STUDY ON THE PREDICTIONS OF GENE FUNCTION AND PROTEIN STRUCTURE USING MULTI-SVM AND HYBRID EDA

CHEN, Benhui

Graduate School of Information, Production and Systems
Waseda University

January, 2011

Abstract

How to efficiently extract the useful information from high throughput biological data is one of the main challenges in bioinformatics. Machine learning techniques such as Support Vector Machines (SVMs), Neural Networks (NNs), Estimation of Distribution Algorithms (EDAs) have been effective in analyzing biological data because of their capabilities in handling uncertain data noise and in generalization. Gene function prediction and protein structure prediction are two important domains where machine learning techniques are applied in bioinformatics. From the view of machine learning, the gene function predictions can be seen as multi-label classification or hierarchical multi-label classification (HMC) tasks; the protein structure predictions can be seen as optimization tasks that search for the protein folding structure with minimum energy based on its amino acid sequence.

For complicated tasks in bioinformatics, applying machine learning methods simply usually cannot obtain expectable results. The different characteristics and challenges of the tasks will require novel modifications of existing machine learning methods. In t
methods based on multi-SVM and hybrid EDA are developed for solving complicated tasks in the predictions of gene function and protein HP model. According to the characteristics of applications, the prior knowledge of tasks and the information of biological data are extracted by some delicate techniques. The obtained knowledge and information are used to enhance machine learning methods. SVMs based methods are developed for solving different applications of the gene function prediction. EDAs based methods are proposed for solving protein HP model problems. The
’s HP-lattice model is a simplified model of protein structure prediction. It is a useful tool for investigating general properties of protein folding.

In gene function classification applications, there are some datasets with characteristics of high noise, large number of input features and relatively small number of training examples. For those datasets, conventional nonlinear classification methods are unavailable due to the over-fitting problem. A new multi-SVM system called composite kernel based SVM (ck-SVM) is proposed to realize a flexibility controllable classifier which can solve nonlinear tasks and avoid over-fitting in certain degree. The proposed ck-SVM method realizes a nonlinear separating boundary by estimating a series of piecewise local linear boundaries. A local linear multi-SVM system is estimated by a composite kernel incorporating prior knowledge of training data. Instead of building multiple local SVM models separately, the prior knowledge of local subsets is used to construct a composite kernel, and the local linear multi-SVM model is realized exactly in the same way as a single SVM model by using the composite kernel.

In multi-label gene function classification, a training example is associated with a set of labels and the task is to predict the label set for each unseen instance. The main challenge of this problem is that classes are usually overlapped and correlated. An improved method based on label ranking and delicate decision boundary SVM is proposed for solving the multi-label gene classification. The label ranking strategy requires a real-valued score for each class (label) to order a label rank, then classifies a new instance into the classes that ranking score above a threshold. In order to obtain a proper label rank, an improved probabilistic SVM with delicate decision boundary is used as the scoring method. It can improve the probabilistic label rank by introducing the information of overlapped training samples into learning procedure. Instead of estimating a static threshold for all testing instances, an instance-independent thresholding strategy is proposed to decide the classification results. It can estimate an appropriate threshold for each testing instance according to the characteristics of instance and label rank.

Hierarchical multi-label classification (HMC) is a special variant of multi-label classification where the classes are organized in a hierarchy. FunCat prediction problem is a difficult task of HMC. In FunCat datasets, there are hundreds of functional classes structured by a predefined hierarchy, and the example distributions of classes are usually high degree skewed (imbalanced) with negative more than positive. An improved method based on over-sampling and hierarchy constraint is proposed for solving the FunCat prediction. Two measures are implemented to improve the HMC performance by introducing the hierarchy constraint into learning procedures. Firstly, for imbalanced functional classes, a hierarchical SMOTE is proposed as over-sampling preprocessing to improve the SVM learning performance. Secondly, an improved True Path Rule consistency approach is introduced to ensemble the results of binary probabilistic SVM classifications. It can correct the classification results and guarantee the hierarchy constraint of classes.

The protein HP model prediction is an NP-complete optimization problem. The task is to predict the minimum energy lattice structure of a protein from its simplified two-letter (H and P) amino acid sequence. Evolutionary Algorithms (EAs) based methods can be used to solve this NP-complete problem. But for long protein sequences, conventional methods can only find the suboptimum solutions. A hybrid EDA is proposed for solving the protein HP model. In order to select better individuals for probabilistic model of EDA, a composite fitness function containing the information of folding structure core (H-Core) is introduced to replace the traditional fitness function of HP model. Then, local search with guided operators is utilized to refine found solutions for improving efficiency of EDA. In addition, an improved backtracking-based repairing method is proposed to repair invalid individuals sampled by the probabilistic model of EDA. It can significantly reduce the number of backtracking searching operation and the computational cost for long sequence protein.

For optimization problems with irregular and complex multimodal landscapes, EDAs always suffer from the drawback of premature convergence similar to other evolutionary algorithms. An adaptive niching EDA with balance searching based on clustering analysis is proposed for solving complex optimization problems. The Affinity Propagation (AP) clustering is used to adaptively partition the population into niches during a run of EDA firstly. Then, a niche capacity selection mechanism based on the Boltzmann scheme is introduced to realize a balance searching between exploration and exploitation. Two different selection strategies, novelty-proportionate selection based on the searching history information and fitness-proportionate selection based on the best fitness in a niche, are assembled by a Boltzmann weight. Tuned by the Boltzmann weight, at the beginning phase of searching, the dominating novelty-proportionate selection can guide the EDA searching to cover the search space as much as possible; at the end phase of searching, the dominating fitness-proportionate selection can enforce the EDA searching to exploit the already found promising areas. The proposed adaptive niching EDA is used for solving the protein HP model prediction. In addition, three benchmark functional multimodal optimization problems are also used to evaluate the proposed method.

Preface

The common theme of this thesis is developing some improved methods based on machine learning techniques for solving the prediction problems of gene function and protein structure. The material is organized in seven chapters. Most of the material has been published or considered to publish in journal papers and conference papers.

The material in Chapter 2 can be found in

- Benhui Chen, Feiran Sun and Jinglu Hu, "Local Linear Multi-SVM Method for Gene Function Classification", in *Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2010)*, pp.183-188, Kitakyushu, Japan, Dec. 2010.

The material in Chapter 3 can be found in

- Benhui Chen, Weifeng Gu and Jinglu Hu, "An Improved Multi-label Classification Method and Its Application to Functional Genomics", International Journal of Computational Biology and Drug Design (IJCBDD), Vol.3 No.2, pp.133-145, Sep. 2010.
- Benhui Chen, Liangpeng Ma and Jinglu Hu. "An Improved Multi-label Classification Method Based on SVM with Delicate Decision Boundary", International Journal of Innovative Computing, Information and Control (IJICIC). Vol.6 No.4, pp.1605-1614, April 2010.
- Benhui Chen, Weifeng Gu and Jinglu Hu, "An Improved Multi-label Classification Based on Label Ranking and Delicate Boundary SVM", in *Proc. of IEEE World Congress on Computational Intelligence 2010 (WCCI 2010) – International Joint Conference on Neural Networks (IJCNN 2010)*, pp. 786-791, Barcelona, Spain, July 2010.
- Benhui Chen, Liangpeng Ma and Jinglu Hu, "A New SVM Based Method for Solving Multi-label Classification Problem", in *Proc. of the 3rd International Symposium on Computational Intelligence and Industrial Applications (ISCIIA 2008)*, pp. 325-334, Dali, China, Nov. 2008.

The material in Chapter 4 can be found in

- Benhui Chen and Jinglu Hu, "Hierarchical Multi-label Classification Incorporating Prior Information for Gene Function Prediction", in *Proc. of the 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010)*, pp.231-236, Cairo, Egypt, Nov. 2010.

which has been extended into a journal paper

- Benhui Chen, Jinglu Hu, "Hierarchical Multi-label Classification Based on Over-sampling and Hierarchy Constraint for Gene Function Prediction", submitted to IEEJ Trans. on Electrical and Electronics Engineering (TEEE), 2010.

The material in Chapter 5 can be found in

- Benhui Chen, Jinglu Hu. "A Hybrid EDA for Protein Folding Based on HP Model", IEEJ Transactions on Electrical and Electronics Engineering (TEEE), Vol.5 No.4, pp.459-466, July 2010.
- Benhui Chen, Long Li and Jinglu Hu, "A Novel EDAs Based Method for HP Model Protein Folding", in *Proc. 2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pp. 309-315, Trondheim, Norway, May 2009.
- Benhui Chen, Long Li and Jinglu Hu, "An Improved Backtracking Method for EDAs Based Protein Folding", in *Proc. of ICROS-SICE International Joint Conference 2009*, pp.4669-4673, Fukuoka, Japan, Aug. 2009.

The material in Chapter 6 can be found in

- Benhui Chen and Jinglu Hu, "An Adaptive Niching EDA with Balance Searching Based on Clustering Analysis", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E93-A, No.10, pp.1792-1799, Oct. 2010.
- Benhui Chen and Jinglu Hu, "An Adaptive Niching EDA Based on Clustering Analysis", in *Proc. of IEEE World Congress on Computational Intelligence 2010 (WCCI 2010) – Congress on Evolutionary Computation (CEC 2010)* , pp. 858-864, Barcelona, Spain, July 2010.
- Benhui Chen and Jinglu Hu, "A Novel Clustering Based Niching EDA for Protein Folding", in *Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009)*, pp. 748-753, Coimbatore, India, Dec. 2009.

Acknowledgements

I would like to take the opportunity to thank the following people who have supported me through my PhD experience at Waseda University.

First of all, I am heartily thankful to my supervisor, professor Jinglu Hu, whose encouragement, guidance and support from the initial to the final level enabled me to complete my PhD research. His tireless pursuit of excellence in research, teaching, and scientific writing is truly inspirational. His remarkable trust and support are essential for my study and research.

I am very grateful to Professor Kotaro Hirasawa, Professor Mizuho Iwaihara and Associate Professor Lianming Sun who proofread all the thesis. Many thanks are for their valuable comments.

I have had many valuable experiences at Waseda University during my graduate study, so I would like to acknowledge my classmates in Neural Computing Lab (Boyang Li, Jun Zhang, Yang Chen, Wenxiang Dou, Qiangwei Wang, Yu Cheng, Lan Wang, Bo Zhou, Zhan Shi, Yuling Lin et al.) for giving constructive suggestions to my thesis and generously supporting my academic career.

I would also like to express my thanks to my research group (Long Li, Liangpeng Ma, Weifeng Gu and Feiran Sun). I have learned so much knowledge and skills from all of you. Your constructive collaborations and evaluations have been very important assets in my PhD research stage.

Finally, I would like to dedicate this work to my parents, Chaogui Chen and Mingfang Kong, my wife Xuefei Hong and my daughter Yi Chen. Without your unending support and encouragement, I would not have finished the degree. Thank you.

Kitakyushu, Japan Benhui Chen
October 20, 2010

Table of Contents

List of Tables

List of Figures

Glossary

Some notations may have different meaning locally.

Notations

x^T	transpose
b	bias
Ω	coordinate parameter vector of local linear boundary in Local Linear Multi-SVM model
α_k	Lagrange multipliers
p	preferences parameter in Affinity Propagation
$\#SV$	number of support vectors
σ	radius of niching clearing procedure
κ	capacity of niching clearing procedure

Operators and Functions

$\text{sign}(x)$	an odd mathematical function that extracts the sign of x
$\mathcal{L}(\cdot)$	Lagrangian function in construction of optimization problems
$K(x_i, x_k)$	kernel function in SVM
$\sum$	sum
$y(x)$	classifier function
$f(x)$	decision value function
$f_P(x)$	local linear multi-SVM model
$\mathcal{R}(x)$	output of basis function in local linear multi-SVM model
$s(i, k)$	similarity function in Affinity Propagation
$r(i, k)$	responsibility function in Affinity Propagation
$a(i, k)$	availability function in Affinity Propagation
$p(y = 1\|x)$	probabilistic output for SVM classifier
$C(\cdot)$	outputs of multi-label classification based on label ranking
$R(\cdot)$	outputs of ranking method for multi-label classification
$T(\cdot)$	outputs of threshold selection for multi-label classification
$\varphi_i(x)$	the children set of node i which have a positive prediction for a given example x in Hierarchical Multi-label Classification

$E(x)$	energy of protein HP model
$Fit_cp(x)$	fitness value of composite fitness function for protein HP model
$p_{MK}(X)$	k-order Markov probabilistic model for EDAs
$\kappa_t(n_i)$	niche capacity selection for the i-th niche in the t-th iteration of adaptive nicheng EDA

Abbreviations

ANN	Artificial Neural Networks
AP	Affinity Propagation
BOA	Bayesian Optimization Algorithm
DAG	Directed Acyclic Graph
EDA	Estimation of Distribution Algorithm
EcGA	Extended compact Genetic Algorithm
EA	Evolutionary Algorithm
FDA	Factorized Distribution Algorithm
FunCat	Functional Catalogue of gene function prediction
GA	Genetic Algorithms
GO	Gene Ontology
HMC	Hierarchical Multi-label Classification
KNN	k Nearest Neighbors
LS-SVM	Least Squares Support Vector Machines
PSO	Particle Swarm Optimization
PSP	Protein Structure Prediction
QP	Quadratic Programming
RBF	Radial Basis Function
SMOTE	Synthetic Minority Over-sampling Technique
SRM	Structural Risk Minimization
SV	Support Vector
SVM	Support Vector Machine
SVR	Support Vector Regression
TPR	True Path Rule
3-D	Three Dimensional
2-D	Two Dimensional

Chapter 1

Introduction and Motivation

1.1 Machine Learning in Bioinformatics

How to extract useful information efficiently from high throughput biological data is an important task in bioinformatics. The tools and softwares for transforming the biological data into the underlying knowledge should be developed and applied. Effective computational methods capable of handling biological data (such as gene sequences, protein expressions and biological pathways) are very important for drug research and disease understanding [4]. Because of well-known generalization characteristic and outstanding capabilities in dealing with data noise, Machine Learning methods such as Support Vector Machines (SVMs), Evolutionary Algorithms (EAs), Neural Networks (NNs), Markov models and graphical models have been remarkable in handling biological data. [65, 100].

There are several biological domains where machine learning techniques are applied for knowledge extraction from data, such as genomics, proteomics, microarrays, systems biology, evolution and text mining, etc. Among them, applications in genomics and proteomics are our focuses in this thesis. 1) Genomics has been an important research topic. The valid gene sequences is increasing exponentially due to the high throughput technologies. And these sequences should be handled effectively for the sake of mining useful knowledge and information [62, 10, 13]. Assigning biological functions to the genome sequences is a key challenge in Genomics. Machine learning techniques, such as SVMs and NNs, are always used to predict gene functions from a predefined set of possible functions. 2) Proteins play a vital role in the life process. In the proteomic domain, the main application of computational techniques is protein structure prediction. The three-dimensional (3-D) structure of protein is a key feature to decide the protein functions. Protein is a kind of complex macromolecules with thousands of atoms and bounds. Therefore, the number of possible structures

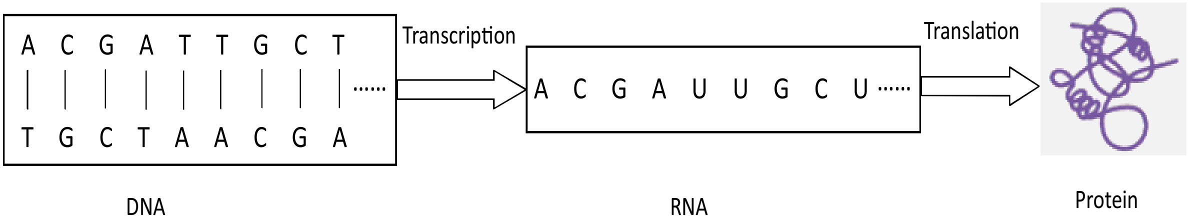


Figure 1.1: Information flows (from DNA to RNA to proteins) in the central dogma of biology.

is huge. This makes protein structure prediction a very complicated combinatorial problem where optimization techniques, such as Genetic Algorithm (GA) and Estimation of distribution algorithm (EDA), are required.

Machine learning strategies always are programming computers to optimize a performance standard by utilizing past experience or example data. The optimized performance standard can be the accuracy index given by a predictive model (in modeling methods, such as NNs, SVM, etc.), and the value of a evaluation or fitness function (in optimization methods, such as GA, EDAs, etc.) [45, 54]. In the modeling method, the "learning" term means to induce a model by utilizing a computer program according to training data or past experience. Statistical theories always are used to build computational models because of the learning objective is to make inferences from training samples. The two important steps in this procedure are to induce the model by processing the training data and to represent the model efficiently. Optimization problem always is the task of finding an optimal solution in a space of multiple (sometimes exponentially sized) possible solutions.

1.2 Gene Function Prediction

1.2.1 Functional Genomics

Functional genomics is a bioinformatics research field that attempts to describe gene (and protein) functions and interactions by utilizing the vast biologic data generated by genomic projects (such as genome sequencing projects). The central dogma of biology is described as that DNA is transcribed into RNA and RNA is translated into proteins. The relationship between the DNA, RNA and protein is showed in Fig.1.1. The gene function usually means the function of the products of genes after transcription and translation, which are proteins.

Genes and ORFs

Genes are defined as the DNA segments which carry the heredity information. This information can produce an organism and decide the organism's characteristics. Gene-finding programs are implemented for finding where the genes locate in a DNA sequence. If a proper stretch of DNA (such as length, start position and end position in DNA sequence, etc.) is identified in gene-finding programs. This stretch of DNA can be signed as an Open Reading Frame (ORF).

DNA

DNA is a long chain molecule which is composed of a backbone of phosphate groups and alternate sugars. A base is utilized for attaching to each sugar. And the base sequences along the backbone produces the code. There are four categories of bases: Adenine (A), Guanine (G), Thymine (T) and Cytosine (C). According to a view of computer science, DNA can be considered as a long letter string with four letters A, G, T and C.

Encoding and replicating the information required to produce proteins are the main functions of DNA. All the approximately twenty amino acids that produce proteins are coded by different combinations of four DNA bases. A codon is defined as each triple of DNA. Three of the triples are utilized for encoding "stop" codons, which indicate the cellular reading code mechanism where to stop.

DNA is double stranded structure which two long chain of molecules entwined together in the well-known double helix. The complementary base pairing exists in two strands. Each A in one strand is paired with a T in the other and each C with a G.

RNA

RNA is utilized for translating DNA to proteins. RNA is a nucleic acid which its structure very likes to DNA but only single stranded. There are four bases of RNA: A, G, C and U (Thymine is replaced by Uracil). RNA undertakes several important roles in a cell. Taking a copy of one of the strands of DNA is the primary role of RNA. This messenger RNA can undertake splicing to remove non-coding regions (*introns*) of a gene. At last, the base sequences of RNA are translated into amino acids for producing the proteins.

1.2.2 Gene Function Prediction

Determining the functions of genes and proteins is a foundation for diseases understanding and drug discovery. It can help for new drug targets identification and reliable diagnostics development. It also can help to understand the molecular and biochemical processes that cause disease or maintain health [78]. The completion of several genome projects in the past decade has generated the full genome sequence of many organisms. Assigning biological functions to the sequences has become a key challenge in modern biology. This step is often implemented through cooperation of computational tool guidance and laboratory experiments. Based on a predefined possible functions sets (such as the FunCat and Gene Ontology), computational techniques are utilized to predict gene functions. Then, the functions with highest confidence prediction are further tested in laboratory experiments [78].

Many computational methods are used in functional genomics. The most common method of determining the function of a protein is to use a sequence similarity program to infer function by orthologous homology. Unfortunately, the sequence similarity method is always unavailable and unpractical. By sequence comparison, gene is annotated as some functions, but it can not indicate clearly where the those functions information came from. In addition, the error of original sequence annotation may be propagated to the future gene function prediction by sequence similarity program.

Recently, some general prediction methods are proposed by using data mining and machine learning to induce rules predicting function from a variety of data sources [3, 74, 6]. Some comprehensive statistic and analysis information, such as percentage of amino acids in the protein and molecular weight of the protein, are abstracted from gene sequence. The obtained statistic and analysis information are utilized as sample attributes for data mining and machine learning techniques. An example of the yeast genome (S. cerevisiae) benchmark dataset attributes is showed in Tab. 1.1. These methods are more general and accurate than conventional sequence similarity methods [25].

From the view of machine learning, the gene function predictions can be seen as multi-label or hierarchical multi-label classification tasks. Different from common machine learning tasks, there are two characteristics of the function predictions: 1) there are multiple functions associated with a single gene; 2) the hierarchy constraint may exist in gene functions: a gene that belongs to a certain function automatically belongs to all its ancestor functions. In practice, if the hierarchical information of functions can not be obtained or provided, the prediction task is a multi-label classification problem.

The Gene Ontology (GO) [3] and the Functional Catalogue (FunCat) [74] are two main taxonomies for gene function. The GO taxonomy is consisted of functional classes organized by a

Table 1.1: Part attributes of the yeast genome (S. cerevisiae) benchmark datasets.

No.	Attribute	Description	Type
1	aa-rat-pair-X-Y	Percentage of the pair of amino acids X and Y consecutively in the protein	real
2	seq-len	Length of the protein sequence	integer
3	theo-pl	Theoretical pl (isoelectric point)	real
4	atomic-comp-X	Atomic composition of X where X is c (carbon), o (oxygen), n (nitrogen), s (sulphur) or h (hydrogen)	real
5	mol-wt	Molecular weight of the protein	integer
6	aliphatic-index	The aliphatic index	real
7	aa-rat-X	Percentage of amino acid X in the protein	real
8	hydro	Grand average of hydrophobicity	real
...	...	...	...

DAG (directed acyclic graph). There are three separated ontologies in GO: “Biological Processes”, “Molecular Function” and “Cellular Component”. Biology knowledge indicate that a gene may participate in different biological processes and may perform different biological functions. For example, a certain gene perform specific molecular functions in specific cellular components (such as binding activities or catalytic that occur at the molecular level in mitochondrion or rough endoplasmic reticulum), at the same time, it also can participate to specific biological processes (such as metabolism, cell cycle and nucleotide biosynthesis) [3]. Different from GO, the FunCat is a more simple gene functional taxonomy. There are 28 main functional branches that contain general fields such as metabolism, cellular transport and cellular communication in FunCat. These main functional branches are divided into a set of subclasses which are organized by a tooted tree structure. There is a hierarchy constraint in FunCat taxonomy: a gene that belongs to a certain function automatically belongs to all its ancestor functions [90]. A small part of the FunCat taxonomies is showed in Fig. 1.2.

1.3 Protein Structure Prediction on HP Model

1.3.1 Protein and Protein Folding

Investigating the proteins’ roles in cell is very important to understand the operation of the whole cell. Biology knowledge indicate that proteins are extremely important molecules. Proteins participate almost all the works in the cell, such as immunity, transportation, muscle structure, metabolism,

```
1 Matabolism
1.1 amino acid metabolism
1.1.3 assimilation of ammonia, metabolism of the glutamate group
1.1.3.1 metabolism of glutamine
1.1.3.1.1 biosynthesis of glutamine
1.1.3.1.2 degradation of glutamine
...
1.2 nitrogen, sulfur, and selenium metabolism
...
14 Protein fate (folding, modification, destination)
14.01 protein folding and stabilization
14.04 protein targeting, sorting and translocation
14.07 protein modification
14.07.01 modification with fatty acids
14.07.02 modification with sugar residues
14.07.02.02 N-directed glycosylation, deglycosylation
14.07.03 modification by phosphorylation
...
14.13 protein/peptide degradation
...
```

Figure 1.2: A small part of the hierarchical FunCat classification scheme [64]

hormones, repair, respiration and control of genes.

As we all know, a protein molecule is a string of amino acids that are connected by peptide bonds. There are approximately twenty different types of amino acids in protein. For computational convenience, a protein always is represented by a character string in computational algorithms.

The protein conformation means how the long chain of amino acids folds in 3-D space. Many research works have been implemented on protein structure determination because of it is associated with the functions of the protein. Investigating protein conformation is a key to understanding its interaction with RNA, DNA and enzyme. The protein conformation information can indicate essential knowledge for protein engineering and drug design.

A protein can be described in terms of its four main structural descriptions as follows. 1) **Primary Structure** refers to the linear sequence of protein amino acid units. These sequences act as instruction for protein, and are results of the human genome project. 2) **Secondary Structure** presents a protein by substructures or regular patterns in the polypeptide backbone. These substructures or regular patterns are defined as one of three motifs (alpha-helix, random coil and beta-sheet).

3) **Tertiary Structure** describes the 3-D conformation of protein molecule. A protein can not realize functional until its tertiary structure is formed. The spatial relationships between the proteins' secondary substructures are presented in the tertiary structure. 4) **Quaternary Structure** describes the molecular relationships and structures when a group of protein molecules are assembled for forming larger molecules.

The protein three-dimensional tertiary structure is defined and folded by the protein primary form. Some environmental conditions (such as the specific chaperon or helper molecules and the correct pH condition) are required in the protein self-folding procedure. Some means of thermal or chemical kinetics can be utilized to unfold a protein. When the agent is withdrawn, some proteins also can refold into their tertiary structures. A protein self-folding process can be completed in a few microseconds. Some large protein molecules may spent minutes or hours for self-folding.

Protein structure prediction is a research area that focuses on accurately predicting the proteins' tertiary structures based on their primary structures. In the conventional structure elucidation biological experiment for protein tertiary structure, Nuclear Magnetic Resonance (NMR) spectroscopy and X-ray crystallography are utilized to identify the positions of atoms within the molecular environment. However, the conventional biological method is a slow and expensive process. As a result, only a few proteins' structures are experimentally determined while the number of known protein sequences exceeds millions. Therefore, utilizing computer programs to predict the protein conformations from protein sequences becomes a key means for uncovering proteins' 3-D structures and functions [4, 48, 9].

Under specific conditions, the protein sequence folds into a unique native 3-D structure. Each possible protein fold has an associated energy. According to the thermodynamic hypothesis, a native protein conformation corresponds to the one which the free energy fulfils the global minimum. Based on this hypothesis, many methods that search for the protein native structure define an approximation of the protein energy and use optimization algorithms that look for the protein fold that minimizes this energy. These approaches mainly differ in the type of energy approximation employed and in the characteristics of the protein modeling. Due to the complexity of PSP task, some simplified models (such as Dill's HP-lattice [56] model) are always used as tools to investigate the properties of protein conformation.

1.3.2 Hydrophobic Polar (HP) Model

According to the assumption of thermodynamical approaches to the protein tertiary structure prediction, a protein folding procedure searches the global minimum of Gibbs free energy. The simplest

Table 1.2: Assignment of residues as Hydrophobic or Polar.

Amino Acid	1-Letter	Assignment	Amino Acid	1-Letter	Assignment
Alanine	A	Hydrophobic	Arginine	R	Polar
Glutamine	Q	Polar	Glycine	G	Hydrophobic
Cysteine	C	Hydrophobic	Glutamic acid	E	Polar
Histidine	H	Polar	Isoleucine	I	Hydrophobic
Proline	P	Hydrophobic	Serine	S	Hydrophobic
Methionine	M	Hydrophobic	Phenylalanine	F	Hydrophobic
Threonine	T	Hydrophobic	Tryptophan	W	Hydrophobic
Asparagine	N	Polar	Aspartic acid	D	Polar
Tyrosine	Y	Hydrophobic	Valine	V	Hydrophobic
Leucine	L	Hydrophobic	Lysine	K	Polar

models of this assumption further simplifies that the interactions between hydrophobic amino acids are the major contributions to the free energy of a protein's native folding [31]. Biology knowledge indicate that the hydrophobic amino acids in a protein sequence tend to push together, on the contrary, the hydrophilic amino acids are inclined to push on the molecule outside [30]. This well-studied simplified model for protein folding is called as hydrophobic polar (HP) model. [56].

Commonly studied proteins consists of approximately 20 different amino acids (as showed in Tab. 1.2). Only two properties of amino acid information in the sequence, whether the amino acid belongs to hydrophobic (H) or belongs to polar (P)(also known as hydrophilic), are considered in HP model. Table 1.2 shows the assignment of residues as hydrophobic or polar [31]. The string of amino acids are folded on a 2-D or 3-D lattice for seeking conformations which the number of topologically adjacent $H - H$ neighbors achieves the global maximum.

In HP model, a protein is considered as a sequence $S \in \{H, P\}^+$, where H represents a hydrophobic residues and P represents a hydrophilic or polar residues. The HP model restricts the space of conformations to self-avoiding paths on a lattice in which vertices are labeled by the residues. Two neighbor relations are considered for a pair of residues. The one is connected neighbor (they are adjacent in the chain). The other is topological neighbor (they are adjacent in the lattice but not connected in the chain). Let ε_{HH} denote the interaction energy between topological neighbor of two H residues, ε_{PP} for two P residues, ε_{HP} for a H residue and a P residue. An energy function is defined as the total energy of topological neighbors with $\varepsilon_{HH} = -1$ and

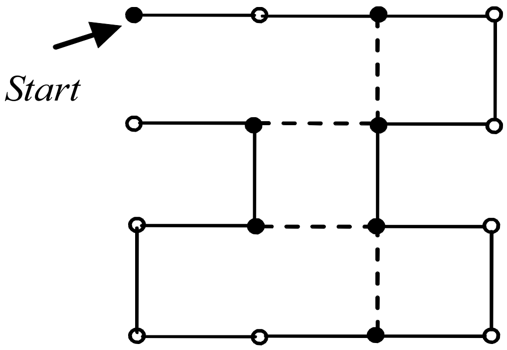


Figure 1.3: One possible configuration of the sequence $HPHPPHHPPHPPPHHP$ in 2-D HP model. There are four HH topological neighbors (represented by broken lines).

$\varepsilon_{PP} = \varepsilon_{HP} = 0$. The HP problem is to find the folding conformation that minimizes the total energy $E(x)$. Figure 1.3 shows the graphical representation of a possible configuration for sequence $HPHPPHHPPHPPPHHP$ in 2-D HP model, hydrophobic residuals are represented by black beads and polar residuals by white beads. The energy that the HP model associates with this configuration is -4. The problem of finding such a minimum energy configuration has been proved to be NP-complete for the 2-D and 3-D lattices [27, 8].

The HP model cannot be used to describe real protein or demonstrate folding of real protein. In HP model, all amino acid properties are simplified by two binary properties and all protein structure properties are simplified by a discrete lattice. Although more complex models have been proposed, the HP model remains a focus of research in computational biology, chemical and statistical physics. By varying the energy function and the bead sequence of the chain (the primary structure), effects on the native state structure and the kinetics (rate) of folding can be explored, and this may provide insights into the folding of real proteins. In particular, the HP model has been used to investigate the energy landscapes of proteins, i.e. the variation of their internal free energy as a function of conformation. In evolutionary computation, the model is still employed as benchmark problem because of its simplicity and its usefulness as a test-bed for new evolutionary optimization approaches [75].

1.4 Challenges

For implementing machine learning techniques into difficult bioinformatics applications, the novel characteristics of the applications will require novel modifications of existing algorithms and procedures, and in some invention of entirely new techniques.

1.4.1 Challenges in Prediction of Gene Function

As mentioned in previous sections, the gene function predictions are considered as multi-label or hierarchical multi-label classification tasks (if hierarchical information of functions are provided). Multi-label classification problem is an extension of traditional multi-class classification problem in which the classes are not mutually exclusive and each sample is related with several classes simultaneously. In this problem, training samples may belong to a label subset and the classification goal is to predict the label subset for the new instances. Hierarchical multi-label classification is a special variant of multi-label classification where the classes are structured by a hierarchy: an instance that is associated with one class automatically associated with all its ancestor classes.

In this thesis, the task of multi-label (or hierarchical multi-label) classification is transferred into a series of separate basis binary classification tasks, and one SVM classifier is trained to learn one class, then results of all binary classifiers are combined by considering the multi-label relationships between classes (and hierarchical constraints in hierarchical multi-label problem). According to the characteristics of gene function prediction, there are many challenges for classification methods in three different levels:

- For the basis binary SVM classifier, there are some special datasets (such as FunCat Yeast datasets [6] and Genbase motif-based datasets [32]), the nonlinear (RBF-kernel or other nonlinear kernels) SVM models can not obtain proper classification results. These tasks are nonlinear problems with characteristics of high noise and large number of input features compared with the relatively small number of training examples, the conventional nonlinear kernel SVM models are severely overfitting at times [6].

- For the multi-label classification problem, the classes are not mutually exclusive and each sample may belong to several classes simultaneously. The main challenge of this problem is that classes are usually overlapped and correlated. Generally, traditional multi-class learning algorithms cannot work with multi-label problem effectively.

- For the hierarchical multi-label classification problem, the task is characterized by hundreds of classes structured according to a predefined hierarchy; and the example distributions of functional classes are usually high degree skewed (imbalanced) with negative more than positive. In such imbalance dataset learning, standard classifiers tend to be overwhelmed by the majority class and ignore the minority one.

