

Uma arquitetura para combinação de classificadores otimizada por métodos de poda com aplicação em *credit scoring*

Luiz Vieira e Silva Filho

Centro de Informática

Universidade Federal de Pernambuco

Recife - PE, Brasil

Email: lvsf@cin.ufpe.br

George D. C. Cavalcanti

Centro de Informática

Universidade Federal de Pernambuco

Recife - PE, Brasil

Email: gdcc@cin.ufpe.br

Resumo—Sistemas de Múltiplos Classificadores (*Multiple Classifiers Systems - MCS*) se baseiam na ideia de que combinar a opinião de vários especialistas pode produzir melhores resultados do que quando se usa apenas um especialista. Diversas técnicas de MCS foram desenvolvidas, apresentando pontos fortes e fracos, a depender do contexto em que são aplicadas. Este trabalho propõe uma arquitetura para MCS que visa potencializar a complementaridade entre essas técnicas, possuindo dois objetivos principais: i) a combinação de métodos de amostragem tradicionais, visando a geração de classificadores de melhor desempenho que componham um *pool* de classificadores; ii) a aplicação de um algoritmo de poda para remover do *pool* aqueles classificadores incompetentes para lidar com o problema em questão, considerando os critérios de seleção adotados. A arquitetura proposta foi avaliada em uma aplicação de *credit-scoring*. Os métodos de amostragem usados foram o *Bagging* e o *Random Subspace* com classificadores-base sendo árvores-de-decisão, construídas com base no algoritmo CART. Para o processamento da poda foi usado o algoritmo *Orientation Ordering*, e para combinação das saídas dos classificadores do *ensemble* adotou-se o método *Majority Vote*. Os experimentos realizados mostraram que a arquitetura proposta alcançou taxas de acerto similares ou superiores às atingidas pelos métodos apresentados na literatura. Esses resultados ainda foram obtidos com *ensembles* cujos tamanhos eram da ordem de 20% dos *pools* originais gerados na fase de treinamento.

I. INTRODUÇÃO

Sistemas de Múltiplos Classificadores (*Multiple Classifiers Systems - MCS*), também chamados de *ensembles* ([1];[2]), vêm sendo estudados desde a segunda metade do século XX, quando, em 1959, foi construído o primeiro modelo de sistemas com múltiplos especialistas, a arquitetura *Pandemonium* [3]. A proposta de combinar classificadores parte do conceito intuitivo de que, na maioria das vezes, conjugar a opinião de diversos especialistas em uma área do conhecimento tende a produzir uma melhor decisão do que a gerada por um único especialista.

É natural supor ainda que combinar classificadores precisos possibilitará a criação de um *ensemble* de melhor desempenho. Contudo, a diversidade entre os classificadores também merece ser considerada nos MCS [2], pois combinar classificadores que cometem os mesmos erros pouco acrescenta ao desempenho do

ensemble ([4];[5]). Logo, quanto menor a correlação entre os classificadores em relação aos erros cometidos mais preciso o *ensemble* tende a ser [6].

Uma estratégia para promover diversidade em MCS é usar classificadores considerados instáveis. Um classificador instável é aquele que pode sofrer grandes variações em suas saídas, a partir de pequenas alterações nos dados usados no seu treinamento, por exemplo: árvores de decisão e redes neurais artificiais (RNA) [1].

Existem na literatura métodos muito utilizados para a construção de MCS, tais como: *Bagging*[7], *Boosting*[8], *Adaboost*[9], *Random Subspace*[10], *Rotation Forest*[11], *Error Correcting Output Coding(ECOC)*[12]), que utilizam diferentes estratégias para classificadores precisos e diversos.

Visando ainda mais diversidade e desempenho em MCS, Wang et al. [13] e Marqués et al. [14] apresentaram métodos de amostragem que denominaram de **estratégia dual de treinamento e treinamento em 2 níveis**, respectivamente. Ambas as técnicas partem da ideia de criar MCS a partir da conjugação de métodos tradicionais de amostragem.

As pesquisas acima são reforçadas pelo teorema **"No Free Lunch"** [15], que afirma não existir um único classificador que possa ser considerado ótimo para todos os problemas. Ou seja, o desempenho elevado em uma classe de problemas é compensado com um mal desempenho em outra classe.

Por sua vez, a cada novo classificador agregado ao sistema, é esperado que sua taxa de erro decresça monotonicamente, tendendo a um nível constante ([16], [17]). Contudo, quanto maior o *ensemble*, maior a necessidade de memória e tempo de processamento [18]. Esses dois aspectos serão aqui referenciados como **custo computacional**.

Para lidar com o custo computacional, foi proposta a abordagem de **poda de ensembles** [18]. A ideia da poda de *ensembles* passa por descartar classificadores que ajam em detrimento do desempenho do *ensemble*, e/ou estejam produzindo informação redundante [19]. Portanto, o processo de poda de um *ensemble* trata de, dado um *ensemble* L , encontrar um *subensemble* S , $S \subset L$. S deve conter classificadores com a precisão desejada para a aplicação em questão, preservando a

diversidade entre os mesmos, mantendo, ou superando, a precisão do *ensemble* L .

Todavia, essa busca se apresenta como um problema *NP-completo* [19], devido ao espaço de busca ser igual a $2^T - 1$, sendo T é a dimensão do MCS. Pesquisas vêm sendo feitas visando identificar heurísticas que tornem factíveis a escolha de um *subensemble* menor e mais preciso do que o *ensemble* completo ([16]; [17]; [19]; [20]; [21]; [22]; [23]; [24]).

