Hierarchical Multi-Label Gene Function Prediction using Adaptive Mutation in Crowding Niching

Mina Moradi Kordmahalleh, Abdollah Homaifar, and Dukka B KC

Abstract— **Computational prediction of protein function is an important field in functional genomics. Gene function prediction is a Hierarchical Multi Label Classification (HMC) problem where each gene can belong to more than one functional class simultaneously, while classes are structured in the form of hierarchy. HMC is becoming a necessity in many domains of applications as well. Crowding niching-Adaptive mutation (CAM) is a new proposed method for solving Hierarchical multi-label gene function prediction problem. The classification in CAM-HMC is structured in three different phases. In the first two phases, a sequential procedure is performed. In the first phase, a full cyclic evolutionary crowding algorithm based on new definition of distance between two individuals, and adaptive mutation is applied in order to find classification rules. In the second phase, all the examples that are covered by these rules are removed from the training data. This sequential procedure is repeated until most of the training examples are covered by CAM-HMC rules. In the third phase, consequent generation is determined to show the probability of coverage of each rule for each hierarchical class. Finally, this ratio is applied to classify testing data. Efficiency of this algorithm is displayed by comparing this algorithm with HMC-GA using Precision-Recall curves for three numerical datasets related to protein functions of the Saccharomyces Cerevisiae organism.**

I. Introduction

Prediction of protein function is an important task in post-genomics era. Especially, assigning biological function to the genes has become a key challenge in this arena. On one hand, various methods [1-4] have been developed to predict functional sites in proteins and on the other hand there are methods to predict overall function [5-6] of the protein.

Databases like GO [7] or FunCat [8] have been developed in order to assist in the gene function prediction. Generally, some machine learning techniques are used to predict function of a gene from a set of possible functions as defined in GO (or FunCat). More precisely, the data in GO have two characteristics that may be different from general machine learning task [9]: i) the functions have hierarchical structure; and ii) each gene can have multiple functions (labels). In this regard, gene function prediction is a hierarchical multi-label classification (HMC) problem.

Various approaches have been developed for solving HMC gene function prediction problem. Obozinski et al. [10] presented a reconciliation approach in which in the first step, SMVs are learned independently for each gene function separately and hierarchy constraints are enforced in the second step. Furthermore, HMC method based on SVM with composite kernel [11] has also been proposed to solve gene function prediction in which HMC problem is resolved into a set of binary classification task, and a composite kernel based classification task is introduced to deal with the binary classification task. Moreover, decision tree based models [9] has also been applied for predicting the multiple functions of the genes.

Furthermore, a system for exploiting the hierarchical structure of functional classification [12] has been developed to improve the accuracy of predictions of individual classifiers. Specifically, circumstances may arise when independent classifiers predict a class positive while its parent predict negative. To avoid such a problem, it is essential to consider all classes in the hierarchy.

A global method called HMC-GA [13] was introduced in which genetic algorithm (GA) is employed for solving the gene-function prediction HMC problem. In this approach, the GA is used to optimize the level of coverage for each antecedent of classification rules in order to build the corresponding consequent of the rules. Additionally, a local classification method based on neural network [14] was proposed to incrementally train a multi-layer perceptron for each level of classification hierarchy where predictions of each level of hierarchy are used as inputs to neural network for prediction of the next level.

In this paper, we propose a new method called CAM-HMC (Crowding niching-Adaptive Mutation in Hierarchical Multi-label Classification) to resolve HMC problem in gene function prediction. In this method, by considering a new definition of distance, crowding is applied in GA in order to find rules with more coverage of the dataset while an adaptive mutation is employed as part of GA operation. Subsequently, the proposed crowding method is applied to the rest of the uncovered data to find the CAM-HMC rules. For the classification, the *consequent* of the given rules is determined. The consequent is the ratio of *number of training example covered by a rule belonging to a class to the number of examples covered by that rule*.