1.4.2 Challenges in Prediction of Protein HP model

The problem of protein structure prediction on HP model has been proved to be NP-complete for the 2-D and 3-D lattices. Therefore, a deterministic approaches is always not practical for this problem. Many GA based methods have been proposed to solve the protein structure prediction on HP model in recent years [89, 41, 82, 26, 22]. Among them, the EDAs that use Markov probabilistic model or other probabilistic models proposed by Ref. [75] outperform other population-based methods when solving the HP model folding problem, especially for the long sequence protein instances. But the EDAs based methods still have some disadvantages as follows.

- For most long sequence protein instances, the chance of finding the global optimum is very low, and the algorithm often need be set by very large generation number and population size for finding the global optimum. For some deceptive sequences, those methods can only find the suboptimum solutions.
- In those EDAs based methods, a backtracking method is used to repair invalid individuals sampled by the probabilistic model of EDAs. For a traditional backtracking algorithm, the computational cost of repairing procedure is very heavy for those long sequence instances.
- For those optimization problems with irregular and complex multimodal landscapes, the EDAs still suffer from the drawback of premature convergence similar to other Evolutionary Algorithms (EAs). A simple yet popular explanation for the occurrence of premature convergence is the loss of diversity. But high diversity can not always guarantee a good performance for EA. It is a challenge which is called as the question of exploration vs. exploitation. At the early stage of evolution, enhancing diversity measures can ensure EA to implement a broad exploration in the search space. But such enhancement may be counterproductive or even destructive for EA at the stage when high exploitation is necessary. Therefore, implementing a modifiable balance between exploration and exploitation is very important for this issue.

1.5 Goals of the Thesis

For complicated tasks in bioinformatics, applying machine learning methods simply usually cannot obtain expectable results. The different characteristics and challenges of the tasks will require novel modifications of existing machine learning methods. In this thesis, improved methods based on machine learning methods are developed and applied to the prediction of gene function and protein structure. More precisely, SVM based multi-label and hierarchical multi-label classification methods are developed for solving the gene function prediction problems, and hybrid EDA based methods are proposed for solving the protein HP model problems. According to the characteristics of applications, the prior knowledge of tasks and the information of biological data are extracted by some delicate techniques. The obtained knowledge and information are used to enhance machine learning methods.

The work presented here aims to assess the performances of the improved methods incorporating prior knowledge of application problems. The thesis also shows how the improved methods handle the aforementioned challenges inherent in the gene function prediction and protein structure prediction.

1.6 Thesis Outlines and Main Contributions

This thesis presents my work that has been done over the last three years. It consists of seven chapters. Chapter 1 gives a background and an outline for the whole thesis. Chapter 2 introduces a novel support vector machine with composite kernel for solving classification tasks in gene function prediction. Chapter 3 is devoted to multi-label classification based on label ranking and delicate boundary SVM for Functional Genomics. Chapter 4 introduces a hierarchical multi-label classification based on over-sampling and hierarchy constraint for FunCat gene function prediction. Chapter 5 proposes a hybrid EDA for protein structure prediction based on HP model. Chapter 6 proposes an adaptive niching EDA with balance searching based on clustering analysis. Finally, Chapter 7 gives a summary for the whole thesis. The flow of this thesis is depicted in Fig. 1.4.

This thesis summarizes the research on the predictions of gene function and protein structure. Thesis outlines and main contributions are listed as follows.

Chapter 2 proposes a composite kernel based SVM (ck-SVM) model for solving nonlinear classification tasks in gene function prediction. The proposed ck-SVM method realizes a nonlinear separating boundary by estimating a series of piecewise local linear boundaries. A local linear multi-SVM model is estimated by a composite kernel with incorporating prior knowledge of training data. Firstly, according to the distribution information of training data, a partitioning approach composed of separating boundary detection and clustering technique is used to obtain local subsets from training data. Secondly, a composite kernel is introduced to realize a local linear multi-SVM model by incorporating prior knowledge mined from each training subsets. Instead of building multiple local SVM models separately, the prior knowledge of local subsets is used to construct a composite kernel, then the local linear multi-SVM model

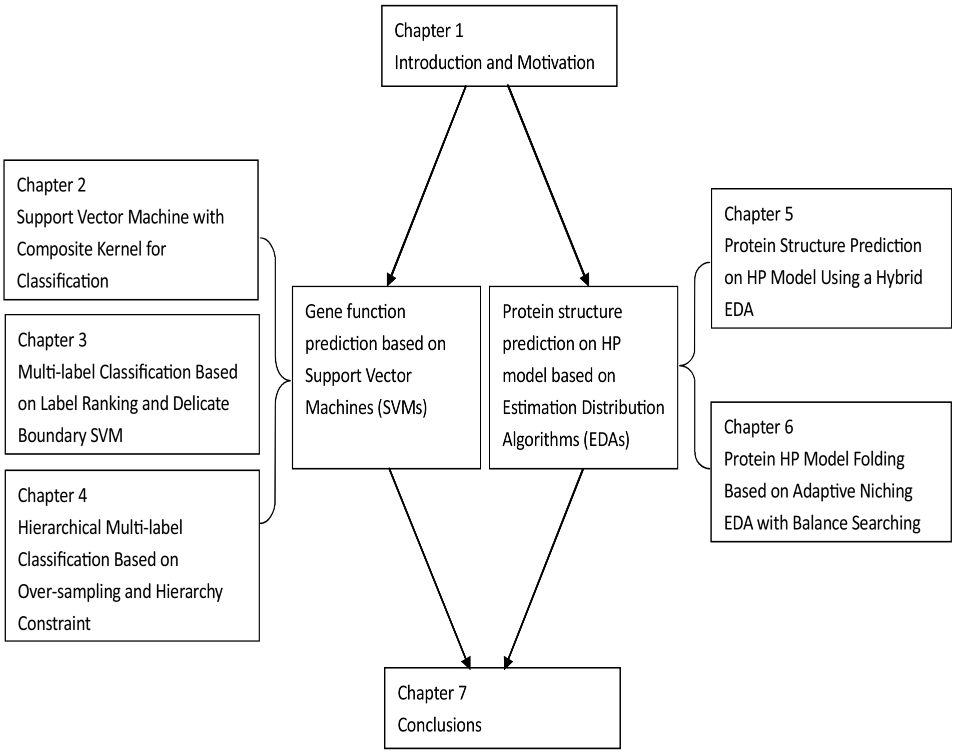

Chapter 1
Introduction and Motivation
Chapter 2
Support Vector Machine with Composite Kernel for Classification
Chapter 3
Multi-label Classification Based on Label Ranking and Delicate Boundary SVM
Chapter 4
Hierarchical Multi-label Classification Based on Over-sampling and Hierarchy Constraint
Gene function prediction based on Support Vector Machines (SVMs)
Protein structure prediction on HP model based on Estimation Distribution Algorithms (EDAs)
Chapter 5
Protein Structure Prediction on HP Model Using a Hybrid EDA
Chapter 6
Protein HP Model Folding Based on Adaptive Niching EDA with Balance Searching
Chapter 7
Conclusions

Figure 1.4: Flow diagram of this thesis

is realized exactly in the same way as a single SVM model by using the composite kernel. The main contributions related to this composite kernel based SVM model are that:

- A novel composite kernel based SVM is proposed to realize a local linear multiple SVM system. Instead of building multiple local SVM models separately, the local subsets is used to estimate a composite kernel, and the composite kernel is utilized to realize the local linear multi-SVM model by implementing the structural risk minimization in the same way as a single standard SVM.
- A partitioning approach composed of separating boundary detection and clustering technique is used to obtain local subsets from training data. By considering sample label changes in neighbor samples, the separating boundary detection can select the training samples which near to separating boundary. It can avoid trivial partitioning produced by implementing unsupervised clustering approach on original training data.

Chapter 3 proposes an improved multi-label classification based on label ranking and delicate decision boundary SVM. Firstly, an improved probabilistic SVM with delicate decision boundary is used as the scoring method to obtain a proper label rank. It can improve the probabilistic label rank by introducing the information of overlapped training samples into learning procedure. Secondly, an instance-dependent thresholding of relating with input instance and label rank is proposed to decide the classification results. It can estimate an appropriate threshold for each testing instance according to the characteristics of instance and label rank.

The main contributions related to this multi-label classification method are that:

- A novel scoring method considering the information of overlapped training samples is proposed to solve the multi-label classification problem. The range of overlapping sample space is identified according to the relative information of two separating surfaces firstly. Then, a bias model with delicate decision boundaries is used to improve the classification accuracy of overlapping samples.
- An instance-dependent thresholding of relating with input instance and label rank is proposed to decide the classification results. By applying d-folds validation on training data, a set of target thresholds for all training samples is determined as teachers, then the instance-dependent thresholds for testing instances are obtained by implementing KNN strategy on this teacher threshold set.

Chapter 4 proposes an improved hierarchical multi-label classification method based on oversampling and hierarchy constraint for solving the gene function prediction problem. The hierarchical multi-label classification task is transferred into a series of binary SVM classification tasks. Then, two measures are implemented to improve the classifier performance by introducing the hierarchy constraint into learning procedures. Firstly, for imbalanced functional classes, a hierarchical SMOTE is proposed as over-sampling preprocessing to improve the SVM learning performance. Secondly, an improved True Path Rule (TPR) consistency approach is introduced to ensemble the results of binary probabilistic SVM classifications. It can improve the classification results and guarantee the hierarchy constraint of classes.

The main contributions related to this hierarchical multi-label classification method are that:

- For imbalance classes in gene function prediction datasets, a hierarchical SMOTE approach is proposed to preprocess the imbalanced training subsets for classifiers. It can improve the performance of classifiers by changing the distribution of the imbalanced datasets.

- An improved TPR consistency approach is used to combine the results of binary probabilistic SVM classifiers. A performance weight is introduced into TPR method, it can restrict the poor performance binary classifiers bring an error propagation effect to other classifiers.

Chapter 5 proposes a novel hybrid Estimation of Distribution Algorithm (EDA) to solve the H-P model problem. Firstly, based on the information of folding structure core (H-Core), a composite fitness function containing H-Core information is proposed to replace the traditional fitness function in HP model. The proposed fitness function is expected to select better individuals for probabilistic model of EDA. Secondly, local search with guided operators is utilized to refine found solutions for improving efficiency of EDA. Thirdly, an improved backtracking-based repairing method is proposed to repair invalid individuals sampled by the probabilistic model of EDA. It can significantly reduce the number of backtracking searching operation and the computational cost for long sequence protein.

The main contributions related to this hybrid EDA are that:

- a composite fitness function containing H-Core information is proposed to replace the traditional fitness function in HP model. The proposed fitness function is expected to select better individuals for probabilistic model of EDAs algorithm. It can help to increase the chance of finding the global optimum and reduce the complexity of EDA (population size and the number of generation needed).
- Local search with guided operators is utilized to refine the found solutions for improving efficiency of EDA. Both the global information about the search space and the local information of solutions found so far can be utilized to enhance the efficiency of evolutionary algorithm.
- For the long sequence protein sequences, an improved backtracking-based repairing method is proposed to repair invalid individuals generated by the EDA probabilistic model. The traditional backtracking repairing procedure will produce heavy computational cost for searching invalid closed-areas of folding structure. To avoid entering invalid closed-areas, a detection procedure for feasibility is introduced when selecting directions for the residues in backtracking searching procedure. It can significantly reduce the number of backtracking searching operation and the computational cost for the long protein sequences.

Chapter 6 proposes an adaptive niching EDA based on Affinity Propagation (AP) clustering analysis. The AP clustering is used to adaptively partition the niches and mine the searching information from the evolution process. The obtained information is successfully utilized to improve the EDA performance by using a balance niching searching strategy. The proposed adaptive niching EDA is used for solving the protein HP model folding. In addition, three benchmark functional multimodal optimization problems based on continuous EDA with single Gaussian probabilistic model are also used to evaluate the proposed method in continuous situation.

The main contributions related to this adaptive niching EDA are that:

- An adaptive niching EDA is proposed, AP clustering is used to adaptively partition the population into niches during a run of EDA. The individuals are clustered before submitting them to niching clearing. A cluster can be seen as a niche, and the niche number and the niche radiuses may vary obviously for different generations.
- A mechanism of niche capacity selection based on the Boltzmann scheme is proposed to realize a balance searching between exploration and exploitation.

Chapter 7 concludes this work, summarizes the thesis and gives suggestions for further research.

Chapter 2

Support Vector Machine with Composite Kernel for Classification

2.1 Introduction

Support Vector Machines (SVMs) have been widely used in different application areas and become the state of the art. Different from most of conventional methods to minimize the training error, the Structural Risk Minimization (SRM) principle is implemented in SVM to minimize an upper bound of the generalization error [91]. This difference eventually results in some remarkable characteristics for SVM, such as the good generalization performance, the absence of local minima and the sparse representation of solution.

By mapping the input data onto a higher-dimensional feature space in a non-linear fashion and seeking an optimal separating hyperplane in the feature space, SVMs can deal with linearly inseparable classification problems, and the feature mapping can be done implicitly through the kernel trick. Nonlinear SVMs employ sophisticated kernel functions, such as radial basis function (RBF), polynomial functions, etc., to fit datasets with complex decision surfaces. However, as many other nonlinear classification methods, nonlinear kernel SVM models also face the potential overfitting issue when the number of training examples is small due to their large VC dimensions.

In functional genomics, there are some classification tasks (such as FunCat Yeast datasets [6] and Genbase motif-based datasets [32]), the nonlinear (RBF-kernel or other non-linear kernels) SVM models can not obtain proper classification results. These tasks are nonlinear problems with characteristics of high noise and large number of input features compared with the relatively small number of training examples, the conventional nonlinear kernel SVM models are severely overfitting at times [6]. Composite kernel techniques are always used to deal with some special classification tasks [71, 57, 12, 50, 86], prior knowledge of training data can be introduced into SVM classifier by composite kernel techniques. In this chapter, a new multi-SVM system called composite kernel based SVM (ck-SVM) is proposed for solving those gene function classification tasks.

A nonlinear separating boundary can be approximately seen as an aggregation of piecewise local linear boundaries. The proposed ck-SVM method realizes a nonlinear separating boundary by estimating a series of piecewise local linear boundaries. A local linear multi-SVM system is estimated by a composite kernel with incorporating prior knowledge of training data. 1) According to the distribution information of training data, a partitioning approach composed of separating boundary detection and clustering technique is used to obtain local subsets from training data. By considering sample label changes in neighbor samples, the separating boundary detection can select the training samples which are near to separating boundary. The selected samples are partitioned into local subsets by Affinity Propagation (AP) clustering, and each subset is utilized to capture prior knowledge of corresponding local linear boundary. 2) A composite kernel is introduced to realize a local linear multi-SVM system by incorporating prior knowledge mined from each training subsets. Instead of building multiple local SVM models separately, the prior knowledge of local subsets is used to construct a composite kernel, then the local linear multi-SVM system is realized exactly in the same way as a single SVM model by using the composite kernel. Experimental results on benchmark datasets demonstrate that the proposed method improves the classification performance efficiently.

The proposed method falls in the category of multiple linear SVM based on subspace pattern recognition. In some previous works, Ref. [21] presented Localized Support Vector Machine which builds multiple linear SVM models from training data and each model is designed to classify a particular test example. Ref. [39] introduced Mixtures of Linear SVMs by packaging linear SVMs into a probabilistic formulation and embedding them in the mixture of experts model. Different from those methods, in the proposed ck-SVM method, first, the subsets for capturing local linear knowledge are only clustered from samples which are near to separating boundaries. It can avoid trivial partitioning produced by implementing unsupervised clustering approach on original training data. Second, a composite kernel incorporating prior knowledge is introduced to realize the local linear multi-SVM system by estimating a series of piecewise linear boundaries. Instead of building multiple local SVM models separately, the partition information of local subsets is used to build a composite kernel, and the composite kernel is utilized to realize the local linear multi-SVM system by implementing the structural risk minimization in the same way as a single standard SVM.

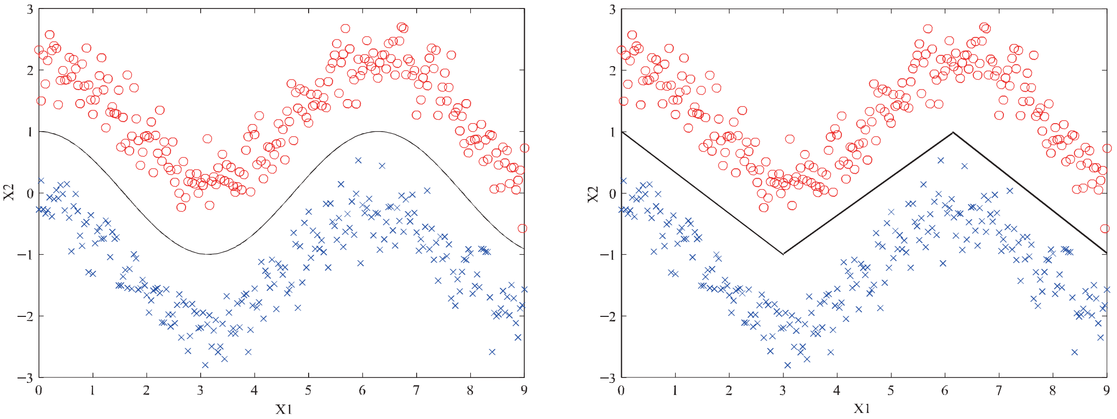


Figure 2.1: An example of multiple linear separating boundaries

The rest parts of the chapter are organized as follows. Section 2.2 formulates the proposed composite kernel based SVM model. Section 2.3 describes estimation of the proposed model. Section 2.4 presents the Functional Genomics benchmark datasets and the results of experimental performance evaluation. Finally, the conclusions and future work directions are discussed.

2.2 A Composite Kernel Based SVM (ck-SVM) Model

2.2.1 Formulations of Model

In this chapter, the binary classification problem is considered. The classifier is built from a given labeled training dataset of N samples

$$(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_N, y_N) \tag{2.2.1}$$

where $x_i \in R^d$ is the input vector corresponding to the i-th sample labeled by $y_i \in \{-1, +1\}$ depending on its class.

As showed in Fig. 2.1, a non-linear separating boundary can be seen as an aggregation of M piecewise local linear boundaries $\Omega_j^T x + b_j, j = 1, \ldots, M$. According to the piecewise linear approximation method, the piecewise linear model $f_P(x)$ can be written compactly as follow.

$$f_P(x) = \sum_{j=1}^{M} (\Omega_j^T x + b_j)\mathcal{R}_j(x) + b \tag{2.2.2}$$

where $\mathcal{R}_j(x)$'s are the basis function, Ω_j's are the coordinate parameter vectors of local linear boundaries. The role of basis function is similar to that of a functional space basis. In some particular situations, they do constitute a functional basis. Typical examples are wavelet basis and RBF basis. The overall performance of the the piecewise linear model is obtained via an interpolation using the basis function $\mathcal{R}(x)$. It also implies that the piecewise linear model $f_P(x)$ can describe any sufficiently smooth nonlinear separating hyperplane function on a compact interval arbitrarily well by merely increasing the value of M.

Two parameter vectors $\Phi(x)$ and Θ are defined as follows.

$$\begin{aligned} \Phi(x) &= [\mathcal{R}_1(x), x\mathcal{R}_1(x), \cdots, \mathcal{R}_M(x), x\mathcal{R}_M(x)]^T \\ \Theta &= \left[b_1, \Omega_1^T, \cdots, b_M, \Omega_M^T\right]^T \end{aligned} \tag{2.2.3}$$

Introducing parameter vectors $\Phi(x)$ and Θ (Eq. 2.2.3) into Eq. 2.2.2, the piecewise linear model $f_P(x)$ can be rewritten as:

$$f_P(x) = \Theta^T\Phi(x) + b \tag{2.2.4}$$

By introducing the Structural Risk Minimization principle [28], the piecewise linear model $f_P(x)$ (Eq. 2.2.4) can be written as the QP optimization problem as

$$\begin{aligned} \min_{\Theta,b,\xi} \mathcal{J}_P &= \frac{1}{2}\Theta^T\Theta + c\sum_{k=1}^{N}\xi_k \\ \text{s.t.} & \quad \begin{cases} y_k[\Theta^T\Phi(x_k) + b] \geq 1 - \xi_k, k = 1, \ldots, N \\ \xi_k \geq 0, k = 1, \ldots, N \end{cases} \end{aligned} \tag{2.2.5}$$

The Lagrangian is constructed:

$$\mathcal{L}(\Theta, b, \xi; \alpha, v) = \mathcal{J}_P(\Theta, \xi) - \sum_{k=1}^{N}(\alpha_k y_k[\Theta^T\Phi(x_k) + b] - 1 + \xi_k) - \sum_{k=1}^{N} v_k\xi_k \tag{2.2.6}$$

with Lagrange multipliers $\alpha_k \geq 0$, $v_k \geq 0$ for $k = 1, \ldots, N$. The solution is given by the saddle point of the Lagrangian:

$$\max_{\alpha,v} \min_{\Theta,b,\xi} \mathcal{L}(\Theta, b, \xi; \alpha, v) \tag{2.2.7}$$

This leads to

$$\begin{cases} \frac{\partial\mathcal{L}}{\partial\Theta} = 0 \rightarrow \Theta = \sum_{k=1}^{N}\alpha_k y_k\Phi(x_k) \\ \frac{\partial\mathcal{L}}{\partial b} = 0 \rightarrow \sum_{k=1}^{N}\alpha_k y_k = 0 \\ \frac{\partial\mathcal{L}}{\partial\xi_k} = 0 \rightarrow 0 \prec \alpha_k \prec c, k = 1, \ldots, N \end{cases} \tag{2.2.8}$$

The dual problem becomes

$$\begin{aligned}\max_{\alpha} \mathcal{J}_D(\alpha) &= -\frac{1}{2}\sum_{k,l=1}^{N} y_k y_l K(x_k, x_l)\alpha_k \alpha_l + \sum_{k=1}^{N}\alpha_k \\ \text{s.t.} &\quad \begin{cases} \sum_{k=1}^{N}\alpha_k y_k = 0 \\ 0 \le \alpha_k \le c, k = 1, \ldots, N \end{cases}\end{aligned} \tag{2.2.9}$$

where $K(x_k, x_l)$ is a composite kernel defined by

$$\begin{aligned} K(x_k, x_l) &= \Phi(x_k)^T \Phi(x_l) \\ &= (1 + x_k^T x_l) \sum_{j=1}^{M} \mathcal{R}_j(x_k)\mathcal{R}_j(x_l) \end{aligned} \tag{2.2.10}$$

for $k = 1, \ldots, N$. Hence, the piecewise linear model $f_P(x)$ (Eq. 2.2.4) is reduced to a standard SVM based on a composite kernel. Finally the nonlinear SVM classifier takes the form

$$y = \text{sign}[\sum_{k=1}^{N} \alpha_k y_k K(x, x_k) + b] \tag{2.2.11}$$

with α_k positive real constants which are the solution to a QP problem.

From the optimal solution, it is not necessary to know the parameter vector $\Phi(x)$ in expansion (Eq. 2.2.3) explicitly, as the solution only depends on the inner product which defines composite kernel $K(x_k, x_i)$ (Eq. 2.2.10). In order to represent an inner product, the kernel is required to satisfy Mercer's condition [91]. In Eq. 2.2.10, factors $\mathcal{R}_j(x_i)\mathcal{R}_j(x)$ are outputs of basis functions. It follows that the kernel $K(x, x_i)$ satisfies Mercer's condition because it consists of a sum of products of Mercer kernels, cf. [91].

2.2.2 Properties of Model

According to the definitions mentioned in the previous subsection, the proposed ck-SVM model can be seen as a novel local linear multi-SVM system that is skillfully realized by a composite kernel. The prior knowledge of each local linear boundaries are introduced into model by basis function parameters of the composite kernel, and the local linear multi-SVM system is realized exactly in the same way as a single SVM model by using the composite kernel.

When the parameter M in Eq. 2.2.2 is set as 1 specifically, the ck-SVM model becomes a linear SVM. By increasing the value of M, the model can describe any sufficiently smooth nonlinear separating hyperplane function. Therefore, the ck-SVM is a flexibility controllable support vector machine model, it can solve nonlinear tasks by introducing multiple local linear boundaries and avoid over-fitting in certain degree by selecting a proper parameter M.

In conventional multi-SVM methods, the training data is partitioned into several subsets, and each subset is used to estimate a local SVM model separately, then outputs of local SVM models are combined by a certain ensemble mechanism. However, the local SVM models, estimated by local information of subsets, are always not suitable for describing whole nonlinear separating boundary because data noise and other reasons. And designing an effective ensemble mechanism is a difficult task in practice.

In the proposed model, instead of building multiple local SVM models separately, the partition information of local subsets is used to construct a composite kernel, and the composite kernel is utilized to realize the local linear multi-SVM system by implementing the structural risk minimization in the same way as a single standard SVM. Comparing with conventional multi-SVM methods, the proposed model can properly estimate nonlinear separating boundary by learning from global information of whole training data. In addition, by implementing the structural risk minimization as a single SVM and introducing the interpolation based on basis function, the ck-SVM can overcome some drawbacks of ensemble mechanism in conventional methods.

2.3 Estimation of the Proposed ck-SVM Model

2.3.1 Affinity Propagation Clustering

Affinity Propagation (AP) [38] is a recently proposed clustering method. A set of measures of similarity between pairs of data points is used as input parameter in AP. And it outputs a set of clusters of the points with their corresponding exemplars. The AP algorithm takes a matrix of similarity metrics between each pair of points $s(i, k)$ as input parameter. Different from some traditional clustering methods to require a predetermined parameter of cluster number, a real number $s(k, k)$ for each data point k, which are called preferences, are used as input to calculate of how likely each point is to be chosen as exemplar in AP clustering.

The AP clustering procedure is implemented by interchanging messages between the points until a stop condition is satisfied. There are two messages types to be interchanged between data points. The availability message $a(i, k)$, sent from the candidate exemplar point k to the point i, describes the accumulated evidence for how well-suited it would be for the point i to choose the point k as its exemplar. The responsibility message $r(i, k)$, sent from the data point i to the candidate exemplar point k, describes the accumulated evidence for how appropriate the point k is to serve as the exemplar for the point i, comparing with other potential exemplars for the point i.

The availability parameters are initialized as zero: $a(i, k) = 0$. Then, the parameters are calculated and updated according to the rules as follows:

$$r(i,k) \leftarrow s(i,k) - \max_{k^{'} s.t. k^{'} \neq k} \{a(i,k^{'}) + s(i,k^{'})\} \tag{2.3.1}$$

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i^{'} s.t. i^{'} \notin \{i,k\}} \max\{0, r(i^{'},k)\}\} \tag{2.3.2}$$

$$a(k,k) \leftarrow \sum_{i^{'} s.t. i^{'} \neq k} \max\{0, r(i^{'},k)\} \tag{2.3.3}$$

The AP method has been praised because of its ability to efficiently and quickly handle very large problems.

2.3.2 Partitioning Based on Separating Boundary Detection

In order to estimate parameters of the proposed ck-SVM model according to the distribution information of training data, clustering techniques can be used to partition training data into subsets. But our partitioning task is somewhat different from conventional unsupervised clustering. In order to capture the prior knowledge of local linear boundaries properly, the partitioning approach in the ck-SVM should ensure each subset contains training examples from two classes simultaneously. But conventional clustering techniques consider only the proximity between examples and often end up grouping training examples from the same class into the same cluster, because such clusters tend to be pure.

A partitioning approach composed of separating boundary detection and clustering technique is proposed to partition training data into subsets. A separating boundary detection considering sample label changes in neighbor area is used to select samples which are near to the separating boundary firstly. Then, the AP clustering is used to partition the selected samples into subsets for estimating parameters of the ck-SVM model.

An example on synthetic data is used to formulate the proposed partitioning method. As showed in Fig. 2.2(a), a two-class dataset is constructed comprising 339 points in each class, the points are uniformly drawn from some shifted cosine signals in the two-dimensional space and perturbed with Gaussian noise in the vertical direction. Implementing unsupervised clustering technique onto the original data may produce trivial partitioning. The clustering results of implementing AP clustering (the cluster number is set as 3 empirically) onto the synthetic data (678 samples) is showed in Fig. 2.2(b), where samples from different clusters are plotted in different colors. It can be found that the

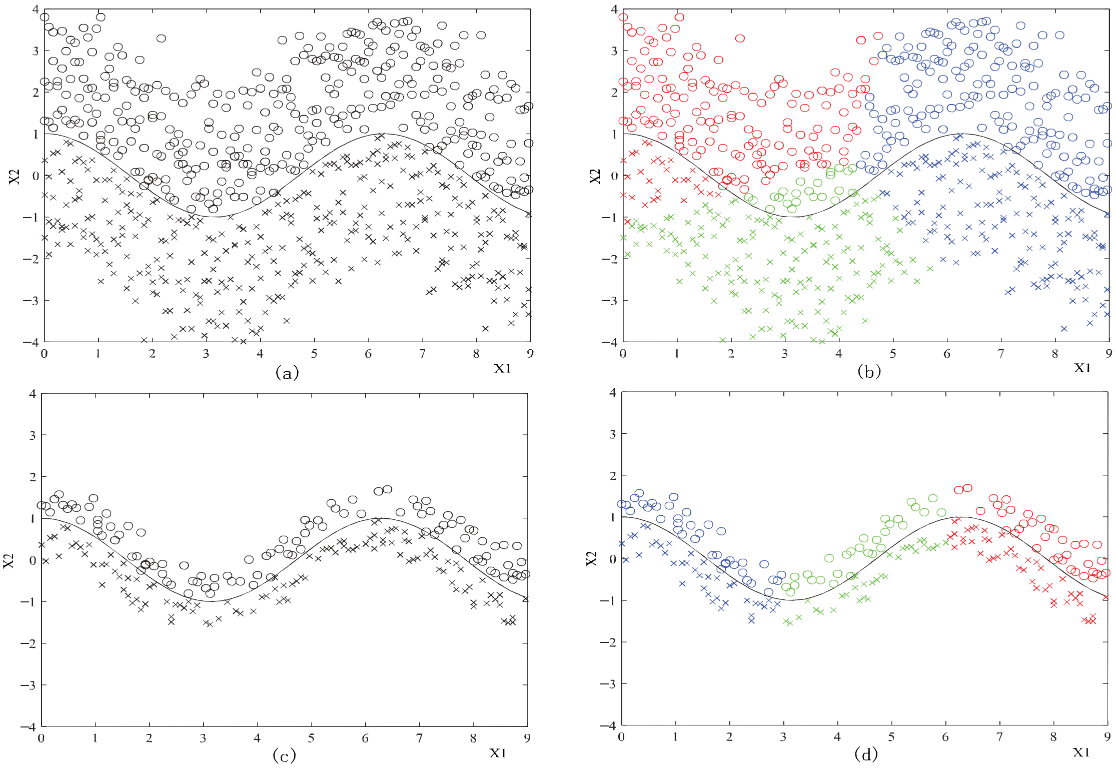


Figure 2.2: An example of partitioning approach. (a) the synthetic data (678 samples) constructed by shifted cosine signals with Gaussian noise; (b) the clustering results of implementing AP clustering onto the original synthetic data; (c) the selected samples (219 samples) by implementing the separation boundary detection; (d) the clustering results of implementing AP clustering onto the selected samples.

partitioning tends to grouping samples from the same class into the same cluster. The partitioned subsets cannot be used to capture distribution information of local separating boundaries properly.

To solve above problem, in the proposed partitioning method, a separating boundary detection is introduced to select samples before implementing clustering. Samples around the boundary are selected by detecting changes of class label. The detection procedure is described as follows. For a certain training data sample P, consider the n nearest neighbor samples, if one of its n neighbor samples has a different class label, then P is selected as a sample around the separating boundary to be reserved. The result of detection on the training data may lead to a subset of samples that distribute around the separation boundary between classes. It can filter out information that may be regarded as less relevant, while preserving the important structural properties near to the separation boundary. The selected result (219 samples) from the synthetic data by implementing the detection is showed in Fig. 2.2(c) and the clustering results on the selected samples is showed in Fig. 2.2(d). It can be found that the subsets of the proposed partitioning is able to capture distribution information in the training data properly.

2.3.3 Estimation of the Proposed ck-SVM Model

According to the definition of the proposed ck-SVM described in previous, basis functions is used to interpolate the piecewise linear hyperplanes, and each basis function corresponds to a local linear hyperplane. Therefore, the basis functions should abstract the distribution information of dataset as accurate as possible. In this chapter, a RBF gaussian function is selected as the basis function to capture distribution information of obtained partitioning subsets.

$$\mathcal{R}(x) = e^{-\frac{(x-\mu)^2}{\lambda\sigma^2}} \tag{2.3.4}$$

where μ is the subset center, σ the radius of subset and λ the scale parameter.

