Hiperplano  $h_l$  que passa pelo ponto médio  $P_l$  mais próximo de  $Xt(j)$ 
   $h_l = \text{sign}(x^T \mathbf{w}_l - b_l)$ 
  if  $h_l > 0$  then
     $CTe(j) = +1$ 
  else
     $CTe(j) = -1$ 
  end if
end for
retorna  $CTe$ 

```

3 kNN

O k-vizinhos mais próximos, kNN (*k Nearest Neighbors*), é um classificador clássico baseado em memória, que é frequentemente usado em aplicações do mundo real devido a sua simplicidade. Apesar de simples, ele tem conseguido considerável exatidão na classificação em diversas aplicações e é conseqüentemente muito usado como uma base para comparação com novos classificadores [13].

O kNN possui duas fases, a primeira denominada de fase de treino simplesmente armazena todos as amostras de treinamento rotuladas. Na segunda fase, ou também chamada de fase de classificação, primeiramente calcula-se as distâncias de uma nova amostra a ser classificada com todos os padrões do conjunto de treinamento. Em seguida, o algoritmo considera os k padrões do conjunto de treinamento com as menores distâncias ao padrão a ser classificado. Por último, o novo padrão é classificado como pertencente a classe da maioria dos k padrões mais próximos do conjunto de treinamento.

Apesar da técnica ser muito simples, o custo computacional do kNN é elevado. Do ponto de vista do custo da memória, o kNN armazena todos os padrões de treinamento e quanto ao desempenho, para cada padrão a ser classificado o kNN deve calcular sua distância para todos os padrões de treinamento. A complexidade desse algoritmo é $O(mn)$ [10] por vetor de teste, onde m é a dimensão das amostras de treino e n o número de tais amostras.

4 Abordagem proposta

A metodologia proposta nesse artigo, denominada NN-clas, consiste em uma extensão do classificador de margem larga baseado em distância, porém, como dito anteriormente, com um menor custo computacional. A diferença da nova abordagem em relação ao CHIP-clas está na regra de decisão. O método tradicional utiliza uma mistura de hiperplanos para estimar a superfície de decisão, já o NN-clas baseia-se na regra de vizinhos mais próximos.

Assim como realizado no CHIP-clas, inicialmente, obtêm-se, no NN-clas, o Grafo de Gabriel G a partir de um conjunto de dados de treinamento $T = \{(x_i, y_i) | i = 1, \dots, N\}$, onde $y_i \in \{+1, -1\}$ e $x_i \in \mathbb{R}^n$, sendo o conjunto de vértices formado por todas as amostras de treinamento, ou seja, $V = \{x_i | i = 1, \dots, n\}$ e o conjunto de arestas A satisfazendo à condição: $D^2(p_i, p_j) \leq [D^2(p_i, z) + D^2(p_j, z)]$.

Em seguida, realiza-se a filtragem dos dados ruidosos, utilizando a técnica que foi devidamente descrita pela Equação 3. Após a eliminação dos dados ruidosos será encontrado o conjunto de vértices das arestas de suporte presentes na borda entre as classes.

Diferentemente da forma como é realizado no CHIP-clas, a classificação realizada no NN-clas calcula a distância entre cada amostra de teste x com todos os vértices das arestas de suporte, em seguida a nova amostra é classificada com o rótulo do vértice mais próximo. Essa fase de classificação é detalhada de maneira pormenorizada pelo algoritmo 3. Observa-se que essa técnica utilizada na regra de decisão é equivalente à aplicação do algoritmo kNN com $k = 1$, porém, considerando apenas o subconjunto de vértices de borda.

Percebe-se portanto, que esta nova metodologia elimina as operações mais custosas do CHIP-clas geradas pela Mistura Hierárquica de Especialistas, tais como os cálculos dos pesos de todos hiperplanos, definido pela Equação 4 e a soma ponderada dos hiperplanos $f(x)$. Facilitando assim, a implementação do NN-clas em hardware.

Algorithm 3 Classificação - NN-clas

Entradas : conjunto de vértices das arestas de suporte V , rótulos das arestas de suporte C e conjunto de teste Xt .