As abordagens para encontrar esses *subensembles* são agrupadas em duas grandes categorias [23]: (1) as que tratam essa busca como um problema de otimização matemática; (2) as que reordenam os classificadores do *ensemble* original, baseando-se em algum critério predefinido (*ranking* [20]). Nesta segunda categoria encontram-se métodos como: *Orientation Ordering* (OO) [19] e o *EPIC - Ensemble Pruning via Individual Contribution Ordering* (EPIC) [23].

Buscando aprimorar técnicas para construção de *ensembles*, esse trabalho tem por objetivo propor uma arquitetura para MCS que: i) combine estratégias de amostragem de dados, tais como *Bagging* e *Random Subspace*, com o intuito de gerar classificadores que irão compor um *pool* de classificadores - deseja-se que esses classificadores sejam diversos e precisos; e em seguida, ii) remova classificadores do *pool* de classificadores usando um método estruturado de poda de *ensembles*, para que o *ensemble* final formado seja mais eficiente e preciso que o *pool* originalmente gerado.

A avaliação da arquitetura proposta foi realizada numa aplicação de *credit-scoring*, exatamente pelos MCS estarem sendo usados em problemas cujos dados tenham elevados níveis de ruído (*noisy data*) e redundância de atributos ([13]; [25]). Ainda assim, espera-se que a arquitetura proposta seja aplicável na solução de outros problemas de classificação de padrões.

O documento está dividido da seguinte forma: na Seção II é apresentada a arquitetura proposta; na Seção III são detalhados os experimentos e os resultados alcançados; e, na Seção IV estão as conclusões e contribuições desta pesquisa.

II. ARQUITETURA PROPOSTA

A arquitetura proposta possui 3 (três) fases - Treinamento, Poda e Teste (Fig. 1). Na primeira fase é gerado um *pool* de classificadores a partir de uma estratégia de treinamento *dual*, ou em 2 níveis, induzindo uma maior diversidade no *ensemble*. Na fase seguinte, o *pool* é submetido a um processo de poda, adotando-se o conceito de agregação ordenada dos classificadores [16], obtendo-se um *subensemble* do *pool*, cujo desempenho espera-se que seja igual ou superior ao do *pool*, sendo mais eficiente quanto ao custo computacional. Na fase de teste o *subensemble* é avaliado usando-se um método de combinação baseado na fusão dos classificadores, onde todos opinam sobre cada instância.

Os arquivos DB_1 , DB_2 e DB_3 usados como entradas nas 3 fases são subconjuntos disjuntos de uma mesma

base de dados. Todos os arquivos, embora de tamanhos diferentes, preservam a mesma proporcionalidade em relação às classes existentes na base de dados completa.

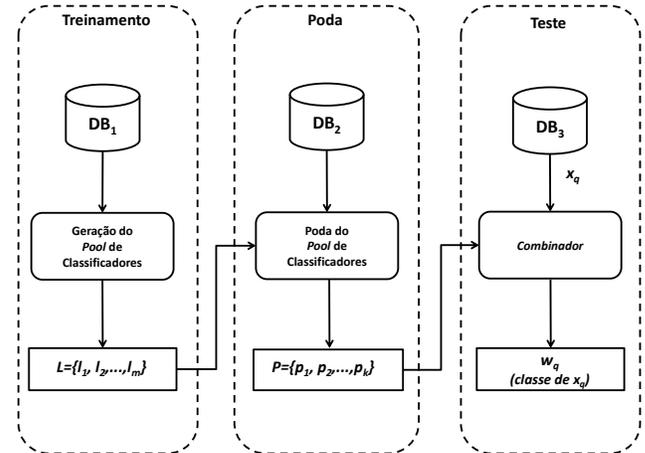


Figura 1: **Visão geral da arquitetura proposta** composta por 3(três) fases: Treinamento do *pool* de classificadores L ; Poda de L com seleção de P , $P \subset L$; Teste de P usando método de combinação não-ponderada de classificadores.

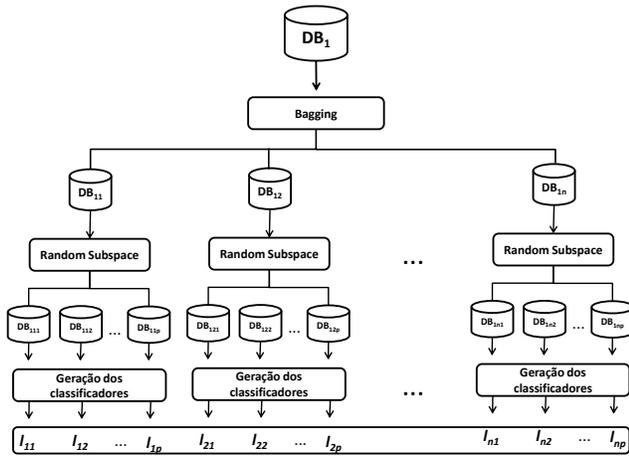
A. Treinamento do *pool* de classificadores

Nesta fase é gerado o *pool* inicial de classificadores $L = \{l_1, \dots, l_m\}$, m indica o total de classificadores. Cada um dos classificadores-membro é treinado tendo como entrada uma base de dados DB_1 , sendo submetido a um método de treinamento baseado numa estratégia *dual*, ou em 2 níveis. Como tratamento dos dados pode ser feito em duas sequências diferentes: i) primeiro reduz os efeitos dos ruídos (*Bagging*) e depois reduz a redundância dos atributos dos dados (*Random Subspace*); ou, ii) na ordem inversa, existem duas possibilidades para aplicar a estratégia *dual*: *Bagging-RS* e o *RS-Bagging* [13].