Another important parameter in the proposed model is the number of basis function M. It is decided by the number of partitioning subsets (clusters). Theoretically, the proposed model $f_P(x)$ (Eq. 2.2.10) can describe any sufficiently smooth nonlinear separating hyperplane function on a compact interval arbitrarily well by merely increasing the value of M. If a proper parameter M is selected according with the data distribution (for example, select $M = 3$ for the cosine synthetic data showed in Fig. 2.2(a)), the proposed model will get a well performance. In practice, the AP clustering used in the proposed model is an adaptive clustering approach, and it also can work by user-specified number of clusters.

2.4 Experiments and Results

2.4.1 Experiment Data Sets

In functional genomics, an important problem is predicting the functions of genes (proteins). Currently, the number of protein sequences is constantly increasing, and these protein sequences always are stored in central protein databases from laboratories all over the world. The experimental determination of protein structure is time-consuming and quite labor-intensive. Therefore, only a fraction of these proteins has been experimentally analyzed for the sake of detecting their structure and uncovering their function in the corresponding organism. The automated tools that can classify new proteins to structural families are needed in functional genomics research field.

Table 2.1: Properties of experiment datasets

Dataset	Attribute	Training	Testing
Sequence (seq)	478	2580	1339
All microarray (expr)	551	2488	1291

Table 2.2: Descriptions of classes (labels).

FunCat ID	Description
14.01	Protein folding and stabilization
14.04	Protein targeting, sorting and translocation
14.07	Protein modification
14.10	Assembly of protein complexes
14.13	Protein/peptide degradation

Generally, gene function predictions always are tasks of multi-label classification [34] or hierarchical multi-label classification [6], and SVMs is often used as efficient basis classifier because of their good generalization ability [34, 93, 6]. The purpose of this chapter is to improve the basis classifier performance for each label (class), so only per-class (binary classification) experiments are implemented for evaluating the proposed method. Two yeast datasets ("Sequence" and "All microarray") from FunCat are used to evaluate the proposed method. The different datasets describe different aspects of the genes in the yeast genome, and the different sources of data highlight different aspects of gene function. The properties of experiment datasets, including instance number D and attribute number, are listed in Tab. 2.1, the detailed description of each dataset can be referred from Ref. [25, 92]. The datasets are downloaded from the following webpage: http://dtai.cs.kuleuven.be/clus/hmcdatasets/. Five classes (labels) in "protein fate" FunCat class (FunCat $ID = 14$) are selected to implement per-class (binary classification) experiments, the descriptions of labels are listed in Tab. 2.2,

2.4.2 Evaluation Metrics

Three classical evaluation metrics of Precision, Recall and F-score are used to evaluate the efficiency of the proposed method. The three metrics are introduced for a traditional binary classification model with negative and positive classes. Precision metric is defined as the proportion of positive predictions that are correct, and Recall metric is defined as the proportion of positive instances that are correctly predicted positive. That is:

$$\begin{aligned} Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN} \\ F-score &= \frac{2*Precision*Recall}{Precision+Recall} \end{aligned} \tag{2.4.1}$$

with TP is the number of true positives (correctly predicted positive instances), FP is the number of false positives (positive predictions that are incorrect), and FN is the number of false negatives (positive instances that are incorrectly predicted negative). Note that the above three metrics ignore the number of correctly predicted negative instances because of only positive instances are concerned in gene function prediction.

2.4.3 Experiment Setting and Results

In our experiments, the LSSVM [85] is taken as a basis. SVM parameters are chosen by cross-validation procedure. In partitioning approach, the neighbor parameter n is set as 5 for separating boundary detection. About the parameters of AP clustering (defined in Ref. [38]), the maximum number of iterations is set as 1000, and early terminate parameter is set as 100 (i.e., the clustering procedure will be terminated if the estimated exemplars stay fixed for continuous 100 iterations); the damping factor, which may be needed if oscillations occur, is set as 0.9; Euclidean distance is used for the similarity metric of samples.

The RBF-kernel SVM and linear SVM are used to compare with the proposed ck-SVM method. The experimental results for five labels on two datasets are presented in the Tab. 2.3 and Tab. 2.4. It can be found that the RBF-kernel SVMs are severely overfitting, most of testing instances are predicted as negative and values of the recall metrics are very low. The experimental results of all evaluation metrics demonstrate that the proposed method improves the classification performance efficiently.

2.5 Conclusions

In this chapter, a composite kernel based SVM method with incorporating prior knowledge is presented. The proposed ck-SVM approximates a nonlinear separating boundary by estimating a series of piecewise linear boundaries. A partitioning approach composed of separating boundary detection and clustering technique is used to obtain local subsets from the training data. The separating

Table 2.3: Results of comparing with the RBF kernel and linear SVM for FunCat "Sequence" datset.

	RBF-kernel SVM			Linear SVM			Proposed ck-SVM		
FunCat ID	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
14.01	0.3963	0.0472	0.0843	0.1667	0.3333	0.2222	0.1921	0.3452	0.2468
14.04	0.5326	0.0825	0.1429	0.3131	0.3690	0.3388	0.3548	0.4013	0.3766
14.07	0.7253	0.0623	0.1147	0.6460	0.5474	0.5926	0.6982	0.6053	0.6484
14.10	0.4039	0.0772	0.1296	0.2474	0.4068	0.3077	0.2731	0.3841	0.3192
14.13	0.4532	0.0451	0.0820	0.3048	0.4507	0.3637	0.3122	0.4864	0.3803

Table 2.4: Results of comparing with the RBF kernel and linear SVM for FunCat "All microarray" datset.

	RBF-kernel SVM			Linear SVM			Proposed ck-SVM		
FunCat ID	Prec.	Rec.	F-score	Prec.	Rec.	F-score	Prec.	Rec.	F-score
14.01	0.4092	0.1175	0.1826	0.2292	0.3929	0.2895	0.2571	0.4285	0.3214
14.04	0.4813	0.1281	0.2023	0.4178	0.4449	0.4309	0.4375	0.4603	0.4486
14.07	0.6947	0.1275	0.2155	0.6081	0.5632	0.5848	0.6424	0.6226	0.6323
14.10	0.4187	0.0984	0.1594	0.2843	0.5075	0.3644	0.3071	0.5684	0.3988
14.13	0.4382	0.0946	0.1556	0.2992	0.4524	0.3602	0.3317	0.4805	0.3925

boundary detection can select the training samples which are near to separating boundary by considering sample label changes in neighbor samples. The selected samples are partitioned into local subsets by AP clustering, and each subset is utilized to capture prior knowledge of corresponding local linear boundary. Then, a composite kernel is introduced to realize the local linear model by incorporating prior knowledge mined from each training subsets.

The experimental results demonstrate that the proposed method can solve the special nonlinear classification tasks in gene function prediction, which conventional nonlinear kernel SVMs can not work efficiently.

Chapter 3

Multi-label Classification Based on Label Ranking and Delicate Boundary SVM for Functional Genomics

3.1 Introduction

Multi-label classification problem is an extension of traditional multi-class classification problem in which its classes are not mutually exclusive and each sample may belong to several classes simultaneously. It is increasingly required by many real-world applications. For instance, in text categorization, a document generally has several different topics, such as *social*, *sport* and *health* [63, 77]; in bioinformatics, each gene may belong to several functional classes, such as *transcription*, *metabolism* and *protein synthesis* [34]; in scene classification problem, each scene image may be associated with a set of semantic classes, such as *city* and *beach* [11]. In these cases, instances in the training dataset are related with a set of labels and the classification task is to predict the label set for the unseen instances. The main challenge of this problem is that classes are usually overlapped and correlated. Generally, traditional multi-class learning algorithms cannot work with multi-label problem effectively.

Label ranking strategy is often used in multi-label classification methods [34, 88, 95]. This strategy requires a real-valued score for each class (label) to order a label rank, the labels of higher score values are more related with the new instance, and then classifies a new instance into the classes that ranking score above a threshold. In a ranking based classification, both the scoring method and the threshold selection can influence the classification results significantly. An ideal ranking based classifier can be imagined as follows, the scoring method can provide a proper label rank (i.e., the labels of higher score values are more related with the instance). And the threshold selection strategy can estimate an appropriate threshold according to the characteristics of instance and label ranking strategy.

About the scoring methods, since Support Vector Machine (SVM) based methods have good generalization ability in single-label multi-class problem[79, 20], more attention also has been paid on such kind of techniques for multi-label problem. At present, there are mainly two types of SVM-based methods to solve the multi-label problem. One is to consider all samples and their labels simultaneously to construct one optimization formulation, for example, rank-SVM [34] and maximal margin labeling algorithm [51]. But it is time-consuming to solve such large scale optimization problems in numerical computation. The other is to decompose a multi-label problem into many binary class sub-problems and to solve them by using SVM-like methods. Generally speaking, the latter runs faster than the former does.

Nowadays the widely used decomposition techniques include one-versus-rest and one-versus-one strategies. In one-versus-rest SVM methods [34, 88], the multi-label training set is simply divided into l (the number of labels) binary class subsets, where each positive class consists of those samples with some given label and the other remained samples belong to the corresponding negative class; and then l traditional binary SVM classifiers are trained. The weakness of this strategy is that binary classifiers are trained independently, ignoring the semantic relationship of labels.

For one-versus-one SVM methods, the first step is to split the training set into $l * (l - 1)/2$ binary class subsets. It is noted that in some subsets samples possess double labels (overlapping samples), i.e. belonging to both positive and negative classes at the same time. Some papers show that the performance of classifier can be improved by introducing the information of those double-label training samples into learning procedure effectively. In [80], those double-label samples are processed by the method of triple class support vector machine. In [93], those double-label samples are considered as the third class besides positive and negative classes and so-called parallel support vector machine is designed to cope with such training subsets. The drawback of this strategy is that many binary classifiers need to be built for those datasets with large label number l.

Different from the above methods, in this chapter, a novel scoring method is proposed to solve the multi-label classification problem. Experientially, the basic Binary-SVM method cannot work very well for those overlapping samples because of the complexity of multi-label problem. But there is an important common characteristic of overlapping (double-label) samples in the essentially overlapping classes of two labels. It is that the majority of double-label samples are distributed near to two binary SVM separating surfaces simultaneously. The motivation of the proposed method is that, the range of overlapping sample space is identified according to the relative information of two separating surfaces firstly. Then, a bias model with delicate decision boundaries is used to improve the classification accuracy of overlapping samples.

The scoring methods are the major focus of research in the multi-label classification. However, the threshold selection is often processed as a unimportant post-processing step and so far few studies have done on this subject. For the document classification applications, in [96] and [58], threshold selection techniques are successfully applied to obtain better performance. A per-class threshold selection strategy has been used to improve the performance of the one-versus-all binary Support Vector Machine (SVM) multi-label classification in [36]. It estimate an appropriate threshold value for each class by optimizing macro-average F-measure.

An improved ranking based multi-label classification method is proposed in this chapter. 1) An improved probabilistic SVM with delicate decision boundary is used as the scoring method. In this method, firstly, utilizing the one-versus-rest strategy and the Platt's sigmoid method [69], l binary probabilistic SVM classifiers are trained for l labels correspondingly. For example, training the p-th ($p \in [1, l]$) binary SVM, the samples labeled by p only are considered as "positive", all others are considered as "negative". Secondly, according to the one-versus-one strategy and l binary probabilistic SVMs, $l * (l - 1)/2$ bias models are built by corresponding each pair of labels. The proposed bias model can capture the characteristics of double-label samples and give the more delicate decision boundaries for samples of overlapping sample space. It is expected to obtain more proper probabilistic classifier results by introducing the information of double-label training samples into classifier. Finally, biased probabilistic classification results are ranked to produce label ranking. 2) An instance-dependent thresholding of relating with input instance and label rank is proposed to decide the classification results. By applying d-folds validation on training data, a set of target thresholds for all training samples is determined as teachers, then the instance-dependent thresholds for testing instances are obtained by implementing KNN strategy on this teacher threshold set. Experimental results on the two Functional Genomics benchmark datasets of Yeast and Genbase show that the proposed method improves the classification performance efficiently, compared with binary SVM method and some existing well-known methods.

The rest parts of the chapter are organized as follows. Section 3.2 gives a brief overview of probabilistic outputs of SVM classification and ranking based multi-label classification. Section 3.3 describes the details of the proposed method based on label ranking and delicate decision boundary SVM. Section 3.4 formulates evaluation metrics of multi-label classification problem and the results of experimental performance evaluation. Finally, the conclusions and future work directions are discussed.

3.2 Probabilistic Outputs of SVM Classification and Ranking Based Multi-label Classification

3.2.1 Probabilistic Outputs for SVMs

SVM is a supervised learning method introduced by Vapnik based on his Statistical Learning Theory and Structural Minimization Principle [91]. The basic idea of using SVM for classification is to find the optimal separating hyperplane between the positive and negative samples. The optimal hyperplane is defined as the maximum margin between the training samples that lie closest to it. The samples that lie closest to the separating hyperplane are defined as support vectors. Once this hyperplane is found, new samples can be classified simply by determining on which side of the hyperplane they fall.

Given training data, $T = \{(x_1, y_1), \cdots (x_m, y_m)\}$, where $x_i \in R^n$ is the input vector of training samples, and $y_i \in \{-1, 1\}$ is the class label of sample x_i. The task of classification is to train a classifier $f(X)$, which minimizes an expected misclassification criterion. A linear SVM classifier $f(X)$ is equivalent to solving a convex quadratic optimization problem defined as follow.

$$\min \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{n}\xi_i \tag{3.2.1}$$

subject to $y_i(\langle W, X_i\rangle + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$. The regularization parameter C is used to balance the classifier complexity and the classification precision on the training dataset T. This convex quadratic problem can be generally solved by its dual formulation [28]. The linear SVM can be converted into a more flexible non-linear SVM by simply replacing the involved vector inner-product with a non-linear kernel function [94, 15].

Constructing a classifier to produce a posterior probability p (class—input) is very useful in practical recognition situations. Posterior probabilities are also required when a classifier is making a small part of an overall decision, and the classification output must be combined for the overall decision. However, the standard SVM do not provide such probabilities. The Platt's sigmoid method through training the parameters of an additional sigmoid function to map the SVM outputs into probabilities [69]. The following gives a brief description of this method of probabilistic output for SVM in two classes' case.

Given training data $T = \{(x_1, y_1), \cdots (x_m, y_m)\}$, labeled by $y_i \in \{-1, 1\}$, the binary SVM can obtain a decision function $f(x)$. The signs of $f(x)$ are used to predict all the test samples x. Instead of predicting a label $y_i \in \{-1, 1\}$ for the the test samples x, many applications require a posterior class probability $p(y = 1|x)$. In [69], the probability $p(y = 1|x)$ is approximated by a sigmoid function with parameters A and B defined as follow.

$$p(y=1|x) = \frac{1}{1+\exp(Af(x)+B)} \tag{3.2.2}$$

For estimating the best values of parameters A and B, any subset of m training data can be used to solve the following maximum likelihood problem,

$$\begin{aligned} &\min_{Z=(A,B)} \{-\sum_{i=1}^{l} (t_i \log(p_i) + (1-t_i)\log(1-p_i))\} \\ &p_i = \frac{1}{1+\exp(Af_i+B)}, \quad f_i = f(x_i) \\ &t_i = \begin{cases} \frac{N_+ +1}{N_+ +2} \, if \, y_i = 1 \\ \frac{1}{N_- +2} \, if \, y_i = -1 \end{cases} \quad i = 1, 2, \cdots m \end{aligned} \tag{3.2.3}$$

where N_+ means the number of positive-labeled samples, and N_- means the number of negative-labeled samples.

The method leaves the SVM error function unchanged. Instead, it adds a trainable post-processing step which is trained with regularized binomial maximum likelihood. A Sigmoid function with two parameters is chosen as the post-processing, since it matches the posterior that is empirically observed. The SVM+Sigmoid combination preserves the spareness of the SVM while producing probabilities that are of comparable quality to the regularized likelihood kernel method.

3.2.2 Ranking Based Multi-label Classification

A formal problem statement for multi-label classification is described as follow. The domain of instances is denoted as $X = IR^d$ and the finite set of labels as $Y = \{1, 2, \ldots, l\}$. Given a multi-label training set T of size m, $T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}(x_i \in X, y_i \subseteq Y)$ where each instance $x_i \in X$ is associated with a subset of relevant labels $y_i \subseteq Y$, the goal of the multi-label classification is to construct a multi-label classifier $C : X \rightarrow 2^l(l = |Y| \geq 2)$ which, for any given instance x_i, determines all its relevant label subset y_i.

For the ranking based multi-label classification problem, the learning model is expected to build a real-valued scoring function as $f(\cdot, \cdot) : X \times Y \rightarrow IR$. And it is supposed that a successful learning model will tend to output larger score values for labels in y_i than those not in y_i. The corresponding multi-label classifier $C(\cdot)$ can also be derived from the real-valued scoring function:

$$C(x_i) = \{y|f(x_i, y) > t, y \in Y\}. \tag{3.2.4}$$

where t is a value estimated by a certain threshold selection strategy.

3.2.3 Threshold Selection Strategies

There are three commonly used threshold selection strategies for classification [96], including the rank-based method (RCut), the proportion-based assignments (PCut) and the score-based local optimization (SCut). Let l denotes the number of classes (labels) in the problem, d denotes the number of instances in the testing (or validation) set, and assume one score is produced by the classifier for each instance-class pair. The threshold selection algorithms are defined as follows.

1. **RCut**, it is a per-instance strategy, for each instance, sort classes by score and assign positive to each of the t top-ranking classes. The parameter t is a value (an integer between 1 and l) can be either specified by the user or automatically tuned using a validation set. That is, the value of t optimizing the global performance of the classifier on the validation set is fixed when applying the classifier on new instance in the testing set. RCut with $t = 1$ is also commonly used for single-label classification.

2. **PCut**, it is a per-class strategy, for each class c_j, sort the testing instance by score and assign positive to each of the l_j top-ranking instances, where $l_j = P(c_j) \times x \times d$ is the number of instance assigned to class c_j. and $P(c_j)$ is the prior probability (estimated using a training set) for an arbitrary instance to be an member of class c_j. The parameter x is automatically tuned in the same fashion as tuning t for RCut, by varying the value of x until the global performance of the classifier is optimized on the validation set.

3. **SCut**, it is also a per-class strategy, score a validation set of documents for each class and tune the threshold over the local pool of score values until the optimal performance of the classifier is obtained for that class. And the per-class thresholds are fixed when applying the classifier to new instance in the testing set.

The three strategies have different properties of information used, suitability, flexibility and overfitting risk. And the choice of threshold selection strategy depends on the characteristics of classifier and application [96].

3.3 Proposed Ranking Based Multi-label Classification

According to the previous analysis, the ranking based multi-label classification consists of two important parts: the scoring method and the threshold selection. And both the scoring and the threshold selection can influence the classification results significantly. The scoring method should

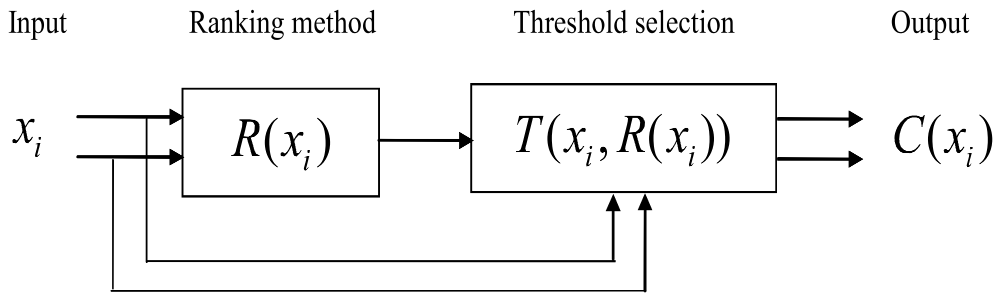


Figure 3.1: Main scheme of the proposed ranking based multi-label classification.

provide a proper label rank, i.e., the labels of high score values are more related with the instance. And the threshold selection strategy can choose appropriate thresholds for the testing instances.

In t -label classification, as showed in Fig. 3.1, instead of estimating a static threshold for all testing instances in conventional methods, threshold selection is considered as a function $T(x_i, R(x_i))$. The threshold function relies on both the input instance x_i and the label rank $R(x_i)$. It is expected to choose appropriate thresholds according to the characteristics of instances and label ranking strategy. The corresponding multi-label classifier $C(\cdot)$ can be described as:

$$C(x_i) = T(x_i, R(x_i)) \cdot R(x_i). \tag{3.3.1}$$

Efforts on both parts are made to improve the classification performance. For the scoring part, an improved probabilistic SVM with delicate decision boundary is used to obtain a proper label ranking results. For the threshold selection part, based on the ideal threshold set of training data estimated by the cross validation method, a KNN learning model is used to realize the threshold function $T(x_i, R(x_i))$ related with both the input instance and the label rank.

3.3.1 Scoring Method Based on SVM with Delicate Decision Boundary

The goal of the multi-label classification is to construct a multi-label classifier $c : X \to 2^l (l = |Y| \geq 2)$ which, for any given sample x_i, determines all its relevant label subset y_i. In this chapter, however, the learning model will produce a real-valued scoring function as the form $f : X \times Y \to IR$. A successful learning system will tend to output larger values (probability) for labels in y_i than those not in y_i. The corresponding multi-label classifier $c(\cdot)$ can also be derived from the real-valued scoring function $f(\cdot, \cdot) : c(x_i) = \{y | f(x_i, y) > t(x_i), y \in Y\}$, where $t(\cdot)$ is a function of threshold.

The main scheme of the proposed method is shown in Fig. 3.2. First of all, l (the number of label) binary SVMs are built by one-versus-rest strategy, i.e., training the p ($p \in [1, l]$) binary SVM,

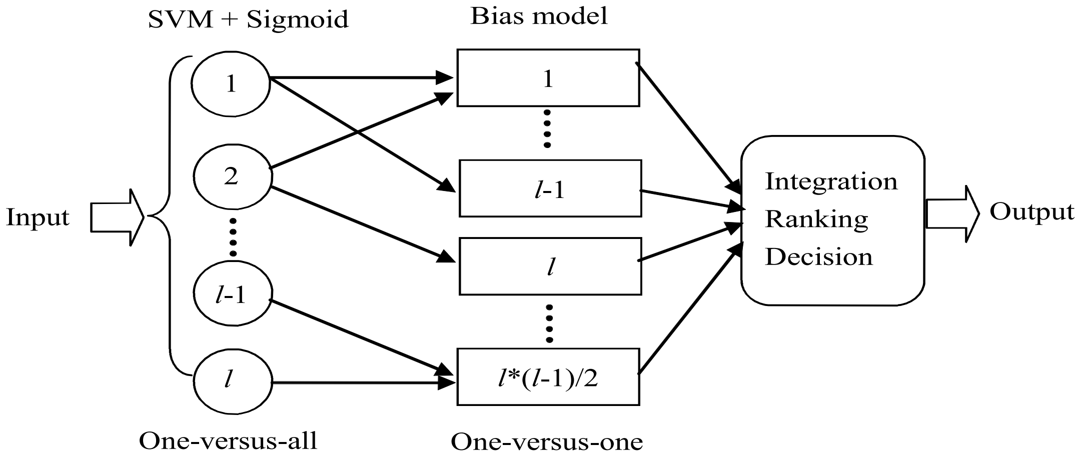


Figure 3.2: Main scheme of the SVM based scoring method

the samples labeled by p only are considered as "positive", all others are "negative". And the Platt's sigmoid method is used to obtain the probabilistic outputs of SVMs. Thus, l probabilistic classifiers are constructed. Then, according to the one-versus-one strategy and probabilistic outputs of l binary SVMs, $l * (l - 1)/2$ bias models are built by corresponding each pair of labels. The detailed mechanism of bias model will be described in next subsection, it can capture the characteristics of double-label samples and give a more delicate decision boundary for samples of overlapping sample space. It can be used to improve the performance of probabilistic classifier. Finally, in integration, ranking and final decision procedure, for each sample, there are $(l-1)$ probabilistic values for every label, the mean value of $(l - 1)$ probabilistic values is used as the final probability for this label. Hence, a probabilistic rank of l values are obtained for all labels. In the testing procedure, a proper threshold estimated based on training set is used to determine the label subset and its size for each testing sample.

Bias Model with Delicate Decision Boundary

Now let us consider the basic problem of classifying the essentially overlapping classes of label p and q. As shown in Fig. 3.3 (a), two separating surfaces are constructed by SVM binary classifiers mentioned in previous subsection. They divide the sample space into four areas: "only p area (1)", "only q area (2)", "p and q area (3)" and "no one's area (4)". It must be noted that in some cases (e.g. for linearly separable classes, or symmetric classes) one or two of these areas may be empty [67].

And two SVM separating surfaces are constructed in different high dimensional space, in order to summarize the probabilistic relativity of two surfaces, they are analyzed in same dimensional space for convenience.

In practice, there is an important common characteristic of double-label samples in the essentially overlapping classes of two labels. It is that the majority of double-label samples are distributed near to two probabilistic separating surfaces simultaneously. A real example of the Yeast benchmark dataset (label 3-rd and 4-t) is shown in Fig. 3.3 (b). Because of the error of SVMs classifiers, there are many samples (especially the double-label samples) can not be classified correctly. Most of double-label samples are distributed near to the separating surfaces and intermixed with other single-label samples.

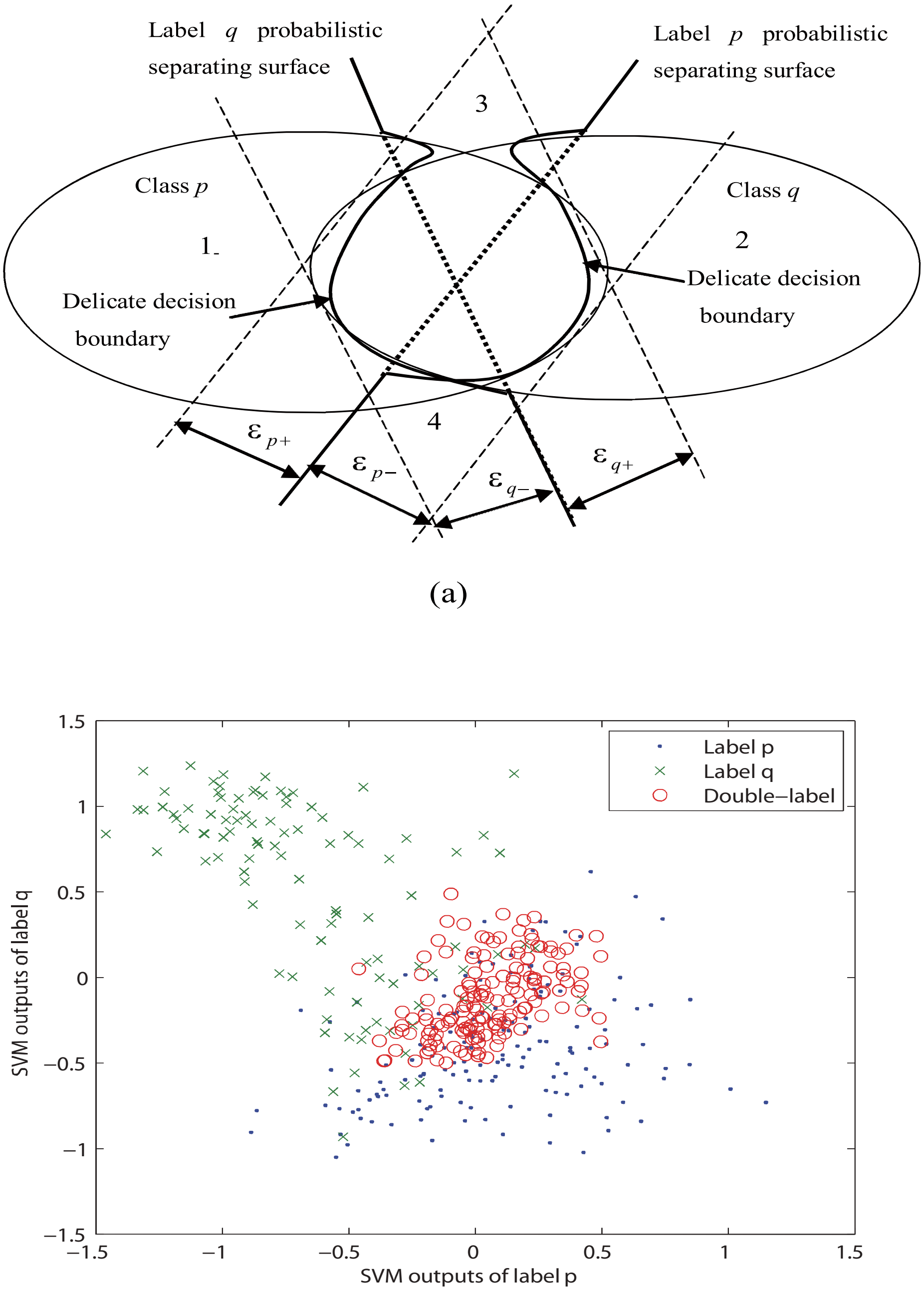
The main idea of the proposed bias model is that, for certain pair of labels, the distributing range of overlapping sample space (specified by the four thresholds ε_{p+}, ε_{p-}, ε_{q+} and ε_{q-}) can be estimated by double-label training samples and relative information of two probabilistic boundaries. Then, training samples in this overlapping sample space are used to estimate two delicate boundaries for two label respectively (bold curves shown in Fig. 3.3 (a)). And the delicate boundaries are used to obtain better classification results of SVMs.

The proposed bias model utilizes the probability value to realize the delicate boundaries. The probabilistic outputs of SVM can be obtained by Sigmoid method mentioned in previous section. Training the first separating surface, the samples labeled by p only are considered as "positive", all others are "negative". Trained classifier estimates the probability $r^{+}_{pq}(x)$ and dual probability $r^{-}_{pq}(x)$:

$$\begin{aligned} r^{+}_{pq}(x) &= P(p \in f(x) \land q \notin f(x) | x \in p \cup q) \\ r^{-}_{pq}(x) = 1 - r^{+}_{pq}(x) &= P(q \in f(x) \land p \notin f(x) | x \in p \cup q). \end{aligned} \tag{3.3.2}$$

The second separating surface is trained for class q can estimates the probability $r^{+}_{qp}(x)$ and dual probability $r^{-}_{qp}(x)$ similarly:

$$\begin{aligned} r^{+}_{qp}(x) &= P(q \in f(x) \land p \notin f(x) | x \in p \cup q) \\ r^{-}_{qp}(x) = 1 - r^{+}_{qp}(x) &= P(p \in f(x) \land q \notin f(x) | x \in p \cup q). \end{aligned} \tag{3.3.3}$$



(a)

(b)

Figure 3.3: Bias model with delicate decision boundary

And the probabilities of a given sample , which belongs to each of four areas, can be derived:

$$\begin{aligned} P_1 &= P(x \in \text{“}only\ p\text{”}) = r^+_{pq}(x) r^-_{qp}(x) \\ P_2 &= P(x \in \text{“}only\ q\text{”}) = r^-_{pq}(x) r^+_{qp}(x) \\ P_3 &= P(x \in \text{“}p\ and\ q\text{”}) = r^+_{pq}(x) r^+_{qp}(x) \\ P_4 &= P(x \in \text{“}no\ one's\text{”}) = r^-_{pq}(x) r^-_{qp}(x) \end{aligned} \tag{3.3.4}$$

The mechanism of bias model is described as follows. Firstly, double-label training samples near to two separating surfaces simultaneously are selected, and their SVM distance values to hyperplane of two labels (denoted as $\{d_{pi}\}$and$\{d_{qi}\}$) are used to estimate the four range thresholds ε_{p+}, ε_{p-}, ε_{q+} and ε_{q-} of two separating surfaces. Those thresholds can be used to decide the range of overlapping sample space. Secondly, according to the four thresholds, training samples in the range of overlapping sample space are selected to train two delicate boundaries for two label p and q respectively. Thirdly, in the testing procedure, the four thresholds and trained delicate boundaries of two separating surfaces are utilized to correct the classification results. For every sample in the testing dataset, if it belong to the overlapping sample space, its final probabilistic outputs are corrected by two trained delicate boundaries of two labels.

Parameters estimation for the proposed method

According to mechanism of the proposed bias model described in previous section, the key procedures of bias model are how to estimate the four thresholds and train the delicate boundaries of separating surfaces.

For the first problem, the double-label training samples near to two separating surfaces simultaneously are utilized to estimate them. Take the ε_{p-} as the example to introduce the estimation algorithm: select the double label training samples subset (denoted S_{ol}). For every sample $i \in S_{ol}$, according to the Eq. 3.3.4 to estimate the probabilities P_1,P_2,P_3 and P_4, then calculate the standard deviation value $r_{std} = \text{stdev}(P_1, P_2, P_3, P_4)$. So the threshold ε_{p-} can be calculated by the equation as follow:

$$\varepsilon_{p-} = \text{mean}(|d_{pi}|), i \in s_{ol} \wedge d_{pi} < 0 \wedge r_{std} < \eta \tag{3.3.5}$$

where d_{pi} is the sample's distance value to label p SVM hyperplane, $r_{std} < \eta$ means the distribution of sample simultaneously near to the two separating surfaces of label p and q in Fig. 3.3 (a), the η is a constant trained based on the overlapping training data.

