

Geração Automática de Conjuntos de Classificadores Através da Seleção de Características não Supervisionada

Marisa Morita, Luiz S. Oliveira e Robert Sabourin

Resumo — Neste artigo propõe-se uma estratégia para a geração automática de conjuntos de classificadores (Ensemble of Classifiers) através da seleção de características não supervisionada. Tal estratégia é baseada em dois níveis de algoritmos genéticos multi-objetivos os quais são responsáveis por gerar diversos classificadores através da seleção de características e então combiná-los para encontrar conjuntos de classificadores. O método proposto é avaliado no contexto de reconhecimento de palavras manuscritas. Para isso, três conjuntos de características diferentes são utilizados para alimentar os classificadores baseados nos Modelos Escondidos de Markov. Experimentos detalhados demonstram a eficácia da metodologia proposta.

Palavras-Chave—Conjuntos de Classificadores, Seleção de Características não Supervisionada, Reconhecimento de Palavras Manuscritas, Algoritmos Genéticos, Modelos Escondidos de Markov.

I. INTRODUÇÃO

A escolha das características para a representação de um problema de reconhecimento de padrões afeta diretamente vários aspectos, como por exemplo, desempenho, tempo necessário para o aprendizado e o número mínimo de exemplos para a aprendizagem. Tendo isso em vista, a seleção das melhores características é muito importante na construção de um classificador. A maioria dos trabalhos sobre seleção de características têm sido realizado no contexto da aprendizagem supervisionada [16], [19], e portanto, pouca atenção têm sido dispensada para o problema de aprendizagem não supervisionada [4], [10].

O objetivo na seleção de características não supervisionada consiste em procurar um subconjunto de características que represente o melhor agrupamento dos dados (clusters¹) de acordo com algum critério. Isso não é uma tarefa trivial, uma

vez que para achar o subconjunto de características que maximiza o critério de desempenho, os clusters devem ser definidos. O problema se torna ainda mais difícil, quando não se conhece a priori o número de clusters, o que acontece na maioria dos problemas reais. Conseqüentemente, é necessário explorar diferentes números de clusters usando métodos tradicionais de agrupamentos, como por exemplo, K-Means [8] e suas variantes. Deste modo, clustering pode se tornar um trabalho massivo de tentativa-e-erro. Além disso, o resultado desse processo pode não ser muito promissor, especialmente quando o número de clusters é grande e difícil de estimar.

Neste contexto, a seleção de características não supervisionada pode ser apresentada na forma de um problema de otimização multi-objetivos, por exemplo, o número de características e um índice que indique a qualidade dos clusters. Sendo assim, uma abordagem adequada para resolver esse tipo de problema são os algoritmos genéticos multi-objetivos, uma vez que eles podem gerir vários objetivos de uma maneira inteligente, e além disso, são geralmente bastante eficientes quando encontram grandes espaços não lineares. Outra vantagem de usar tais algoritmos é a possibilidade de obter mais do que um conjunto de soluções (trade-offs entre as funções que estão sendo otimizadas) no final do processamento.

Tais soluções podem ser úteis de diferentes maneiras, porém neste trabalho elas serão utilizadas na construção automática de conjuntos de classificadores (CdC). Em outras palavras, a partir de um classificador base, vários classificadores serão gerados e vários CdC construídos de forma automática. A literatura mostra diversas estratégias para a geração de CdC, por exemplo, bagging [1], boosting [5], sub-espacos aleatórios [7], input decimation [18] e seleção de características[6].

O método apresentado neste trabalho utiliza a seleção de características não supervisionada e opera em dois diferentes níveis. O primeiro é responsável pela produção de vários classificadores através da seleção de características, enquanto o segundo deve escolher o melhor CdC dentre todos os classificadores. Esta estratégia foi baseada no trabalho de Oliveira et al [17], o qual considera o contexto da aprendizagem supervisionada. Do melhor do conhecimento dos autores, este é o primeiro trabalho que contempla a

Luiz S. Oliveira é professor visitante no Programa de Pós-Graduação em Informática Aplicada da Pontifícia Universidade Católica do Paraná, e professor adjunto no curso de Ciência da Computação da Universidade Tuitui do Paraná, Curitiba, PR, Brasil (e-mail: soares@ppgia.pucpr.br).