This paper is organized as follows. In section II, proposed CAM-HMC method is described. Section III discusses metrics for evaluation of the results and Section IV presents simulation and results followed by Conclusion.

Manuscript received July 30, 2013. This work was supported in part by the National Science Foundation (NSF) under Cooperative Agreement No. DBI-0939454. Any opinions, findings and conclusions are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. DBKC is partly supported by a startup fund from North Carolina A&T State University

M.M.K is with Electrical and Computer Engineering Department, North Carolina A&T State University, NC, 27411 (e-mail: mmoradik@aggies.ncat.edu).

A.H is with Electrical and Computer Engineering Department, North Carolina A&T State University, NC, 27411 (phone: 336-334-3151, e-mail: homaifar@ncat.edu).

D.B.KC is with Computational Science & Engineering Department, North Carolina A&T State University (phone: 336-285-3210, e-mail: dbkc@ncat.edu).

978-1-4799-3163-7/13/$31.00 ©2013 IEEE

II. PROPOSED CAM-HMC METHOD

There are many niching method in evolutionary and genetic algorithms. Niching is a way to maintain diversity of individuals in GA [15]. Niching was introduced to detect multiple picks in a multi-modal optimization problem, and to prevent genetic drift [16]. Effective niching techniques are critical to the success of GA in classification and machine learning [17]. The focus of this paper is on the application of Crowding niching-Adaptive Mutation to the classification of HMC problem. In the method, in evolutionary cycle, deterministic crowding is used to select new offspring based on adaptive mutation and a proposed procedure to gain distance between two individuals. Individuals with shorter distance are chosen as the winner of the competition. To the best of our knowledge, the introduced procedure for the computation of distance between two individuals, when different types of numbers (binary, integer, and real) in each individual exist, is a new one.

Before describing the proposed implementation, it is important to discuss the representation of individual rules (strings, candidate solutions) as used in the proposed CAM-HMC method. One example of a multi-label classification rule is shown in (1),

$$\text{If } (A_1 \text{OP } \Delta) \text{ AND } \dots \text{ AND } (A_m \text{OP } \Delta) \text{ Then } \{C_1, C_{1,3}, C_{1,4}, C_{1,4,2}\} \quad (1)$$

Each individual is a concatenation of AND clauses where each clause represent an attribute A_i along with {Flag, OP, Δ_1, Δ_2}. An example is provided to illustrate the individual rule format for two sequential attributes, as in (2)

$$\dots/\text{Flag, OP, } \Delta_1, \Delta_2/\text{Flag, OP, } \Delta_1, \Delta_2/\dots \quad (2)$$

where for each real valued attribute; Flag is a binary number which shows the activation/deactivation of the clause; operation OP is related to the type of attribute; in this paper we consider numerical dataset so related operations consist of $>, \geq, <, \leq$; Δ_1, Δ_2 are two essential real numbers to limit the bound on each attribute; and $\{C_i, C_{i,j}, \dots\}$ is the set of hierarchical classes related to examples covered by each rules.

For evolution of the CAM-HMC method, we create initial population via random generation of Flag (0 and 1), operations which are indexed by integers 1, 2, 3, and 4 assigned to $>, \geq, <, \leq$, and real value thresholds Δ_1 and Δ_2. When operations are $>, \geq$, then Δ_2 is considered as null, and also, when the operations are $<, \leq$, then Δ_1 is considered as null. In general, the length of each individual is equal to number of attributes multiplied by four.

After initial population is created, evolutionary algorithm follows by assigning fitness to each individual. Fitness of each individual is related to the number of active clauses within individual and the corresponding number of satisfied attributes in the training dataset as in (3),

$$fitness = K/(M \times N) \quad (3)$$

where K is the total number of attributes on the training dataset assigned by active clauses and satisfied by the corresponding operations. M is the total number of active clauses in the individual, and N is the number of current training examples in each evolutionary cycle. Fitness assignment in this form has the tendency toward the individuals that covers majority of the data in training set.