For the second problem, training samples in the overlapping sample space decided by four range thresholds (those samples stimulatingly meet two condition of $\varepsilon_{p-} \leq d_{pi} \leq \varepsilon_{p+}$ and $\varepsilon_{q-} \leq d_{qi} \leq$

ε_{q+}) are selected to train two curve functions f_p and f_q (delicate boundaries) for two label p and q respectively.

Take f_p as the example to introduce the curve function fitting algorithm. A statistical method and the relative information of two separating surfaces are used to fit the curve function f_p. The decision values for function f_p is denoted by the distances between the training samples and separation boundary of the label q. As shown in Fig. 3.3 (a), let us consider the decision value region $[\varepsilon_{q-}, \varepsilon_{q+}]$, the length is L. The whole decision value domain is divided into ki intervals, then the distance between two neighbor intervals d can be calculated as $d = L/ki$. Using statistical method, the numbers of samples in overlapping sample space belonging to double-label or single-label respectively in each interval are counted. In the $j-$th interval, the number of samples belonging to double-label is denoted as Num_d^j, belonging to single label is denoted as Num_s^j, where $j = 1, 2, ..., ki$ is the index of interval. The bias probabilistic value is defined as following

$$p_{de\_j} = \frac{Num_d^j}{Num_d^j + Num_s^j}. \tag{3.3.6}$$

Then the data set $\{(d_{qj}, p_{de\_j})\}$ is used to fit the delicate decision boundary curve function f_p, d_{qj} as the input variable and p_{ad_j} as the value of function.

In the testing procedure, for every sample i of the test data, if the values of d_{pi} and d_{qi} meet two conditions of $\varepsilon_{p-} \leq d_{pi} \leq \varepsilon_{p+}$ and $\varepsilon_{q-} \leq d_{qi} \leq \varepsilon_{q+}$ simultaneously, the probabilities are biased as $p_{pi} = p_{pi} + f_p(d_{qi}), p_{qi} = p_{qi} + f_q(d_{pi})$ for the final probabilistic outputs.

3.3.2 Instance-dependent Thresholding Strategy

Different from the conventional methods to estimate a static threshold for all testing instances, the proposed instance-dependent thresholding strategy is considered as a function that relies on input instance and label rank. It is expected to choose appropriate thresholds according to the characteristics of instance and label ranking strategy. Based on the ideal threshold set of training data estimated by the cross validation strategy, a KNN learning model is used to realize the threshold function related with the input instance and the label rank.

As showed in Fig. 3.4, firstly, the d-folds validation method is utilized to estimate the target threshold values for training samples. The training data is split into several training/validation subsets. Training subsets are used to train the classifiers, and estimate the target threshold value for each sample in validation subset by optimizing a certain evaluation measure. Secondly, according to the teacher set of target thresholds for training samples, the KNN strategy is utilized to obtain the

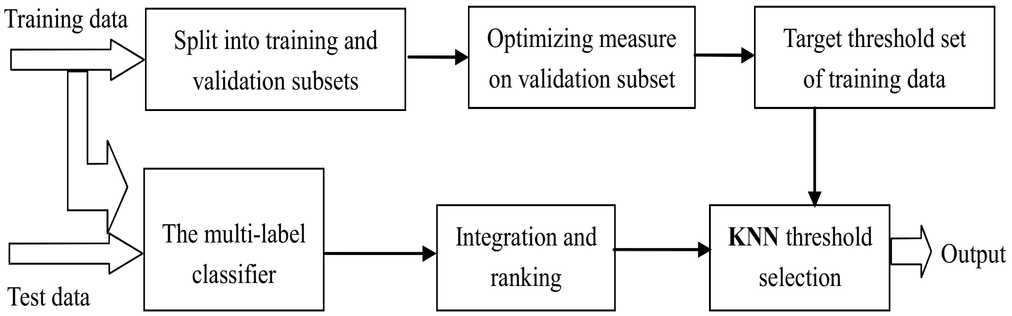

Training data
Split into training and validation subsets
Optimizing measure on validation subset
Target threshold set of training data
The multi-label classifier
Integration and ranking
KNN threshold selection
Output
Test data

Figure 3.4: Proposed instance-dependent threshold selection based on KNN.

instance-dependent thresholds for testing instances. The outline of the proposed instance-dependent thresholding strategy is described as follows.

1. Split training data into d folds. According to the validation strategy, for fold $i = 1, \ldots, d$.

   (a) Train the SVM+Sigmoid multi-label classifier (mentioned in previous subsection) on the $d-1$ folds (except the validation fold i).

   (b) Based to the trained multi-label classifier, consider the fold i as the testing data, obtain the probabilistic test outputs (score values for each classes) for samples in the fold i.

   (c) For each sample I_p in the validation fold i, sort the score values of classes (labels) as a descending rank $(c^{(1)}, c^{(2)}, \cdots\cdots, c^{(l)})$, where $c^{(1)}$ is the label of maximum score and l is the number of labels. Then, respectively take subset $\{c^{(1)}\}$, $\{c^{(1)} \quad c^{(2)}\}$, $\{c^{(1)} \quad c^{(2)} \quad c^{(3)}\} \cdots\cdots$ (from the subset contains the top label, add a new label one by one sequentially) as the candidate result label set for the sample I_p and evaluate it by a certain evaluation measure. Select the best candidate label set $\{c^{(1)} \quad c^{(2)} \quad \cdots c^{(j)}\}, 1 < j < l$. Then, take the mean of score values of $c^{(j)}$ and $c^{(j+1)}$ as the target threshold value of the sample I_p.

2. The target threshold values for all training samples are obtained by validation of Step 1. This target threshold set of training data is utilized as teacher set to estimate the thresholds of testing instances by KNN strategy described in Step 3.

3. In test procedure, for each instance I_q in the testing data, the similarities with all training samples are calculated by a certain distance metric (Euclidean distance, Hamming distance or other measures). The mean value of k nearest samples' thresholds is assigned as the threshold of the testing instance I_q, where k is the parameter of KNN strategy.

The proposed threshold selection is a per-instance strategy similar with the RCut method. But different from RCut method, instead of using a static parameter t as the threshold for all testing instances, the proposed method estimates threshold for different testing instance by a lazy learning method of KNN according to the ideal threshold values of training data. For the d-folds validation method in the proposed thresholding strategy, the parameter d should be chosen by considering the trade-off between the performance and computational cost.

3.4 Experiments

3.4.1 Benchmark Datasets

Nowadays, many protein sequences are stored in central protein databases from labs all over the world. And the number of sequences is constantly increasing. Only a fraction of these protein sequences has been experimentally analyzed for detecting their structure and their functions in the corresponding organism. The main reason is that experimental determination of protein structure is time-consuming and labor-intensive. Therefore, the automatic computer program tools that can classify new proteins to their corresponding structural families are important and imperative. With the contribution of modern data analysis techniques, such as machine learning and knowledge discovery, the issue has been approached computationally, thus providing fast and more flexible solutions.

The proposed improved multi-label classification method is used to solve two benchmark biology functional prediction problems of Yeast functional genomics [34] and Genbase motif-based protein classification [32].

Yeast functional genomics data. The yeast Saccharomyces cerevisiae is one of the best-studied organisms. The yeast functional genomics dataset from Ref. [34] is studied in our experiments. Each gene is described by the phylogenetic profile and concatenation of microarray expression data. And each gene is associated with a set of functional labels whose maximum size is very large (more than 190). The whole set of functional classes is structured by hierarchies up to four levels deep. In order to make it simplified, the dataset is preprocessed by Elisseeff and Weston. Only the known structure of the functional classes are utilized in dataset. In this chapter, the same data set as used in [34] is adopted. In this data set, only functional classes in the top hierarchy are considered.

Genbase motif-based protein classification data. This problem is studied in [32]. The protein chain can be mapped into a proper motif sequence representation according to attributes. A very important issue in the data mining process is the efficient choice of attributes. The motif sequence representation supports the efficient function of data-driven algorithms, which represent samples as classified part of a fixed set of attributes. In this problem, protein chains are represented utilizing a proper motif sequence vocabulary. Suppose there are N motifs in the vocabulary. Given a protein sequence typically contains a few motifs in the vocabulary. A protein sequence is encoded as an N-bit binary vector. The i-th bit is encoded as 1 if the corresponding motif is present in the sequence; otherwise the corresponding bit is encoded as 0. And each N-bit binary vector is associated with a set of functional label (if known).

Two benchmark datasets are downloaded from website of [88]. The detailed information about them, such as the number of samples, attributes, classes and their average number of labels or classes are listed in Tab. 3.1.

Table 3.1: Characteristics of the experimental benchmark datasets

Dataset	Total labels	Total features	Avg. labels	Training/Testing set
Yeast	14	103 (Numeric)	4.25	1500/917
Genbase	27	1186 (Discrete)	1.35	463/199

3.4.2 Evaluation metrics for multi-label classification

Multi-label classification requires different metrics than those used in traditional single-label classification. The evaluation multi-label dataset D contains $|D|$ instances $(x_i, y_i), i = 1 \dots |D|, y_i \subseteq Y$. $Y = \{1, 2, \dots, l\}$ is the set of labels. Let C be a multi-label classifier and $z_i = C(x_i)$ be the set of labels predicted by C for example x_i. The following evaluation metrics for label ranking used in [77, 34, 88, 98] are adopted in this chapter:

Hamming loss metric describes how many times a sample label pair is misclassified (a label not belonging to the sample is predicted or a label belonging to the sample is not predicted). Hamming loss is defined as follow.

$$HammingLoss = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{1}{l} |y_i \Delta z_i| \tag{3.4.1}$$

where Δ means for the XOR operation of two sets. In $E1$, Δ means the XOR operation of two sets. The smaller this measure is, the better the method performs.

Multi-label Precision (MPrec), Multi-label Recall (MRec) and Multi-label F-score (MF-score) evaluation metrics are defined as follows.

$$
\begin{aligned}
MPrec &= \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|y_i \cap z_i|}{|z_i|} \\
MRec &= \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|y_i \cap z_i|}{|y_i|} \\
MF-score &= \frac{2*MPrec*MRec}{MPrec+MRec}
\end{aligned}
\tag{3.4.2}
$$

There is a common characteristic for above three measures, the lager measure value, the better the classification performance. And the performance is perfect when its value equal 1.

3.4.3 Experimental Setting and Results

In our experiments, the LSSVM [85] is taken as a basis. SVM parameters are chosen by cross-validation procedure. In the training procedure, the methods mentioned in Section 3.3 are utilized to train the SVM classifiers, calculate the parameters of sigmoid functions and correction models and estimate the ideal threshold values for all training samples. In the testing procedure, for each test instance, a probabilistic rank of l values for all labels is obtained by final decision procedure. Then, according to the ideal threshold values of training samples, a proper threshold estimated based on KNN method. And this proper threshold is used to determine the label subset finally.

For the threshold selection, the *MF-score* evaluation metric is selected as the optimizing measure for estimating the ideal threshold set of training data. In addition, there are two important parameters for the proposed threshold selection. The first is the parameter d of d-fold cross validation, considering the trade-off between the performance and computational consume, 10-folds strategy is used for Yeast dataset and 5-folds for Genbase dataset. The second is the parameter k of KNN method, by experimentalism, $k = 4$ is chosen for Yeast dataset and $k = 3$ for Genbase dataset. The Fig. 3.5 shows the relation of the parameter k and the performance of classification for Yeast dataset.

The following popular multi-label classification methods: C4.5, Naive Bayes and Binary-SVM [77, 88] are utilized to compare with the proposed method. The experiment results for two datasets are presented in the Tab. 3.2 and Tab. 3.3. For Genbase dataset, the nonlinear (RBF-kernel or other non-linear kernels) SVM models are severely overfitting. Therefor, the linear SVM and ck-SVM (proposed in Chapter 2) are used for Genbase dataset in our experiments. The results of the methods of C4.5 and Naive Bayes are cited from [88]. The winning results are marked with bold font.

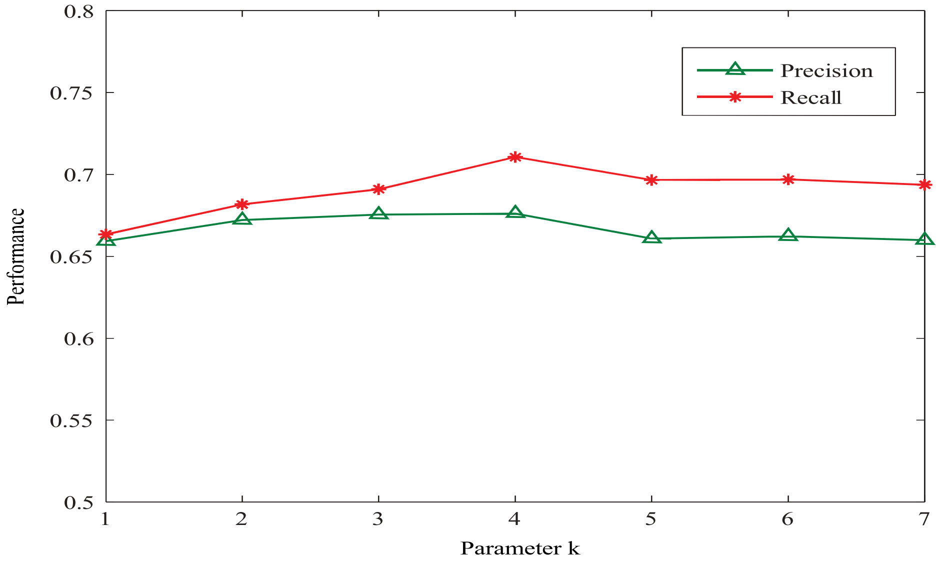


Figure 3.5: Relation between the parameter k of KNN in threshold selection and the performance of classification for Yeast dataset

Table 3.2: Experimental results for Yeast dataset.

Algorithms	Hamming loss	MPrec	MRec	MF-score
C4.5	0.259	0.561	0.593	0.5766
Naive Bayes	0.301	0.610	0.531	0.5678
Binary-SVM (based on RBF kernel)	0.2021	0.5862	0.6332	0.6088
Proposed (based on RBF kernel)	**0.1915**	**0.6761**	**0.7106**	**0.6929**

Table 3.3: Experimental results for Genbase dataset.

Algorithms	Hamming loss	MPrec	MRec	MF-score
C4.5	0.001	0.992	0.995	0.9935
Naive Bayes	0.035	0.273	0.276	0.2745
Binary-SVM (Based on linear SVM)	0.0011	0.9933	**0.9958**	0.9945
Proposed (based on linear SVM)	0.00006	0.9947	0.9950	0.9949
Proposed (based on ck-SVM)	**0.00006**	**0.9968**	0.9942	**0.9955**

Figure 3.6 shows the MF-score results of the thirteen bias models for 12-th label in Yeast dataset comparing with the Binary-SVM method.

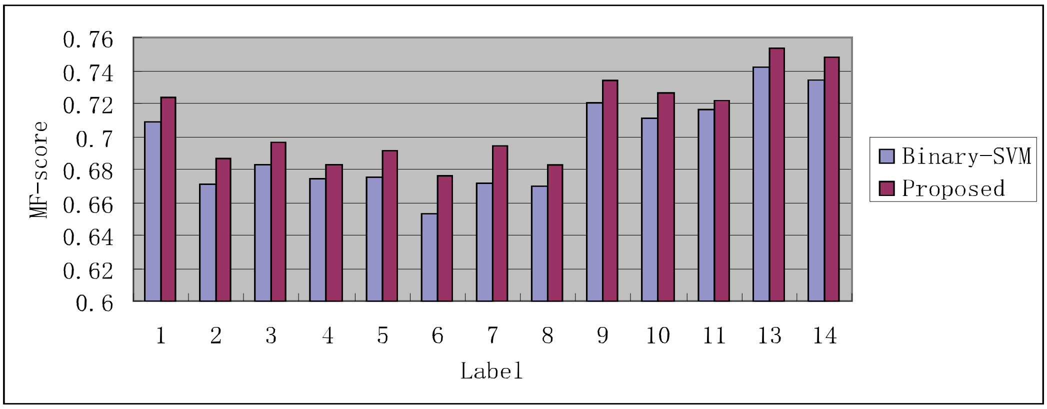


Figure 3.6: Comparing results of the bias models for 12-th label in Yeast dataset

The experimental results of all evaluation metrics demonstrate that the proposed method improves the performance of classification efficiently.

3.5 Conclusions

In a ranking based multi-label classification, both the scoring method and the threshold selection can influence the classification results significantly. In t -label classification is proposed to solve functional genomics applications. Firstly, the improved probabilistic SVM with correction model is used as the scoring method. It can improve the probabilistic label ranking by introducing the information of multi-label training samples in overlapping sample space into learning procedure. Secondly, an instance-dependent thresholding is proposed to decide the classification results. It can estimate appropriate threshold value according to the characteristics of input instance and label ranking strategy. Experimental results on the two functional genomics benchmark datasets of Yeast and Genbase show that the proposed method improves the performance of classification efficiently, compared with binary SVM method and some existing well-known methods.

Chapter 4

Hierarchical Multi-label Classification Based on Over-sampling and Hierarchy Constraint for Gene Function Prediction

4.1 Introduction

Multi-label classification problem is an extension of traditional multi-class classification problem in which the classes are not mutually exclusive and each sample is associated with several classes simultaneously. In multi-label classification, training samples are belong to a label subset and the classification goal is to predict label subset for a new instance. Hierarchical multi-label classification (HMC) is a special variant of multi-label classification where the classes are organized by hierarchy structure, which an sample that is labeled by one class automatically is labeled by all its super-classes (this is called hierarchy constraint).

Gene function prediction [6] is a difficult task of HMC. It is known that a single gene may have multiple functions, and these functions are organized in a hierarchy. Gene function prediction is characterized by hundreds or thousands of functional classes structured according to a predefined hierarchy; and the example distributions of functional classes are usually high degree skewed (imbalanced) with negative more than positive. There are two hierarchy structures: a directed acyclic graph for the Gene Ontology (GO) [3] and a tree forest for FunCat [74]. The classification on FunCat taxonomy of rooted tree structure is our focus in this chapter.

Many approaches have been proposed to deal with the HMC tasks [72, 73, 83, 6, 70, 78], and the approaches can be categorized as two different research lines. The first research line is to develop a single multi-label learning model that predicts all the classes of an example at once. Clare et al [25] presented a decision tree method for multi-label classification in the context of functional genomics.

A decision tree is used to predict not a single class but a vector of Boolean class variables. Vens et al. [92] proposed a method based on the predictive clustering trees for HMC tasks. The second research line is to develop an ensemble model that consisted of binary classifiers. An HMC task is transferred into a series of separate binary classification tasks, and each classifier is trained to learn one class, then results of all binary classifiers are combined by considering the hierarchical relationships between classes [5, 6, 66, 42, 14, 90]. Barutcuoglu et al. [6] presented a two-step method where SVMs are used to learn for each class separately and combined by a Bayesian network model so that the prediction results are compliance with the hierarchy constraint. Valentini et al. [90] proposed a True Path Rule (TPR) ensemble method to combine the binary classifications. This approach is implemented by a two-way asymmetric information flow that traverses the graph-structured ensemble. The positive predictions for a node influence its ancestor nodes by a recursive way. And the negative predictions influence its offspring nodes.

Approaches of the first research line, which most of are based on the decision tree model with a vector of Boolean class variables, can predict all the classes by learning a single model. But for complicated HMC tasks (datasets) with a large number of attributes and classes, it is difficult and unpractical to tackle all classes efficiently by a single learning model. The approaches of the second research line, which transfer an HMC task into many binary classifications, always have advantages for solving such complicated HMC tasks with high dimension attributes. Efficient classification method (such as SVM), which is adept at high dimension learning task, can be used as binary classifier to improve the classification performance.

There are two main challenges that need to be solved for the second research line approaches. Firstly, most of binary classifiers at lower hierarchy levels often need to learn from strongly skewed training datasets. Because of the hierarchy constraint in typical HMC problems, the frequency of classes at higher levels often tends to be high, while the frequency of classes at lower hierarchy levels tends to be small. In such imbalance dataset learning, standard classifiers tend to be overwhelmed by the majority class and ignore the minority one. Secondly, the hierarchy constraint of classes is not taken into account in separate binary classifiers. It can not guarantee that a sample belongs to a class should belongs to all its super-classes. Most of traditional HMC methods [6, 66, 42, 14, 90] mainly focus on the second challenge by developing ensemble strategies considering the hierarchy constraint. In our research, the two challenges are simultaneously considered: a hierarchical imbalanced dataset preprocessing approach is proposed to enhance the performance of the separate binary classifiers; and an improved ensemble strategy based on classifier performance is introduced to deal with the second challenge.

How to efficiently introduce the information of hierarchy constraint between classes into learning model is a vital problem for the HMC task. Along the second research line, this chapter proposes an improved HMC method based on over-sampling and hierarchy constraint for solving the gene function prediction problem. The HMC task is transferred into binary SVM classification tasks for classes in the hierarchy. Then, two measures are implemented to improve the HMC performance by introducing the hierarchy constraint into learning procedures. 1) A hierarchical over-sampling approach based on Synthetic Minority Over-sampling Technique (SMOTE) is proposed to preprocess the imbalanced training subsets for binary SVM classifiers. It can improve the performance of classifiers by changing the distribution of the imbalanced datasets. 2) An improved TPR consistency approach is used to combine the results of binary probabilistic SVM classifiers. The consistency ensemble can improve the classification results and guarantee the hierarchy constraint of classes (an instance belonging to a class should belong to all its super-classes).

The rest of the chapter is organized as follows. Section 4.2 gives a brief overview of the hierarchical multi-label classification, True Path Rule consistency ensemble and imbalanced datasets learning methods. Section 4.3 formulates the proposed HMC method based on over-sampling and consistency ensemble. Section 4.4 presents the experiments on FunCat benchmark datasets. Finally, the conclusions are introduced.

4.2 Hierarchical Multi-label Classification

4.2.1 Notation and Definition

In a hierarchical multi-label classification problem, an instance x can be assigned to one or more classes of the set $C = \{c_1, c_2, \cdots, c_m\}$. The assignments can be coded through a vector of multilabels $y = \langle y_1, y_2, \cdots, y_m \rangle \in \{0, 1\}^m$, where if x can be labeled as class c_i, then $y_i = 1$, otherwise $y_i = 0$. And the parameter $i, 1 \leq i \leq m$ means the indices corresponding to the m classes in the set C.

In HMC problem, the classes are structured by a hierarchy and can be described by a rooted tree. Each node of the rooted tree corresponds to a class, and paths correspond to relationships of classes. The class hierarchy is denoted by $(C, \leq_h)$, where $\leq_h$ is a partial order in class tree representing the superclass relationship (for all $c_1, c_2 \in C, c_1 \leq_h c_2$ if and only if c_1 is a superclass of c_2). For convenience, the node corresponding to class c_i is simply denoted by i. The child nodes set of c_i is denoted by $child(i)$, and the parent nodes set of c_i is denoted by $par(i)$. The children classes of node c_i is represented by $y_{child(i)}$, and the parent classes of c_i is represented by $y_{par(i)}$. A classifier

$D : X \to \{0,1\}^m$ is builded to predict the label set corresponding to each new instance $x \in X$, and $d_i(x) \in \{0,1\}$ is the label set for class c_i predicted by the classifier. For simplicity, $d_i(x)$ is represented simply by d_i if there is no ambiguity.

4.2.2 Imbalanced Dataset Learning

The imbalanced dataset learning is a difficult challenge in a classification problem. There are much more samples of some classes than others. And the standard classifiers for solving such imbalanced dataset are inclined to ignore the small classes and be overwhelmed by the large classes [17, 87]. The solutions to imbalanced dataset learning problem can be divided into data and algorithmic levels categories. The methods at data level change the distribution of the imbalanced dataset, and then the balanced dataset are provided to the learner to improve the detection rate of minority class. The methods at the algorithm level modify the existing data mining algorithms or put forward new algorithms to resolve the imbalance problem [44].

For the algorithmic level, solutions for imbalanced dataset learning include adjusting the probabilistic estimation of the leaf nodes based on decision tree methods, adjusting the costs of the various classes for dealing with the imbalance problem, and adopting recognition-based methods (learning from one class) rather than discrimination-based methods (learning from two classes) [17].

For the data level, different forms of re-sampling methods were proposed [17]. The simplest re-sampling methods are random over-sampling and random under-sampling. The former augments the minority class by exactly duplicating the examples of the minority class, while the latter randomly takes away some examples of the majority class. However, random over-sampling may make the decision regions of the learner smaller and more specific, thus cause the learner to over-fit. Random under-sampling can reduce some useful information of the data sets. Many improved re-sampling methods are thus presented, such as heuristic re-sampling methods [53], combination of over-sampling and under-sampling methods [7], embedding re-sampling methods into data mining algorithms [43], and so on.

SMOTE (Synthetic Minority Over-sampling Technique) proposed in Ref. [16] is an efficient re-sampling method at the data level. It generates new synthetic examples along the line between the minority examples and their selected nearest neighbors. The advantage of SMOTE is that it makes the decision regions larger and less specific. SMOTE method produces synthetic examples for over-sampling the minority class. Firstly, for each minority example, its k nearest neighbors with same class are considered. The default value of parameter k is set as $k = 5$ in SMOTE. Secondly, some examples are randomly selected from the considered neighbors by a over-sampling rate. Finally, according to the line between the minority example and its selected nearest neighbors, some new synthetic examples are produced by over-sampling.

4.2.3 TPR Consistency Ensemble

TPR consistency ensemble method has proposed in Ref. [90] for solving the HMC problem. The True Path Rule can guarantee the uniformity of gene function annotations in FunCat taxonomies: "If the child term describes the gene product, then all its parent terms must also apply to that gene product". That is to say, if a gene is labeled by a specific functional class, then it should be labeled by all its "parent" classes, and also should be labeled by all its ancestor classes in a recursive way. The ensemble method is implemented by a two-way asymmetric information flow that traverses the graph-structured classes. The positive predictions for a node influence its ancestor nodes by a recursive way. And the negative predictions influence its offspring nodes.

For a given example x and a class node c_i, a classifier in TPR consistency ensemble should obey the rules as follows:

$$\begin{cases} d_i = 1 \Rightarrow d_{par(i)} = 1 \\ d_i = 0 \Rightarrow d_{child(i)} = 0 \end{cases} \tag{4.2.1}$$

The TPR method realizes an ensemble that respects the "true path rule" by putting together the classification results predicted at each node by local "base" classifiers. Positive predictions of local classifiers are propagated from bottom to top across the graph in a recursive way. They effect the predictions of their ancestors nodes by traversing the graph towards higher level nodes. Negative predictions for a given node are propagated to their descendants nodes for preserving the consistency of the hierarchy according to the true path rule.

4.3 Improved HMC Method

4.3.1 Main Frame of the Proposed Method

Main frame of the proposed HMC method based on over-sampling and hierarchy constraint ensemble is shown in Fig. 4.1. Firstly, the training dataset is partitioned into m (the number of classes) subsets for each binary classifier, and the imbalanced subsets are preprocessed by over-sampling strategy to improve the data distribution firstly. Then, the m binary probabilistic SVM classifiers are trained to predict the classification results. Finally, the probabilistic prediction results of m binary classifiers are assembled by the consistency strategy of taking into account the hierarchical relationships between classes.

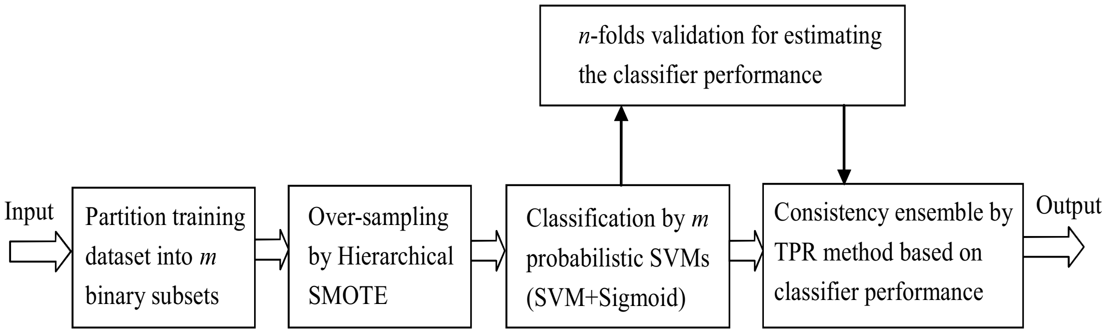

n-folds validation for estimating the classifier performance
Input
Partition training dataset into m binary subsets
Over-sampling by Hierarchical SMOTE
Classification by m probabilistic SVMs (SVM+Sigmoid)
Consistency ensemble by TPR method based on classifier performance
Output

Figure 4.1: Main frame of the proposed HMC method.

The SVM+Sigmoid method is used as the basic binary probabilistic classifier. Constructing a classifier to produce a posterior probability ps (class—input) is very useful when a classifier is making a small part of an overall decision, and the classification output must be combined for the overall decision. The standard SVM do not provide such probabilities, and the Platt's sigmoid method is commonly used to estimate the probabilistic outputs of SVM. It maps the SVM outputs into probabilities through training the parameters of an additional sigmoid function by validation set in training data [69].

$$ps(y=1|x)=\frac{1}{1+\exp(Af(x)+B)} \tag{4.3.1}$$

In order to obtain the best values of parameters A and B, a validation subset including n' training data can be used to solve the following maximum likelihood problem,

$$\begin{aligned} &\min_{Z=(A,B)}\{-\sum_{i=1}^{l}(t_i\log(ps_i)+(1-t_i)\log(1-ps_i))\} \\ &ps_i=\frac{1}{1+\exp(Af_i+B)},\ \ f_i=f(x_i) \\ &t_i=\begin{cases}\frac{N_{+}+1}{N_{+}+2} & if\, y_i=1\\ \frac{1}{N_{-}+2} & if\, y_i=-1\end{cases} \quad i=1,2,\cdots n'\end{aligned} \tag{4.3.2}$$

where N_{+} means the number of positive-labeled samples, and N_{-} means the number of negative-labeled samples. The SVM+Sigmoid combination preserves the spareness of the SVM while producing probabilities that are of comparable quality to the regularized likelihood kernel method.

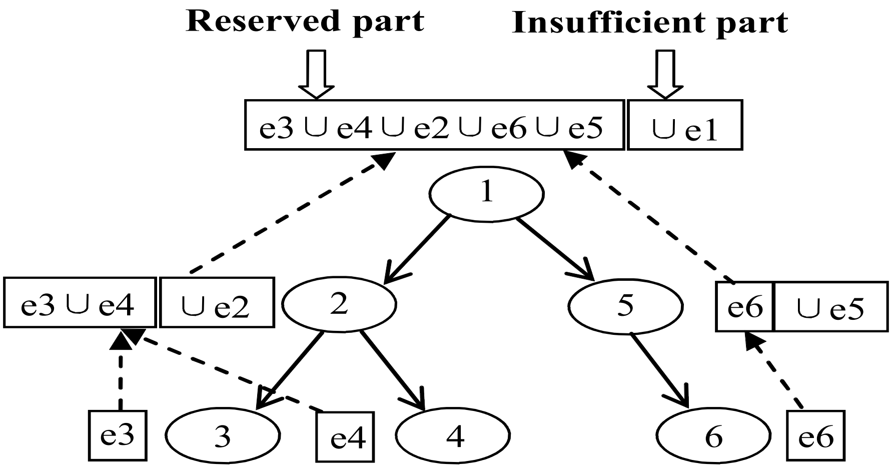


Figure 4.2: Illustration of the hierarchical SMOTE.

4.3.2 Proposed Hierarchical SMOTE

Preprocessing for Training Subsets

In the preprocessing procedure of partitioning the training dataset into m subsets for binary classifiers. According to the hierarchy relations between classes, the negative examples in subset for each class have been selected in such a way described as follows. The negative examples are not annotated for the class, but belong to the parent class (i.e. positive for the parent class). For the training subset of class c_i, examples that belong to c_i become the positive examples, and the other examples which belong to $par(i)$ but not to c_i) become the negative examples. This negative example selection method is often used in many HMC methods [92, 90], in this way only negative examples that are not too dissimilar to the positive ones are selected as the training subsets for binary classifiers.

Even using the above mentioned partition preprocess, there are still many high degree imbalanced training subsets for binary classifiers. A hierarchical SMOTE is proposed to tackle this problem. SMOTE is an over-sampling approach in which the minority class is over-sampled by creating "synthetic" examples. The dataset of minority class is over-sampled by introducing synthetic examples generated according to each minority class sample and its k nearest neighbors.