Marisa Morita é analista de sistemas senior no HSBC Bank, Curitiba, Pr, Brasil (e-mail: marisa.e.morita@hsbc.com.br)

Robert Sabourin é professor titular da Ecole de Technologie Superieure, Montreal, QC, Canadá. (e-mail: robert.sabourin@etsmtl.ca)

¹ Neste artigo os termos cluster e clustering são usados como sinônimos de grupos e agrupamentos, respectivamente.

seleção de características não supervisionada para a criação automática de CdC. A robustez do método proposto é demonstrada através de experimentos realizados no contexto do reconhecimento de palavras manuscritas, onde ambas as taxas de reconhecimento e confiabilidade foram melhoradas.

O restante deste artigo está organizado da seguinte maneira. Seção II discute a otimização multi-objetivos usando algoritmos genéticos, Seção III descreve os classificadores e base de dados usados e a Seção IV detalha a metodologia proposta. Finalmente, a Seção V reporta os experimentos realizados enquanto a Seção VI conclui este trabalho.

II. OTIMIZAÇÃO MULTI-OBJETIVOS USANDO ALGORITMOS GENÉTICOS

Desde que o conceito de otimização multi-objetivos será explorado no restante deste artigo, essa seção faz uma breve introdução ao assunto.

Um problema de otimização multi-objetivos é composto por um número de objetivos associados a um número de restrições. As soluções deste tipo de problema podem ser expressas matematicamente em termos de *pontos não-dominados*, ou seja, uma solução domina outra se e somente se ela é superior a outra em todos os objetivos. A solução pertence ao *Pareto-ótimo* se ela não é dominada por qualquer outra solução pertencente ao espaço de busca. Nos experimentos apresentados aqui, o algoritmo escolhido foi o NSGA (Non-Dominated Sorting Genetic Algorithm) com elitismo, proposto por Srinivas e Deb [20].

Tal algoritmo utiliza um esquema de ordenação para enfatizar os bons indivíduos da população e uma técnica de nicho para evitar convergência prematura. Ele difere do algoritmo genético simples somente na forma em que faz a seleção. As operações de cruzamento e mutação permanecem as mesmas. Antes que a seleção aconteça, a população é ordenada com base na não-dominância dos indivíduos. Os indivíduos não-dominados da população são identificados, e então se assume que eles constituem a primeira frente não-dominada. Para garantir que todos os indivíduos tenham a mesma chance de reprodução, se atribui a eles um alto valor de fitness, o qual é conhecido por *dummy-fitness*. Para manter a diversidade da população, os indivíduos desta frente são forçados a compartilhar suas *dummy-fitness*. Esse compartilhamento é feito através da divisão da *dummy-fitness* de um indivíduo pelo número de indivíduos próximo a ele. Ou seja, se existem vários indivíduos próximos uns dos outros, eles terão suas *dummy-fitness* reduzidas para manter a diversidade da população. Após esse compartilhamento, os indivíduos dessa frente são ignorados temporariamente para processar o resto da população afim de encontrar os indivíduos da segunda frente não-dominada. Para esses indivíduos dessa nova frente, se atribui um novo valor de *dummy-fitness*, o qual é mantido menor que o mínimo valor encontrado na frente anterior. O processo continua até que toda população esteja classificada em frentes não-dominadas.

Na seqüência, a população é reproduzida de acordo com os valores das *dummy-fitness*. Uma vez que os indivíduos da primeira frente possuem valores maiores de fitness, os

mesmos têm maiores chances de reprodução. A eficiência do NSGA está na maneira em que vários objetivos são convertidos em frentes não-dominadas e reduzidos a uma função da *dummy-fitness*. Mais detalhes sobre NSGA podem ser encontrados em [3] e [20].

III. CLASSIFICADORES E BASE DE DADOS

Para avaliar a metodologia proposta, foram utilizados três classificadores baseados nos modelos escondidos de Markov (HMM – Hidden Markov Models), os quais foram treinados para reconhecer palavras manuscritas, cujo léxico é composto dos doze meses do ano (janeiro, fevereiro, março, abril, maio, junho, julho, agosto, setembro, outubro, novembro e dezembro). As bases de dados de treinamento (TRDB), validação (VLDB) e teste (TSDB) são compostas de 1200, 400 e 400 imagens, respectivamente. Afim de aumentar as bases de treinamento e validação, também foram consideradas 8300 e 1900 palavras, respectivamente, extraídas da base de dados de valores extensos de montantes de cheques bancários Brasileiros. Isto é possível, pois os HMMs considerados aqui são baseados em modelos de caracteres. O método de validação empregado durante o treinamento dos modelos é baseado em uma sub-amostragem aleatória. Para cada experimento, a base de dados é dividida como ilustrado na Figura 1, os modelos re-treinados e o erro calculado. O erro final estimado é então a média dos erros encontrados nos K experimentos realizados.