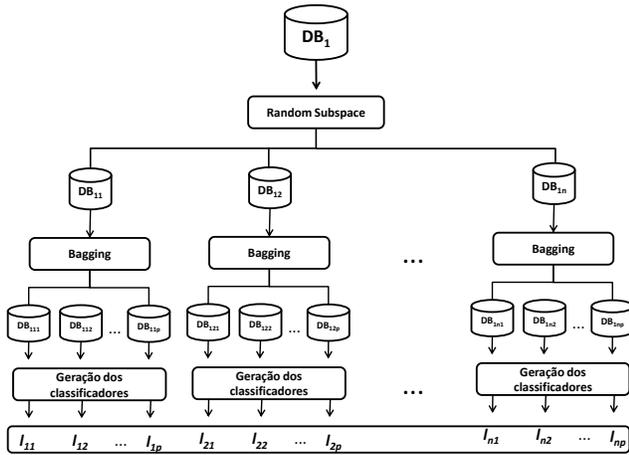
Portanto, para cada uma das sequências da estratégia *dual*, são informados na fase de treinamento os seguintes parâmetros de entrada: a) números de iterações n e p que cada um dos métodos de amostragem serão processados. Logo, a quantidade de classificadores de L é dado por $m = n \times p$; b) taxa de seleção usada pelo *Random Subspace* na redução da dimensionalidade das características dos padrões de entradas, visando tratar eventual redundância nos dados.

Conforme pode ser visto na Fig. 2, o primeiro método da sequência processará a base de dados DB_1 produzindo subamostras DB_{1_i} , $i = 1 \dots n$. A partir de cada subamostra DB_{1_i} , o segundo método produzirá p subamostras $DB_{1_{ij}}$, $j = 1 \dots p$, que serão utilizadas no treinamento dos classificadores. Ao final, $n \times p$ classificadores estarão treinados, compondo um único *pool* de classificadores $L = \{l_1, \dots, l_m\}$.

A diferença básica entre as técnicas de amostragem propostas por Wang et al. [13] e Marqués et al. [14], está na quantidade de *ensembles* envolvidos na classificação final da instância. A primeira usa n *subensembles* com p classificadores, e a segunda usa apenas um *ensemble* com $n \times p$ classificadores. Além disso, ao contrário da abordagem proposta, nenhuma das duas técnicas anteriores adotou procedimentos de poda de *ensembles*.



(a) Bagging-RS



(b) RS-Bagging

Figura 2: Estratégias *dual* de treinamento (ou treinamento em 2 níveis) para geração do Pool de Classificadores inicial

A arquitetura proposta baseou-se em [14], pois, como na fase de poda e processamento se dará sobre um único *pool* de classificadores, o uso de vários *ensembles* tornaria a poda mais custosa.

B. Poda do pool de classificadores

A partir da base de dados DB_2 o *pool* completo L é podado conforme método de agregação ordenada – *Orientation Ordering* (OO) [19]. Esse processo produz

o *ensemble* $P = \{p_1, \dots, p_q\}$, $P \subset L$, sendo q igual à quantidade de classificadores selecionados pelo OO. Espera-se que P alcance um desempenho igual ou superior a L , a um menor custo computacional.

O algoritmo *Orientation-Ordering* (OO) é um método de poda baseado em *ranking* [20]. A estratégia adotada pelo OO é a de modificar a agregação original dos classificadores no *ensemble* completo, estabelecendo uma ordem de importância desses classificadores quanto à contribuição dos mesmos na otimização do desempenho do *ensemble* final. Esta abordagem tem como premissa ser possível se construir, de forma incremental, *subensembles* cada vez mais eficientes, agregando um classificador de cada vez, conforme um determinado critério de contribuição do classificador para o desempenho do *ensemble* [19]. Após a ordenação, uma fração desses classificadores é selecionada, de acordo com um limiar estabelecido, para então formar o *ensemble* podado P .

O método OO encontrou *subensembles* com quantidades intermediárias de classificadores, variando entre 15% e 30% do *ensemble* original, com taxas de erro de generalização abaixo do valor da constante assintótica (v. Seção I), obtida no caso do *ensemble* cujos classificadores foram agregados aleatoriamente pelo *bagging* [19].

C. Teste do ensemble final

Na terceira fase da arquitetura proposta são feitos os testes no *ensemble* P , produzido na fase de poda. Na fase de teste são apresentados novos padrões (conjunto de dados DB_3) a fim de se avaliar a capacidade de generalização do *ensemble* produzido. É utilizado o *majority vote* [4] como método de combinação das previsões individuais dos classificadores de P .

III. EXPERIMENTOS E RESULTADOS

A eficácia da arquitetura proposta foi avaliada sobre as seguintes bases de dados de *credit scoring*: *Australian* e *German*, ambas disponíveis no *UCI Machine Learning Repository* [26]. A base *Australian* possui 14 atributos, 690 instâncias, sendo 307 (44,5%) de bons clientes (avaliação de crédito positiva), e 383 (55,5%) de maus clientes (avaliação de crédito negativa). A base *German* possui 24 atributos, 1000 instâncias, das quais 70% são de bons clientes e 30% de maus clientes.