According to the hierarchy constraint characteristic of HMC problem, for the hierarchical class tree, the positive examples belong to training subset of class (node) c_i automatically belong to all training subset of its super-classes (nodes) $par(i)$. Therefore, in the proposed hierarchical SMOTE method, over-sampling procedures for m subsets are implemented from bottom to top across the class tree, and the "synthetic" examples created by children classes are automatically reserved and regarded as the "synthetic" examples of their parent classes. For example, as showed in Fig. 4.2, considering the node 2, all "synthetic" examples $\{e3 \cup e4\}$ created by its children nodes $child(2) = \{3, 4\}$ are combined and reserved as its "synthetic" examples firstly, the over-sampling procedure of node 2 only need create the insufficient "synthetic" examples. Detailed procedures of the proposed hierarchical SMOTE are described in next subsection.

Comparing with implementing the basic SMOTE over-sampling for each subsets respectively, the proposed hierarchical SMOTE method has two advantages. Firstly, because of the "synthetic" examples created by children classes are automatically reserved for their parent classes, the proposed method can remarkably reduce the over-sampling operations of "synthetic" examples for whole HMC problem. Secondly, the basic SMOTE method has a drawback in some special situations [1], it can not obtain good performance because of its assumptions about the training set. For instance, the space between two positive instances is assumed to be positive and the neighborhood of a positive instance is also assumed to be positive, but those assumptions may not always be true for some special distribution datasets. In the proposed hierarchical SMOTE method, considering the "synthetic" examples for class c_i, those reserved "synthetic" examples from children classes $child(i)$ are all distributed in the positive space of class c_i definitely, therefore, the proposed method can solve the drawback of basic SMOTE in certain degree.

Hierarchical SMOTE for Imbalanced Subsets

According to the rooted tree of m labels, from bottom to top across the tree, implement over-sampling processing for subsets of imbalanced classes. For node c_i ($1 \leq i \leq m$) in the class tree, denote the positive (minority) class is P and the negative (majority) class is N as follows,

$$P = \{p_1, p_2, \cdots, p_{pn}\}, N = \{n_1, n_2, \cdots, n_{nn}\} \tag{4.3.3}$$

where pn and nn are the number of positive and negative examples in training subset of label c_i. The class c_i is considered as imbalanced if $\frac{pn}{nn} \leq \eta$, η ($\eta = 0.6$ is chosen in experiments) is a skewed parameter to evaluate the degree of imbalance. The hierarchical SMOTE method is described as follows.

Step 1, calculate the number of "synthetic" examples need be over-sampled $O_p = nn * \mu - pn$, where μ ($\mu = 0.8$ is chosen in experiments) is a cost sensitive parameter of SMOTE, it means the ratio of positive and negative examples after over-sampling.

Step 2, if c_i is a leaf node in class tree, generate O_p "synthetic" positive examples from the data P as follows. Calculate the parameter vector $S = \{s_1, s_2, \cdots, s_{pn}\}$ according to an average strategy, where s_j is an integer and $O_p = \sum_{j=1}^{pn} s_j$. For each $p_j \in P$, s_j nearest neighbors are randomly selected from its k ($k = \max(S) + 5$ is chosen in experiments) nearest neighbors in P. Firstly, the differences $dif_q(q = 1, 2, \cdots, s_j)$ between example p_j and its s_j nearest neighbors from P are calculated, then multiply dif_q by a random number $r_q(q = 1, 2, \cdots, s_j)$ between 0 and 1, finally, s_j new "synthetic" positive examples are generated between p_j and its nearest neighbors:

$$sysnthetic_q = p_j + r_q \times dif_q, q = 1, 2, \cdots, s_j. \tag{4.3.4}$$

The above procedure is repeated for each p_j in P, and O_p "synthetic" positive examples are generated for node c_i. This step is similar with SMOTE, more details can be seen in Ref. [16].

Step 3, if c_i is not a leaf node in class tree, combine all "synthetic" examples created by children nodes $child(i)$ as set P_{child}. If $|P_{child}| \geq O_p$, randomly select O_p examples from the set P_{child} as "synthetic" positive examples set of node c_i. If $|P_{child}| < O_p$, generate $O_p - |P_{child}|$ "synthetic" positive examples for node c_i using similar procedure Step 2, and combine the generated examples and the examples of P_{child} as "synthetic" positive examples of node c_i.

4.3.3 Improved TPR Consistency Ensemble

Disadvantage of the basic TPR Ensemble

The TPR ensemble method [90] combines the local decisions of the base classifiers related to each node with the positive predictions that propagate from the lower level nodes of hierarchy, and with the negative predictions that propagate from the higher level nodes. Given a new instance x, the base classifiers predict local probabilities $\overline{ph}_i(x)$ that it associated with the class c_i, and the TPR ensemble generates a "consensus" global probabilities $ph_i(x)$. $\varphi_i(x)$ is used to denote the children set of node c_i which have a positive decision for the instance x:

$$\varphi_i(x) = \{j | j \in child(i), d_j(x) = 1\} \tag{4.3.5}$$

Based on the local probabilities $\overline{ph}_i(x)$ and the decisions of the child nodes in $\varphi_i(x)$, the global consensus probabilities $ph_i(x)$ of the TPR ensemble can be obtained as described in Eq. 4.3.6. The decision $d_i(x)$ at node c_i is set as 1 if $ph_i(x) > t$ (a default value for t is 0.5), and as 0 otherwise. The positive decisions propagate from bottom to top through this ensemble way. On the contrary, if $d_i(x) = 0$ is obtained, then this negative prediction is propagated to its subtree. For all leaf nodes, the Eq. 4.3.6 becomes $ph_i(x) = \overline{ph}_i(x)$. To balance the local predictions with the positive predictions coming from the ensemble, a parent weight $w, 0 \leq w \leq 1$ is introduced. If $w = 1$

the prediction at node c_i is provided only by the local predictor, otherwise the decision is provided proportionally to w and $1 - w$ between respectively the local predictor and its child nodes.

$$ph_i(x) = w \cdot \overline{ph}_i(x) + \frac{1-w}{|\varphi_i(x)|} \sum_{j \in \varphi_i(x)} ph_j(x). \tag{4.3.6}$$

There is a disadvantage in basic TPR ensemble method. In practice, the performances of different binary SVM classifiers for hierarchy classes are obviously varied for real-world datasets. For some classes at lower level of hierarchy rooted tree, the classifier performance is very poor because of data noise or other reasons. But in the basic TPR consistency ensemble method, as described in Eq. 4.3.6, each classifier in the positive prediction children set $\varphi_i(x)$ contributes same propagation effect for estimating the local probabilities $\overline{ph}_i(x)$. Because the propagation effects from children classes are represented by probabilistic values, this equal propagation mechanism may produce an error propagation: if an error prediction occurs in a poor performance child node classifier, i.e., this child classifier predicts a high probabilistic value for a negative instance that not belongs to it. This error prediction will give an obvious effect to other node classifiers because of its high probabilistic value.

Improved TPR Ensemble Based on Classifier Performance

To restrict the poor performance binary classifiers bring an error propagation effect to other classifiers. In the improved consistency ensemble method, another weight ν_j of classifier performance is introduced to the TPR ensemble as follows:

$$\begin{aligned} ph_i(x) &= w \cdot \overline{ph}_i(x) + \frac{1-w}{|\varphi_i(x)|} \sum_{j \in \varphi_i(x)} \nu_j \cdot ph_j(x) \\ \nu_j &= \frac{E_j}{\sum\limits_{k \in \varphi_i(x)} E_k}, j \in \varphi_i(x) \end{aligned} \tag{4.3.7}$$

where E_j is the performance evaluation of child node classifier $j \in \varphi_i(x)$ by a certain metric.

The introduced performance weight ν_j are decided by a certain performance metric of classifiers, this classifier performance metric can be estimated by a validation strategy according to the train data. It should be noted that the meaning of two weight w and ν_j are different. The parent weight w is used to balance the local predictions and the positive predictions coming from the children nodes, Ref. [90] indicates that this parent weight is a global parameter that affect the general precision/recall characteristics of the ensemble. The performance weight ν_j in the improved method is used to restrict the poor performance binary classifiers bring an error propagation effect to other classifiers.

Similar to the negative prediction propagation method in Ref. [90], negative predictions for a given node are propagated to its descendants, to preserve the consistency of the hierarchy. In the event that a negative prediction for the node c_i, all the nodes belonging to the subtree rooted at c_i are predicted as negative, and if their probabilities are larger than $ph_i(x)$, their probabilities are changed to $ph_i(x)$.

4.4 Experiments

4.4.1 Evaluation Metrics

Considering the multi-label and hierarchy characteristics of HMC problem, two categories of metrics are adopted to evaluate the proposed method.

The first metric category is the classical Precision (Prec), Recall (Rec) and F-score for each class of the hierarchy. F-score metric is used to estimate the classifier performance parameters in the improved TPR ensemble method (Eq. 4.3.7). The three metrics are traditionally introduced for the binary classification with positive and negative classes. The recall metric means the proportion of positive instances that are correctly predicted as positive. The precision metric means the proportion of positive predictions that are correct. That is:

$$\begin{aligned} Prec &= \frac{TP}{TP+FP}, Rec = \frac{TP}{TP+FN} \\ F-score &= \frac{2*Prec*Rec}{Prec+Rec} \end{aligned} \tag{4.4.1}$$

with FN is the number of false negatives (positive instances that are incorrectly predicted as negative), FP is the number of false positives (positive predictions that are incorrect), and TP is the number of true positives (correctly predicted positive instances). Note that these measures ignore the number of correctly predicted negative examples.

Moreover, in order to take into account the multi-label characteristics of HMC problem, the second metric category, which includes Multi-label Precision (MPrec), Multi-label Recall (MRec) and Multi-label F-score (MF-score) [34, 88], is also used to evaluate the proposed method. Let D be a multi-label evaluation data set, consisting of $|D|$ multi-label examples $(x_i, y_i), i = 1 \ldots |D|, y_i \subseteq Y$. $Y = \{1, 2, \ldots, k\}$ is the set of labels. Let z_i be the set of labels predicted by HMC classifier for

Table 4.1: Properties of experiment datasets

Dateset	Attribute	Training	Testing
D_1 Sequence (seq)	478	2580	1339
D_2 Spellman et al. (cellcycle)	77	2476	1281
D_3 Gasch et al. (gasch1)	173	2480	1284
D_4 All microarray (expr)	551	2488	1291

example x_i. The three metrics are described as follows.

$$\begin{aligned} MPrec &= \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|y_i \cap z_i|}{|z_i|} \\ MRec &= \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|y_i \cap z_i|}{|y_i|} \\ MF-score &= \frac{2 * MPrec * MRec}{MPrec + MRec} \end{aligned} \tag{4.4.2}$$

There is a common characteristic for above mentioned metrics, the lager measure value, the better the classification performance. The performance is perfect when its value equals 1.

4.4.2 Data Sets and Experiment Setting

Four yeast data sets from FunCat are used to evaluate the proposed method. The different data sets describe different aspects of the genes in the yeast genome, and the different sources of data highlight different aspects of gene function. The properties of experiment datasets, including instance number D and attribute number, are listed in Tab. 4.1, the detailed description of each dataset can be referred from Ref. [25, 92]. The four datasets are downloaded from the following webpage: http://dtai.cs.kuleuven.be/clus/hmcdatasets/.

Two different ensemble strategies is used to compare with the proposed method. The first is *Flat* ensemble, that does not take into account the hierarchical structure of the data. The second is basic TPR hierarchical ensemble, the parent weight w in Eq. 4.3.6 is set as 0.5. For each ensemble strategies, linear LSSVM method [85] is taken as base learner (the nonlinear (RBF-kernel or other non-linear kernels) SVM models are severely overfitting for FunCat datsets), the Platt's sigmoid method is utilized to estimate the probabilistic outputs of SVM, and the threshold t of all binary classifier output are set as 0.5 in all the experiments. About the parameters of the proposed method, the hierarchical SMOTE parameter setting are described in Subsection 3.2.2; in the improved TPR ensemble, 5-folds validation strategy on training data is used to estimate the F-score metric as the performance weight ν_j.

4.4.3 Experiment Results and Analysis

To observe the detailed performance of the proposed hierarchical SMOTE and the improved TPR ensemble, firstly, some specific experiments are implemented based on the dataset of Sequence (D_1) and the subtree of "protein fate" FunCat class (FunCat $ID = 14$). The subtree is composed by 20 nodes (as showed in Fig. 4.3). In the dataset of Sequence (D_1), there are 625 training examples and 337 testing instances relating with the class subtree of $ID = 14$. For each class, the number of positive and negative training examples are listed in Tab. 4.2, it can be found that most of classes are significantly imbalanced.

In order to evaluate the performance of the proposed hierarchical SMOTE over-sampling approach, three experiment strategies including *Flat* ensemble, basic SMOTE over-sampling and the proposed hierarchical SMOTE over-sampling are implemented and evaluated. Experiment results about per-class metrics (Prec, Rec) for 20 classes (nodes) are listed in Tab. 4.2. We can find that two over-sampling strategies can effectively improve the performances of classifiers for imbalanced datasets, and the proposed hierarchical SMOTE over-sampling approach outperforms the basic S-MOTE for all classes. The basic SMOTE assumes the space between two positive instances is to be positive and the neighborhood of a positive instance is also to be positive, but those assumptions may not always be true for some special distribution datasets. The proposed hierarchical SMOTE can overcome this drawback in certain degree. According to the strategies of the proposed hierarchical SMOTE, most of the reserved "synthetic" samples from children classes $child(i)$ are all distributed in the positive space of class c_i definitely.

To observe the improved TPR ensemble approach, two experiment strategies including basic TPR ensemble and the improved method with classifier performance weight are implemented and evaluated. As listed in Tab. 4.3, the experiment results about per-class metrics (Prec, Rec) for 20 classes (nodes) show that the improved TPR outperforms the basic TPR ensemble method. The classifier performance weight in the proposed method can effectively restrict the error propagation effects from poor performance binary classifiers.

To evaluate the performances of combination methods based on different over-sampling and ensemble strategies, four combination strategies including M1 (*Flat* ensemble), M2 (basic TPR ensemble), M3 (the proposed hierarchical SMOTE over-sampling and *Flat* ensemble) and the proposed HMC (the proposed hierarchical SMOTE over-sampling and the improved TPR ensemble)

Table 4.2: Experiment results of different over-sampling strategies based on the FunCat subtree of "Protein fate" (ID=14).

FunCat ID	Positive	Negative	Flat Prec	Flat Rec	SMOTE Prec	SMOTE Rec	H-SMOTE Prec	H-SMOTE Rec
14.01	61	564	0.1667	0.3333	0.1721	0.3784	0.2407	0.4415
14.04	158	467	0.3131	0.3690	0.3371	0.3814	0.3558	0.4405
14.07	306	319	0.6460	0.5474	0.6460	0.5474	0.6460	0.5474
14.07.01	14	292	0.1020	0.4167	0.2141	0.4072	0.2111	0.4833
14.07.02	41	265	0.0822	0.2500	0.0978	0.2571	0.1486	0.4583
14.07.02.01	11	30	0.0714	0.7500	0.1521	0.4860	0.0882	0.7500
14.07.02.02	27	14	0.0606	0.8571	0.0606	0.8571	0.0606	0.8571
14.07.03	91	215	0.2619	0.6226	0.2987	0.6233	0.2736	0.6604
14.07.04	32	274	0.1019	0.5500	0.0925	0.3672	0.1400	0.7000
14.07.05	44	262	0.0532	0.2174	0.0872	0.2556	0.0860	0.3478
14.07.07	11	295	0.0121	0.1453	0.0195	0.1670	0.0263	0.3333
14.07.09	6	300	0.0153	0.1475	0.0205	0.1892	0.0385	0.2500
14.07.11	49	257	0.0690	0.2857	0.0944	0.3034	0.0909	0.5000
14.07.11.01	13	36	0.0244	0.2500	0.0215	0.1931	0.0357	0.3750
14.10	118	507	0.2474	0.4068	0.2850	0.4768	0.2700	0.4576
14.13	150	475	0.3048	0.4507	0.2715	0.5272	0.3380	0.5070
14.13.01	112	38	0.1688	0.7407	0.1275	0.4248	0.1688	0.7407
14.13.01.01	75	37	0.1147	0.7143	0.1147	0.7143	0.1147	0.7143
14.13.04	16	134	0.0333	0.1429	0.0314	0.1762	0.0556	0.2857
14.13.04.02	6	16	0.0127	0.2542	0.0127	0.2542	0.0127	0.2542

are implemented and evaluated. Experiment results about multi-label metrics (MPrec, MRec and MF-score) for the whole subtree dataset are showed in Fig.4.4. The results show that the proposed HMC method outperforms other three strategies. The combination of two proposed strategies (the hierarchical SMOTE over-sampling and the improved TPR with classifier performance weight) can effectively enhance the HMC performance.

Finally, the three ensemble strategies experiment results on four datasets are listed in Tab. 4.4. The results of the proposed HMC method based on ck-SVM (proposed in Chapter 2) are also listed. It can be found that the proposed method significantly outperforms the basic TPR ensemble method and the *Flat* ensemble method.

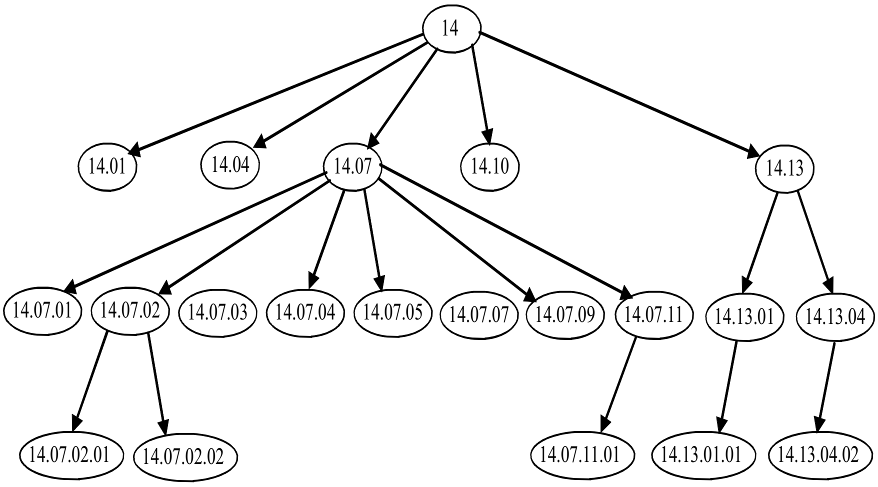


Figure 4.3: Rooted tree of the FunCat classes of ID=14 (Protein fate).

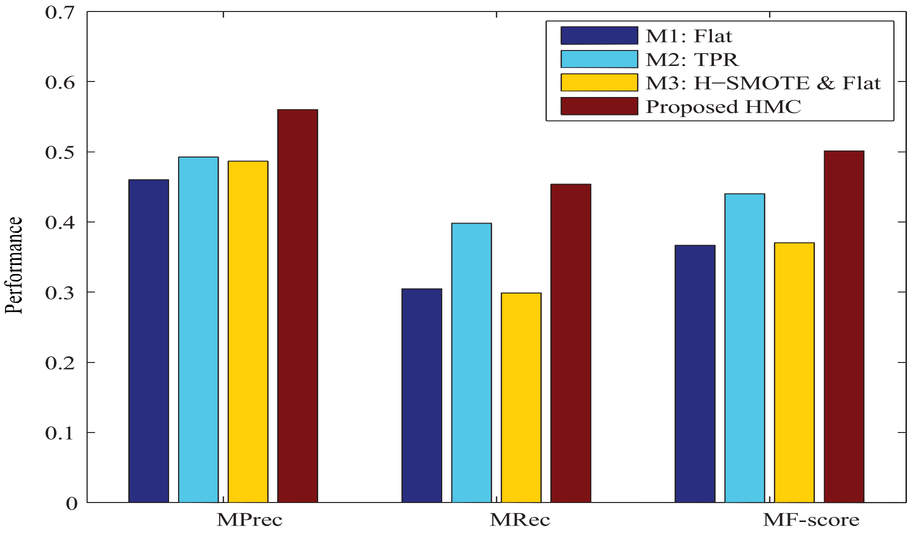


Figure 4.4: Three multi-label metrics on the FunCat subtree of ID=14 (Protein fate).

Table 4.3: Experiment results of different ensemble strategies based on the FunCat subtree of "Protein fate" (ID=14).

FunCat ID	TPR Prec	TPR Rec	Improved TPR Prec	Improved TPR Rec
14.01	0.2016	0.3792	0.2578	0.4356
14.04	0.3228	0.3929	0.3521	0.4568
14.07	0.5490	0.8263	0.6705	0.8947
14.07.01	0.2000	0.4167	0.2176	0.5116
14.07.02	0.0923	0.2500	0.2275	0.4835
14.07.02.01	0.1667	0.5000	0.2483	0.5758
14.07.02.02	0.0794	0.3571	0.1829	0.7683
14.07.03	0.2857	0.6038	0.2837	0.6487
14.07.04	0.0824	0.3500	0.1285	0.6982
14.07.05	0.0685	0.2174	0.0920	0.4276
14.07.07	0.0121	0.1453	0.0303	0.3333
14.07.09	0.0153	0.1475	0.0657	0.3150
14.07.11	0.0749	0.2786	0.1267	0.546
14.07.11.01	0.0169	0.1436	0.0465	0.3542
14.10	0.2474	0.4068	0.3043	0.4792
14.13	0.2516	0.5634	0.3357	0.5547
14.13.01	0.1825	0.4630	0.1864	0.6932
14.13.01.01	0.1221	0.7429	0.1237	0.7265
14.13.04	0.0378	0.1865	0.0567	0.2487
14.13.04.02	0.0127	0.2542	0.0476	0.5000

4.5 Conclusions

In this chapter, an improved HMC method based on over-sampling and consistency ensemble is proposed for solving the gene function prediction problem. According to the characteristics of HMC task, two measures are implemented to improve the HMC performance by introducing the hierarchy constraint into learning procedures.

Firstly, in typical HMC problems, the frequency of classes at higher levels in hierarchy tends to be high, while classes at lower levels often have very small frequencies. Therefore, most of training datasets for binary classifiers at lower hierarchy levels are always imbalanced. For such imbalance dataset learning, a hierarchical SMOTE approach is proposed to preprocess the imbalanced training subsets for binary SVM classifiers. It can improve the performance of classifiers by changing the distribution of the imbalanced datasets.

Table 4.4: Classification results on the four FunCat datasets.

Dataset	Method	MPrec	MRec	MF-score
D1	Flat	0.4771	0.3049	0.3720
	TPR	0.6056	0.3568	0.4490
	Proposed	0.6832	0.4362	0.5324
	Proposed + ck-SVM	0.6905	0.4487	0.5439
D2	Flat	0.4598	0.2876	0.3539
	TPR	0.5529	0.3624	0.4378
	Proposed	0.6513	0.4292	0.5174
	Proposed + ck-SVM	0.6597	0.4472	0.5331
D3	Flat	0.4694	0.2917	0.3598
	TPR	0.5883	0.3342	0.4263
	Proposed	0.6728	0.4318	0.5260
	Proposed + ck-SVM	0.6704	0.4342	0.5270
D4	Flat	0.4628	0.2932	0.3590
	TPR	0.5785	0.3427	0.4304
	Proposed	0.6593	0.4186	0.5121
	Proposed + ck-SVM	0.6685	0.4276	0.5216

Secondly, the hierarchy constraint of classes is not taken into account in separate binary classifiers. In other words, it can not greentree that an instance is predicted as a class should be predicted as all its super-classes. For this problem, an improved TPR consistency approach is used to combine the results of binary probabilistic SVM classifiers. It can improve the classification results and guarantee the hierarchy constraint of classes.

Four yeast data sets from FunCat are used to evaluate the proposed method. Experiment results show that the proposed method is an efficient hierarchical multi-label classification method, it significantly outperforms the basic TPR ensemble and the *Flat* ensemble.

Chapter 5

Protein Structure Prediction on HP Model Using a Hybrid EDA

5.1 Introduction

Protein structure prediction (PSP) is one of the most important problems in computational biology. A protein is a chain of amino acids (also called as residues) that folds into a specific native tertiary structure under certain physiological conditions. Understanding protein structures is very important for determining a protein's function and its interaction with RNA, DNA and enzyme. Some vital knowledge for protein engineering and drug design can be obtained from the protein conformation information. There are over a million known protein sequences. However, only a limited number of protein structures are experimentally determined due to expensive and time-consuming. Therefore, prediction protein structure from its sequences utilizing computer software is an important means for understanding proteins' functions.

Due to the complexity of PSP task, simplified models (such as Dill's HP-lattice [56] model) are always used as tools to investigate the general properties of protein folding. In HP model, 20-letter alphabet of residues is simplified to a two-letter alphabet, namely H (hydrophobic) and P (polar). Experiments on small protein suggest that the native state of a protein corresponds to a free energy minimum. This hypothesis is widely accepted, and forms the basis for computational prediction of a protein's conformation from its residue sequence. The problem of finding such a minimum energy configuration has been proved to be NP-complete for the bi-dimensional (2-D) [27] and tri-dimensional (3-D) lattices [8]. Therefore, a deterministic approache is always not practical for this problem.

Many genetic algorithm (GA) based methods have been proposed to solve the PSP on HP model in recent years [89, 41, 82, 26, 22]. However, it has been acknowledged that the crossover operators, particularly one-point crossover and uniform crossover, in a conventional GA do not perform well for this problem [52, 37]. On the other hand, R. Santana et al. (2008) pointed out that the evolutionary algorithms able to learn and use the relevant interactions that may arise between the variables of the problem can perform well for this kind of problems. Estimation of distribution algorithm (EDA) is known as one of such kind of evolutionary algorithms.

In the EDAs [55, 59], instead of using conventional crossover and mutation operations, probabilistic models are used to sample the genetic information in the next population. The use of probabilistic models, especially, models taking into account bivariate or multivariate dependencies between variables, allows EDAs to capture genetic tendencies in the current population effectively. In brief, these algorithms construct, in each generation, a probabilistic model that estimates the probability distribution of the selected solutions. Dependency regulars are then used to generate next generation solutions during a simulation step. It is expected that the generated solutions share a number of characteristics with the selected ones. In this way, the search leads to promising areas of the search space.

In Ref. [75], the EDAs that use Markov probabilistic model or other probabilistic models outperform other population-based methods when solving the HP model folding problem, especially for the long sequence protein instances. But those methods have three obvious disadvantages as follow. 1) For most long sequence protein instances, the chance of finding the global optimum is very low, and the algorithm often need be set by very large generation number and population size for finding the global optimum. 2) For some deceptive sequences, those methods can only find the suboptimum solutions. 3) In those methods, a backtracking method is used to repair invalid individuals sampled by the probabilistic model of EDAs. For a traditional backtracking algorithm, the computational cost of repairing procedure is very heavy for those long sequence instances.

This chapter proposes a hybrid method to solve above problems. Firstly, based on the information of folding structure core (H-Core), a composite fitness function containing H-Core information is proposed to replace the traditional fitness function in HP model. The proposed fitness function is expected to select better individuals for probabilistic model of EDAs algorithm. It can help to increase the chance of finding the global optimum and reduce the complexity of EDA (population size and the number of generation needed). Secondly, local search with guided operators is utilized to refine the found solutions for improving efficiency of EDA. Local search with guided operators generates offspring through combination of global statistical information and the location information of solutions found so far. Thirdly, for the long sequence protein sequences, an improved backtracking-based repairing method is proposed to repair invalid individuals generated by the EDA probabilistic model. The traditional backtracking repairing procedure will produce heavy computational cost for searching invalid closed-areas of folding structure. In the improved method, to avoid entering invalid closed-areas, a detection procedure for feasibility is introduced when selecting directions for the residues in backtracking searching procedure. It can significantly reduce the number of backtracking searching operation and the computational cost for the long protein sequences.

The rest of the chapter is organized as follows. Section 5.2 gives a brief overview of protein HP model and the EDAs. Section 5.3 describes the proposed hybrid EDA for HP model protein folding. It includes the proposed composite fitness function and local search with guided operators. Section 5.4 formulates the improved backtracking repairing algorithm for invalid solutions. Section 5.5 presents the experiment results of the introduced method. Finally, the conclusions and further work directions are given.

5.2 Protein HP Model and Problem Representation

5.2.1 Protein Folding and HP model

Proteins are macromolecules made out of 20 different residues. A residue has a peptide backbone and a distinctive side chain group. The peptide bond is defined by an amino group and a carboxyl group connected to an alpha carbon to which a hydrogen and side chain group is attached. Residues are combined to form sequences which are considered as the primary structure of proteins. And the locally ordered structure generated by hydrogen bounding mainly within the peptide backbone is named as the secondary structure of protein. The beta sheet and the alpha helix are two common secondary structure elements in proteins. The global folding of a single polypeptide chain is considered as the tertiary structure of protein.

Under specific conditions, a protein sequence folds into a unique native 3-D structure. Each possible protein fold has an associated energy. According to the thermodynamic hypothesis, the native structure of a protein corresponds to the folding which the free energy reaches the minimum. Based on this hypothesis, many methods are proposed to search for the protein native structure by defining an approximation of the protein energy and utilizing the optimization methods. These approaches mainly differ in the type of energy approximation employed and in the characteristics of the protein modeling.

The well-known Dills HP model is used in this chapter. The HP model takes into account the hydrophobic interaction as the main driving force in protein folding. In HP model representation,

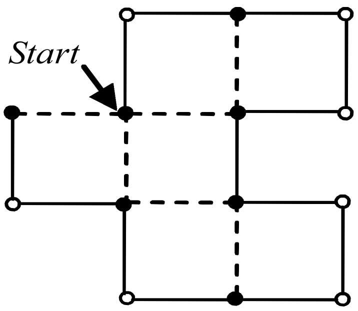


Figure 5.1: One possible configuration of the sequence $HPHPPHHPPHPHPH$ in 2-D HP model. There are six HH topological neighbors (represented by broken lines).

beads denotes the amino acids, and lines of beads denotes the connecting bonds. In this model, a protein is considered as a sequence $S \in \{H, P\}^{+}$, where H represents a hydrophobic residue and P represents a hydrophilic or polar residue. The HP model restricts the space of conformations to self-avoiding paths on a lattice in which vertices are labeled by the residues.

Two neighbor relations are considered for a pair of residues. The one is connected neighbor (they are adjacent in the chain). The other is topological neighbor (they are adjacent in the lattice but not connected in the chain). Let ε_{HH} denote the interaction energy between topological neighbor of two H residues, ε_{PP} for two P residues, ε_{HP} for a H residue and a P residue. An energy function is defined as the total energy of topological neighbors with $\varepsilon_{HH} = -1$ and $\varepsilon_{PP} = \varepsilon_{HP} = 0$. The HP problem is to find the folding conformation that minimizes the total energy $E(x)$. Figure 5.1 shows the graphical representation of a possible configuration for sequence $HPHPPHHPPHPHPH$ in 2-D HP model, hydrophobic residuals are represented by black beads and polar residuals by white beads. The energy that the HP model associates with this configuration is -6.

Although more complex models have been proposed, the HP model remains a focus of research in computational biology, chemical and statistical physics. By varying the energy function and the bead sequence of the chain (the primary structure), effects on the native state structure and the kinetics (rate) of folding can be explored, and this may provide insights into the folding of real proteins. In particular, the HP model has been used to investigate the energy landscapes of proteins, i.e. the variation of their internal free energy as a function of conformation. In evolutionary computation, the model is still employed because of its simplicity and its usefulness as a test-bed for new evolutionary optimization approaches [75].

5.2.2 Problem Representation

There are neither mutation nor crossover operators in EDAs [55]. A probabilistic model is used to capture the probability distribution from a database that includes the selected individuals from previous generation. And this probabilistic model is utilized to sample the new population for EDA. Therefore, through the joint probability distribution of the selected individuals at each generation, the interrelations between the different variables that represent the individuals are explicitly expressed by the probabilistic model.

EDAs suppose that it is feasible to obtain a probabilistic model for the promising areas of search space. And this model can be utilized to guide the search for the optimum solutions. A condensed representation of the features shared by the selected individuals is used to build the probabilistic graphical model in EDAs. By the probabilistic graphical model, different patterns of interactions between subsets of the problem variables can be captured efficiently, and this knowledge can be utilized to sample new solutions conveniently.

In the algorithm of protein folding optimum, one of the important problems is how to present a specific conformation. To embed a hydrophobic pattern $S \in \{H, P\}^+$ into a lattice, there are three methods of Cartesian Coordinate, Internal Coordinate and Distance Matrix [52] as follows.

1) Cartesian Coordinate representation. The residue position is represented by its Cartesian Coordinate. The positions of residues are independently.

2) Internal Coordinate representation. The residue position is decided by its predecessor residues in the sequence. Two types of internal coordinate are often used in lattice model. The one is relative direction representation where the residue directions depend on the direction of the previous move. The other is absolute direction representation where the residue directions depend on the axes defined by the lattice.

3) Distance Matrix representation. A distance matrix is introduced to represent the residue positions. Given a residue, its location can be computed by the distance matrix.

In Ref. [52], the authors implemented a detailed comparative research on absolute and relative directions utilizing evolutionary algorithms. According to the experimental results, relative directions always have better performances than absolute directions based on square and cubic lattice. On the contrary, absolute directions outperform relative directions based on triangular lattices. Evaluation the effectiveness of direction encoding on an evolutionary algorithm is uncertain in general.