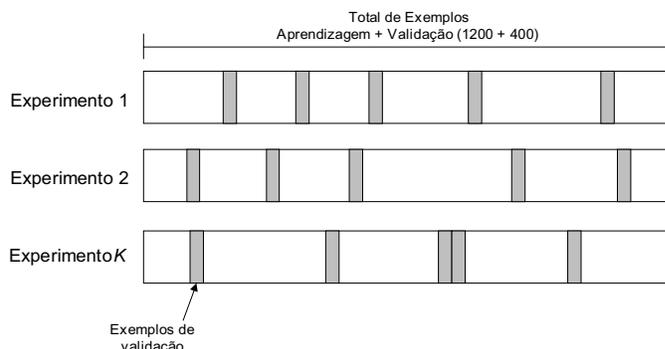


Fig 1. Procedimento utilizado para a validação durante o treinamento.

Uma segunda base de dados de validação (VLDB2), composta por 500 palavras de meses, também foi considerada [9]. Ela é utilizada durante a seleção do melhor CdC (Seção V).

O conjunto de características que alimenta o primeiro classificador é uma mistura de informações de concavidades e contorno (CC) [16]. Nesse caso, cada sub-imagem é dividida em duas partes iguais (horizontalmente), e então um vetor de 17 componentes é extraído. Portanto, o vetor de características final contém 34 componentes. Os outros dois classificadores usam características baseadas em medidas de distâncias [15]. Um usa o mesmo zoneamento discutido anteriormente, mas nesse caso para cada região 16 componentes são extraídos, o que leva a um vetor final de 32 componentes (DDD32). Já para o outro, um diferente

zoneamento foi tentado, no qual a sub-imagem é dividida em quatro zonas usando as linhas de base como referência (Fig. 2). Desta maneira, o vetor de características contém 64 componentes (DDD64).

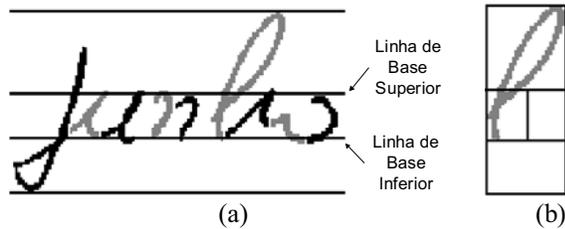


Fig 2. Zoneamento baseado nas linha de base: (a) linhas de base e (b) zoneamento em quatro regiões.

A Tabela I reporta o desempenho de todos os classificadores na base de dados de teste (TSDB), bem como com o número de componentes no vetor de características e o tamanho do codebook (definido empiricamente). Além disso, mostra ainda a taxa de reconhecimento sem rejeição e taxa de reconhecimento para uma taxa de erro fixada em 1%. Essa última, é muito mais significativa quando se considera uma aplicação real, uma vez que ela descreve a taxa de reconhecimento em relação à uma taxa específica de erro, incluindo assim, uma taxa de rejeição implicitamente. Essa taxa ainda permite calcular a confiabilidade dos sistemas para uma data taxa de erro (1).

$$\text{Confiabilidade} = \frac{\text{Taxa Rec}}{\text{Taxa Rec} + \text{TaxaErro}} \cdot 100 \quad (1)$$

Pode-se observar na Tabela I que as taxas de reconhecimento com erro fixo em 1% são baixas, conseqüentemente, o número de exemplos rejeitados é bastante alto.

TABELA I
DESEMPENHO DOS CLASSIFICADORES NA BASE DE TESTE

Conjunto de Características	No. de Componentes	Tamanho do Coodebook	Taxa. de Rec.	Taxa de Rec. (1%)
CC	34	80	86,1%	61,0%
DDD32	32	40	73,0%	30,0%
DDD64	64	60	64,5%	24,7%

IV. METODOLOGIA PROPOSTA

Esta seção descreve a abordagem hierárquica proposta. Como mencionado anteriormente, ela é baseada em algoritmos genéticos multi-objetivos (MOGA) organizados em dois níveis. O primeiro nível produz vários classificadores através da seleção de características não supervisionada e o segundo nível procura os melhores CdC dentre os classificadores. Em ambos os níveis, os algoritmos genéticos utilizam uma representação binária, cruzamento em um ponto e mutação de bit. O elitismo implementado aqui combina a nova população com a população corrente (*generational*

algorithm) [3]. Em síntese, a metodologia considera cinco passos: (i) Seleção de características utilizando TRDB, (ii) Treinamento dos HMMs produzidos durante a seleção de características usando TRDB e VLDB como bases de dados de treinamento e validação, respectivamente, (iii) Procura pelo melhor CdC usando VLDB, (iv) Seleção do melhor CdC usando VLDB2 e (v) Testar o melhor CdC na base de teste (TSDB). Nas próximas subseções, os passos supracitados são discutidos em detalhes. Figura 3 ilustra a estratégia proposta.