Nos experimentos adotou-se o método *10-fold Cross-Validation* (*10-fold CV*). As bases foram particionadas em 10 (*folds*) de mesma dimensão, com a mesma proporção entre as classes. Em seguida essas partes foram reagrupadas em 10 formações diferentes de 3 blocos na regra 7-2-1: a) 7 *folds* usados no treinamento do *ensemble*; b) 2 *folds* na poda; e, c) 1 *fold* no teste. Para cada uma dessas 10 formações, foram avaliadas 189 configurações diferentes. Essa parametrização é resultante da combinação dos 21 cenários da Tab. I com 9 dimensões do *pool* original. Em cada uma das

configurações foram treinados, podados e testados 10 *ensembles*, sendo calculada uma taxa média de acerto por configuração. Essas taxas foram usadas para avaliar o desempenho da arquitetura proposta em relação ao uso de métodos encontrados na literatura.

Foram usados 9 tamanhos diferentes de *pool* de classificadores, todos eles gerados a partir da mesma quantidade de iterações realizadas pelo *Bagging* como pelo *RS*, ou seja $n = p$. Os tamanhos foram os seguintes: 25, 49, 81, 121, 169, 225, 289, 361 e 441.

Em [19] foi constatada a eficácia do algoritmo do OO em *ensembles* com quantidades entre 100 e 200 classificadores do tipo árvores de decisão, inclusive sobre as bases usadas nestes experimentos. Por isso, na avaliação desta arquitetura foi arbitrado um intervalo de tamanhos de *ensembles* que possibilitasse, com razoável margem de segurança, que o OO realizasse todo seu potencial. Por sua vez, no treinamento em 2 níveis, visando reduzir a eventual predominância de um método de treinamento sobre o outro, adotou-se o mesmo número de iterações para ambos os métodos usados nessa fase.

Conforme apresentado na Tab. I, os 21 cenários avaliados foram definidos a partir de 3 parâmetros: i) **Estratégia de amostragem de dados** adotada na geração do *pool* na fase de treinamento; ii) **Taxa de seleção de atributos** adotada quando do uso do algoritmo *RS*; iii) **Método usado na poda** que define se, e como, o *pool* original será podado.

Estratégia de Treinamento	Taxa de Seleção de Atributos	Métodos de Poda	Cenários Possíveis
RS-Bagging ou Bagging-RS	0,5; 0,6; 0,7	OO; <i>Single Best</i> ; Sem Poda	18
Bagging	1,0 (sem RS)	OO; <i>Single Best</i> ; Sem Poda	3

Tabela I: Parâmetros informados na configuração dos cenários dos experimentos.

O classificador base do *ensemble* foi a árvore de decisão, tanto pelo seu uso frequente em MCS [4], como pela sua instabilidade. As árvores de decisão foram construídas com base no algoritmo CART.

Os resultados da arquitetura proposta foram analisados em função dos parâmetros de entrada, além de comparados com outros métodos de geração de *ensembles*, como: i) amostragem usando apenas o *Bagging* (1 nível), associada ou não à poda do *ensemble*, ou ainda usando o conceito de *Single Best* - seleção do melhor classificador após ordenação do OO; ii) amostragem em 2 níveis desassociada da poda, ou usando o *single Best*.

De acordo com a Fig. 3, a arquitetura proposta apresentou melhores taxas de acerto que as demais configurações, principalmente quando na sequência *Bagging-RS*, e ao se adotar uma taxa de seleção igual a 0,5. Isso vem ao encontro do apresentado em [10] sobre melhores resultados alcançados com árvores de decisão, quando construídas a partir de espaços de características com metade da dimensão do espaço

total. Quanto ao tamanho do *ensemble*, observou-se uma maior estabilidade na taxa de acerto, na medida em que o tamanho do *ensemble* original aumenta. Nas demais configurações testadas, não se constata o mesmo nível de estabilidade obtida pela arquitetura proposta. A Taxa Média de Poda (TMP) alcançada em ambas as sequências de treinamento, para as duas bases de dados, reduziu o *Pool L* a cerca de 20% do seu tamanho.