But the internal coordinates with relative directions can be suggested based on the experimental evidence.

In this chapter, the representation of internal coordinates with relative direction is used. The position of each residue depends upon the previous move. Relative direction representation presents the direction of each residue relative to the main chain next turn direction. This representation can reduce the direction number of each position. For 2-D HP model, the set of direction is left, right and forward (L, R, F). And it is left, right, forward, up and down, (L, R, F, U, D) for the 3-D HP model. For example, by relative direction representation, the representation of the protein structure shown in Fig. 5.1 is $s = (RFRRLLRRFRLR)$.

It can be noted that the backward direction is not used, because the backward direction will cause overlap in this representation. Thus, this representation can reduce the position collision in a certain degree to guarantee the self-avoiding walk folding procedure. There are other advantages of the relative direction representation. One is that the sequence conformation can be presented as one dimension array. The most important is that the change of a start direction will not influence the structure of other part in sequence.

5.2.3 Probabilistic Model of EDA

It is very important for EDAs to select an appropriate probabilistic model according to a given application problem. The probabilistic model is represented by conditional probability distributions for each variable and estimated from the genetic information of selected individuals in the current generation. Therefore, the type of probabilistic model also influences the number and strength of the interactions learned by the model.

In Ref. [75], three probabilistic models for EDAs are proposed to solve the HP model problem: k-order Markov model, tree model and mixtures of trees model. In practice, the k-order Markov model is an appropriate probabilistic model for the HP model problem, where $k \geq 0$ is a parameter of the model. It can effectively embody the self-avoiding folding characteristics of the HP model problem, because it is assumed that positions of adjacent residues are related in the protein folding procedure.

The k-order Markov model can encode the dependencies between the move of a residue and the moves of the previous residues in the sequence, and this information can be used in the generation of solutions. It is described as follow. The joint probability mass function of X is denoted as $p(X)$. Given $X_j = x_j$, the conditional probability distribution of $X_i = x_i$ is denoted by $p(X_i = x_i | X_j = x_j)$ or its simplified form $p(x_i | x_j)$.

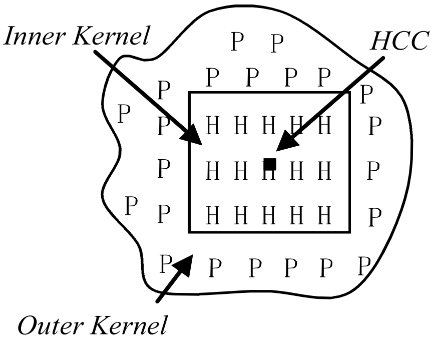


Figure 5.2: Illustration of the protein folding H-Core in 2-D HP model.

In the k-order Markov model, the value of variable X_i depends on the values of the previous k variables. The joint probability distribution can be factorized as follows:

$$p_{MK}(X) = p(x_1, \cdots, x_{k+1}) \prod_{i=k+2}^{n} p(x_i | x_{i-1}, \cdots, x_{i-k}) \tag{5.2.1}$$

Since the structure of the Markov model is given, it can be used to construct the probabilistic model through computing the marginal and conditional probabilities of the set of selected individuals and to sample the new generation. To sample a new solution, first variables in the factor $(x_1, \cdots, x_{k+1})$ are generated and the rest of variables are sampled according to the order specified by the Markov factorization.

5.3 A Hybrid EDA for Protein Folding Based on HP Model

5.3.1 Proposed Composite Fitness Function

In order to increase the chance of finding the global optimum and reduce the complexity of EDA (population size and the number of generation needed), a composite fitness function based on the information of folding structure core (H-Core) is proposed to replace the traditional fitness function in HP lattice model.

It is well known that the hydrophobic residues have a potential propensity to develop a hydrophobic core in protein folding. The Hs (hydrophobic residues) form the protein core and the Ps (hydrophilic or polar residues) tend to remain in the outer surface. As shown in Fig. 5.2, the inner kernel, called the H-Core [46], is compact and mainly formed of Hs while the outer kernel consists mostly of Ps. The H-Core Center is called HCC. The H-Core is a rectangle-like area in 2-D lattice

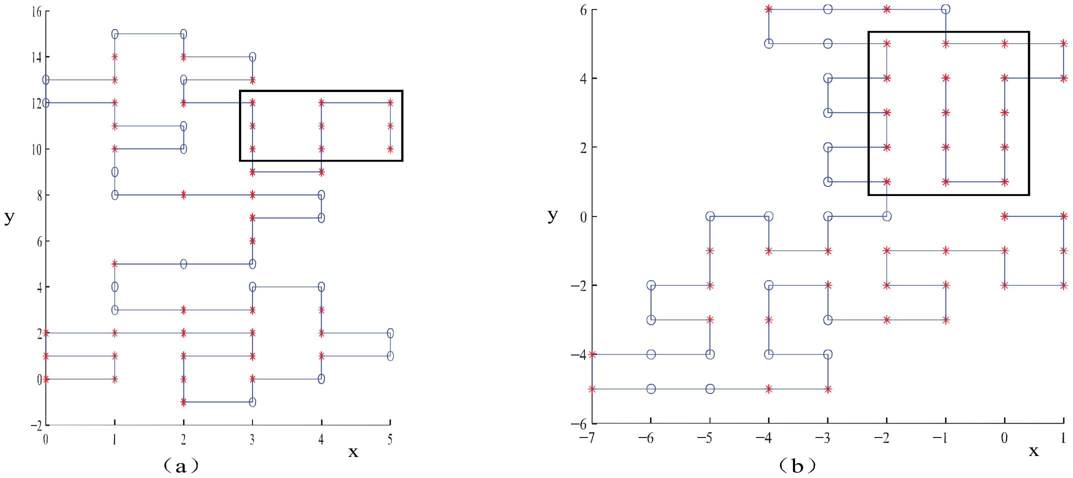


Figure 5.3: Two example solutions with same energy (Instance $s8$ in Tab. 5.1, length: 64, energy: -25).

and cube-like space in 3-D lattice. The coordinates of HCC can be calculated by follows equations.

$$\begin{aligned} x_{HCC} &= \frac{1}{n_H}\sum_{i=1}^{n_H} x_i, \quad y_{HCC} = \frac{1}{n_H}\sum_{i=1}^{n_H} y_i \\ z_{HCC} &= \frac{1}{n_H}\sum_{i=1}^{n_H} z_i \end{aligned} \tag{5.3.1}$$

where n_H is the sum of hydrophobic residues in solution, x_i, y_i and z_i (for 3-D HP model) are the coordinates of hydrophobic residues position in lattice. The number of Hs in inner kernel H-Core (denoted as $N_{HC}(x)$) can be calculated through search surrounding rectangle area (cube space for 3-D HP model) of HCC.

The number of Hs in inner kernel H-Core is an important characteristic for the folding solution. It also reflects the optimum degree of solution. In practice, as showed in Fig. 5.3, for two solutions with same basic HP model energy $E(x)$ (defined by the number of topological neighbor residues in lattice), the solution with bigger H-Core has more similar to the optimum solution, and it also has more biology significance. The two possible solutions of the Instance $s8$ (length is 64) have same basic HP model energy (-25), but they have different $N_{HC}(x)$ values (9 and 15 respectively). Obviously, the solution Fig. 5.3(b) has more similar to the optimum solution.

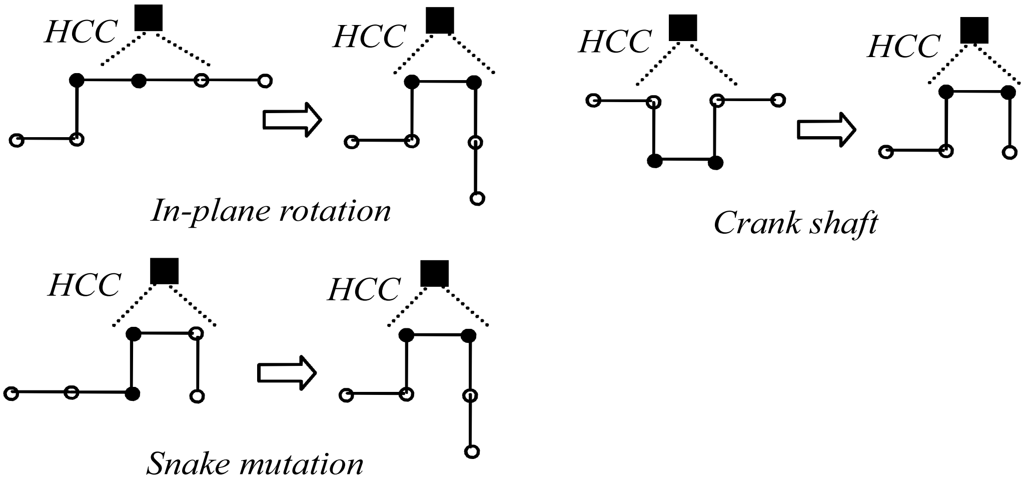


Figure 5.4: Guided operators for local search.

In the proposed method, a novel composite fitness function containing the information of H-Core for the k-order Markov EDA is introduced as follow.

$$Fit_{cp}(x) = \omega(-E(x)) + (1 - \omega)N_{HC}(x) \tag{5.3.2}$$

where, $E(x)$ is the total energy of the interaction between topological neighbor residues of HP model ($\varepsilon_{HH} = -1$, $\varepsilon_{PP} = \varepsilon_{HP} = 0$). $N_{HC}(x)$ is the number of Hs in inner kernel H-Core. ω is weight parameter of the fitness function, and $\omega > 0.5$ is chosen in practice, because the interaction energy $E(x)$ is the dominant characteristic of protein folding solution.

5.3.2 Local Search with Guided Operators

The search in traditional EDAs is mainly based on the global information, but local search is an exploitation method based on local information. Both the global information about the search space and the local information of solutions found so far can be utilized to enhance the efficiency of evolutionary algorithm. The global information can supervise the search for exploring in promising areas. And the local information of solutions can be useful for exploitation in promising areas. Therefore, it is worthwhile investigating whether combining local search with EDA could improve the performance of the EDA [99, 98].

Local search with a set of guided operators is implemented in the proposed hybrid EDA. Some of these operations have been utilized as mutations in the previous GA and ant colony optimizations studies of protein folding [82]. But in this chapter, "guided operators" means that those operations are implemented only under some special conditions.

Take 2-D HP model as example, the special conditions defined as follow. 1) Guided operation should guarantee the validity of individual, i.e. it can not produce position collision in lattice. If the positions of residues in lattice are changed, the object positions must be empty. 2) Guided operation should follow a basic principle that make Hs as near as possible to the HCC and Ps far away from the HCC according to the relative position in lattice, as shown in Fig. 5.4.

The way of choosing individuals to implement local search is described as follow. In each iteration procedure of EDAs, use the composite fitness function (described by Eq. 5.3.2) to sort the selection individuals. According to the distribution of individuals' fitness, randomly select some individuals (the number is a certain percentage of the population) in each fitness domain to implement the local search with guided operators.

EDAs extract globally statistical information from the previous search and then build a probabilistic model for modeling the distribution of best solutions visited in the search space. However, the information of the locations of best individual solutions found so far is not directly used for guiding the further search. Local search with guided operator generates offspring through combination of global statistical information and the location information of solutions found so far. The resultant solution can (hopefully) fall in or close to a promising area which is characterized by the probabilistic model.

5.4 Improved Backtracking-based Repairing Method

5.4.1 Backtracking Method

In HP model, A $collision$ is the embedding of two different peptides onto the same vertex of the lattice. As each member of the initial population of EA based method is randomly generated, it may represent an illegal conformation resulting one or more collisions when embedded. Similarly, $crossover$ and $mutation$ operations of GAs, $sampling$ from probabilistic model operation of EDAs and other genetic operations may produce additional collisions.

In Ref. [26], a backtracking algorithm was introduced to repair the positional collisions. It utilizes backtracking strategy to search feasible positions for collision residues in folding procedure. This particular algorithm constitutes a simple yet efficient approach for the purposed task. Its pseudocode is showed in Fig. 5.5.

The algorithm receives three parameters. The first one is λ, a table containing the allowed moves for each residue in the protein; thus, λ_k is a list of allowed moves for $(k+1)$-th residue and $\lambda_{k,r}$

is the r-th move. Although λ may contain in principle the full set of moves, in general $|\lambda|$ will not be the same for every k. The second parameter s is a partial conformation involving $|s|$ residues. As to the third parameter, it is a Boolean flag used to finalize the execution of the algorithm as soon as a feasible conformation is found. Notice finally that the operator :: represents the sequence concatenation operator.

5.4.2 Disadvantage of Traditional Backtracking-based Method

The basic backtracking method mentioned in the previous section has been shown to be a simple and efficient means of positional collision repairing for protein folding. But in practice, the repairing computational cost is very heavy for long sequence instances of more than 50 residues.

In the generic search procedure of protein folding, especially for the long protein sequences, the EAs based algorithm will produce a lots of valid and invalid individuals that contain closed-areas (or closed-spaces in 3-D circumstance). The Fig. 5.6 shows a valid 2-D individual's conformation contains two closed-areas. When the basic backtracking method is used to repair invalid individuals contain closed-areas, some invalid closed-areas made by backtracking searching folding procedure will produce computational cost wastes.

Take the 2-D circumstance as example, as showed in Fig. 5.7(a), there is a closed-area formed by residues from 1 to n. If the folding procedure select right (R) as the next direction for $n+1$ residue, it will enter the closed-area. Thus, even if the size of this closed-area can not satisfy the length of remain residues (called it as invalid closed-area), the traditional backtracking method will still search all empty position in closed-area by backtracking operation. This will produce a large number of computational cost wastes. According to our experiment, this phenomenon takes place with a high probability in repairing procedures for long sequence proteins.

5.4.3 Improved Method

To solve the above problem, in the improved method, a detection for feasibility is introduced. The detection procedure is implemented before selecting direction for next residue to avoid entering an invalid closed-area (The procedure $Detect - fea$ in Line 9 of Fig. 5.5).

The pseudocode of the introduced detection algorithm is shown in Fig. 5.8. The main idea of the detection procedure is described as follow.

1) The current boundaries in lattice is defined as shown in Fig. 5.7(a), the scale of boundary coordinates is larger one position than current filled area and will be changed with current folding procedure. For example, four current boundaries of the 2-D solution shown in Fig. 5.6 are $x = -11$

```
R-Backtracking ($\downarrow \lambda$:MOVE[], $\downarrow\uparrow s$:MOVE[],
$\uparrow SolutionFound$:bool).
  if $Feasible(s)$ then
    if $|s| = n - 1$ then
      $SolutionFound \leftarrow TRUE$
    else
      $SolutionFound \leftarrow FALSE$
      $i \leftarrow 1$
      while $\neg SolutionFound \wedge (i \leq \left|\lambda_{|s|}\right|)$ do
        if $Detect - fea$ ($\left\langle\lambda_{|s|,i}\right\rangle$, $s$) then
          $s' \leftarrow s :: \left\langle\lambda_{|s|,i}\right\rangle$
          R-Backtracking ($\lambda, s', SolutionFound$)
        endif
        $i \leftarrow i + 1$
        if $SolutionFound$ then
          $s \leftarrow s'$
        endif
      endwhile
    endif
  else
    $SolutionFound \leftarrow FALSE$
  endif
```

Figure 5.5: Pseudocode of the backtracking repairing algorithm.

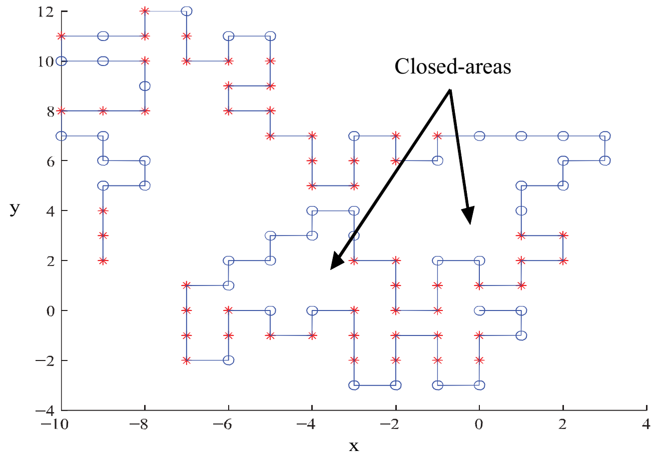


Figure 5.6: Illustration of the closed-areas in 2-D HP model.

(left), $x = 4$ (right), $y = 13$ (up) and $y = -4$ (down). They can be used to check the folding procedure whether enter a closed-area. If the detection meet the current boundaries, the folding procedure will not enter a closed-area.

2) A search approach, similar to Floodfill strategy, is utilized to count possible empty positions connected to the detected direction $\langle \lambda_{|s|,i} \rangle$, i.e. those empty positions which could be arrived through this direction. The Floodfill-like search approach count and label the possible empty positions based on a queue Q. If the queue Q becomes empty, it means that all possible empty positions are labeled.

3) In the operation of detection procedure, if the current boundaries are met or the number of counted empty positions is larger than the length of remain residues, it means that the folding will not enter a closed-area or entered closed-area is not invalid. Under such circumstance, the detected direction $\langle \lambda_{|s|,i} \rangle$ could be chosen for the next residue.

For long length protein sequences, there are many invalid closed-area in folding procedure. The improved method can significantly reduce the computational cost. Although the detection procedure has some computational cost, it is far less than the cost of backtracking searching operations for invalid closed-areas.

The main reason of the improvement is that the improved method can significantly reduce the number of backtracking operation. The folding procedure implements backtracking operation only under few special circumstances. As shown in Fig. 5.7(b), if the folding procedure has selected the right (R) direction for the $n + 1$ residues. But at $n + i$ position, the folding procedure produce two

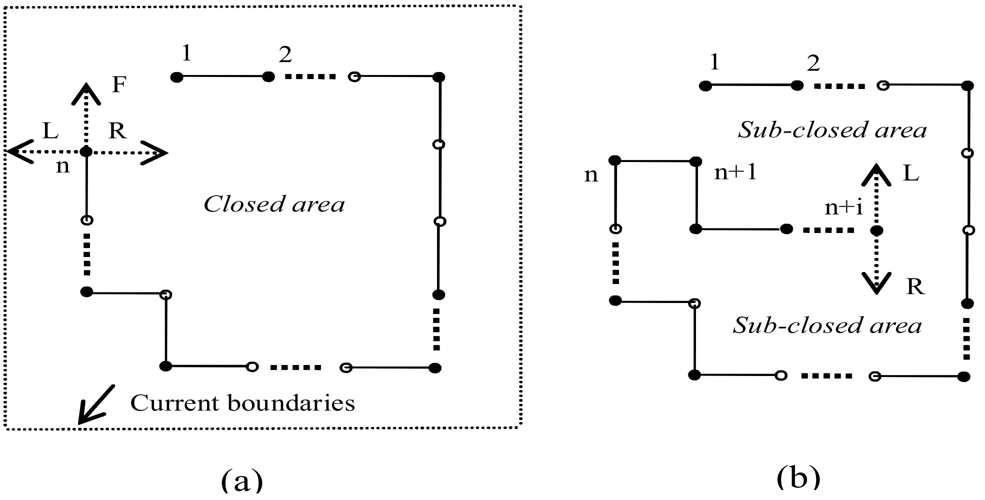


Figure 5.7: (a) Illustration of closed-area detection. (b) The situation that need to implement backtracking.

sub-closed-areas and all of two are invalid closed-area for remain residues. The folding procedure should implement a backtracking operation under this situation. It will back to the $n + i - 1$ residue and search other possible directions.

5.5 Experiments

5.5.1 Problem Benchmark

Eleven benchmark HP instances are used to evaluate the searching capability of the proposed method. The first nine instances are taken from the *Tortilla* 2-D HP Benchmarks [1], and the last two are taken from Ref. [75]. In Tab. 5.1, E^* is the optimal or best-known energy value, H_i, P_i and $(\cdots)_i$ indicate i repetitions of the relative symbol or subsequence. It is important to highlight that most randomly generated amino acid sequences do not behave like natural proteins, because the latter are products of natural selection. Likewise, most randomly generated sequences of H and P residues in the HP model do not fold to a single conformation [75].

5.5.2 Results of the Hybrid EDA for HP Model

In order to test the effects of the composite fitness function and the local search with guided mutation, different experiments are implemented by using one of them independently. The composite

[1] http://www.cs.sandia.gov/tech-reports/compbio/tortilla-hp-benchmarks.html

```
1.  ↑ Detect-fea (↓ $\langle\lambda\rangle$, ↓ $s$:MOVE[]):bool.
2.    Calculate current boundaries according to $s$
3.    Set label for every positions in $s$ (i.e. not empty).
4.    if $Feasible(s :: \langle\lambda\rangle)$ then
5.      $counter = 0$ and set an empty queue $Q$
6.      Add $\langle\lambda\rangle$ to the end of $Q$
7.      while $Q$ is not empty do
8.        $x$=first element of $Q$
9.        if position $x$ is unlabeled
10.         Set label for position $x$
11.         $counter = counter + 1$
12.       endif
13.       if (position $x$ meet the current boundaries) or ($counter$ is larger than the length of remain residues)
14.         Return TRUE
15.       endif
16.       Remove the first element of $Q$
17.       if west neighbor of $x$ is unlabeled
18.         Set label for $west - x$
19.         Add $west - x$ to the end of $Q$
20.       endif
21.       Check and process other three (five for 3D) neighbor positions of $x$ using similar strategies Step (17)-(20)
22.     endwhile
23.   endif
24.   Return FALSE
```

Figure 5.8: Pseudocode of the detection procedure.

Table 5.1: HP instances used in the experiments.

No.	Size	E^*	Sequence
s1	20	-9	$HPHP_2H_2PHP_2HPH_2P_2HPH$
s2	24	-9	$H_2P_2(HP_2)_6H_2$
s3	25	-8	$P_2HP_2(H_2P_4)_3H_2$
s4	36	-14	$P_3H_2P_2H_2P_5H_7P_2H_2P_4H_2P_2HP_2$
s5	48	-23	$P_2H(P_2H_2)_2P_5H_{10}P_6(H_2P_2)_2HP_2H_5$
s6	50	-21	$H_2(PH)_3PH_4P(HP_3)_2HPH_4(PH)_4H$
s7	60	-36	$P_2H_3PH_8P_3H_{10}PHP_3H_12P_4H_6PH_2PHP$
s8	64	-42	$H_{12}(PH)_2(P_2H_2)_2P_2H(P_2H_2)_2P_2H(P_2H_2)_2P_2(HP)_2H_{12}$
s9	85	-53	$H_4P_4H_{12}P_6(H_{12}P_3)_3H(P_2H_2)_2P_2HPH$
s10	100	-48	$P_6HPH_2P_5H_3PH_5PH_2P_4H_2P_2H_2PH_5PH_{10}PH_2PH_7P_{11}H_7$ $P_2HPH_3P_6HPHH$
s11	100	-50	$P_3H_2P_2H_4P_2H_3(PH_2)_3H_2P_8H_6P_2H_6P_9HPH_2PH_{11}P_2H_3P$ $H_2PHP_2HPH_3P_6H_3$

fitness function can help to reduce the complexity of EDA, it can obtain same results with basic k-order Markov EDA (MK-EDA) by using less population size and generation number. The local search with guided mutation can help to obtain the global optimum for some instances. But it seems that combination of two strategies can get much better results in practice. The performance of MK-EDA for $k \in \{2, 3, 4\}$ are investigated, and the algorithm performs very well when $k = 3$.

In the experiments of the hybrid EDA, all algorithms use a population size of 2000 individuals. Truncation selection is used as selection strategy. In this strategy, individuals are ordered by fitness, and the best $T * PopSize$ are selected where T is the truncation coefficient. The parameter $T = 0.15$ is used in our algorithms. The best elitism scheme is also implemented in algorithms, the set of selected solutions in the current generation are passed to the next generation. The stop criteria considered are a maximum number of generation $G = 1000$ or that the number of different individuals in the population falls below 5. For the protein instances of $s6$ to $s11$, the improved backtracking-based method is used to repair the invalid solutions.

The results of the proposed method comparing with the MK-EDA for the 2-D HP Model shown in Tab. 5.2. It includes the best solution and the percentage of times the best solution has been found in 100 experiments. The results of MK-EDA are also obtained by our experiments with same EDA parameters ($Pop = 2000$, $G = 1000$ and $T = 0.15$) as the proposed method. The experiment results show that the proposed method has more chance to find global optimum or suboptimum solution for long sequences. The MK-EDA cannot find the global optimum of the deceptive sequences and long sequences $s7$, $s9$, $s10$ and $s11$, but the proposed method can find the global optimum of the sequences $s7$, $s9$ and $s10$, and can find the second best solution for sequence $s11$. Figure 5.9 shows the best fitness for one representative run of the instance $S7$.

The performance of the proposed method comparing with the best results achieved with other evolutionary and Monte Carlo optimization algorithms is shown in Tab. 5.3 (2-D HP model) and Tab. 5.4 (3-D HP model). The results of other method are cited from Ref. [75, 46]. The experiment results show that none of the algorithms are able to outperform the rest of algorithms for all the instances. The PERM is one of the best contenders in all cases except $s8$ in which its result is very poor. It shows that the proposed method is very competitive with the other existing algorithms for the PSP on HP models. It should be noted that all fitness values of the proposed method in the comparing results are calculated by basic HP-model fitness definition. The composite fitness function is only used in optimization procedure of EDA.

Table 5.2: Results of comparing with MK-EDA for 2-D HP model.

		Proposed Method		MK-EDA	
No.	E^*	$H(X)$	$Percentage$	$H(X)$	$Percentage$
s1	-9	-9	100	-9	100
s2	-9	-9	100	-9	100
s3	-8	-8	100	-8	100
s4	-14	-14	16	-14	5
s5	-23	-23	22	-23	7
s6	-21	-21	92	-21	57
s7	-36	-36	24	-35	12
s8	-42	-42	16	-42	4
s9	-53	-53	8	-52	3
s10	-48	-48	12	-47	4
s11	-50	-49	6	-48	2

5.5.3 Results of Comparing Computational Cost

The hybrid EDA is an improved method based on the MK-EDA. The detailed computational cost analysis of the MK-EDA method can be found in the Ref. [75]. Comparing with the MK-EDA, t

local search with guided operations; 3) the improved backtracking-based repairing method for the

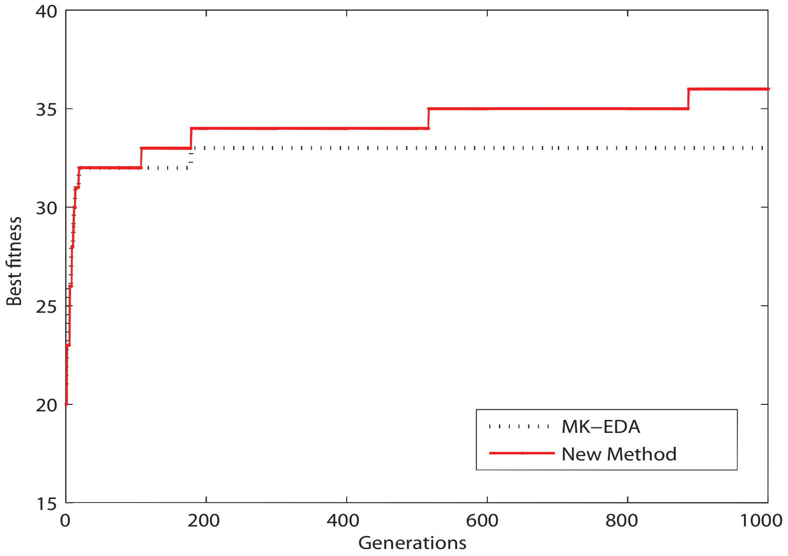


Figure 5.9: The best fitness for one representative run of instance $s7$.

Table 5.3: Results achieved by different search methods for 2-D HP model.

No.	Proposed Method $H(X)$	MK-EDA $H(X)$	GA $H(X)$	NewACO $H(X)$	PERM $H(X)$
s1	-9	-9	-9	-9	-9
s2	-9	-9	-9	-9	-9
s3	-8	-8	-8	-8	-8
s4	-14	-14	-14	-14	-14
s5	-23	-23	-22	-23	-23
s6	-21	-21	-21	-21	-21
s7	-36	-35	-34	-36	-36
s8	-42	-42	-37	-42	-38
s9	-53	-52		-51	-53
s10	-48	-47		-47	-48
s11	-49	-48		-47	-50

Table 5.4: Results achieved by different search methods for 3-D HP model.

No.	Proposed Method $H(X)$	MK-EDA $H(X)$	Hybrid GA $H(X)$	IA $H(X)$
s1	-11	-11	-11	-11
s2	-13	-13	-11	-13
s3	-9	-9	-9	-9
s4	-18	-18	-18	-18
s5	-29	-29	-28	-28
s6	-30	-29	-22	-23
s7	-49	-48	-48	-41
s8	-51	-50	-46	-42

long protein instances. As far as the computational cost is concerned, modifications of 1) and 2) will produce some additional computational cost. The modification 3) can significantly reduce the repairing costs for EDA invalid individuals.

To demonstrate the computational cost of the hybrid EDA comparing with MK-EDA, some practical experiments in 2-D are implemented. Two comparing methods are implemented with same parameters (population:1000, generation:100, the truncation selection of parameter $T = 0.15$) and same computational environment [2]. Because there are few closed-areas existing in short protein folding, the improved backtracking-based repairing method can not improve the EDA efficiency for short instances. In the comparing experiments for the short instances of $s1$ to $s5$, same basic backtracking repairing methods are used in two comparing methods. For the long instances of $s6$ to $s10$, the improved backtracking-based repairing method is used in the hybrid EDA.

The number of backtracking searching operation and computer CPU-Time are recorded. The average backtracking searching operations and the CPU-Times of 10 runs are shown in Tab. 5.5. According to the results of the short instances of $s1$ to $s5$, it can be found that the local search operations and the composite fitness calculation in the hybrid EDA produce some additional computational costs. But it is not very serious. The results of the long instances of $s6$ to $s10$ show that t It not only covers the additional computational costs caused by local search and composite fitness calculation, but also improves the algorithm efficiency remarkably.

The backtracking searching operations of each generation for sequence $s8$ (the length is 64),

[2] All experiments are performed on the computers with Intel Xeon 2.20 GHz processor, and 1 GB of RAM.

Table 5.5: Comparing results of the improved backtracking repairing method in 2-D HP model.

		AVG-Backtracking operation		AVG-CPUTime (Hour)	
No.	Size	MK-EDA	Proposed Method	MK-EDA	Proposed Method
s1	20	2.1492E+4	2.1496E+4	0.2181	0.2309
s2	24	2.5715E+4	2.5715E+4	0.2659	0.2761
s3	25	2.6834E+4	2.7044E+4	0.2774	0.2805
s4	36	3.8898E+4	3.8797E+4	0.3059	0.3203
s5	48	5.2577E+5	5.2617E+5	0.4341	0.4659
s6	50	5.7842E+6	*5.4887E+6	0.6249	0.5768
s7	60	7.4545E+6	*6.7478E+6	0.9276	0.7516
s8	64	9.6731E+6	*7.2236E+6	1.1276	0.8661
s9	85	2.0829E+8	*1.1196E+7	12.3077	1.6086
s10	100	2.6531E+8	*1.9765E+7	15.7125	2.0007

$s9$ (the length is 85) and $s10$ (the length is 100) are also be counted and shown in Fig. 5.10, Fig. 5.11 and Fig. 5.12. It can be found that the improved backtracking-based repairing method can significantly reduce the number of backtracking searching operation. And the longer the protein sequence length is, the more remarkable the improvement achieves.

5.6 Conclusions

In this chapter, a novel hybrid EDA method is introduced to solve the HP model problem. For the basic k-order Markov EDA, it has very low chance to find the general optimum for those long sequence and deceptive protein instances. A composite fitness function based on the information of folding structure core (H-Core) is proposed to replace the traditional fitness function. It can help to select better individuals for probabilistic model of EDA algorithm. In addition, local search with guided operators is utilized to refine found solutions for improving efficiency of EDA.