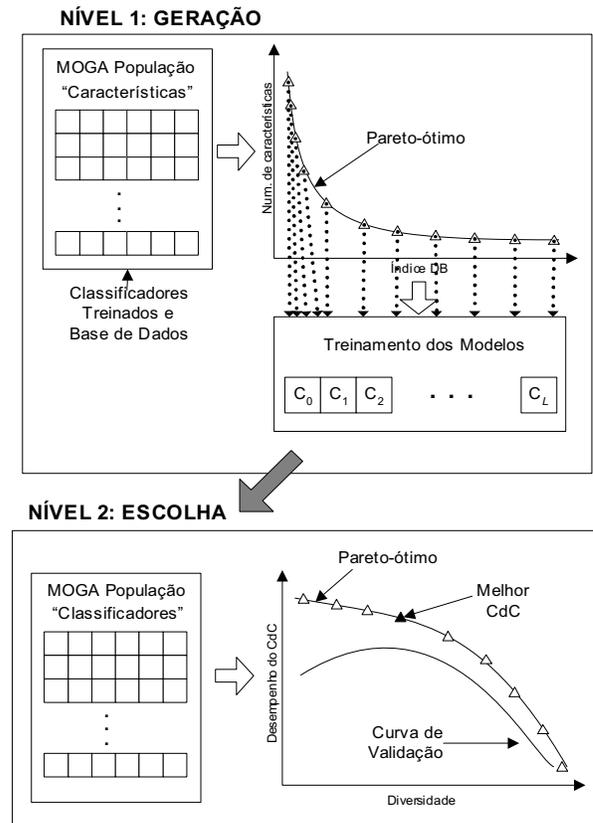


Fig 3: Visão global da estratégia proposta.

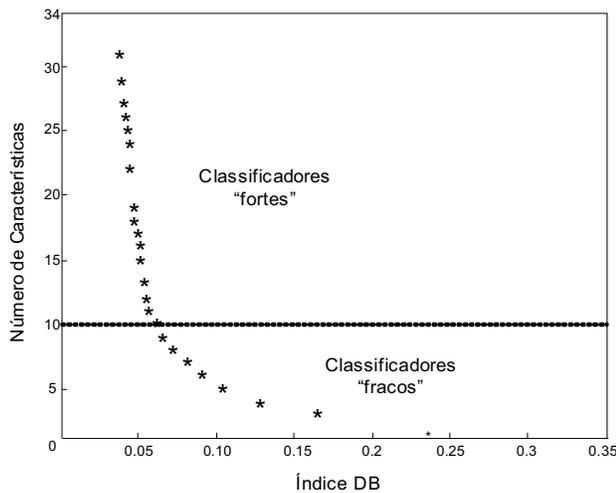
A. Seleção de Características

O algoritmo de seleção de características usado neste trabalho foi introduzido primeiramente em [14]. Entretanto, para fazer com que esse artigo fique completo, uma breve descrição do mesmo é apresentada nessa seção. Ele considera um algoritmo genético multi-objetivos para otimizar dois critérios: minimização do número de características e minimização de um índice que qualidade dos clusters.

Para medir a qualidade dos clusters durante o processo de *clustering*, o índice DB proposto por Davies e Bouldin [2], foi utilizado sobre 80000 vetores de características extraídos de 9500 palavras da TRBD. Para tornar esse índice adequado para o problema de seleção de características, ele deve ser normalizado pelo número de características selecionadas. Isso se deve ao fato do mesmo ser baseado em distâncias

geométricas e, portanto, não pode ser aplicado diretamente aqui, uma vez que ele pode ser influenciado pela dimensionalidade do espaço, o qual é variável em problemas de seleção de características.

Durante os experimentos, notou-se que o valor do índice DB diminuía a medida que o número de características aumentava. Tal problema está diretamente relacionado à normalização do índice pelo número de características. Por esse motivo optou-se pela utilização do número de características como segundo objetivo. Um exemplo do Pareto-ótimo encontrado ao fim da seleção de características é mostrado na Fig. 4. Nessa figura, cada ponto representa um classificador que utiliza um subconjunto de características diferente do conjunto original.