There are many ways of selecting individuals to create offspring such as roulette wheel, tournament selection, elitism and others. Here, elitism is applied to select a fraction of individual with the highest fitness for the next generation. The remainders of individuals are selected by employing deterministic crowding. This ensures diversity in rules that covers the data in the training set. In deterministic crowding, two individual are picked randomly as parents, and then offspring are created through one point crossover, and adaptive mutation. Competition between parents and offspring is done through the following steps (1-5) [18].

1. Randomly select two parents p_1 and $p_2$with replacement from the current population.
2. Generate two offspring c_1 and c_2 using crossover and adaptive mutation.
3. Perform fitness value evaluation of offspring, $f(c_1)$ and $f(c_2)$, and calculate their distance to parents, $d(p_1, c_1)$, $d(p_1, c_2)$, $d(p_2, c_1)$, and $d(p_2, c_2)$.
4. If $|d(p_1, c_1) + d(p_2, c_2)| \leq |d(p_1, c_2) + d(p_2, c_1)|$, the competition is between (p_1, c_1) and (p_2, c_2). Otherwise, the competition is between (p_1, c_2) and (p_2, c_1).
5. Choose individuals with the higher fitness values as winner in the competition and keep them in the population. Discard the losers.

In order to calculate distance between two individuals, we have introduced a new procedure, since each attribute of individual consists of quadruple value; Flag is a binary number, OP is an integer that indexes four algebraic operations, and Δ_1, Δ_2 are real values. In the following, we have outlined the steps to calculate the distance.

1. For two individuals p_1 and p_2, assume the three counters s_1, s_2 *and* S to be equal to zero.
2. Repeat steps (3-6) for each set of {Flag, OP, Δ_1, Δ_2} corresponding to each attribute in p_1 and p_2.
3. If $Flag(p_1) \neq Flag(p_2)$, then $s_1 = s_1 + 1/4$,
4. If $OP(p_1) \neq OP(p_2)$, then $s_1 = s_1 + 1/16$,
5. If $Flag(p_1) = Flag(p_2) = 1$, and

- If $OP(p_1) = OP(p_2) = [1 \text{ or } 2]$, then $s_2 = s_2 + |\Delta_{1p_1} - \Delta_{1p_2})|/|ub - lb|$

- If $OP(p_1) = OP(p_2) = [3 \text{ or } 4]$, then
  $s_2 = s_2 + |\Delta_{2p_1} - \Delta_{2p_2})|/|ub - lb|$

6. $S = S + s_1 + s_2$
7. Stop the procedure until the last set of {Flag, OP, Δ_1, Δ_2} assigned to the last attribute of the individual is compared.
8. $Distance, d(p_1, p_2) = S$.

where ub and lb are the maximum and minimum value in the training dataset.

Investigating many different mutation schemes such as uniform and non-uniform mutations, we conclude that non-uniform mutation introduced in Janikow et al. [19] is a remarkable scheme for the type of problem studied here since likelihood of large mutation decreases as the search proceeds.

Hence in this paper, mutation is applied only to the Flag and threshold Δ of each attribute with probability of mutation pm for each attribute. If flag is 1 then it changed to 0 and vice versa. For mutation of the real value Δ, adaptive range mutation introduced in Austin et al. [20] is employed and defined in (4-8). This is done to ensure uniform exploration of the search space.

In illustration of adaptive mutation, $\mathrm{P}(t, y)$ is the perturbation function returns a value in the range $[0, y]$ such that as t increases it approaches 0, see (4-6), where y is a fixed preset value and is the difference between upper bound (ub) and lower bound (lb) of training dataset, t is generation number, and T is the maximum number of generation. In these equations, α is a uniform random number in [0,1], β is a system parameter that determines the degree of dependency on generation number, and $\gamma(t)$ provides the fine-tuning capability in the generation t.

