Aplicação de Multiclassificadores Heterogêneos no Reconhecimento de Classes Estruturais de Proteínas

Valnaide G.Bittencourt*, Marjory C. C. Abreu†, Marcílio C. P. de Souto†, José Alfredo F. Costa, Anne M P Canuto†

*Departamento de Computação e Automação**
Departamento de Engenharia Elétrica
Departamento de Informática e Matemática Aplicada†
valnaide@dca.ufrn.br,majo_breu@yahoo.com.br,marcilio@dimap.ufrn.br,alfredo@dee.ufrn.br, anne@dimap.ufrn.br

Resumo

O reconhecimento de dobras de proteína é um dos principais problemas em aberto da biologia molecular e uma importante abordagem para a descoberta de estruturas de proteínas desconsiderando a similaridade de suas seqüências. Neste contexto, as ferramentas computacionais, principalmente as técnicas da Aprendizagem de Máquina (AM), tornaram-se alternativas essenciais para tratar esse problema, considerando o grande volume de dados empregado. Este trabalho apresenta os resultados obtidos com a aplicação de diferentes sistemas multiclassificadores heterogêneos (Stacking, StackingC e Vote), empregando tipos distintos de classificadores base (Árvores de Decisão, K-Vizinhos Mais próximos, Naive Bayes, Máquinas de Vetores Suporte e Redes Neurais), à tarefa de predição de classes estruturais de proteína.

Abstract

Protein fold recognition is one of the main open problems of molecular biology and a important approach to proteins structures discovery without relying on sequence similarity. In this context, computer-based tools, mainly the techniques from Machine Learning (ML), have became essential considering the large volume of data. This paper presents the results obtained with the application of diferents heterogeneous multiclassifier systems (Stacking, StackingC e Vote) using different types of base classifiers (Decision Trees, k-Nearest Neighbor, Naive Bayes, Support Vector Machines and Neural Networks), to the task of protein structural class prediction.

1. Introdução

Atualmente, um dos principais objetivos da área de bioinformática é a predição da função que uma proteína exerce em uma célula [1]. Tal conhecimento é essencial para, por exemplo, o desenvolvimento de novos medicamentos e métodos de diagnóstico. Dois tipos de pesquisas mais comuns na identificação da função de uma proteína são a análise de seqüência e a análise de estrutura.

A análise de seqüência é baseada na comparação entre seqüências (estruturas primárias) de proteínas com funções não conhecidas e aquelas para as quais já se sabe a função. Entretanto, muitas seqüências resultantes do projeto genoma não são rotuladas. Além disso, muitas delas, embora possuam um alto grau de similaridade, não compartilham da mesma função. Por outro lado, duas proteínas podem ter um baixo grau de similaridade e, no entanto, apresentarem funções semelhantes [2].

A análise de estrutura, por sua vez, diz respeito à inferência da forma da proteína (estrutura tridimensional) a partir de sua seqüência para, assim, prever-se sua função. Este método de pesquisa vem sendo bastante estudado pelo fato de as funções das proteínas estarem intrinsecamente relacionadas às suas diferentes conformações espaciais [3] e estas conformações não serem obtidas, de modo fácil, experimentalmente em laboratório.

Neste trabalho, analisamos o problema de determinar a similaridade estrutural de proteínas sem

considerar a similaridade das seqüências, usando métodos de Aprendizagem de Máquina (AM) [1] [4] [5] [6], como Máquinas de Vetores Suporte e sistemas multiclassificadores. Mais especificamente, consideramos o problema de reconhecimento de classes estruturais de dobras de proteína. As proteínas são ditas terem uma dobra comum, que é um padrão tridimensional, se tiverem a mesma estrutura secundária[1] principal no mesmo arranjo e com a mesma topologia, apresentando ou não uma mesma origem evolucionária [7].