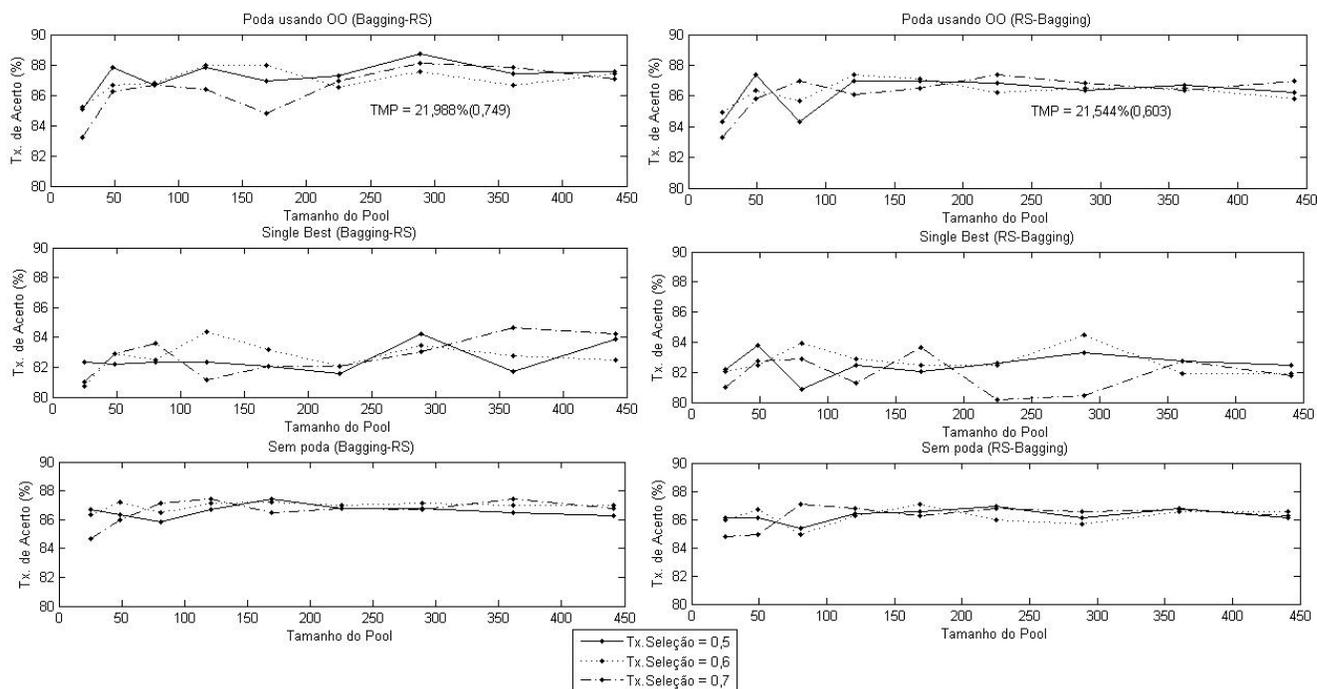
Observou-se que o desempenho da configuração usando apenas o *Bagging* durante o treinamento, ou seja, treinamento em 1 nível, não associado a um método de poda, apresenta taxas de acerto ligeiramente superiores em relação às demais configurações, inclusive a que usa o OO como método de poda. É possível inferir que sem a extração da redundância dos atributos, o algoritmo de poda do OO não contribui tanto para o desempenho. Contudo, quando usado o OO, a taxa média de poda (TMP) se mantém no patamar de 20%, o que significa que apesar de taxas de acerto menores, há uma redução no custo computacional. O desempenho das configurações que utilizaram o conceito do *Single Best* ficaram bastante abaixo dos resultados das demais configurações que usaram *ensembles*.

As observações acima estão resumidas na Tab. II, na qual encontram-se detalhadas os valores das taxas médias para cada um dos 21 cenários possíveis de configuração. Essas taxas médias foram calculadas a partir dos resultados das 189 combinações de configuração possíveis, agrupadas por cenário, sem considerar o tamanho do *pool*.

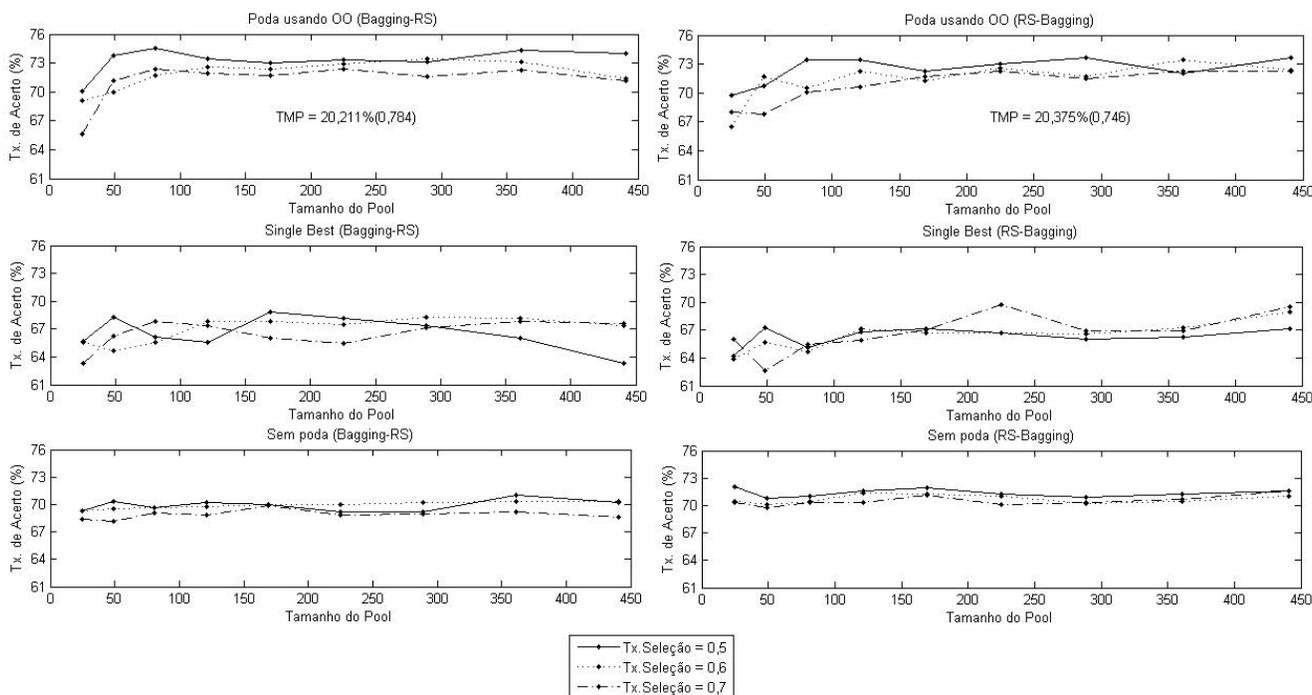
Na sequência, foram avaliados os 10 melhores resultados obtidos, para cada estratégia *dual* treinamento (*Bagging-RS* e *RS-Bagging*). Observou-se que 85% dos 10 melhores resultados alcançados na base *Australian* e 100% da base *German* foram obtidos pela arquitetura proposta.