For the disadvantage of heavy computational cost of the traditional backtracking method which used to repair the invalid individuals in population. It will produce heavy computational cost for searching invalid closed-areas of folding structure. For the long protein sequences, an improved method is proposed to reduce the repairing computational cost. When selecting directions for the residues, a detection procedure for feasibility is introduced to avoid entering invalid closed-areas. Therefore, the number of backtracking searching operation and the computational cost can be significantly reduced for long sequence protein. It can be noted that the improved backtracking repairing

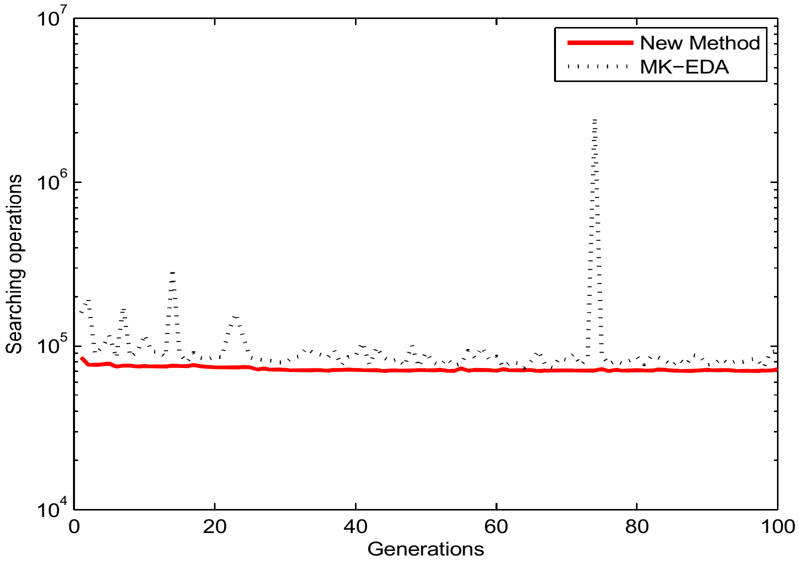


Figure 5.10: The number of backtracking searching operations for instance $s8$.

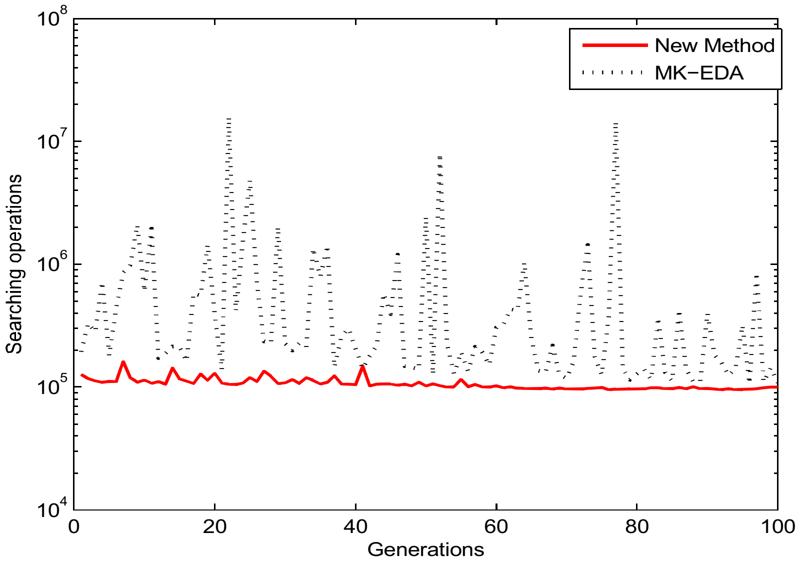


Figure 5.11: The number of backtracking searching operations for instance $s9$.

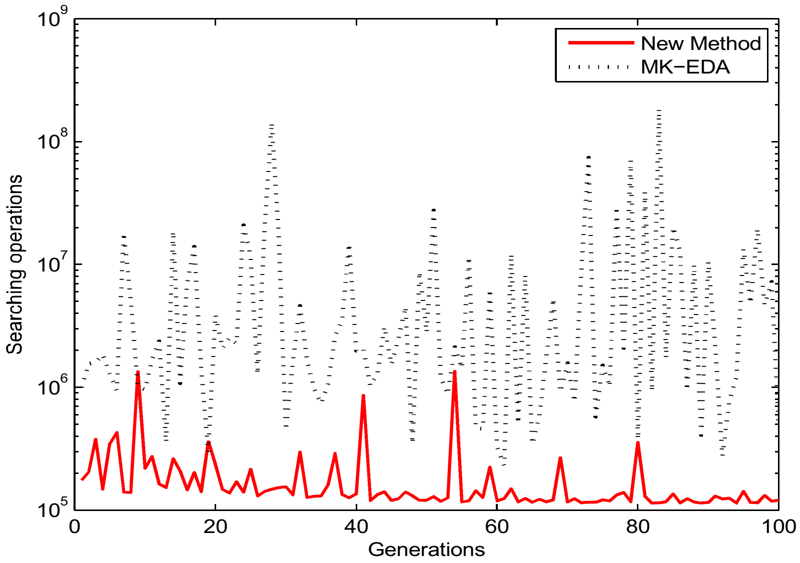


Figure 5.12: The number of backtracking searching operations for instance $s10$.

method can be used in all EA based PSP methods that need to repair invalid individuals. And the underlying mutations are implemented for individuals in repairing procedure.

Experimental results show that the proposed method has better performance than the basic EDA approach. Meanwhile, the proposed method is very competitive with other existing methods for the HP model. Further research is needed to determine more efficient local search strategies and probabilistic models of EDA for protein HP model problem.

Chapter 6

Protein HP Model Folding Based on Adaptive Niching EDA with Balance Searching

6.1 Introduction

Estimation of Distribution Algorithms (EDAs) [55] are a new class of evolutionary algorithms. Such approaches, instead of using conventional crossover and mutation operations, use probabilistic models to sample the genetic information in the population. The use of probabilistic models, especially those models taking into account bivariate or multivariate dependencies between variables, allows EDAs to capture genetic tendencies in current population effectively. In brief, these algorithms construct, in each generation, a probabilistic model that estimates the probability distribution of selected solutions. Dependency regulars are then used to generate next generation solutions during a sampling step. It is expected that the generated solutions share a number of characteristics with the selected ones. In this way, the search leads to promising areas of the search space [55].

By explicitly handling interactions between different variables that represent the individuals, the EDA based methods can get much better results than other population-based methods when solving some optimization problems. But for those optimization problems with irregular and complex multimodal landscapes, the EDAs still suffer from the drawback of premature convergence similar to other Evolutionary Algorithms (EAs). A simple yet popular explanation for the occurrence of premature convergence is the loss of diversity. But high diversity can not always guarantee a good performance for EA. It is a challenge which is called as the question of exploration vs. exploitation [61, 19]. At the early stage of evolution, enhancing diversity measures can ensure EA to implement a broad exploration in the search space. But such enhancement may be counterproductive or even destructive for EA at the stage when high exploitation is necessary. Therefore, implementing a modifiable balance between exploration and exploitation is very important for this issue [2, 35, 84].

For solving complex optimization problems, an efficient EA searching procedure with a balance between exploration and exploitation can be imagined as follows. At the beginning phase, the searching procedure should cover the search space as much as possible to avoid the premature convergence. But at the end phase of searching, the exploitation should be enforced to find good solutions. In other words, the degree of exploration should be monotonically decreasing, while the degree of exploitation should be monotonically increasing during the searching procedure. Inspired by this thought, in this chapter, an adaptive niching EDA based on Affinity Propagation (AP) clustering analysis [38] is proposed to improve the performance of EDA. As a machine learning method, the AP clustering is used as an intelligent and efficient way to mine the information of EDA search procedure. It is not only used to partition the niches for population adaptively, but also used to mine the searching history information of EDA to guide the EDA searching.

The clearing procedure [68] is used as a basic niching frame in the proposed method. 1) AP clustering is used to adaptively partition the population into niches during a run of EDA. The individuals are clustered before submitting them to niching clearing. A cluster can be seen as a niche, and the niche number and the niche radiuses may vary obviously for different generations. 2) A mechanism of niche capacity selection based on the Boltzmann scheme is proposed to realize a balance searching between exploration and exploitation. A niche novelty metric is proposed to record the searching history information by detecting the overlapping distributions of the niches in successive generations. The metric is used to evaluate the degree of search space that has been explored by the previous searching. Two different selection strategies, novelty-proportionate selection based on the novelty metric and fitness-proportionate selection based on the best fitness in a niche, are assembled by a Boltzmann weight in the niche capacity selection mechanism. Tuned by the Boltzmann weight, at the beginning phase of searching, the dominating novelty-proportionate selection can guide the EDA searching to cover the search space as much as possible; at the end phase of searching, the dominating fitness-proportionate selection can enforce the EDA searching to exploit the already found promising areas.

The proposed adaptive niching EDA is used for solving the protein HP model folding. For the protein HP model problems, the k-order Markov EDA proposed in Ref. [75] outperforms other population-based methods. But for most of the long protein sequences, the chance of finding the global optimum is very low, and for some deceptive sequences, it can only find the suboptimum solutions. In this chapter, the proposed method is used to improve the performance of the k-order Markov EDA for protein HP problem. In addition, three benchmark functional multimodal optimization problems based on continuous EDA with single Gaussian probabilistic model are also used to evaluate the proposed method in continuous situation. Experiment results show that the proposed adaptive niching EDA is an efficient method.

The rest of the chapter is organized as follows. Section 6.2 briefly introduces the premature convergence and the AP clustering. Section 6.3 formulates the proposed adaptive niching EDA based on AP clustering analysis. Section 6.4 carries out experiments on solving the optimization problem of HP model protein folding based on a k-order Markov probabilistic model. Finally, Section 6.5 presents the conclusions.

6.2 Premature Convergence and Niching Clearing

6.2.1 Premature Convergence

A critical problem when dealing with EAs is the phenomenon of premature convergence. Premature convergence occurs when the population of an evolution searching reach a suboptimal situation. In such suboptimal situation, the search process of EA is trapped in a local suboptimal region, and the genetic operators can not generate better offspring solutions than their parents. The loss of diversity is always regarded as the main reason for the occurrence of premature convergence. The diversity means a genetic variation of the population members in this context [40].

Various enhanced strategies for diversity maintenance have been proposed which target different stages of the evolution process. Niching method is one of the well-known strategies proposed to reduce the effects of genetic drift resulting from the selection mechanism, to allow the formation and the maintenance of different solutions and to prevent the EA from being trapped in local optima [40]. In the niching method, the analogy with nature is straightforward. As in an ecosystem there are different subsystems (niches) that contain many diverse species (subpopulations). The number of elements in a niche is determined by its resources and by the efficiency of each individual in taking profit of these resources. Using this analogy, it is possible for a EA to maintain its population diversity during the generations.

6.2.2 Niching Clearing Procedure

Among niching methods, fitness sharing based methods are widely known and used [23, 24, 81, 47, 29]. In sharing algorithms, the number of individuals in a niche is limited to the caring capacity of the niche. Individuals within a niche must share the fitness payoff offered by their niche. By treating fitness as a resource and sharing it among all niche members it restricts the maximum number of individuals that can be supported by the niche. When a niche becomes overcrowded, payoff is spread too thin and it in each individual's best interests to seek out another less crowded niche where they can get more payoff (fitness). Niches are a subdividing of the environment, or the fitness landscape. A population can be thought to specialize in different areas of the environment, where the idea of sharing resources between individuals induces a powerful niching behavior. It has been said [47] that through sharing, a form of localized competition for local resources leads to implicit cooperation between subpopulations (niches).

The clearing strategy [68] is another niching method. Different from the fitness sharing, the clearing strategy is derived from the concept of limited resources of the environment. The available resources are shared among all members of a subpopulation in the fitness sharing. But in clearing strategy, those limited resources are only used to supply the best individuals of each subpopulation.

As well as sharing, clearing is applied after the fitness evaluation and before selection, using a dissimilarity measure to separate the individuals in niches. Similar to in the sharing strategies, individuals are classified to a same niche if their distance in the search space is less than a dissimilarity threshold σ (is called as clearing radius).The maximum number of individuals that a niche can contain is defined as the capacity κ. In the clearing strategy, the κ best individuals (are called as dominant individuals) of a niche are preserved, and the fitness of others that belong to the same nich (are called as dominated individuals) are reset. The elitism strategies can also used in clearing naturally to preserve the best elements of niches during the generations. If the two parameters (the niche radius and the niche capacity) are correctly estimated, the clearing method can be considered as an excellent niching method [76].

6.3 Adaptive Niching EDA with Balance Searching

The clearing procedure is used as a basic niching frame in the proposed adaptive niching EDA. To implement the niching clearing in the selection procedure of EDAs, the individuals are firstly grouped into many niches by a certain partition strategy. A static radius σ is used to partition the niches in a conventional niching clearing method. Then, the κ best individuals are selected from

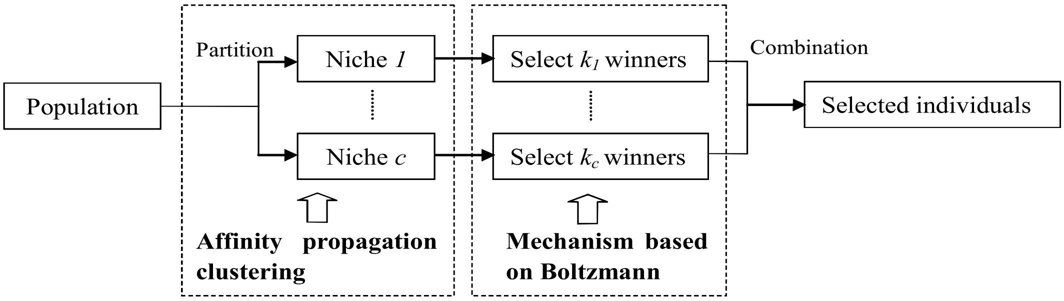


Figure 6.1: Selection procedure of the adaptive niching EDA.

each niche respectively and a combination of selected individuals is used to estimate probabilistic model of EDA. The capacity parameter κ is defined as the maximum number of winners that a niche can accept. In the conventional method, the capacity κ is set as a static value for all niches, and it can be set in a range from 1 to the population size, so that the niching effect can be, respectively, maximized or minimized as convenient.

In the proposed adaptive niching EDA, as showed in Fig. 6.1, a balance niching searching strategy based on adaptive clustering is introduced into the selection procedure to improve the performance of EDA. 1) The values of the clearing radius σ are determined adaptively by AP clustering during a run of EDA. The individuals are clustered before submitting them to the niching clearing and a cluster can be seen as a niche. 2) A niche capacity κ selection mechanism based on the Boltzmann scheme is utilized to realize a balance searching. Two different selection strategies of considering exploration and exploitation are assembled by the Boltzmann scheme in the selection mechanism. A combination of the selected individuals from all niches is used to estimate the probabilistic model of EDA.

6.3.1 A Mechanism of Niche Capacity Selection

In each iteration of the niching EDA, a combination of the selected individuals from all niches is used to estimate a probabilistic model to capture genetic tendencies in the current population. For a certain niche, the number of winners (the κ selected individuals) represents the degree of EDA searching for this niche area. Therefore, the niche capacity selection can be utilized to tune the searching behavior of EDA. An efficient EDA with a balance searching should have the behaviors described as follows. The degree of exploration is monotonically decreasing, while the degree of exploitation is monotonically increasing during the search run. To realize an efficient searching procedure with a balance between exploration and exploitation, a mechanism of niche capacity selection based on the Boltzmann scheme is defined as follows.

$$\begin{aligned} \kappa_t(n_i) &= M \times [\omega_t \times Ei_t(n_i) + (1-\omega_t) \times Er_t(n_i)] \\ \omega_t &= \left[\frac{e^{\left(\frac{t}{T_{\max}}\right)} - 1}{e-1} \right]^{\alpha}, \quad 1 \le i \le cl_t \end{aligned} \tag{6.3.1}$$

where M is the number of selected individuals in each iteration, n_i denotes the i-th niche, ω_t the Boltzmann weight of the t-th iteration, $T_{\max}$ the maximum number of iterations, α the scaling factor used to tune the Boltzmann scheme, cl_t the cluster number of the t-th iteration population, $Ei_t(n_i)$ and $Er_t(n_i)$ the selection probabilities of the i-th niche in the t-t

. The detail definitions of $Ei_t(n_i)$ and $Er_t(n_i)$ will be described in the following subsections.

The round value $\lfloor \kappa_t(n_i) \rfloor$ is used as the winner number of niche n_i. A combination of the winners from all niches is used as the selected individuals (the number is $\sum_{i=1}^{cl_t} \lfloor \kappa_t(n_i) \rfloor$) to estimate the probabilistic model of the t-th generation.

In the proposed mechanism of niche capacity selection, two different selection strategies are assembled by a Boltzmann weight ω_t related with the generation t. Tuned by the Boltzmann weight ω_t, at the beginning phase of EDA, the searching procedure explores the search areas as more as possible by dominating the exploration strategy; at the end phase of searching, the searching exploitation is enforced to refine the already found solutions by dominating the exploitation strategy.

6.3.2 The Exploration Probability $Er_t(n_i)$

According to the definition of the niche capacity selection mechanism described in the previous subsection, the exploration strategy (represented by the probability $Er_t(n_i)$) is expected to preserve a diversity in the population of EDA. It should guide the EDA searching to cover the solution space as much as possible to avoid premature convergence. Selecting the same number of winners from all niches is a simple exploration strategy, i.e., the niche capacities for all niches are set as the average value decided by the selected population size and the cluster number of current population.

But the average selection is not a good strategy suitable to the adaptive niching scheme. Because in the adaptive niching scheme, the individuals in the population are separated into niches adaptively; in addition, the niche number and the niche radiuses vary obviously in different generations. The varied niche radiuses may produce a problem of irregular niche distribution, i.e., at some generations, too many niches are distributed in a search space that has already been explored by the previous searching. A simple experiment can be used to explain this problem. An experiment of

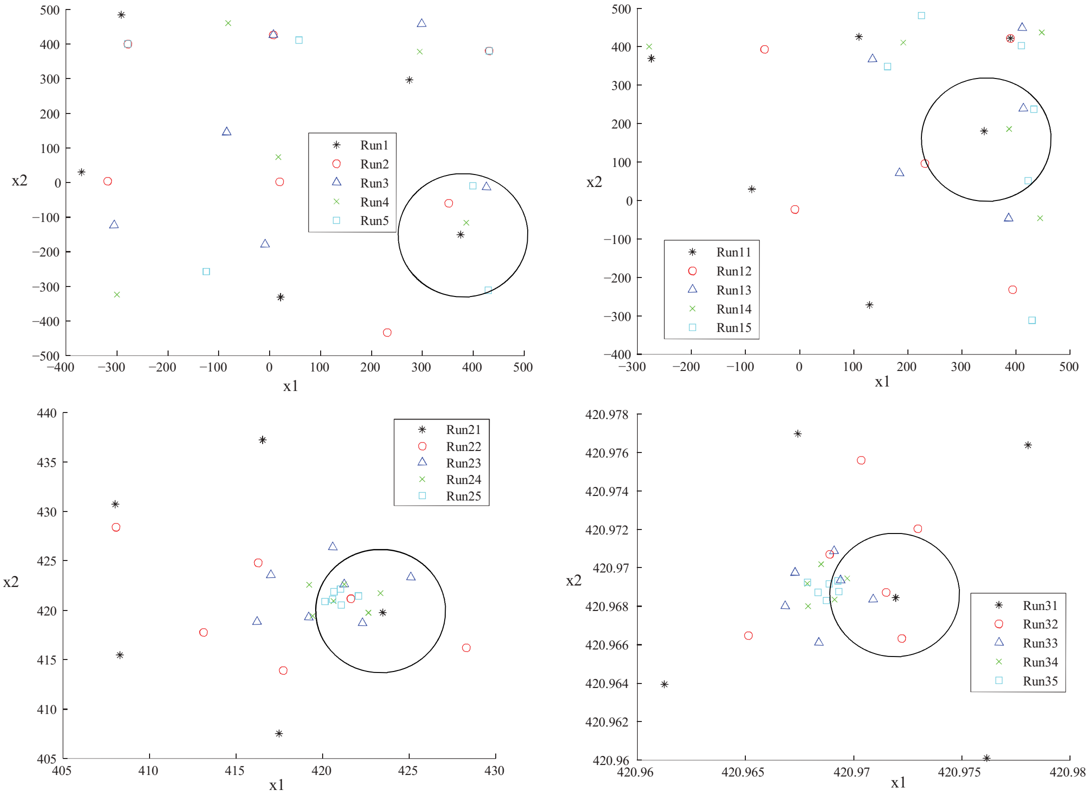


Figure 6.2: Population clustering center exemplars of Schwefel function (dimemsion=2) optimized by UMDA$_c$.

optimizing the 2-dimension Schwefel function (Eq. 6.4.3 and the function landscape showed in Fig. 6.5) is implemented by the continuous Univariate Marginal Distribution Algorithm (UMDA$_C$) [55]. The function domain is [-500 500] and the population size is 50. The populations of four different searching phases (generations of 1-5, 11-15, 21-25 and 31-35) are stored and analyzed by AP clustering. The obtained cluster center exemplars are showed in Fig. 6.2. From the distributions of cluster center exemplars in different generations, it can be found that the adaptive niching scheme often produces the irregular niche distribution problems. To solve this problem, two levels of diversity preservation are considered in the adaptive niching scheme: one is the diversity preservation at individual level; the other is the one at niche (subpopulation) level.

In the proposed exploration searching strategy, the population clustering information is utilized to record the EDA searching history for solving the irregular niche distribution problem. From the overlapping information of the cluster center exemplar distribution in successive generations, the search space parts which have been explored by the previous searching can be found. A niche *novelty* metric is proposed to record the overlapping information, and it is used to evaluate the degree of the corresponding niche explored by the previous searching. The niches with smaller novelty metric values will be punished moderately at the exploration searching strategy to realize the diversity preservation at the niche level.

The niche *novelty* metric $nov_t(n_i)$ of niche n_i in the t-t

$$nov_t(n_i) = \ {}^{O_t(n_i)}, \quad 0 < \beta < 1 \tag{6.3.2}$$

where n_i denotes the i-th niche, β is the penalty parameter for niche novelty, and $O_t(n_i)$ the overlapping parameter of niche n_i in the t-th generation.

Let us describe a niche n_i at generation t by its center $C_t(n_i)$ and radius $\sigma_t(n_i)$ where the center determining its position and the radius determining its area. If the center of a niche n_i of t generation is inside the niche n_j of $t-1$ generation, that is, the distance between two centers $C_{t-1}(n_j)$ and $C_t(n_i)$ is less than the radius $\sigma_{t-1}(n_j)$, the niche n_i of t generation is said to be overlapped by the niche n_j of $t-1$ generation.

Set $O_t(n_i) = 0$ as the initial value. If the niche n_i of t generation is one of the $k(k > 1)$ niches that are overlapped by the niche n_j of $t-1$ generation, then set $O_t(n_i) = O_{t-1}(n_j) + 1$ with a probability of $(k-1)/k$. That is, among the k niches only one keeps 0, while other $k-1$ niches are set to $O_{t-1}(n_j) + 1$.

The niche novelty metric reflects the information of search space explored by the EDA. The exploration searching probability $Er_t(n_i)$ is defined by a novelty-proportionate strategy as follows.

$$Er_t(n_i) = \frac{nov_t(n_i)}{\sum_{j=1}^{cl_t} nov_t(n_j)} \tag{6.3.3}$$

where cl_t is the cluster number of the t-th iteration population.

6.3.3 The Exploitation Probability $Ei_t(n_i)$

A popular fitness-proportionate method is used as the exploitation strategy in the proposed adaptive niching EDA. The exploitation searching probability $Ei_t(n_i)$ is defined as follows.

$$Ei_t(n_i) = \frac{f_t(n_i)}{\sum_{j=1}^{cl_t} f_t(n_j)} \tag{6.3.4}$$

where $f_t(n_i)$ is the maximum fitness value of the i-th niche in the t-th iteration.

The fitness-proportionate niche capacity selection is defined based on the best fitness value in a niche. It can guide the EDA to search the already found promising areas of searching space.

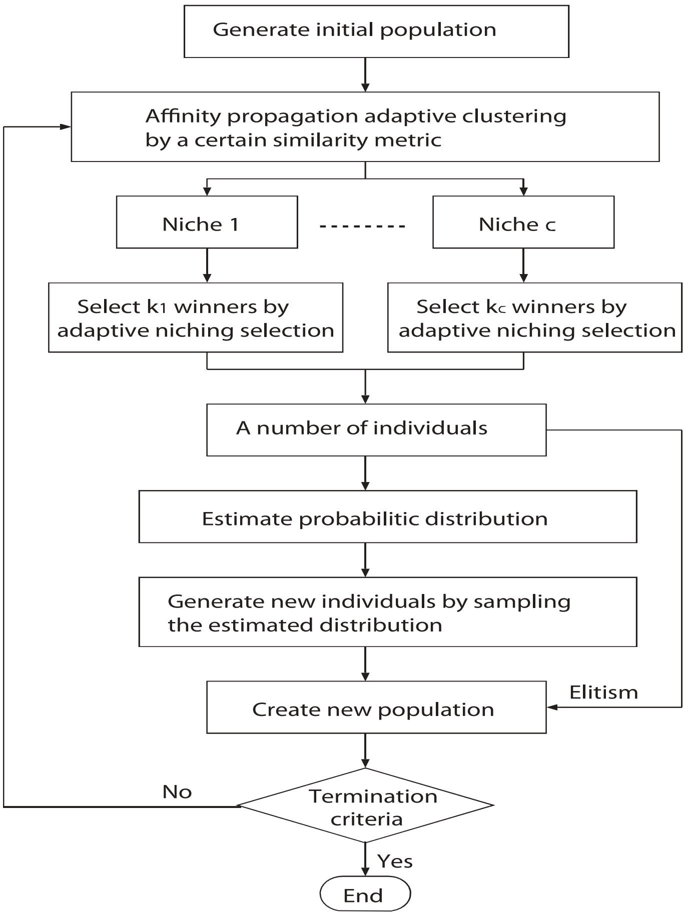

Generate initial population
Affinity propagation adaptive clustering
by a certain similarity metric
Niche 1
Niche c
Select k1 winners by
adaptive niching selection
Select kc winners by
adaptive niching selection
A number of individuals
Estimate probabilitic distribution
Generate new individuals by sampling
the estimated distribution
Create new population
Elitism
No
Termination
criteria
Yes
End

Figure 6.3: Framework of the proposed adaptive Niching EDA

6.3.4 Framework of the Proposed Method

Following a typical EDA framework, as showed in Fig. 6.3, the major steps of the proposed adaptive niching EDA can be summarized as follows.

1) Generate the first population of N individuals randomly. The individuals should be initialized to cover different areas in the search space.

2) Repair invalid individuals and evaluate each individuals of the current population. Then the individuals of current population is grouped into clusters by using AP method based on a certain similarity metric.

3) Calculate the novelty metric for each cluster (niche) by Eq. 6.3.2 and evaluate each niche by its best solution found. According to the proposed niche capacity κ selection mechanism

of the Boltzmann scheme defined by Eq. 6.3.1, a number of $\sum_{i=1}^{cl_t} \lfloor \kappa_t(n_i) \rfloor$ individuals are selected from the niches obtained in Step 2).

4) Estimate the probabilistic model from the selected individuals.

5) Generate a new population of individuals based on the probabilistic model obtained in Step 4). The elitist preservation strategy is implemented, so a set of good solutions in the current generation is passed to the next generation.

6) Repeat Steps 2) to 5) until certain stop criterion is met (e.g. a maximum number of generations, a homogeneous population, or no improvement after a certain number of generations).

6.4 Experiments

6.4.1 Experiments on Continuous Benchmark Functions

To evaluate the proposed method for continuous optimization problems, the proposed adaptive niching EDA is applied to solving three benchmark functional multimodal problems firstly. In cases of continuous EDAs, Gaussian model is the most widely used probabilistic model. But there are difficulties in solving those multimodal functions with irregular and complex landscapes, since conventional EDAs are said to be easily misled by such kinds of deceptive landscapes and then converge to poor results.

Benchmark Functions

The following three benchmark functions from [97] are used to evaluate the proposed method.

1) Kowalik function

$$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2 \tag{6.4.1}$$

where $-5 \le x_i \le 5$, and a_i is [0.1957 0.1947 0.1735 0.1600 0.0844 0.0627 0.0456 0.0342 0.0323 0.0235 0.0246], b_i^{-1} is [0.25 0.5 1 2 4 6 8 10 12 14 16].

2) Shekel function

$$f(\vec{x}) = \sum_{i=1}^{n} [(\vec{x} - \vec{a_i})(\vec{x} - \vec{a_i})^T + \vec{c_i}]^{-1} \tag{6.4.2}$$

where n is set to 5 in the experiment and the parameters of a_{ij} and c_i are listed in Table 6.1.

Table 6.1: Parameters of the Shekel function ($n = 5$)

i	a_{i1}	a_{i2}	a_{i3}	a_{i4}	$\cdots$	c_i
1	2	2	2	2	$\cdots$	0.1
2	4	4	4	4	$\cdots$	0.2
3	8	8	8	8	$\cdots$	0.2
4	6	6	6	6	$\cdots$	0.4
5	3	7	3	7	$\cdots$	0.4

Table 6.2: Settings for the three test functions

	Dimension	Domain	Type	Optimum
Kowalik	4	[-5 5]	Min.	0.0003075
Shekel(n=5)	30	[0 10]	Max.	10.0134
Schwefel	30	[-500 500]	Min.	-12569.5

3) Schwefel function.

$$f(\overrightarrow{x}) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|}) \tag{6.4.3}$$

Table 6.2 lists the specifications of three test functions. And the particular two-dimension cases for the Shekel and Schwefel are showed in Fig. 6.4 and Fig. 6.5.

Experiment Results

Three EDA based methods of UMDA$_C$, CEGDA$_{BGe}$ [60] and NichingEDA [33] are used to compare with the proposed method. Among them, the results of CEGDA$_{BGe}$ and NichingEDA are cited from their original papers, the results of the UMDA$_C$ are obtained by our experiments. In Ref. [60, 33], the maximal evaluation number for CEGDA$_{BGe}$ and NichingEDA are 4×10^5 and 5×10^5, respectively; all results have been averaged over 30 independent runs.

In our experiments, the maximal evaluation number for UMDA$_C$ and the proposed method are set as 4×10^5 (population: 1000 and generation: 400). A Gaussian probabilistic model similar to the UMDA$_C$ is used in the proposed method. The initial populations are sampled by randomly in the function domain. A truncation selection with size of 500 is used in all algorithms. And the elitism strategy is utilized in the evolution processing. For those invalid individuals sampled in each generation, a repairing procedure is used to change the values of each out of range variable

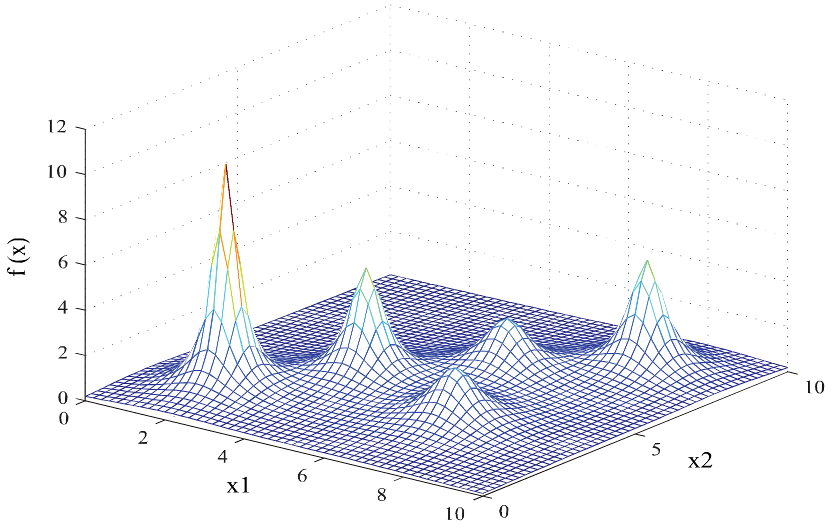


Figure 6.4: Plot of the Shekel function (dimension=2).

to the minimum (respectively maximum) bounds of the function domain. The penalty parameter α in Eq. 6.3.1 for the niche novelty and the scaling factor β in Eq. 6.3.2 for the Boltzmann scheme are set as: $\alpha = 2$ and $\beta = 0.95$, respectively. About the parameters of AP clustering (defined in Ref. [38]), Euclidean distance is used as the similarity metric of individuals; the maximum number of iterations is set as 1000, and early terminate parameter is set as 100 (i.e., the clustering procedure will be terminated if the estimated exemplars stay fixed for continuous 100 iterations); the damping factor, which may be needed if oscillations occur, is set as 0.9.

The results on three functions are listed in Table 6.3, Table 6.4 and Table 6.5. Based on the best solutions that the algorithms have found in 30 independent runs, the best values, mean values and standard errors are calculated for comparison. The results show that the proposed method outperforms other methods for solving the three functional optimizations. To observe the computational overheads of the proposed method, the CPU-Time costs (the computers with Intel Xeon 2.20 GHz processor, and 1 GB of RAM) are listed in Table 6.6.