Em nossa análise, como em [4], supomos que o número de dobras é restrito. Conseqüentemente, o foco está em predições estruturais no contexto de uma classificação particular de dobras 3-D. Há diversas bases de dados de classificação, tais como Structure Classification of Protein (SCOP) [8] e Class, Architecture, Topology, and Homologous superfamily (CATH) [9]. Essas bases de dados de proteínas possuem características próprias. Por exemplo, a base de dados SCOP, que é usada neste trabalho, é uma classificação hierárquica de estruturas conhecidas de proteína, organizada de acordo com seu relacionamento evolucionário e estrutural. Tal base de dados é dividida em quatro níveis hierárquicos: classe, dobra, superfamília e família. Neste trabalho, concentramos nossos estudos em nível de classe, em que as proteínas podem ser rotuladas em uma das seguintes principais classes estruturais: *all-α, all-β, α/β α+ β ou small.*

2. Trabalhos relacionados

Em termos de reconhecimento de dobras de proteína, [4] usaram três métodos de multi-classificação ("um-contra-outros", "único-um-contra-outros", "todos-contra-todos"), com Máquinas de Vetores Suporte (SVM) e Redes Neurais (RN), para classificar as proteínas em 27 tipos de dobras diferentes. Seu estudo utiliza um grande número de classificadores e introduz SVM ao problema de classificação de proteína. Em [10], os autores usaram comitês de *Discretized Interpretable Multi-Layer Perceptrons* (DIMLP) para aprender duas bases de dados relacionadas à predição de dobras de proteína. Em [10], cada rede aprende as 27 dobras simultaneamente. Bagging e Arcing foram usados para combinar as saídas das redes.

Em [4], observou-se que os métodos tradicionais de aprendizado não tiveram uma performance tão boa devido à quantidade desbalanceada de dados pertencentes a cada uma das classes. Motivados por este problema, em [6] os autores criaram um sistema multiclassificador chamado *Knowledge for Imbalance Sample Sets* (eKISS) e aplicaram-no a uma nova versão da base de dados que foi utilizada em [4]. O eKISS é um sistema que integra o conhecimento induzido por diferentes classificadores, gerados pelas abordagens "um-contra-outros" e "todos-contra-todos", através de regras para melhorar a performance dos classificadores diante de uma base de dados cujas classes estão desbalanceadas.

Em [11] é proposta uma arquitetura hierárquica de aprendizagem (HLA). No primeiro nível do HLA, as redes neurais classificam os dados em uma das 4 principais classes do SCOP. No segundo nível, o sistema tem um outro conjunto de redes, que classificam os dados nas 27 dobras.

O trabalho mais semelhante ao nosso é o encontrado em [12]. Nele, os autores aplicaram o algoritmo *K-Local Hyperplane Distance Neighbor* (HKNN), entre outras coisas, ao problema da predição da classe estrutural de proteínas (4 classes diferentes). Para este problema, obtiveram uma exatidão média de 82%. Entretanto, não podemos comparar este resultado diretamente com os obtidos em nosso trabalho porque as bases de dados e a metodologia para desenvolver os experimentos são diferentes - por exemplo, a fim medir o desempenho dos classificadores, [12] usaram o método do *holdout* e nós, o *10-fold stratified cross validation.*

3. Materiais e métodos

A fim de avaliar a aplicabilidade de sistemas de multiclassificação para a predição de classes estruturais de proteínas, foram selecionados três tipos de multiclassificadores heterogêneos: *Stacking*, *StackingC* e *Vote*, empregando cinco tipos diferentes de classificadores base: Árvore de Decisão (AD), Naive Bayes (NB), K-Vizinhos Mais Próximos (KNN), Máquinas de Vetores Suporte (SVM) e Redes Neurais Artificiais (RN). Todos os métodos usados em nosso estudo foram obtidos do pacote de aprendizagem da máquina do WEKA [13] (http://www.cs.waikato.ac.nz/~ml/weka/) e tiveram seus resultados comparados entre si.

3.1. Multiclassificadores heterogêneos

Neste trabalho foram usadas 3 diferentes estratégias de multiclassificação heterogênea (que combinam distintos algoritmos de aprendizado):

[1] Estrutura secundária se refere a elementos estruturais locais, como α-helices e β-folhas.

- *Vote*: Combina classificadores usando a média não ponderada das probabilidades estimadas, através de um esquema de votação por maioria simples.
- *Stacking*: Constrói um meta conjunto de dados, originado da saída dos classificadores base, para ser usado no treinamento do meta classificador, responsável pela predição final do sistema [14].
- *StackingC* (*Stacking with Confidences*): Variação do *Stacking* baseada na remoção prévia de características irrelevantes e na redução da dimensionalidade do meta conjunto de dados, de tal modo a ser independente do número de classes [15].