Na base *Australian* a arquitetura proposta teve como seu melhor resultado uma taxa de acerto de $88,696 \pm 4,573$, usando a estratégia *dual RS-Bagging*, taxa de seleção de 0,5 podando o *ensemble* a cerca de 22% do original, cujo tamanho era de 289 classificadores. Na base *German* sua melhor taxa de acerto foi de $74,5 \pm 2,014$, também na estratégia *dual RS-Bagging*, taxa de seleção de 0,5, e uma poda em torno de 20%, em um *ensemble* com 81 classificadores.

Na Tab. III são apresentadas as médias dos 10 melhores resultados de todas as configurações, considerando as estratégias usadas na fase de treinamento (em 1 ou 2 níveis), e os métodos usados na fase de poda. Quando a fase de poda é considerada **ativada** significa que foram considerados para entrar no cálculo os resultados obtidos por qualquer uma das 3 possibilidades de parametrização da fase de poda: OO, *Single Best*, Sem Poda, desde que alguns destes estejam entre os 10 melhores. Quando a fase de poda está como **desativada** significa que foram considerados apenas os 10 melhores resultados da parametrização sem poda. Na escolha desses resultados para cálculo da taxa média não foram considerados como filtro, nem as taxas de seleção nem



(a) Base de dados *Australian*



(b) Base de dados *German*

Figura 3: Desempenho da arquitetura em função do tamanho do *pool* original de classificadores e da taxa de seleção de características. Os gráficos variam quanto: à estratégia *dual* adotada e à taxa de seleção adotada na fase de treinamento; e, ao método usado na fase de poda. A taxa média de poda (TMP) indica para quanto foi reduzido o tamanho do *Pool* original.

o tamanho do *pool* de classificadores.

Nota-se que entre os resultados apresentados na Tab. III, as melhores taxas de acerto são aquelas alcançadas

pela arquitetura proposta, quando usada a sequência Bagging-RS em ambas as bases. Aplicou-se então o *t-Student* e constatou-se que as duas melhores taxas

Fase de Poda (Método)	Treinamento em 1 nível (Bagging)	Treinamento em 2 níveis					
		Bagging - RS (Tx. de seleção)			RS - Bagging (Tx. de seleção)		
		0,5	0,6	0,7	0,5	0,6	0,7
Poda com OO	84,742 ± 1,552	87,246 ± 1,001	86,973 ± 0,865	86,360 ± 1,533	86,232 ± 1,120	86,280 ± 0,750	86,248 ± 1,196
Single Best	82,955 ± 1,364	85,512 ± 0,917	82,705 ± 1,004	82,738 ± 1,281	82,496 ± 0,819	82,711 ± 0,913	81,851 ± 1,214
Sem Poda	85,974 ± 0,557	86,586 ± 0,441	86,940 ± 0,303	85,586 ± 0,858	86,264 ± 0,474	86,167 ± 0,640	86,232 ± 0,829

(a) Base de dados *Australian*

Fase de Poda (Método)	Treinamento em 1 nível (Bagging)	Treinamento em 2 níveis					
		Bagging - RS (Tx. de seleção)			RS - Bagging (Tx. de seleção)		
		0,5	0,6	0,7	0,5	0,6	0,7
Poda com OO	66,716 ± 2,383	73,277 ± 1,301	71,844 ± 1,464	71,144 ± 2,130	72,422 ± 1,419	71,377 ± 2,015	70,733 ± 1,781
Single Best	63,766 ± 1,220	66,600 ± 1,740	66,989 ± 1,324	66,522 ± 1,486	66,300 ± 1,049	66,411 ± 1,496	66,667 ± 2,132
Sem Poda	68,094 ± 1,205	69,900 ± 0,606	69,889 ± 0,362	68,900 ± 0,497	71,378 ± 0,452	70,689 ± 0,468	70,500 ± 0,550

(b) Base de dados *German*Tabela II: Taxas médias de acerto (%) calculadas a partir das taxas de acerto alcançadas pelos *ensembles* de tamanho variado para cada um dos 21 cenários avaliados nos experimentos

Fase de Treinamento	Estratégia de Treinamento	Fase de Poda	Método Predominante na Fase de Poda	Tx. Média de Acerto(%)	Tx. Média de Poda(%)
2 Níveis	Bagging-RS	Ativada	100% OO	87,870 ± 0,368	22,258 ± 0,499
2 Níveis	Bagging-RS	Desativada	N/A	87,188 ± 0,170 ⁽⁻⁾	N/A
2 Níveis	RS-Bagging	Ativada	70% OO	87,130 ± 0,191 ⁽⁻⁾	21,547 ± 0,739
2 Níveis	RS-Bagging	Desativada	N/A	86,797 ± 0,210 ⁽⁻⁾	N/A
1 Nível	Bagging	Ativada	60% Sem Poda	86,507 ± 0,241 ⁽⁻⁾	20,391 ± 0,137
1 Nível	Bagging	Desativada	N/A	86,362 ± 0,383 ⁽⁻⁾	N/A