$$\mathrm{P}(t, y) = y[1 - \alpha^{\gamma(t)}] \tag{4}$$

$$\gamma(t) = (1 - t/T)^{\beta} \tag{5}$$

$$y = ub - lb \tag{6}$$

To ensure that mutation remains bounded by the search range, the mutation range is redefined as in (7).

$$\begin{cases} \delta_L = \max(lb, \Delta^t - \mathrm{P}(t, y)) \\ \delta_U = \min(ub, \Delta^t + \mathrm{P}(t, y)) \end{cases} \tag{7}$$

Where Δ^t is the threshold value at generation t. By defining δ_L, δ_U boundary, the adaptive range mutation returns Δ^{t+1} as the new random threshold that has symmetry about Δ^t as shown in (8).

$$\Delta^{t+1} = \begin{cases} \Delta^t - (1 - 2p)(\Delta^t - \delta_L), & if\ p \le 0.5 \\ \Delta^t + (2p - 1)(\delta_U - \Delta^t), & otherwise \end{cases} \tag{8}$$

At the end of the full evolutionary CAM cycle, because of huge number of examples and too many attributes in HMC problem, it is still possible to have uncovered data by the generated rules. For solving this problem, CAM is applied again to the remaining examples of the training data that were not covered by the previous rules. In the end, sequential covering will produce enough rules to cover most of the examples in different steps.

After producing enough CAM rules, it is necessary to use a metric to classify test data. Since training data consists of a set of examples and their corresponding hierarchical classes, we used consequent of each rules [13] as a deterministic metric for classification. For rule r produced by CAM, consequent generation gives the probability of belonging to a distinct hierarchical class i. This ratio is the number of training examples covered by the rule r belonging to class i (S_{ri}), to the number of training examples covered by stated rule (S_r) as in (9)

$$consequent_{ri} = S_{ri}/S_r \tag{9}$$

In prediction phase, when an example is covered by a set of rules, classification is achieved by choosing a threshold δ. In cases where an example is covered with more than one rule, the rule with highest fitness is chosen as a covered rule, and classification is done based on the consequent generated by that rule. In fact, for the given threshold, one can decide which hierarchical classes are assigned to the new example. For the covered rule r and hierarchical class i, if $consequent_{ri}$ is more than the selected threshold, then this example belongs to that class.

III. Evaluation Criteria

In HMC problems, often classes are skewed meaning that there are a large number of hierarchical classes but each example is assigned to few classes. In this situation, even a relatively low false positive rate can produce a large number of false positives and hence low precision [21]. For example, in medical diagnosis, (highly skewed cancer detection datasets) only a small proportion of the population has a specific disease at any given time [22]. For these cases, classification error criteria such as Hamming loss cannot make a good sense.

PRC (Precision-Recall curves) and ROC (Receiver Operating Characteristic) are widely used to evaluate the results of machine learning techniques [23]. In case of skewed class distribution, PRC is often preferred to ROC curves [24]. In fact, area under the PR curve is guaranteed to be obtained in any algorithm. In conclusion, area under the precision-recall often serves as an evaluation criterion for statistical relational learning

In the result it is necessary to define Precision-Recall as evaluation criteria for HMC. Since in HMC an example that belongs to a given class automatically belongs to all of its super classes, we use calculation of hierarchical precision and recall known as hP and hR as in [13]. Given an example of testing data, C_i is the set of its predicted classes, and $\acute{C}_i$ is

the set of its real classes. C_i, and $\acute{C}_\iota$ can be extended to contain their corresponding ancestor classes $\hat{C}_i$ and $\hat{\acute{C}}_i$

$$\hat{C}_i = \cup_{c_k \in C_i} Ancestors(c_k)$$
$$\hat{\acute{C}}_i = \cup_{c_l \in \acute{C}_i} Ancestors(c_l) \quad (10)$$

$$hP = \sum_i \left|\hat{C}_i \cap \hat{\acute{C}}_i\right| \Big/ \sum_i \hat{C}_i$$
$$hR = \sum_i \left|\hat{C}_i \cap \hat{\acute{C}}_i\right| \Big/ \sum_i \hat{\acute{C}}_i \quad (11)$$

Applying different threshold δ in the prediction phase, different values of precision and recall are obtained. Thus, by changing the threshold between 0 and 1, a precision and recall curve can be plotted. The area under the precision-recall curve is a metric to evaluate performance of the proposed algorithm.