3.2. Base de dados

Neste trabalho, usamos a base de dados disponibilizada pelos autores em [6] (*http://www.brc.dcs.gla.ac.uk/~actan/eKISS/data.htm*). Esta base de dados é uma modificação da base de dados original criada e usada em [4] e [5] (*http://www.nersc.gov/_cding/protein/*), que é formada por um conjunto do treinamento (Ntrain) e conjunto de teste (Ntest). O conjunto de treinamento foi extraído do PDB [16] e compreende 313 proteínas de 27 dobras mais povoadas do SCOP (mais de sete exemplos para cada dobra) com todas as seqüências de proteínas com menos de 35% de similaridade entre si e representando suas principais classes estruturais (all-α, all-β, α/β e α+β). O conjunto de teste foi extraído de PDB 40D [8] que contêm 385 representantes (excluindo as seqüências já usadas no conjunto de treinamento) das mesmas 27 dobras de SCOP, em que, novamente, duas proteínas não têm mais do que 35% da seqüência iguais uma das outras.

As características usadas no sistema de aprendizagem são extraídas das seqüências da proteína de acordo com o método descrito em [17], onde uma proteína é representada por um conjunto de vetores baseados em várias propriedades físico-químicas e estruturais dos aminoácidos ao longo da seqüência. Estas propriedades são: hidrofobicidade, polaridade, polarizabilidade, predição da estrutura secundária, volume de Van der Waals normalizado e a composição de aminoácidos da seqüência da proteína.

As propriedades de cada seqüência são descritas por um vetor de 21 atributos contínuos, com exceção da última propriedade (composição de aminoácidos), que contém 20 atributos. Assim, no total, combinando todos os 6 vetores de características, formamos um único vetor de dimensão 125 (ou 125 atributos). Além dessas propriedades, o comprimento da proteína também é considerado para cada dobra da proteína. Desse modo, a dimensionalidade das características consideradas como um todo passa para 126.

É importante ressaltar que [6] ajustaram sua base de dados em relação à base de dados original apresentada em [4] e [5] removendo os erros tanto do conjunto de treinamento como do de teste. Os autores também aplicaram a classificação de dobra de proteína de acordo com SCOP 1.61 [8] e Astral 1.61 [18], com todas as seqüências de proteínas com menos de 40% de similaridade entre elas, removendo-se aquelas classes de dobras com menos de 8 exemplos. Feito isso, a base de dados resultante de dobra de proteína conteve 582 exemplos distribuídos em 25 classes de dobras SCOP e, em um nível hierárquico mais alto, em 5 classes estruturais (distribuídos como visto na Tabela 1). A classe acrescentada (*small*), em relação aos dados de [4] e [5], abrange as proteínas que não se enquadravam em nenhuma das demais classes estruturais (all-α, all-β, α/β e α/β).

Tabela 1: Distribuição de proteínas em classes estruturais.

Classes	all-α	all-β	α/β	α+β	*Small*	TOTAL
Quant. de Exemplos	111	177	203	46	45	582

Como as 27 dobras de proteína podem ser agrupadas em 4 classes estruturais, como indicado em [11], um reconhecimento em dois níveis pode ser feito. No nível 1, uma proteína a ser reconhecida é atribuída a uma das 4 classes estruturais (all-α, all-β, α/β e α+ β) e, no nível 2, ela é classificada em uma das 27 dobras. O reconhecimento pode incluir um desses níveis ou ambos os níveis. Neste trabalho, foi realizada a predição de proteínas no nível 1, considerando as 5 classes estruturais da classificação SCOP (com a adição da classe *small* em relação a [4] e [5]) a ser aprendidas pelas técnicas de AM empregadas.

3.3. Avaliação

Tradicionalmente, a comparação de dois métodos de aprendizagem supervisionada é realizada analisando a significância estatística da diferença entre a média da taxa do erro de classificação, em conjuntos independentes de teste, dos métodos avaliados. Para esta avaliação, diversos conjuntos (distintos) de dados são necessários. Entretanto, a quantidade de dados disponíveis é normalmente limitada. Uma forma de superar este problema é dividir a base de dados em conjuntos de treinamento e de teste pelo uso do procedimento do *k-fold cross validation* [19], [20]. Neste trabalho, é usado o *10-fold stratified cross*

validation, garantindo que cada um dos 10 *folds* apresentam a mesma proporção das diferentes classes.