Saída : classes do conjunto de teste CTe

```
for  $j$  in  $Xt$  do
  for  $i$  in  $V$  do
    Calcula a distância entre o novo padrão e os vértices da aresta de suporte
     $d(i) = D(Xt(j), V(i))$ 
  end for
   $Cte(j) =$  rótulo  $C(l)$  do vértice mais próximo do padrão  $Xt(j)$ 
end for
retorna  $Cte$ 
```

5 Experimentos e Resultados

Com o intuito de se validar a metodologia aqui proposta e avaliar o desempenho da técnica por hiperplanos mais próximos foram realizados alguns experimentos, inicialmente com testes em dados sintéticos de *benchmarks* e no momento posterior com dados reais. Os testes realizados em dados sintéticos serviram para que se fosse feita uma comparação entre a superfície de decisão obtida pela metodologia proposta e pelo CHIP-clas. A Figura 4 mostra as superfícies de decisão das bases sintéticas. Os contornos exibidos pelas linhas tracejadas representam as superfícies de separação geradas pela nova abordagem. Já os contornos representados pelas linhas contínuas, as superfícies geradas pelo CHIP-clas.

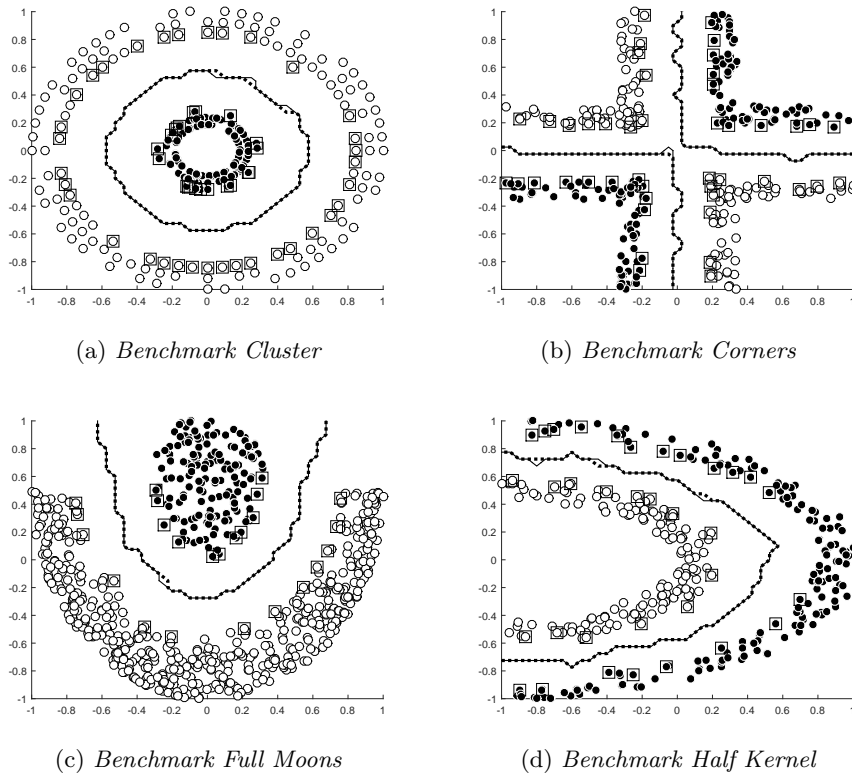


Figura 4: Superfícies de decisão geradas pelo método NN-clas e pelo CHIP-clas

Como se pode observar na Figura 4, alguns dados ruidosos foram eliminados para se garantir que os vértices das arestas de suporte estejam localizados na fronteira de separação entre as classes. Na figura, os pontos destacados com quadrados representam os tais vértices das arestas de suporte. Percebe-se, portanto

com o experimento que as superfícies de separação dos *benchmarks* geradas pelos dois métodos são similares, em alguns casos é quase imperceptível a diferença, apesar de ter havido uma redução do custo computacional no novo método. É importante ressaltar também que, embora o método proposto não se baseie em um princípio de margem larga, ele tende a gerar superfícies equidistantes das classes, inclinando-se assim para a maximização da margem.

Os demais experimentos foram realizados em 13 bases de dados reais retiradas do repositório (UCI - *Machine Learning Repository*) [9], e também em 2 problemas de expressão gênica: “Golub”[4] e “BcrHess”[6]. Primeiramente, realizou-se um pré-processamento, removendo-se as amostras que continham atributos faltantes e normalizando-se os dados entre $\{-1, 1\}$. Utilizou-se uma validação cruzada 10-folds, objetivando-se a garantia da relevância estatística. Escolheu-se, como medida de desempenho a AUC (área abaixo da curva ROC) por serem desbalanceadas algumas das bases que se pretende testar.