IV. Result

In the simulation part, we considered three numerical data sets (Cellcycle, Derisi, Eisen) related to protein function of the Saccharomyces cerevisiae organism. These datasets are organized according to FunCat scheme and are available at http://www.cs.kuleuven.be/~dtai/clus/hmcdatasets.html. In table I number of examples, number of attributes, and number of hierarchical classes related to both training and testing datasets is provided. GA and also proposed CAM method are applied on these three datasets to show the effectiveness of the aforementioned techniques to the HMC problem. Table II shows the essential parameters used in these algorithms during the simulations. Where Min-coverage is the minimum number of covered example by an accepted rule, and Max-uncovered is a stopping criteria which shows maximum number of uncovered example ending the sequential procedure of producing rules.

In classification part, there are cases in which an example is covered with more than one rules. In this situation covering rule with highest fitness is selected. According to section III, because of skewed datasets limitations, where only a few classes assigned to an example while there are too many hierarchical classes especially at much deeper level of hierarchy, in this paper area under PR curve is used for evaluation of performance of CAM-HMC. Higher area under the curve proves efficiency of the method. By considering 100 individual and 500 different thresholds δ between [0-1], area under the curve is computed by summing the trapezoidal areas between each point.

We apply HMC-GA by considering two different evolutionary parameters (tables III-IV). By using parameters of table III, we consider range between minimum and maximum of each attribute in producing the initial population of HMC-GA. In other simulations, we just consider the minimum and maximum of all data to make initial population. Results consist of mean and standard deviation for number of rules for covering, and area under the PR curves after 10 repetition. Tables V ,VI show results when two different set of parameter are applied. Table (VII-XI) display results obtained by the proposed CAM-HMC method for $\beta = 2, 3, 4, 5, 6$ respectively. The bold numbers in the tables represent that the Area Under PR Curve (AUC-PR) of HMC-CAM of the table is better than AUC-PR of HMC-GA. According to these results, it is obvious that for each type of dataset there is an appropriate β for which the area under PR curve is higher than HMC-GA. For example, area under PR curve for Cellcycle in the case of $\beta = 6$ is more than result of HMC-GA, also results show when β is equal to 2, Eisen dataset has higher area under the PR curve in contrast to HMC-GA.

TABLE I
SUMMARY OF DATASETS

Dataset	No. of Attributes	No. of Hierarchical Class	No. of training Examples	No. of testing Examples
Cellcycle	77	499	1628	1281
Derisi	63	499	1608	1275
Eisen	79	461	1058	837

TABLE II
FIXED PARAMETER

Parameter	Values	Parameter	Values
Tournament Size	17	Min-coverage	10
Max-uncovered	0.01*maxExample	Number of testing	500

TABLE III
HMC-GA PARAMETERS

Parameter	Values	Parameter	Values
Crossover Rate	0.9	Mutation Probability	0.5
Elitism Rate	0.05	Number of Generations	1000
Flag Rate	0.2	Mutation Rate	0.1

TABLE IV
CAM-HMC PARAMETERS

Parameter	Values	Parameter	Values
Crossover Rate	0.7	Mutation Probablity	0.02
Elitism Rate	0.1	Number of Generations	200
Flag Rate	0.2		

TABLE V
RESULTS OF HMC-GA, USING PARAMETERS IN TABLE III

Dataset	No. of Rules	Area under PR curve
Cellcycle	25.90±12.58	0.1409±0.0086
Derisi	9±3.65	0.1315±0.0138
Eisen	10.80±3.99	0.1429±0.0110