4. Experimentos

Os melhores parâmetros de cada um dos classificadores base foram escolhidos de acordo com o seguinte procedimento: para um algoritmo, por exemplo, com somente um parâmetro a ser *setado*, um valor inicial para tal parâmetro é escolhido e o algoritmo executado. Então, experimentos com valor maior e menor que ele são também realizados. Se com o valor inicialmente escolhido o classificador obteve o melhor resultado (em termos da média de erro de classificação), então outros experimentos não precisam mais ser executados. Caso contrário, o mesmo processo é repetido para o valor do parâmetro com o melhor resultado até então, e assim por diante. Naturalmente, este procedimento consome mais tempo com o aumento do número dos parâmetros a serem investigados.

Usando tal procedimento, os seguintes valores dos parâmetros de cada um dos métodos empregados foram obtidos (os parâmetros não citados foram *setados* para os seus próprios valores *default*):

- NB: *KernelEstimator* foi setado para *true*;
- KNN: *distance Weighting* = 1/*distance*;
- AD: todos os parâmetros foram *setados* para os seus valores *default*;
- SVM: c= 2^{-6} e expoente = 2;
- RN: número de neurônios na camada escondida = 10; taxa de aprendizado = 0.001; máximo número de iterações = 1000; momento = 0.9; tamanho do conjunto de validação = 10%;

Por representarem diferentes paradigmas de aprendizado, esses métodos foram escolhidos para compor os classificadores base dos multiclassificadores *Stacking*, *StackingC* e *Vote*, [21]. Na formação de cada um dos multiclassificadores, esses métodos foram empregados com os melhores parâmetros encontrados quando executados individualmente (citados acima).

Foram avaliadas algumas variações quanto ao uso do meta classificador no *Stacking* e *StackingC*. Para o *Satcking*, foram utilizados como meta os métodos AD, NB e KNN (métodos simples e de rápido aprendizado); no *StackingC*, o Linear Regression (LR) e KNN (a AD e o NB não foram utilizados porque o meta classificador desse método necessita ter como saída um valor numérico, e na implementação do WEKA, a AD e o NB são categóricos). A AD e o LR foram usados com os valores *default* para seus parâmetros; o KNN, com distance Weighting = 1/distance; e o NB com *KernelEstimator = true*.

Cada um dos métodos, como já mencionado, foi treinado com o *Stratified 10-fold Cross-Validation* da base de dados, de acordo com os melhores parâmetros encontrados. Então, considerando todos os experimentos, a média da porcentagem de classificação incorreta nos conjuntos de testes independentes foi calculada. Em seguida, essas médias foram comparadas duas a duas pelo teste de hipótese (como descrito em [19] e [20]) com o nível de significância (α) igual a 0,05.

5. Resultados computacionais

Antes de começarmos nossa investigação de performance dos sistemas multiclassificadores, analisamos a performance dos classificadores base aplicados ao problema, de acordo com a média da taxa de erro de classificação total.

5.1. Classificadores base

A Tabela 2 a seguir apresenta a média da porcentagem incorreta de classificação de cada um dos métodos de AM empregados (Média), de acordo com a melhor configuração para cada um deles, e o Desvio Padrão (DP) verificado em cada caso. De acordo com essa tabela e com o teste de hipótese aplicado, pode-se verificar que o SVM apresenta um desempenho superior aos outros métodos aplicados (taxa de erro igual a 17,60%), com exceção a RN, cujo teste de hipótese indicou que não há evidência de diferença estatisticamente significativa entre os resultados desses dois métodos, uma vez que a hipótese nula não é rejeitada.

Tabela 2: Taxa de erro dos classificadores base.

Algoritmo	Média	DP
AD	25,21%	5,74%
KNN	24,34%	4,82%
NB	22,04%	5,27%
SVM	17,60%	4,54%
RN	18,79%	2,62%

5.2. Multiclassificadores

As Tabelas 3, 4 e 5 mostram a média (Média) e o desvio padrão (DP) da taxa de erro de classificação para, respectivamente, as técnicas *Stacking*, *StackingC* e *Vote*. Os multiclassificadores gerados usaram como

classificadores base os métodos previamente apresentados. O *Stacking* e o *StackingC* foram treinados com diferentes meta classificadores: NB, AD e KNN para o *Stacking* e LR e KNN para o *StackinC*.