Os métodos comparados foram: o kNN com $k = 1$ (NN), o método aqui proposto (NN-clas), o CHIP-clas e o CHIP-Clas reduzido, nomeado nos experimentos de RCHIP-clas. Optou-se por realizar a comparação com kNN pela semelhança do NN-clas com tal método. Entretanto é importante ressaltar que o kNN calcula a distância entre cada amostra de teste x com todo o conjunto de treinamento, sendo necessário assim o armazenamento de todas as amostras de treinamento ao invés de guardar apenas os vértices das arestas de suporte como ocorre no NN-clas. Além disso, como o kNN não possui nenhuma fase de filtragem de ruído, a classificação de bases com sobreposição pode ser afetada pelo *overfitting*. Uma forma de minimizar o *overfitting* seria através de uma busca exaustiva pelo melhor k , contudo essa prática é inviável para sistemas embarcados.

A Tabela 1 mostra a AUC média e o desvio padrão obtidos pelos métodos em cada base de dados. Nela, também são apresentados o número total de amostra N , a dimensionalidade N_d , o número de amostras de treinamento N_t , o número de atributos de cada base de dados N_a e o número total dos vértices das arestas de suporte N_b , com o intuito de se exemplificar a redução de armazenamento de dados exigido pelo método proposto em relação ao kNN, bem como mostrar a complexidade de cada base de dados. Os valores destacados em negrito demonstram o método que obteve o melhor desempenho em cada base de dados.

Para realizar uma avaliação estatística dos resultados obtidos pelos classificadores, utilizou-se o teste estatístico de *Friedman*, o qual é considerado por [1] o teste mais indicado para se comparar mais de dois métodos. O teste de *Friedman* cria um ranqueamento entre os métodos para cada base de dados. A última linha da Tabela 1 mostra o *rank* médio obtido por cada um dos classificadores. É importante ressaltar que quanto menor o valor do *rank* melhor o desempenho do algoritmo. Assumiu-se como hipótese nula H_0 a equivalência entre os quatro métodos com um nível de significância de $\alpha = 0.05$. O valor p obtido com esse teste foi de 0.04, ou seja, não é possível afirmar que os métodos sejam equivalentes estatisticamente.

Tabela 1: Resultados dos algoritmos

Base de dados	CHIP-clas	NN-clas	RCHIP-clas	NN	N	N_d	N_t	N_a	N_b
Australian Cr.	0.846 ± 0.041	0.848 ± 0.041	0.846 ± 0.04	0.790 ± 0.047	690	14	395	14	62
Banknote Auth.	0.980 ± 0.026	0.999 ± 0.003	0.979 ± 0.028	0.998 ± 0.003	1372	4	1235	4	61
BcrHess	0.811 ± 0.117	0.814 ± 0.125	0.807 ± 0.123	0.714 ± 0.228	133	30	71	30	6
B. Cancer W.P	0.956 ± 0.029	0.964 ± 0.028	0.963 ± 0.026	0.949 ± 0.031	683	9	520	9	20
Climate M.S.C.	0.840 ± 0.068	0.771 ± 0.129	0.768 ± 0.131	0.594 ± 0.088	540	18	274	18	199
Fertility	0.588 ± 0.259	0.553 ± 0.162	0.559 ± 0.163	0.578 ± 0.224	100	9	48	9	9
German Cr.	0.672 ± 0.042	0.685 ± 0.033	0.668 ± 0.043	0.605 ± 0.064	1000	24	463	24	263
Golub	0.774 ± 0.172	0.788 ± 0.160	0.784 ± 0.160	0.789 ± 0.086	72	50	39	50	9
Haberman's S.	0.569 ± 0.089	0.559 ± 0.108	0.579 ± 0.084	0.539 ± 0.082	306	3	143	3	49
ILPD	0.560 ± 0.086	0.568 ± 0.087	0.559 ± 0.090	0.595 ± 0.072	579	10	275	10	117
Liver disorders	0.614 ± 0.102	0.598 ± 0.114	0.566 ± 0.124	0.597 ± 0.079	345	6	164	6	110
P. ind. diabetes	0.721 ± 0.036	0.728 ± 0.046	0.716 ± 0.049	0.672 ± 0.058	768	8	392	8	124
Parkinsons	0.898 ± 0.149	0.894 ± 0.146	0.895 ± 0.147	0.945 ± 0.125	195	22	175	22	62
Sonar. M against R.	0.876 ± 0.079	0.872 ± 0.061	0.846 ± 0.067	0.872 ± 0.061	208	60	187	60	163
Stalog heart	0.798 ± 0.082	0.788 ± 0.071	0.784 ± 0.160	0.789 ± 0.086	270	13	134	13	36
Média do rank $R(\mathcal{L})$	2.0333	2.0333	3.1000	2.8333					

Como o teste de *Friedman* demonstrou que os métodos não são equivalentes estaticamente, um teste *post-hoc* pareado (*Wilcoxon*) foi realizado para verificar significância estatística entre cada par de algoritmos. Para todos os testes utilizou-se como hipótese nula H_0 a equivalência entre os métodos comparados com um nível de significância de $\alpha = 0.05$. A Tabela 2 mostra o p -valor obtido em cada um dos testes realizados. Percebe-se que o NN-clas e o CHIP-clas são equivalentes estatisticamente, já o RCHIP-clas possui desempenho inferior ao CHIP-clas.