Outras regras de combinação, como por exemplo, produto, max e min foram testadas, mas a média produziu melhores resultados. Para avaliar a ambigüidade e o poder de generalização dos classificadores durante o processo de otimização, foi usada a base de dados VLDB.

V. EXPERIMENTOS E DISCUSSÃO

Todos os experimentos realizados neste trabalho são baseados em algoritmos genéticos multi-objetivos, com uma única população, e estratégia mestre-escravo, na qual o computador mestre executa a operações genéticas (seleção, mutação e cruzamento), e os computadores escravos são responsáveis pela avaliação da fitness de cada indivíduo. Nestes experimentos, foi utilizado um cluster de computadores Beowulf, com 17 máquinas (um mestre e 16 escravos) de 1.1GHz, 512Mb RAM.

Os seguintes parâmetros foram usados nos dois níveis do algoritmo: tamanho da população = 128, número de gerações = 1000, probabilidade de cruzamento = 0.8, probabilidade de mutação $1/L$ (na qual L é o tamanho do cromossomo) e distância para o cálculo do nicho = 0.35. Esses valores foram definidos empiricamente e são muito próximos daqueles apresentados por Miki et al [21].

O tamanho do cromossomo no primeiro nível é igual ao número de componentes do conjunto de características (veja Tabela I), enquanto no segundo nível é igual ao número de classificadores do conjunto selecionado do Pareto-ótimo resultante do primeiro nível. Para definir as probabilidades de cruzamento e mutação em ambos os níveis, foi considerado a função *one-max*, a qual é provavelmente a função de teste mais usada em pesquisas de algoritmos genéticos, devido a sua simplicidade e similaridade com o problema de seleção de características [22]. Ela simplesmente mede o número de bits iguais a 1 no cromossomo.

Após definir todos os parâmetros, o primeiro passo, como definido na Seção IV, consiste em executar a seleção de características para um dado conjunto de características. Como ilustrado na Fig. 4, esse processo produz um grande número de classificadores, os quais devem ser treinados para serem usados no segundo nível do algoritmo. Após alguns experimentos, foi constatado que o segundo nível do algoritmo sempre escolhe classificadores “fortes” (veja Fig. 4) para compor os CdC. Desta maneira, afim de acelerar o processo de aprendizagem e o segundo nível de busca do algoritmo, somente os classificadores pertencentes ao grupo “fortes” são utilizados. Para treinar tais classificadores, as mesmas bases de dados descritas na Seção III foram consideradas. A Tabela II resume os classificadores “fortes” (após treinamento) produzidos durante a seleção de características para os três conjuntos de características discutidos anteriormente.

Para ler a Tabela II, considere por exemplo o conjunto de características CC. Nesse caso, o algoritmo de seleção de características (1º nível do algoritmo) gerou 15 classificadores “fortes”, os quais têm o número de características variando entre 10 a 32, tamanho do *codebook* (número de clusters para

tornar as características discretas para os HMMs) variando entre 29 a 39 e taxa de reconhecimento variando entre 68,1 a 88,6 na VLDB.

TABELA II
RESUMO DOS CLASSIFICADORES PRODUZIDOS PELA SELEÇÃO DE
CARACTERÍSTICAS

Conjunto de Caract.	Número de Classificadores	Qtde de Caract. (variação)	Codebook (variação)	Taxa de Rec. (variação)
CC	15	10-32	29-39	68,1-88,6
DDD32	21	10-31	20-30	71,7-78,0
DDD64	50	10-64	52-80	60,6-78,2

O tamanho do *codebook* para um determinado conjunto de características é fornecido pelo algoritmo genético. Como visto anteriormente, na primeira fase dois objetivos são otimizados: o índice DB e o número de características. Sendo assim, cada solução encontrada ao fim da otimização (Fig 4) possui um *codebook* associado, o qual representa o número de clusters que minimiza o índice DB.

A Tabela 2 mostra a grande diversidade de classificadores produzida pelo algoritmo de seleção de características. Baseado nesses classificadores, quatro conjuntos de classificadores básicos podem ser derivados da seguinte maneira: $F_1 = \{CC_0, \dots, CC_{14}\}$, $F_2 = \{DDD32_0, \dots, DDD32_{20}\}$, $F_3 = \{DDD64_0, \dots, DDD64_{49}\}$ e $F_4 = \{F_1 \cup F_2 \cup F_3\}$. Todos esses conjuntos poderiam ser chamados CdC, porém neste artigo o termo CdC é reservado para caracterizar os conjuntos produzidos de forma automática durante o segundo nível do algoritmo.