Em relação aos resultados obtidos com a técnica *Stacking* (Tabela 3), verifica-se que o emprego do KNN como meta classificador ocasiona uma maior taxa de classificação incorreta (22,15%) do que os sistemas multiclassificadores obtidos tanto com NB como com AD usados como meta, não apresentando estes dois últimos evidências estatisticamente significativas para diferenciá-los. Em relação aos resultados obtidos com a técnica *StackingC* (Tabela 4), não é verificada nenhuma diferença estatisticamente significativa com a variação do meta classificador (LR e KNN).

Tabela 3: Taxa de erro do método *Stacking*.

Meta	NB	AD	KNN
Média	19,18%	19,77%	22,15%
DP	4,48%	4,88%	4,75%

Tabela 4: Taxa de erro do método *StackingC*.

Meta	LR	KNN
Média	16,31%	16,52%
DP	4,07%	4,74%

Tabela 5: Taxa de erro do método *Vote*.

Média	17,47%
DP	4,29%

Entre as diferentes técnicas de multiclassificação apresentadas acima, podemos concluir, com a aplicação do teste de hipótese, que o *StackingC* apresenta um menor erro de classificação do que os outros multiclassificadores, seguido do *Vote* e do *Stacking*. A superioridade do *StackingC* em relação ao *Stacking* pode ter sido possível devido à quantidade de classes de nossa base de dados. Como mostrado em [15], o *StackingC* supera, normalmente, o *Stacking* para bases de dados com mais de duas classes e tem a tendência de se tornar ainda melhor quando o número de classes aumenta.

A Figura 1 a seguir apresenta um gráfico que resume os resultados obtidos com os classificadores base e os melhores multiclassificadores encontrados (embora não haja diferença estatística entre o *Stacking* com NB e com AD usados como meta classificadores, o *Stacking* com NB é considerado por ter apresentado um menor custo computacional – tempo – na sua construção do que com AD; do mesmo modo, é apresentado o *StackingC* tendo como meta o LR).

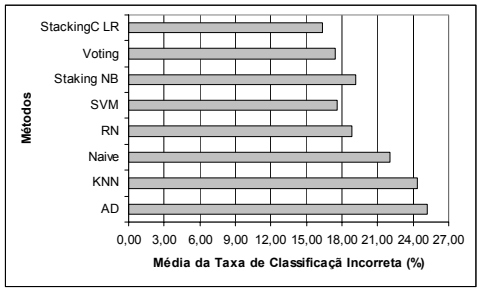

Figura 1: Resumo dos resultados em termos de taxa de erro.

Observando a Figura 1, pode-se verificar a melhoria no resultado obtido com o método *StackingC* (16,31%) em relação a todos os classificadores base. Já o *Stacking*, mostrou-se melhor do que os métodos AD, NB e KNN, mas não apresentou melhor desempenho que a RN (nenhuma diferença estatística foi detectada); e quando comparado ao SVM, mostrou um desempenho significativamente inferior a ele (19,18%). O método *Vote*, apesar de simples, apresentou uma boa performance quando comparado com os classificadores base utilizados, com uma menor taxa de classificação incorreta (17,47%), com exceção do resultado obtido com SVM, que através da aplicação do teste de hipótese, nenhuma diferença estatística é detectada entre eles.

6. Considerações finais

Com os resultados apresentados, podemos observar que, de modo geral, um sistema de multiclassificação tem globalmente um melhor desempenho do que os classificadores individuais usados como base para sua combinação, independente do classificador usado como meta (no caso do *Stacking* e *StackingC*). Além disso, evidenciamos a maior eficiência do *StackinC*, apresentando um melhor desempenho, em termos de taxa de erro de classificação, em relação às outras técnicas empregadas. O método *Vote*, por sua vez, apesar de ser o multiclassificador utilizado de mais baixo custo computacional, em termos de tempo e de complexidade, também apresentou uma boa performance – na comparação entre o *StackingC* e o *Vote*, a hipótese nula foi rejeitada a um nível de significância 0,05, sendo que o *p-value* foi de 0,049.