(a) Base de dados *Australian*

Fase de Treinamento	Estratégia de Treinamento	Fase de Poda	Método Predominante na Fase de Poda	Tx. Média de Acerto(%)	Tx. Média de Poda(%)
2 Níveis	Bagging-RS	Ativada	100% OO	73,590 ± 0,530	20,736 ± 0,489
2 Níveis	Bagging-RS	Desativada	N/A	70,250 ± 0,292 ⁽⁻⁾	N/A
2 Níveis	RS-Bagging	Ativada	100% OO	73,030 ± 0,058 ⁽⁻⁾	20,540 ± 0,723
2 Níveis	RS-Bagging	Desativada	N/A	71,500 ± 0,323 ⁽⁻⁾	N/A
1 Nível	Bagging	Ativada	60% Sem Poda	69,160 ± 0,276 ⁽⁻⁾	21,978 ± 1,727
1 Nível	Bagging	Desativada	N/A	68,950 ± 0,450 ⁽⁻⁾	N/A

(b) Base de dados *German*Tabela III: Taxas médias de acerto dos 10 melhores resultados obtidos em cada uma das configurações utilizadas pela arquitetura, usando na fase de treinamento estratégias em 1 ou 2 níveis, independentemente da taxa de seleção utilizada no *Random Subspace* e do tamanho do *pool* de classificadores, variando na fase de poda o método utilizado, isto é: OO, *Single Best*, Sem poda. Em negrito, a taxa de acerto usada como base de comparação para o teste estatístico (*t-Student*). O símbolo ⁽⁻⁾ indica que cada uma dessas taxas é significativamente menor do que a taxa em negrito.

médias de acerto, por base, eram significativamente superiores às das demais configurações.

As melhores taxas médias de acerto alcançadas pelo *Single Best* foram 82,652 ± 1,041 para a base *Australian* e 66,704 ± 1,482 para a base *German*, ambas muito abaixo das apresentadas na Tab. III.

Os resultados foram comparados com os apresentados por [13] e [19]. Apesar das diferenças metodológicas nos experimentos e na apresentação dos resultados, esta comparação pode servir como mais um referencial para

avaliação da eficiência da arquitetura proposta. Ambos os trabalhos usaram as bases *Australian* e *German*.

A melhor taxa de acerto apresentada por [13] para a base de dados *Australian* foi 88,01%, obtida pela estratégia *Bagging-RS*, com taxa de seleção de 0,7, e *ensemble* com 150 classificadores. Para a base *German* as estratégias e taxa de seleção foram as mesmas, porém com taxa de acerto de 78,52% e 100 classificadores por *ensemble*. Em [19] os resultados foram apresentados na forma de taxa de erro. A menor taxa de erro obtida na

base *Australian* foi de $14,1\% \pm 2,2$, o que corresponderia a um acerto próximo de $85,9\%$. A poda reduziu o *ensemble* a 19% da sua dimensão original. Para a base *German* a menor taxa de erro foi de $25,4\% \pm 1,7\%$, correspondendo a um acerto em torno de $74,6\%$. A poda reduziu o *ensemble* a 16,5% do seu tamanho original. Para ambas as bases, o *ensemble* original continha 200 classificadores.

A arquitetura proposta alcançou para a base *Australian* taxas de acerto superiores ou similares a ambos os trabalhos acima. Na comparação específica com os resultados de [13], é importante observar ainda que: i) as taxas da obtidas pela arquitetura proposta foram alcançadas por *ensembles* com 40% do tamanho do *ensemble* de melhor resultado de [13]; ii) os melhores resultados também aparecem na sequência *Bagging-RS*. Na comparação com os resultados de [19], observa-se que as reduções na dimensão dos *ensembles* decorrente da poda encontram-se na mesma ordem de grandeza, próximas a 20%.

Para a base de dados *German*, considerando o desvio padrão ali apresentado, as taxas de acerto de [13] aparecem ligeiramente superiores às obtidas pela arquitetura proposta. Contudo, esta última alcançou seus melhores resultados com *ensembles* de tamanho 50% menor do que o tamanho do *ensemble* de melhor resultado de [13]. Na comparação com os resultados de [19], as taxas de acerto praticamente se equivalem, com um pequena superioridade da taxa de poda alcançada por [19].

IV. CONCLUSÕES

Foi proposto nesse artigo uma arquitetura para MCS cujos objetivos são: i) gerar um *pool* de classificadores a partir de uma estratégia de treinamento em 2 níveis (ou *dual*); e em seguida, ii) aplicar um processo de poda que remova desse *pool* os classificadores incompetentes para lidar com o problema em questão, tornando-o assim mais eficiente e eficaz.

As taxas de acerto da arquitetura proposta: i) superaram as alcançadas pelos métodos de treinamento em 1 nível (*Bagging*), com ou sem poda, ou no conceito *Single Best*; ii) se igualaram ou superaram as obtidas quando usada estratégia de treinamento em 2 níveis, sem aplicação de poda, ressaltando que esses resultados da arquitetura proposta foram obtidos com *ensembles* com tamanhos em torno de 20% do *pool* original.