Assim como no primeiro nível, o segundo também gera um Pareto-ótimo, o qual mostra os possíveis *trade-offs* entre o poder de generalização (representado aqui pela taxa de reconhecimento do CdC) e a diversidade. Deste modo, o problema agora consiste em escolher o melhor CdC do Pareto-ótimo. A Fig. 5 mostra a grande quantidade de CdC gerados pelo segundo nível do algoritmo para os quatro conjuntos de classificadores básicos citados no parágrafo anterior. Os números sobre os pontos na Fig. 5 representam a quantidade de classificadores que compõe os CdC em questão. Para facilitar a tarefa de escolher qual classificador escolher, e garantir que o mesmo tenha um bom poder de generalização em uma base de dados desconhecida, foi utilizada uma nova base de validação, VLDB2. Uma vez que o objetivo principal é o desempenho, a escolha mais lógica é escolher o CdC que produz a maior taxa de reconhecimento na VLDB2.

Após selecionar o melhor CdC, o passo final consiste em avaliar o mesmo sobre a base de testes (TSDB). A Tabela III mostra os resultados dos quatro CdC na TSDB.

A Fig. 5 ainda mostra o desempenho do CdC gerado a partir do conjunto de todos os classificadores básicos disponíveis (F_4). Neste experimento, o desempenho mínimo esperado seria um desempenho igual ao do melhor CdC (aquele baseado no conjunto F_1), pois o segundo nível do algoritmo deveria ser capaz de encontrar pelo menos o mesmo CdC,

uma vez que todos os classificadores básicos estão

disponíveis.

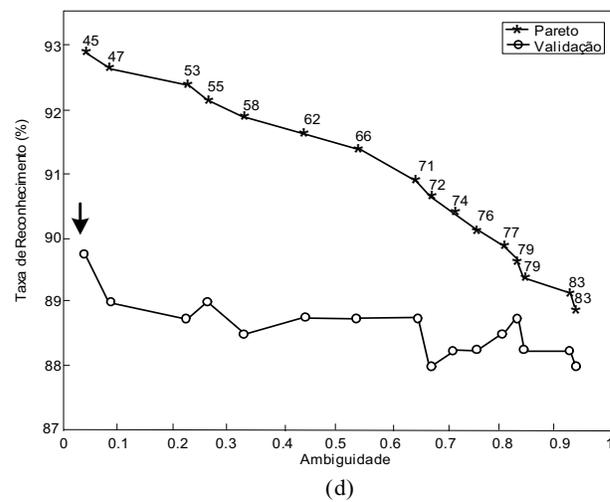
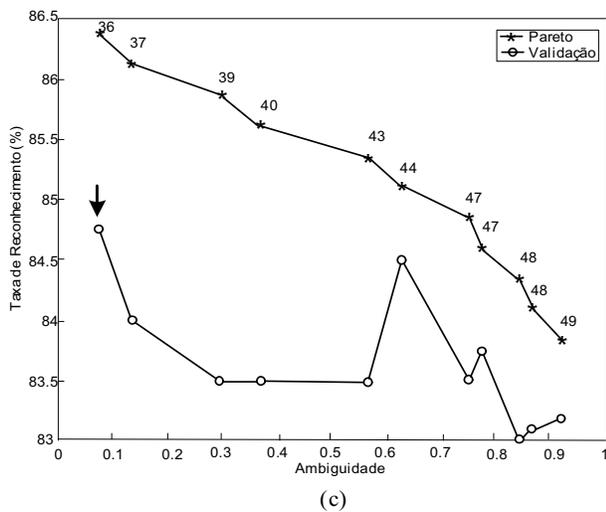
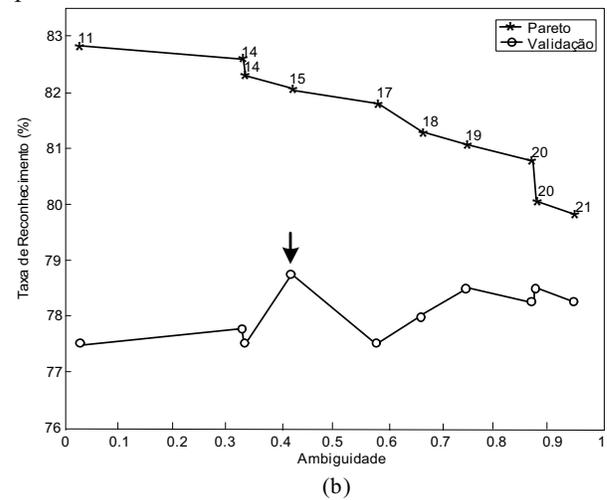
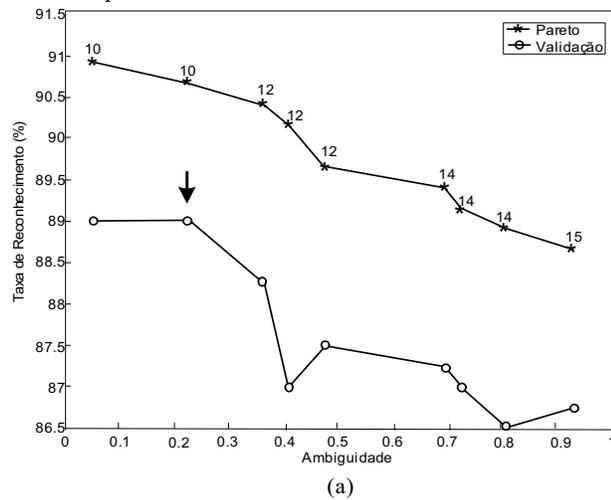