TABLE VI
RESULTS OF HMC-GA, USING PARAMETERS IN TABLE IV

Dataset	No. of Rules	Area under PR curve
Cellcycle	45.58±19.62	0.1479±0.0052
Derisi	42.45±24.58	0.1299±0.0001
Eisen	34.45±23.23	0.1497±0.0029

TABLE VII
RESULTS OF CAM-HMC, β =2

Dataset	No. of Rules	Area under PR curve
Cellcycle	97.50±32.87	0.1383±0.0160
Derisi	85.2±4.61	0.1310±0.0000
Eisen	**78.85±14.57**	**0.1512± 0.0092**

TABLE VIII
RESULTS OF CAM-HMC, β =3

Dataset	No. of Rules	Area under PR curve
Cellcycle	89.05±23.76	0.1412±0.0129
Derisi	**86.05±3.60**	**0.1320±0.0000**
Eisen	85.80±24.54	0.1402±0.0109

TABLE IX
RESULTS OF CAM-HMC, β =4

Dataset	No. of Rules	Area under PR curve
Cellcycle	94.95±27.26	0.1391±0.0173
Derisi	84.40±4.77	0.1299±0.0000
Eisen	100.85±38.76	0.1402±0.0136

TABLE X
RESULTS OF CAM-HMC, β =5

Dataset	No. of Rules	Area under PR curve
Cellcycle	**99.30±35.51**	**0.1498±0.0125**
Derisi	88.70±3.09	0.1300±0.0000
Eisen	84.80±25.38	0.1479±0.0046

TABLE XI
RESULTS OF CAM-HMC, β =6

Dataset	No. of Rules	Area under PR curve
Cellcycle	**99.20±32.18**	**0.1502±0.0150**
Derisi	85.80±3.29	0.1299±0.0000
Eisen	98.55±35.24	0.1426±0.0190

V. CONCLUSION

In conclusion, we have proposed CAM-HMC, a novel method for HMC gene function prediction problem. This method obtains classification rules by applying evolutionary Crowding niching (where we define a new distance between two individuals) and adaptive mutation. In essence, keeping the diversity in search space increases chance to find decision rules with more coverage on the data. Classification rules contain AND clauses between parts of rule, while each part try to cover the assigned attribute by using of operation $>, \geq, <, \leq$ and thresholds Δ_1, Δ_2 operations. In a sequential manner, until the coverage of most of the data, CAM-HMC tries to find rules with more coverage on the remained examples of the previous evolutionary cycle. In the end, consequent rules that show the probability of belonging to each hierarchical class that has coverage by a rule are used to classify examples of test dataset. The results show that proposed CAM-HMC is more beneficial than HMC-GA.

Application of other niching techniques (e.g. fitness sharing) and comparison of these techniques are important future works.

ACKNOWLEDGMENT

Authors would like to thank Anil Khanal, Seifemichael Bekele and Dr. Abrham T. Wokineh for helpful discussions.

REFERENCES

[1] D.B. KC and D. Livesay, "Improving position specific predictions of protein functional sites using phylogenetic motifs," *Bioinformatics*, 24, pp. 2308-2316, 2008.

[2] D.B. KC and D. Livesay, "Topology improves phylogenetic motif functional site predictions," *IEEE/ACM Trans. Comput. Biol and Bioinf.*, 8, pp. 226-233, 2011.

[3] O. Lichtarge, H.R. Bourne and F.E. Cohen, "An evolutionary trace method defines binding surfaces common to protein families," *J. Mol. Biol.*, 257(2), pp. 342-358, 1996.

[4] J.A. Capra, R.A. Laskowski, J.M. Thornton, and M. Singh, "Predicting protein ligand binding sites by combining evolutionary sequence conservation and 3D structure," *Plos Comput. Biol*., 5(12), 2009.

[5] D. Lee, O. Redfern, and C. Orengo, "Predicting protein function from sequence and structure," *Nature Reviews*, 8, pp.995-1005, 2007.