Principais contribuições: i) formulação de um modelo eficaz de uso conjugado e complementar de métodos de treinamento em 2 níveis com poda de *ensembles*; ii) eficácia da arquitetura em uma aplicação de *credit-scoring*; iii) demonstração da superioridade dos MCS sobre métodos baseados em um único classificador; iv) método eficaz para uso de árvores de decisão em *credit-scoring* por meio de MCS.

V. TRABALHOS FUTUROS

É possível configurar a arquitetura aqui proposta de diferentes maneiras, visando identificar conjunções

que alcancem desempenho interessante, a depender da aplicação. Dessa forma, são sugeridas algumas linhas de pesquisa, tais como: i) avaliar como se comporta a arquitetura proposta, ao variarmos o método usado na fase de poda. Por exemplo, o método de poda *Ensemble Pruning via Individual Contribution Ordering (EPIC)* [23] usa critérios de seleção de classificadores diferentes do *Orientation-Ordering*, chegando a superá-lo em desempenho em problemas de classificação binária. É possível que, neste caso, a arquitetura proposta apresente ainda um melhor desempenho do que com o OO; ii) Experimentar o uso da arquitetura em outros tipos de aplicações, inclusive aquelas que envolvam problemas com múltiplas classes, adotando para isso outros modelos de classificadores base, tais como redes neurais artificiais (RNA) ou máquinas de vetor de suporte (SVM). Nestas situações, pode ser interessante na fase de teste trabalhar com métodos de combinação com abordagens de *rank* ou *medição*; iii) Analisar como alternativa para tratar da questão da redundância de atributos, o uso de outros métodos de seleção ou extração de características na fase de treinamento.

REFERÊNCIAS

- [1] M. P. Ponti, "Combining classifiers: from the creation of ensembles to the decision fusion," in *Conference on Graphics, Patterns and Images Tutorials*, 2011, pp. 1–10.
- [2] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion*, vol. 16, pp. 3 – 17, 2014.
- [3] O. G. Selfridge, "Pandemonium: a paradigm for learning in mechanisation of thought processes," in *Symposium Held at the National Physical Laboratory*, Nov. 1958, pp. 513–526.
- [4] L. I. Kuncheva, *Combining Pattern Classifiers: methods and algorithms*. Wiley-Interscience, 2004.
- [5] L. E. A. Santana, D. F. de Oliveira, A. M. Canuto, and M. C. P. de Souto, "A comparative analysis of feature selection methods for ensembles with different combination methods," in *International Joint Conference on Neural Networks*, 2007, pp. 643–648.
- [6] K. Tumer and J. Ghosh, "Error correlation and error reduction in ensemble classifiers," *Connection science*, vol. 8, no. 3-4, pp. 385–404, 1996.
- [7] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.
- [8] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.
- [9] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119 – 139, 1997.
- [10] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, pp. 832–844, 1998.
- [11] J. Rodríguez, L. Kuncheva, and C. Alonso, "Rotation forest: A new classifier ensemble method," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 10, pp. 1619–1630, Oct 2006.
- [12] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *CoRR*, vol. cs.AI/9501101, 1995.
- [13] G. Wang, J. Ma, L. Huang, and K. Xu, "Two credit scoring models based on dual strategy ensemble trees," *Knowledge-Based Systems*, vol. 26, pp. 61–68, 2012.
- [14] A. Marqués, V. García, and J. Sánchez, "Two-level classifier ensembles for credit risk assessment," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10916 – 10922, 2012.

- [15] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.
- [16] G. Martínez-Munoz and A. Suárez, "Aggregation ordering in bagging," in *IASTED International Conference on Artificial Intelligence and Applications*, 2004, pp. 258–263.
- [17] G. Martínez-Muñoz, D. Hernandez-Lobato, and A. Suarez, "An analysis of ensemble pruning techniques based on ordered aggregation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 245–259, 2009.
- [18] D. D. Margineantu and T. G. Dietterich, "Pruning adaptive boosting," in *International Conference on Machine Learning*, vol. 97, 1997, pp. 211–218.
- [19] G. Martínez-Muñoz and A. Suárez, "Pruning in ordered bagging ensembles," in *International Conference on Machine Learning*, 2006, pp. 609–616.
- [20] G. Tsoumakas, I. Partalas, and I. Vlahavas, "An ensemble pruning primer," in *Applications of Supervised and Unsupervised Ensemble Methods*, ser. Studies in Computational Intelligence, O. Okun and G. Valentini, Eds. Springer Berlin Heidelberg, 2009, vol. 245, pp. 1–13.
- [21] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: Many could be better than all," *Artificial Intelligence*, vol. 137, no. 1–2, pp. 239 – 263, 2002.
- [22] Z.-H. Zhou and W. Tang, "Selective ensemble of decision trees," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, ser. Lecture Notes in Computer Science, G. Wang, Q. Liu, Y. Yao, and A. Skowron, Eds. Springer Berlin Heidelberg, 2003, vol. 2639, pp. 476–483.
- [23] Z. Lu, X. Wu, X. Zhu, and J. Bongard, "Ensemble pruning via individual contribution ordering," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 871–880.
- [24] Q. Dai, "A competitive ensemble pruning approach based on cross-validation technique," *Knowledge-Based Systems*, vol. 37, pp. 394 – 414, 2013.
- [25] A. Marqués, V. García, and J. Sánchez, "Exploring the behaviour of base classifiers in credit scoring ensembles," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10 244 – 10 250, 2012.
- [26] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: <http://archive.ics.uci.edu/ml>