Figura 5: Paretos produzidos pelo segundo nível do algoritmo e suas respectivas curvas de validação. Os melhores CdC de cada Pareto-ótimo estão assinalados por uma flecha. (a) F_1 , (b) F_2 , (c) F_3 , (d) F_4 .

Como pode-se notar na Tabela III, o desempenho alcançado nesse experimento foi ainda melhor. Isso mostra a habilidade do algoritmo em encontrar bons CdC quando mais classificadores básicos estão disponíveis.

TABELA III
DESEMPENHO DOS CdC NA BASE DE TESTE

CdC	Número de Classif.	Taxa. de Rec.	Taxa de Rec. (1%)	Confiab. (Eq. 1)
F_1	10	89,2	70,0	98,5
F_2	15	80,2	45,9	97,8
F_3	36	80,7	43,7	97,7
F_4	45*	90,2	73,2	98,6

*Este CdC é composto por 9, 11 e 25 classificadores de F_1 , F_2 e F_3 respectivamente

Tendo em vista os experimentos discutidos até aqui, pode-se afirmar que a seleção de características não supervisionada é

uma boa técnica para a geração de classificadores com alta diversidade.

É importante salientar que os resultados apresentados na Tabela III devem ser comparados com aqueles apresentados na Tabela I, levando-se em consideração os respectivos conjuntos de características, ou seja, F_1 deve ser comparado com CC, F_2 com DDD32 e assim por diante.

Sendo assim, considere por exemplo que o único conjunto de características disponível é o DDD64, o qual fornece de acordo com a Tabela I uma taxa de reconhecimento de 64,5%. Utilizando o método proposto, é possível gerar um conjunto de classificadores (que mesmo conjunto de características) e melhorar a taxa de reconhecimento em aproximadamente 15%. Ou seja, a taxa de reconhecimento passou de 65% para 80%. Isso demonstra que a estratégia proposta é capaz de produzir uma melhora expressiva no desempenho do sistema.

VI. CONCLUSÃO

Uma metodologia para a geração automática de CdC foi proposta neste artigo. Ela considera dois níveis de algoritmos genéticos multi-objetivos, onde o primeiro produz um conjunto de classificadores através da seleção de características não supervisionada e o segundo combina tais classificadores afim de obter CdC com alto poder de discriminação.

Experimentos em três diferentes conjuntos de características mostraram a eficiência e validade da metodologia proposta. Foi demonstrado que a partir de um simples conjunto de característica pode-se aumentar consideravelmente o desempenho de um classificador.

VII. AGRADECIMENTOS

Esse trabalho foi financiado em parte pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo #150542/2003/8.