[6] T. Hawkins, M. Chitale, and, D. Kihara, "Functional enrichment analyses and construction of functional similarity networks with high confidence function prediction by PFP," *BMC Bioinf*., 11: 265, 2010.

[7] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al. "Gene ontology: tool for the unification of biology," *Nature Genet*., 25, pp. 25-29, 2000.

[8] A. Ruepp, A. Zollner, D. Maier, K. Albermann, J. Hani, M. Mokrejs, and H.W. Mewes, "The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes," *Nucleic Acids Res.* 32, no 18, pp. 5539-5545, 2004.

[9] L. Schietgat, C. Vens, J. Struyf, H. Blockeel, D. Kocev, and S. Džeroski, "Predicting gene function using hierarchical multi-label decision tree ensembles," *BMC Bioinformatics*, 11, no. 1, 2010.

[10] G. Obozinski, G. Lanckriet, C. Grant, M. Jordan, W. Noble, consistent probabilistic output for protein function prediction," 2008, 9(Suppl 1):S6.

[11] B. Chen, L. Duan, J. HU, "Composite Kernel Based SVM for Hierarchical Multi-label Gene Function Classification," in *Conf. Rec. WCCI 2012 IEEE World Congress on Computational Intelligence*, pp. 10-15, Brisbane, Australia, 2012.

[12] Z. Barutcuoglu, RE. Schapire, OG. Troyanskaya, "Hierarchical multilabel prediction of gene function," *Bioinformatics*, 22, pp. 830-836, 2006.

[13] R. Cerri, R. C. Barros and A.C.P.L.F. de Carvalho, "A genetic algorithm for hierarchical multi-label classification," in *Proceedings of the 27th annual ACM Symposium on Applied Computing (SAC 12)*, pp. 250-255, 2012.

[14] R., Cerri, R.C., Barros, A.C.P.L.F. de Carvalho, "Hierarchical multi-label classification using local neural networks," *Journal of Computer and System Sciences*, 2013.

[15] A.Â. Workineh, And A. Homaifar, "Fitness Proportionate Niching: Maintaining Diversity in a Rugged Fitness Landscape", in *Conf. Rec. The 2012 International Conference on Genetic and Evolutionary Methods*, *GEM'12*, Las Vegas, July 16-191, 2012.

[16] O.J. Mengshoel, D.E. Goldberg, "The Crowding Approach to Niching in Genetic Algorithms," *Journal of Evolutionary Computation*, 16 Issue 3, pp. 315-354,2008.

[17] S.W. Mahfoud, "Niching method for genetic algorithm," PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, USA, IlliGAL Report 95001,1995.

[18] X. Yu, M. Gen, "*Introduction to Evolutionary Algorithms*," ISSN 1619-5736, 2010.

[19] Z. Janikow and Z. Michalewicz, "An experimental comparison of binary and Floating point representations in genetic algorithms," In *Proceedings of the 4th International Conference on Genetic Algorithms (ICGA 1991),* pp. 31–36, 1991.

[20] K. Austin, "Evolutionary Design of Robust Flight Control for a Hypersonic Aircraft," PhD Dissertation, The Univeristy of Queensland, Department of Mechanical Engineering, 2002.

[21] J. Davis, and M. Goadrich, "The relationship between precision-recall and ROC curves," In *ICML 2006*, pp. 233-240, 2006.

[22] J, Davis, E. Burnside, I. Dutra, D. Page, R. Ramakrishnan, V.S. Costa, and J. Shavlik, "View learning for statistical relational learning: With an application to mammography," In *Proceeding of the 19th International Joint Conference on Artificial Intelligence*, 2005.

[23] K. Boyd, V.S. Costa, J. Davis, C.D. Page, "Unachievable region in Precision-Recall space and its effect on empirical evaluation," In *the Proceedings of the Twenty-Ninth International Conference on Machine Learning*, 2012.

[24] S. Kok, and P. Domingos, "Learning Markov logic networks using structural motifs," In *ICML 2010*, pp. 551-558, 2010.