Tabela 2: Resultados dos teste de Wilcoxon

Métodos	NN-clas	RCHIP-clas
CHIP-clas	0.476	0.036

6 Conclusão

Nesse artigo, foram avaliadas duas técnicas para redução das operações mais complexas utilizadas na regra de decisão do método CHIP-clas [14]. Os resultados mostraram que a redução obtida através do NN-clas não gera perdas significativas no desempenho do método, tornando-o ainda mais atrativo para implementação em sistemas embarcados. Já abordagem utilizando a formulação do CHIP-clas reduzido, apesar de conseguir reduzir as referidas operações complexas, possui perdas consideráveis de desempenho.

Também foi realizada uma comparação do NN-clas com o kNN com $k = 1$, uma vez que essa nova metodologia baseia-se nesse algoritmo. Com essa comparação foi possível observar que, apesar do kNN também ser um método factível de ser implementado em sistemas embarcados, ele exige um maior consumo de memória, pelo fato de armazenar todo o conjunto de treinamento, além de apresentar um desempenho inferior ao do NN-clas.

Agradecimentos

O presente trabalho foi realizado com o apoio financeiro da FAPEMIG, CNPQ e CAPES - Brasil.

Referências

1. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research* 7(Jan), 1–30 (2006)
2. Gabriel, K.R., Sokal, R.R.: A new statistical approach to geographic variation analysis. *Systematic Biology* 18(3), 259–278 (1969)
3. Garcia, L.P., de Carvalho, A.C., Lorena, A.C.: Effect of label noise in the complexity of classification problems. *Neurocomputing* 160, 108–119 (2015)
4. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science* 286(5439), 531–537 (1999)
5. He, M., Klein, J.O., Belhaire, E.: Design and electrical simulation of on-chip neural learning based on nanocomponents. *Electronics Letters* 44(9), 575–576 (2008)
6. Hess, K.R., Anderson, K., Symmans, W.F., Valero, V., Ibrahim, N., Mejia, J.A., Booser, D., Theriault, R.L., Buzdar, A.U., Dempsey, P.J., et al.: Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of clinical oncology* 24(26), 4236–4244 (2006)
7. Holanda, P.C.: Desenvolvimento de uma Rede Neural Multilayer Perceptron Embarcada em FPGA. Master's thesis, Universidade Federal do Ceará (2013)
8. Holler, M., Tam, S., Castro, H., Benson, R.: An electrically trainable artificial neural network (etann) with 10240 floating gate synapses. In: *International Joint Conference on Neural Networks*. vol. 2, pp. 191–196 (1989)
9. Lichman, M.: UCI machine learning repository (2013), <http://archive.ics.uci.edu/ml>
10. Manolakos, E.S., Stamoulias, I.: Ip-cores design for the knn classifier. In: *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*. pp. 4133–4136. IEEE (2010)
11. Mohammed, E.Z., Ali, H.K.: Hardware implementation of artificial neural network using field programmable gate array. *International Journal of Computer Theory and Engineering* 5(5), 780 (2013)
12. Muthuramalingam, A., Himavathi, S., Srinivasan, E.: Neural network implementation using fpga: issues and application. *International journal of information technology* 4(2), 86–92 (2008)
13. da Silva Júnior, E.R.: Investigação de Técnicas de Extração e Seleção de Características e Classificadores Aplicados ao Problema de Classificação de Dígitos Manuscritos de Documentos Históricos. Master's thesis, Universidade de Pernambuco (2007)
14. Torres, L., Castro, C., Coelho, F., Torres, F.S., Braga, A.: Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters* 51(24), 1967–1969 (2015)
15. Torres, L.C.B.: Classificador por arestas de suporte (CLAS): Métodos de aprendizado baseados em grafos de Gabriel. Tese de doutorado, UFMG (2016)