VIII. REFERÊNCIAS

- [1] L. Breiman. Stacked regressions. *Machine Learning*, 24(1):49-64, 1996.
- [2] D. L. Davies and D.W. Bouldin. A cluster separation measure. *IEEE Trans. on PAMI*, 1:224-227, 1979.
- [3] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley&Sons Ltd, 2001.
- [4] J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, A. M. Aisen, Unsupervised feature selection applied to content-based retrieval of lung images. *IEEE Trans. on PAMI*, 25(3):373-378, 2003.
- [5] Y. Freund and R. Schapire. Experiments with a new boosting algorithm. In *Procs. 13th Int. Conf. on Machine Learning*, pages 148-156, 1996.
- [6] S. Gunter and H. Bunke. Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. *Pattern Recognition Letters*, 25(11):1323-1336, 2004.
- [7] T. K. Ho. The random subspace method for constructing decision forests. *IEEE Trans. on PAMI*, 20(8):832-844, 1998.
- [8] X. D. Huang, Y. Ariki, and M. A. Jack. *Hidden Markov Models for Speech Recognition*, Edinburgh Univ. Press, 1990.
- [9] J. J. Oliveira Jr, J. M. Carvalho, C. Freitas, and R. Sabourin. Evaluating NN and HMM classifiers for handwritten word recognition. In *Procs of the 15th Brazilian Symposium on Computer Graphics and Image Processing*, pages 210-217, 2002.
- [10] Y. S. Kim, W. N. Street, and F. Menczer. Feature selection in unsupervised learning via evolutionary search. In *Procs. of the 6th Int. Conf. on Knowledge Discovery and Data Mining*, pages 365-369, 2000.
- [11] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on PAMI*, 20(3):226-239, 1998.
- [12] A. Krogh and J. Vedelsby. Neural networks ensembles, cross validation, and active learning. In *Procs. NIPS-7*, pages 231-238, 1995.
- [13] L. I. Kuncheva and C. J. Whitaker, Ten measures of diversity in classifiers ensembles: limits for two classifiers. In *Procs. IEE Workshop on Intelligent Sensor Processing*, pages 1-10, 2001.
- [14] M. Morita, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Procs. 7th Int. Conf. on Document Analysis and Recognition*, pages 666-670, 2003.
- [15] I-S. Oh and C. Y. Suen. Distance features for neural network-based recognition of handwritten characters. *International Journal on Document Analysis and Recognition*, 1(2):73-88, 1998.
- [16] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Automatic recognition of handwritten numerical strings: A recognition and verification strategy. *IEEE Trans. on PAMI*, 24(11):1438-1454, 2002.

- [17] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach. In *Procs. of the 7th Int. Conf. on Document Analysis and Recognition*, pages 676-680, 2003.
- [18] N. C. Oza and K. Tumer. Input decimation ensembles: decorrelation through dimensionality reduction. In *Procs of the 2nd International Workshop on Multiple Classifiers Systems*, pages 238-247, 2001.
- [19] Z. Sun, G. Bebis, R. Miller, Object recognition using feature subset selection, *Pattern Recognition*, 37(11):2165-2176, 2004.
- [20] N. Srinivas and K. Deb. Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221-248, 1995.
- [21] M. Miki, T. Hiroyasu, K. Kaneko, and K. Hatanaka, A parallel genetic algorithm with distributed environment scheme. In *Proc. of International Conference on System, Man, and Cybernetics*, Vol. 1, pages 695-700, 1999.
- [22] E. Cantu-Paz. *Efficient and Accurate Parallel Genetic Algorithms*. Kluwer Academic Publishers, 2000

IX. BIOGRAFIAS



Marisa Morita obteve o título de bacharel em ciência da computação pela PUC/PR em 1994, mestre em informática industrial e engenharia elétrica pelo CEFET-PR em 1998 e Ph.D. em Engenharia da Produção Automática pela École de Technologie Supérieure, Université du Quebec, Montreal, Canadá, em 2003. Atualmente ela é analista de sistemas senior do banco HSBC, Curitiba, PR, Brasil, lotada em IT Payment Clearing Systems. Seus interesses de pesquisa estão

voltados ao reconhecimento de padrões, aprendizagem de máquina não supervisionada e modelos escondidos de Markov.



Luiz S. Oliveira obteve o título de bacharel em informática pelas Faculdades Positivo, em 1995, mestre em informática industrial e engenharia elétrica pelo CEFET-PR em 1998 e Ph.D. em Engenharia da Produção Automática pela École de Technologie Supérieure, Université du Quebec, Montreal, Canadá, em 2003. De 1994 a 1998 ele foi analista de sistemas do Banco HSBC onde trabalhou com sistemas de recursos humanos e financeiros. Atualmente ele é professor visitante da

Pontifícia Universidade Católica do Paraná, e professor adjunto do curso de ciência da computação da Universidade Tuitui do Paraná. Suas pesquisas estão voltadas ao reconhecimento de padrões, aprendizagem de máquina e suas aplicações.



Robert Sabourin obteve os títulos de engenheiro, mestre e PhD em engenharia elétrica pela Ecole Polytechnique de Montreal em 1977, 1980 e 1991, respectivamente. Atualmente ele é professor titular do departamento de engenharia de produção automática da École de Technologie Supérieure, Montreal, Canadá. Ele também é membro sênior do CENPARMI (Centre for Pattern Recognition and Machine Intelligence) e membro da IEEE. Seus interesses de pesquisas são o reconhecimento

de padrões e suas diversas aplicações.