7. Referências

[1] P. Baldi and S. Brunak, *Bioinformatics: the Machine Learning Approach*, second edition ed. MIT Press, 1998.

[2] D. T. Jones, "GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences," *J. Mol. Biol.*, vol. 287, pp. 797–815, 1999.

[3] K. Guimarães e J. Melo, "Uma Introdução à Análise de Seqüências e Estruturas Biológicas*." Cap. 1. In: III Jornada de Mini-Curso de Inteligência Artificial – Livro Texto*, Editora SBC.

[4] C. H. Q. Ding and I. Dubchak, "Multi-class protein fold recognition using support vector machines and neural networks." *Bioinformatics*, vol. 17, no. 4, pp. 349–358, 2001.

[5] I. Dubchak, I. Muchnik, S. R. Holdbrook, and S. H. Kim, "Prediction of protein folding class using global description of amino acid sequence," *Proc. Natl. Acad. Sci.*, vol. 92, pp. 8700–8704, 1995.

[6] A. C. Tan, D. Gilbert, and Y. Deville, "Multi-class protein fold classification using a new ensemble machine learning approach," *Genome Informatics*, vol. 14, pp. 206–217, 2003.

[7] M. W. Craven, R. J. Mural, L. J. Hauser, and E. C. Uberbacher, "Predicting protein folding classes without overly relying on homology," in *Proc. of ISBM*, vol. 3, 1995, pp. 98–106.

[8] L. LoConte, B. Ailey, T. J. P. Hubbard, S. E. Brenner, A. G. Murzin, and C. Chothia, "Scop: a structural classification of proteins database." *Nucleic Acids Research*, vol. 28, no. 1, pp. 257–259, 2000.

[9] F. M. G. Pearl, D. Lee, J. E. Bray, I. Sillitoe, A. E. Todd, A. P. Harrison, J. M. Thornton, and C. A. Orengo, "Assigning genomic sequences to CATH." *Nucleic Acids Research*, vol. 28, no. 1, pp. 277–282, 2000.

[10] G. Bologna and R. D. Appel, "A comparison study on protein fold recognition," in *Proc. of 9th International Conference on Neural Information Processing*, vol. 5, 2002, pp. 2492–2496.

[11] C. D. Huang, C. T. Lin, and N. R. Pal, "Hierarchical learning architecture with automatic feature selection for multiclass protein fold classification," *IEEE Trans. on Nanobioscience*, vol. 2, no. 4, 2003.

[12] O. Okun, "Protein fold recognition with k-local hyperplane distance nearest neighbor algorithm," in *Proc. of Second European Workshop on Data Mining and Text Mining for Bioinformatics*, 2004, pp. 51–57.

[13] I. H. Witten and E. Frank, *Data mining: practical machine learning tools and techniques with Java implementation*. USA: Morgan Kaufman Publishers, 2000.

[14] D. H. Wolpert, "Stacked generalization". *Neural Networks* (1992), 5:241-259, Pergamon Press.

[15] A.K. Seewald, "How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness", in Proceedings of the Nineteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, 2002, pp. 554-561.

[16] U. Hobohm and C. Sander, "Enlarged representative set of proteins," *Protein Sci.*, vol. 3, pp. 522–524, 1994.

[17] I. Dubchak, I. Muchnik, and S. H. Kim, "Protein folding class predictor for SCOP: approach based on global descriptors," in *Proc. of 5th ISBM*, 1997, pp. 104–107.

[18] J.-M. Chandonia, N. S. Walker, L. L. Conte, P. Koehl, M. Levitt, and S. E. Brenner, "ASTRAL compendium enhancements," *Nucleic Acids Research*, vol. 30, no. 1, pp. 260–263, 2002.

[19] T. Mitchell, *Machine Learning*. New York: McGraw Hill, 1997.

[20] T. G. Dietterich, "Approximate statistical test for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998.

[21] A. K. Seewald, "Towards Understanding Stacking - Studies of a General Ensemble Learning Scheme." PhD thesis, Institute for Med. Cybernetics and Artificial Intelligence, University of Vienna, 2003.