6.4.2 Experiments on Protein HP Model

In this section, the proposed adaptive niching EDA is applied to solving a real complicated discrete EDA optimization problem, the HP model protein folding based on a k-order Markov (MK-EDA)

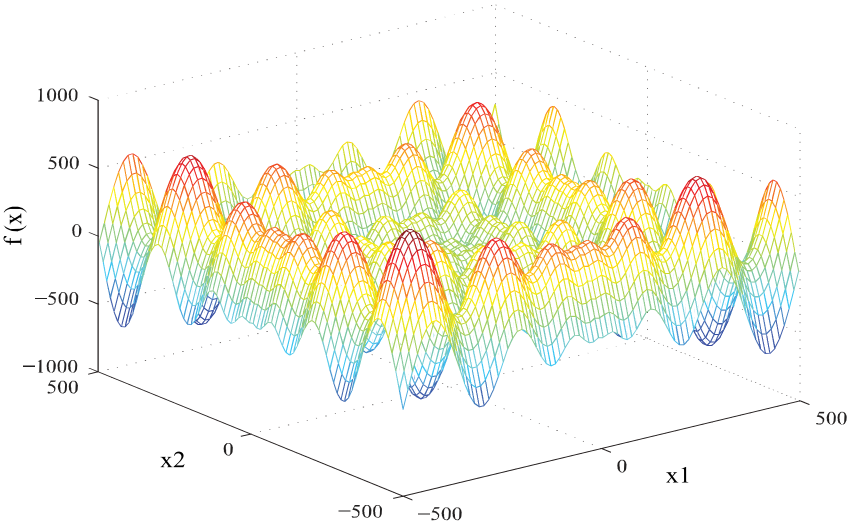


Figure 6.5: Plot of the Schwefel function (dimension=2).

Table 6.3: Experimental results for the Kowalik function

Algorithm	Best	Mean	Std
UMDA$_c$	0.00085553	0.0031	0.0017
NichingEDA	0.00030750	0.00030874	2.7899e-006
Proposed	0.00030750	0.00030791	4.173e-008

probabilistic model.

Experiment Setting

In this chapter, the representation of internal coordinates with relative direction [52] is used. The position of each residue depends upon the previous move. Relative direction representation presents the direction of each residue relative to the main chain next turn direction. This representation can reduce the direction number of each position. For 2-D HP model, the set of direction is left, right and forward (L, R, F). It is left, right, forward, up and down, (L, R, F, U, D) for the 3-D HP model.

During the course of EDA search procedure, initialization and sampling from probabilistic model operations may produce a lot of invalid individuals including positional collisions. A positional collision is the embedding of two different residues onto the same vertex of the lattice. The improved

Table 6.4: Experimental results for the Shekel function (Dim=30)

Algorithm	Best	Mean	Std
UMDA$_c$	5.1325	4.3108	0.9326
NichingEDA	0.3550	0.2230	0.0564
CEGDA$_{BGe}$	10.0134	10.0134	8.8818e-015
Proposed	10.0134	10.0134	4.7622e-015

Table 6.5: Experimental results for the Schwefel function

Algorithm	Best	Mean	Std
UMDA$_c$	-5928.24	-5424.81	202.437
NichingEDA	-8733.51	-8005.39	270.755
CEGDA$_{BGe}$	-10773.1	-6760.35	2624.33
Proposed	-11064.82	-8741.61	2124.33

backtracking-based repairing method proposed in Ref. [18] is used to repair invalid individuals for the long sequence protein instances. In this repairing method, to avoid entering invalid closed-areas, a detection procedure for feasibility is introduced when selecting directions for the residues in backtracking searching procedure. It can significantly reduce the number of backtracking searching operation and the computational cost for the long protein sequences.

$$p_{MK}(X) = p(x_1, \cdots, x_{k+1}) \prod_{i=k+2}^{n} p(x_i | x_{i-1}, \cdots, x_{i-k}). \tag{6.4.4}$$

The k-order Markov model of EDA is described by Eq. 6.4.4. It can encode the dependencies between the move of a residue and the moves of the previous residues in the sequence, and this information can be used in the generation of solutions. To sample a new solution, first variables in the factor $(x_1, \cdots, x_{k+1})$ are generated and the rest of variables are sampled according to the order specified by the Markov factorization.

Table 6.6: Computing costs of the proposed method and the UMDA$_c$.

Algorithm	Kowalik	Shekel	Schwefel
UMDA$_c$ (second)	11.27	13.34	14.95
Proposed (second)	1551.65	1618.73	1692.39

Experiment Results

In our experiments, the benchmark protein instances ($s4$ to $s11$ in Chapter 5 Tab. 5.1) from the *Tortilla* 2-D HP Benchmarks and Ref. [75] are used to evaluate the searching capability of the proposed method.

Table 6.7: Results of comparing with MK-EDA and the proposed method for 2-D HP model.

			Proposed method				MK-EDA			
No.	E^*	$Runs$	$Best$	$Perc.$	$Mean$	$Time(H)$	$Best$	$Perc.$	$Mean$	$Time(H)$
s4	-14	100	-14	32%	-13.87	1.5324	-14	5%	-11.05	0.5321
s5	-23	100	-23	49%	-22.72	1.9113	-23	7%	-20.45	0.7112
s6	-21	100	-21	96%	-20.83	2.5884	-21	57%	-18.32	0.9125
s7	-36	100	-36	34%	-35.02	3.1270	-35	12%	-32.65	1.5278
s8	-42	100	-42	28%	-41.71	4.5327	-42	4%	-36.24	2.1232
s9	-53	15	-53	13.33%	-51.91	11.4751	-52	6.67%	-46.62	8.5321
s10	-48	15	-48	6.67%	-46.77	16.6680	-47	6.67%	-43.24	10.2117
s11	-50	15	-49	6.67%	-47.81	15.9712	-48	6.67%	-44.38	9.5449

The results of MK-EDA are also obtained by our experiments, the maximal evaluation number of the proposed method and MK-EDA are set as 2×10^6, all algorithms use a population size of 2000 individuals. Truncation selection parameter $T = 0.15$, i.e., the best $T * PopSize$, are selected for learning probabilistic model. The penalty parameter α in Eq. 6.3.1 for the niche novelty and the scaling factor β in Eq. 6.3.2 for the Boltzmann scheme are set as: $\alpha = 1$ and $\beta = 0.80$. The stop criteria considered are a maximum number of generation $G = 1000$ or that the number of different individuals in the population falls below 5. The performance of MK-EDA for $k \in \{2, 3, 4\}$ are investigated in our experiments, and the algorithm performs very well when $k = 3$. In the AP clustering, Hamming distance is used as the similarity metric of individuals, other parameters are set same as the previous section experiments.

The results of the proposed method compared with the MK-EDA for the 2-D HP Model are listed in Table 6.7. It includes the best solution, the percentage of times the best solution has been found, the mean fitness of selected solutions in the last generation has been found and computing cost (CPU-T). The experiment results show that the proposed method has more chance to find global optimum or suboptimum solution for protein sequences than MK-EDA. The MK-EDA cannot find the global optimum of the deceptive sequences and long sequences $s7$, $s9$, $s10$ and $s11$.

The proposed method also can find the global optimum of the sequences $s7$, $s9$ and $s10$, and can find the second best solution for sequence $s11$. But the proposed method need more computing cost than the MK-EDA because of the clustering and niching operations. Figure 6.6 shows the number of different individuals in the selected population for one representative run of instance $s10$ in 2-D model. One 3-D conformation of the best results for sequences $s5$, $s7$ and $s8$ are shown in Fig. 6.7, Fig. 6.8 and Fig. 6.9 respectively.

Table 6.8: Results of different search methods for 2-D HP model.

	Proposed	MK-EDA	GA	NewACO	PERM
No.	$H(X)$	$H(X)$	$H(X)$	$H(X)$	$H(X)$
s4	-14	-14	-14	-14	-14
s5	-23	-23	-22	-23	-23
s6	-21	-21	-21	-21	-21
s7	-36	-35	-34	-36	-36
s8	-42	-42	-37	-42	-38
s9	-53	-52		-51	-53
s10	-48	-47		-47	-48
s11	-49	-48		-47	-50

Table 6.9: Results of different search methods for 3-D HP model.

	Proposed	MK-EDA	Hybrid GA	IA
No.	$H(X)$	$H(X)$	$H(X)$	$H(X)$
s4	-18	-18	-18	-18
s5	-29	-29	-28	-28
s6	-30	-29	-22	-23
s7	-49	-48	-48	-41
s8	-51	-50	-46	-42

The performance of the proposed method compared with the best results achieved with other evolutionary and Monte Carlo optimization algorithms is shown in Table 6.8 (2-D HP model) and Table 6.9 (3-D HP model). The results of other methods are cited from Ref. [75]. The experiment results show that none of the algorithms are able to outperform the rest of algorithms for all the instances. The PERM (Pruned-Enriched Rosenbluth Method) [49] is a biased chain growth algor -sampling (“population control”) with depth-first implementation. It is one of the best

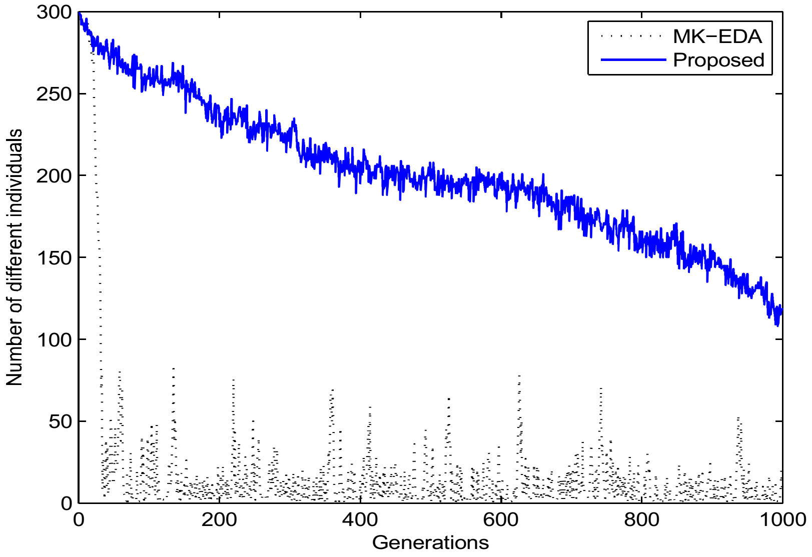


Figure 6.6: The number of different individuals in the selected population for one representative run of instance $s10$.

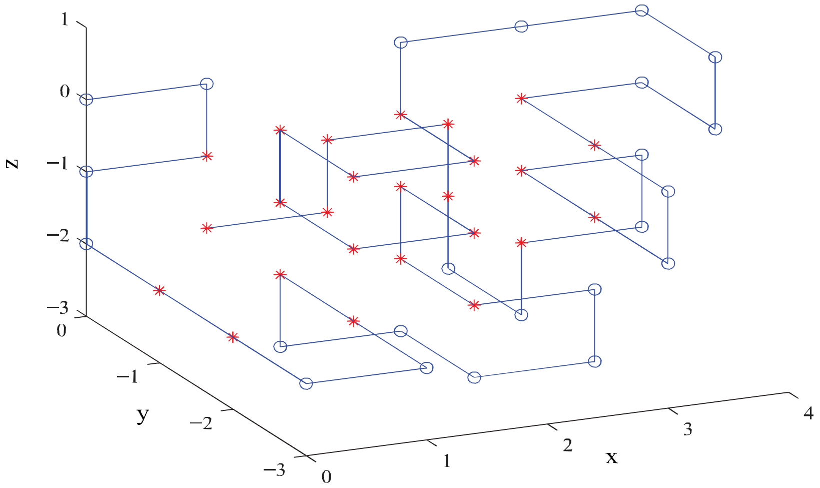


Figure 6.7: One conformation of the best results for the instance $s5$ in 3-D HP model (fitness=-29, star denotes H and circle denotes P).

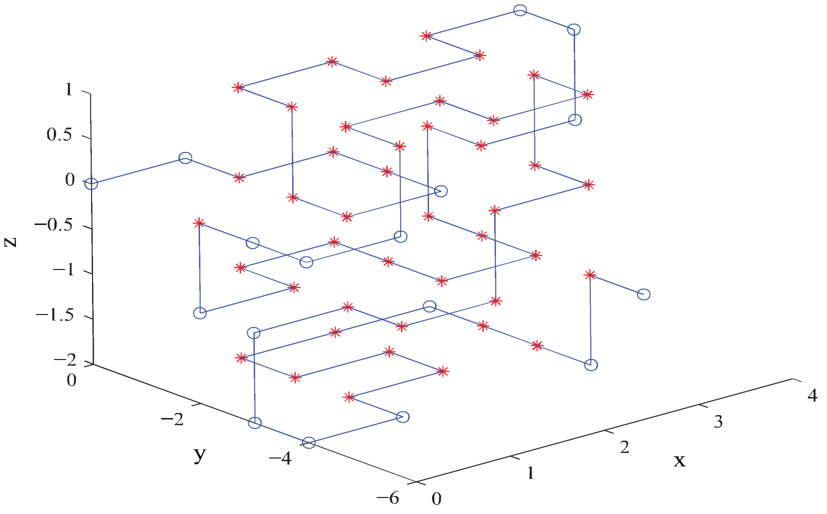


Figure 6.8: One conformation of the best results for the instance $s7$ in 3-D HP model (fitness=-49, star denotes H and circle denotes P).

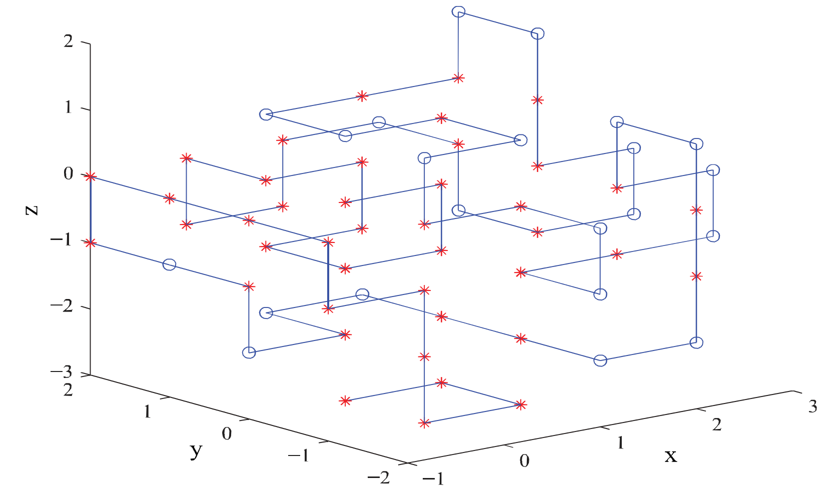


Figure 6.9: One conformation of the best results for the instance $s8$ in 3-D HP model (fitness=-51, star denotes H and circle denotes P).

contenders in all cases except $s8$ in which its result is very poor. This shows that our method is very competitive with other existing algorithms for the PSP on lattice HP models.

6.5 Conclusions

This chapter proposed a novel adaptive niching EDA based on AP clustering analysis. The clearing procedure is utilized as a basic niching frame in the proposed method. The AP clustering is not only used to partition the niches for population adaptively, but also used to mine the searching history information of EDA. A niche novelty metric is introduced to evaluate the degree of search space that has been explored by detecting the niche overlapping information of continuous generations. A Boltzmann scheme niche capacity selection is proposed to improve the EDA performance by a balance searching between exploration and exploitation. At the beginning phase of the search, the dominating novelty-proportionate selection based on the niche novelty metric can guide the EDA searching to cover the search space as much as possible. At the end of the search process, the dominating fitness-proportionate selection based on the best fitness value in a niche can enforce the EDA searching to exploit the already found promising areas.

Two different categories of optimization problems are used to evaluate the proposed adaptive niching EDA. Experiment results show that the proposed adaptive niching EDA is an efficient method for solving the optimization problems with irregular and complex multimodal landscapes.

Chapter 7

Conclusions

7.1 Summary

The gene function prediction and protein structure prediction are two important domains where machine learning techniques are applied in bioinformatics. Generally, the first is a classification problem and the second is an optimization problem. Supervised and unsupervised classification methods are often used to predict gene function, such as Clustering, SVM, NN, etc. Evolutionary algorithm (EA) based methods are the main optimization technologies for protein structure prediction, such as genetic algorithm (GA), estimation distribution algorithms (EDAs), particle swarm optimization (PSO), etc.

For the complicated tasks, such as the hierarchical multi-label gene function classification and the long protein sequence structure prediction, applying machine learning methods simply usually cannot obtain expectable results. The different characteristics of the applications will require novel modifications of existing machine learning methods.

In this thesis, improved machine learning methods are explored for solving complicated genomics and proteomic applications. More precisely, SVM based multi-label and hierarchical multi-label classification methods are developed for solving the gene function prediction problems, and novel EDAs based methods are proposed for solving the protein HP model problems. The information of the biology data is extracted and used to improve the performances of machine learning methods, and some delicate techniques are also combined according to the characteristics of application problem.

The following are the main conclusions of this thesis:

1. A composite kernel based SVM (ck-SVM) model can be used for solving nonlinear classification tasks in gene function prediction, which conventional nonlinear kernel SVMs can not

work efficiently.

2. A multi-label classification based on label ranking and delicate decision boundary SVM can be used for solving multi-label gene function classification.

3. A hierarchical multi-label classification (HMC) method based on over-sampling and hierarchy constraint can be used for solving the FunCat gene function prediction problem.

4. A hybrid Estimation of EDA can be used for solving the protein structure problem on HP model.

5. An adaptive niching EDA with balance searching based on clustering analysis can be used to enhance the performance of EDA.

7.2 Topics for Future Research

Although a lot of progress has been made, there are still many aspects that need further investigations.

1. More characteristics and properties of the proposed composite kernel based SVM are need be investigated, such as the efficient subset partition methods and composite kernel estimation techniques.

2. Applications of the proposed hierarchical multi-label method in other domains should be studied, such as in text classification and object recognition.

3. Applications of the proposed adaptive niching EDA in other protein structure prediction problems should be researched, such as in protein side chain placement problem and tag single nucleotide polymorphism (SNP) selection.

Publication List

1. Benhui Chen and Jinglu Hu, “An Adaptive Niching EDA with Balance Searching Based on Clustering Analysis”, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E93-A, No.10, pp.1792-1799, Oct. 2010.

2. Benhui Chen, Weifeng Gu and Jinglu Hu, “An Improved Multi-label Classification Method and Its Application to Functional Genomics”, International Journal of Computational Biology and Drug Design (IJCBDD), Vol.3 No.2, pp.133-145, Sep. 2010.

3. Benhui Chen, Jinglu Hu. “A Hybrid EDA for Protein Folding Based on HP Model”, IEEJ Transactions on Electrical and Electronics Engineering (TEEE), Vol.5 No.4, pp.459-466, July 2010.

4. Benhui Chen, Liangpeng Ma and Jinglu Hu. “An Improved Multi-label Classification Method Based on SVM with Delicate Decision Boundary”, International Journal of Innovative Computing, Information and Control (IJICIC). Vol.6 No.4, pp.1605-1614, April 2010.

5. Benhui Chen, Jinglu Hu, Lihua Duan and Yinglong Gu. “Network Administrator Assistance System Based on Fuzzy C-means Analysis”, Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII), Vol.13 No.2, pp. 91-96, Mar. 2009.

6. Benhui Chen, Feiran Sun and Jinglu Hu, “Local Linear Multi-SVM Method for Gene Function Classification”, in *Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2010)*, pp.183-188, Kitakyushu, Japan, Dec. 2010.

7. Benhui Chen and Jinglu Hu, “Hierarchical Multi-label Classification Incorporating Prior Information for Gene Function Prediction”, in *Proc. of the 10th International Conference on Intelligent Systems Design and Applications (ISDA 2010)*, pp.231-236, Cairo, Egypt, Nov. 2010.

8. Benhui Chen and Jinglu Hu, “An Adaptive Niching EDA Based on Clustering Analysis”, in *Proc. of IEEE World Congress on Computational Intelligence 2010 (WCCI 2010) – Congress on Evolutionary Computation (CEC 2010)* , pp. 858-864, Barcelona, Spain, July 2010.

9. Benhui Chen, Weifeng Gu and Jinglu Hu, “An Improved Multi-label Classification Based on Label Ranking and Delicate Boundary SVM”, in *Proc. of IEEE World Congress on Computational Intelligence 2010 (WCCI 2010) – International Joint Conference on Neural Networks (IJCNN 2010)* , pp. 786-791, Barcelona, Spain, July 2010.

10. Benhui Chen and Jinglu Hu, “A Novel Clustering Based Niching EDA for Protein Folding”, in *Proc. of World Congress on Nature and Biologically Inspired Computing (NaBIC 2009)*, pp. 748-753, Coimbatore, India, Dec. 2009.

11. Benhui Chen, Long Li and Jinglu Hu, “An Improved Backtracking Method for EDAs Based Protein Folding”, in *Proc. of ICROS-SICE International Joint Conference 2009*, pp.4669-4673, Fukuoka, Japan, Aug. 2009.

12. Benhui Chen, Long Li and Jinglu Hu, “A Novel EDAs Based Method for HP Model Protein Folding”, in *Proc. of 2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pp. 309-315, Trondheim, Norway, May 2009.

13. Benhui Chen, Liangpeng Ma and Jinglu Hu, “A New SVM Based Method for Solving Multi-label Classification Problem”, in *Proc. of the 3rd International Symposium on Computational Intelligence and Industrial Applications (ISCIIA 2008)*, pp. 325-334, Dali, China, Nov. 2008.

14. Weifeng Gu, Bebhui Chen and Jinglu Hu, “Combining Binary-SVM and Pairwise Label Constraints for Multi-label Classification”, in Proc of 2010 IEEE International Conference on Systems, Man and Cybernetics (SMC 2010) (Istanbul), Oct. 2010 (to appear).

15. Jinglu Hu and Benhui Chen, “A New Method for Identifying Nonlinear Polynomial Model Using Genetic Algorithm”, in *Proc. of the 3rd International Symposium on Computational Intelligence and Industrial Applications (ISCIIA 2008)*, pp. 75-84, Dali, China, Nov. 2008. (Best Paper Award)

Bibliography

[1] Akbani, R., Kwek, S., Japkowicz, N.: Applying support vector machines to imbalanced datasets. In: Proc. of the 15th European Conference on Machine Learning (ECML)(Pisa, Italy), pp. 39–50 (2004)

[2] Amor, H.B., Rettinger, A.: Intelligent exploration for genetic algorithms: using self-organizing maps in evolutionary computation. In: Proc. of the Conference on Genetic and Evolutionary Computation (GECCO'05)(Washington, USA), pp. 1531–1538 (2005)

[3] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J., Davis, A., Dolinski, K., ss. Dwight: Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature genetics **25(1)**, 25–29 (2000)

[4] Baldi, P., Brunak, S.: Bioinformatics: The Machine Learning Approach, second ed. MIT Press, Cambridge, MA (2001)

[5] Barutcuoglu, Z., DeCoro, C.: Hierarchical shape classification using bayesian aggregation. In: Proc. of the IEEE International Conference on Shape Modeling and Applications (Matsushima, Japan), pp. 44–50 (2006)

[6] Barutcuoglu, Z., Schapire, R.E., Troyanskaya, O.G.: Hierarchical multi-label prediction of gene function. Bioinformatics **22(7)**, 830–836 (2006)

[7] Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. Sigkdd Explorations **6(1)**, 20–29 (2004)

[8] Berger, B., Leight, T.: Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. Journal of Computational Biology **5(1)**, 27–40 (1998)

[9] Blazewicz, J., Lukasiaka, P., Milostan, M.: Application of tabu search strategy for finding low energy structure of protein. Artificial Intelligence in Medicine **35(1-2)**, 135C145 (2005)

[10] Bockhorst, J., Craven, M., Page, D., Shavlik, J., Glasner, J.: A bayesian network approach to operon prediction. Bioinformatics **19(10)**, 1227C1235 (2003)

[11] Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition **37(9)**, 1757–1771 (2004)

[12] Camps-Valls, G., Gomez-Chova, L., Munoz-Mari, J., Vila-Frances, J., Calpe-Maravilla, J.: Composite kernels for hyperspectral image classification. IEEE Geoscience and Remote Sensing Letters **3(1)**, 93–97 (2006)

[13] Carter, R.J., Dubchak, I., Holbrook, S.R.: A computational approach to identify genes for functional rnas in genomic sequences. Nucleic Acids Research **29(19)**, 3928C3938 (2001)

[14] Cesa-Bianchi, N., Gentile, C., Zaniboni, L.: Incremental algorithms for hierarchical classification. Journal of Machine Learning Research **7**, 31–54 (2006)

[15] Chang, B., Tsai, H.: Training support vector regression by quantum-neuron-based hopfield neural net with nested local adiabatic evolution. International Journal of Innovative Computing, Information and Control **5(4)**, 1013–1026 (2009)

[16] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research **16**, 321–357 (2002)

[17] Chawla, N.V., Japkowicz, N.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations **6(1)**, 1–6 (2004)

[18] Chen, B., Li, L., Hu, J.: A novel EDAs based method for HP model protein folding. In: Proc. of the IEEE Congress on Evolutionary Computation (CEC'09) (Trondheim, Norway), pp. 309–315 (2009)

[19] Chen, J., Xin, B., Peng, Z., Dou, L., Zhang, J.: Optimal contraction theorem for explorationc-exploitation tradeoff in search and optimization. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans **39(3)**, 680–691 (2009)

[20] Chen, R., Chen, S.: Intrusion detection using a hybrid support vector machine based on entropy and TF-IDF. International Journal of Innovative Computing, Information and Control (IJICIC) **4(2)**, 413–424 (2008)

[21] Cheng, H., Tan, P., Jin, R.: Efficient algorithm for localized support vector machine. IEEE Transactions on Knowledge and Data Engineering **22(4)**, 537–549 (2010)

[22] Cheng, J., Tegge, A.N., Baldi, P.: Machine learning methods for protein structure prediction. IEEE Reviews in Biomedical Engineering **1**, 41–49 (2008)

[23] Cioppa, D., Stefano, D., Marcelli, A.: On the role of population size and niche radius in fitness sharing. IEEE Transactions on Evolutionary Computation **8(6)**, 580–592 (2004)

[24] Cioppa, D., Stefano, D., Marcelli, A.: Where are the niches? dynamic fitness sharing. IEEE Transactions on Evolutionary Computation **11(4)**, 453–465 (2007)

[25] Clare, A.: Machine learning and data mining for yeast functional genomics. PhD dissertation, Dept. of Computer Science, Univ. of Wales Aberystwyth (2003)

[26] Cotta, C.: Protein structure prediction using evolutionary algorithms hybridized with backtracking. In: J. Mira (ed.) Evolutionary Computation in Bioinformatics, pp. 321–328. Springer Verlag Berlin (2003)

[27] Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., Yannakakis, M.: On the complexity of protein folding. Journal of Computational Biology **5(3)**, 423–466 (1998)

[28] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)

[29] Darwen, P., Yao, X.: Every niching method has its niche: Fitness sharing and implicit sharing compared. In: H. Voigt, W. Ebeling, I. Rechenberg, H. Schwefel (eds.) Lecture Notes in Computer Science, pp. 398–407. Springer-Verlag, Berlin, Germany (1996)

[30] Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry **24(8)**, 1501–1509 (1985)

[31] Dill, K.A.: Dominant forces in protein folding biochemistry. Biochemistry **29(31)**, 7133–7155 (1990)

[32] Diplaris, S., Tsoumakas, G., Mitkas, P., Vlahavas, I.: Protein classification with multiple algorithms. In: Proc. of the 10th Panhellenic Conference on Informatics (PCI'05)(Volos, Greece), pp. 448–456 (2005)

[33] Dong, W., Yao, X.: NichingEDA: utilizing the diversity inside a population of EDAs for continuous optimization. In: Proc. of the IEEE Congress on Evolutionary Computation (CEC'08) (Hongkong, China), pp. 1260–1267 (2008)

[34] Elisseeff, A., Weston, J.: A kernel method for multi-labelled classification. In: S.B. T. G. Dietterich, Z. Ghahramani (eds.) Advances in Neural Information Processing Systems, pp. 681–687. MIT Press, USA (2001)

[35] Emmendorfer, L., Pozo, A.: A clustering-based approach for linkage learning applied to multimodal optimization. In: Y. Chen (ed.) Linkage in Evolutionary Computation, pp. 225–248. Springer Verlag Berlin (2008)

[36] Fan, R., Lin, C.: A study on threshold selection for multi-label classification. Technical Report of National Taiwan University, Taipei, Taiwan pp. 1–23 (2008)

[37] Flores, S., Smith, J.: Study of fitness landscapes for the HP model of protein structure prediction. In: Proc. 2003 the IEEE Congress on Evolutionary Computation (CEC'03) (Canberra, Australia), pp. 2338–2345 (2003)

[38] Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science **315**, 972–976 (2007)

[39] Fu, Z., Robles-K .: On mixtures of linear SVMs for nonlinear classification. Lecture Notes in Computer Science **5342**, 489–499 (2010)

[40] Goldberg, D.E., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: J.J. Grefenstette (ed.) Genetic Algorithms and Their Applications, pp. 41–49. L. Erlbaum Associates, USA (1987)

[41] Greenwood, G.W., Shin, J.M.: On the evolutionary search for solutions to the protein folding problem. In: G.B. Fogel (ed.) Artificial Neural Nets Problem Methods, pp. 115–136. Elsevier Science and Technology Books (2002)

[42] Guan, Y., Myers, C., Hess, D., Barutcuoglu, Z., Caudy, A., Troyanskaya, O.: Predicting gene function in a hierarchical context with an ensemble of classifiers. Genome Biology **9(s3)** (2008)

[43] Guo, H., Viktor, H.: Learning from imbalanced data sets with boosting and data generation: The databoost-im approach. Genome Biology **6**, 40–49 (2004)

[44] Han, H., Wang, W., Mao, B.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: Proc. of the International Conference on Intelligent Computing (Hefei, China), pp. 878–887 (2005)

[45] Haykin, S.: Neural Networks: A Comprehensive Foundation (Second Edition). Prentice Hall (1998)

[46] Hoque, T., Chetty, M., Dooley, L.S.: A guided genetic algorithm for protein folding prediction using 3D hydrophobic-hydrophilic model. In: Proc. of the IEEE Congress on Evolutionary Computation (CEC'06) (Vancouver, Canada), pp. 2339–2346 (2006)

[47] Horn, J., Goldberg, D.E., Deb, K.: Implicit niching in a learning classifier system: Nature's way. Evolutionary Computation **2(1)**, 37–66 (1994)

[48] Hsu, H., Mehra, V., Grassberger, P.: Structure optimization in an off-lattice protein model. Physical Review E **68(2)**, 1–4 (2003)

[49] Hsu, H., Mehra, V., Nadler, W., Grassberger, P.: Growth algorithms for lattice heteropolymers at low temperatures. Journal of chemical physics **118(1)**, 444–451 (2003)

[50] Jiang, T., Wang, S., Wei, R.: Support vector machine with composite kernels for time series prediction. In: D.L. et.al. (ed.) Lecture Notes in Computer Science: Advances in Neural Networks (LNCS 4493), pp. 350–356. Morgan Kaufmann (2007)

[51] Kazawa, H., Izumitant, T., Taira, H., Maeda, E.: Maximal margin labeling for multi-topic text categorization. In: Proc. of the Advances in Neural Information Processing Systems (Canada), pp. 647–656 (2004)

[52] Krasnogor, N., Hart, W.E., Smith, J., Pelta, D.A.: Protein structure prediction with evolutionary algorithms. In: Proc. of the Genetic Evolutionary Computation Conference (Orlando, USA), pp. 1596–1601 (1999)

[53] Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proc. of the 14th International Conference on Machine Learning (ICML)(Nashville, USA), pp. 179–186 (1997)

[54] Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Armananzas, J.A.L.R., Santafe, G., Perez, A., Robles, V.: Machine learning in bioinformatics. Briefings in Bioinformatics **7(1)**, 86–112 (2006)

[55] Larranaga, P., Lozano, J.A.: Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation. Kluwer Academic (2002)

[56] Lau, K.F., Dill, K.A.: A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Marcromolecules **22(10)**, 3986–3997 (1989)

[57] Lauer, F., Bloch, G.: Incorporating prior knowledge in support vector machines for classification: A review. Neurocomputing **71**, 1578C–1594 (2008)

[58] Lewis, D., Yang, Y., Rose, T.G., Dietterich, G., Li, F.: RCV1: A new benchmark collection for text categorization research. Journal of Machine Learning Research **5**, 361–397 (2004)

[59] Lozano, J.A., Larranaga, P., Inza, I., Bengoetxea, E.: Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms. Springer Verlag Berlin (2006)

[60] Lu, Q., Yao, X.: Clustering and learning Gaussian distribution for continuous optimization. IEEE Transactions on Systems, Man, and Cybernetics **35(2)**, 195–204 (2005)

[61] March, J.G.: Exploration and exploitation in organizational learning. Organization Science **2(1)**, 71–87 (1991)

[62] Mathe, C., Sagot, M.F., Schiex, T., Rouze, P.: Current methods of gene prediction, their strengths and weaknesses. Nucleic Acids Research **30(19)**, 4103C–4117 (2002)

[63] McCallum, A.: Multi-label text classification with a mixture model trained by EM. In: Proc. of the Working Notes Am. Assoc. Artificial Intelligence Workshop Text Learning (AAAI'99)(Florida, USA), pp. 1–7 (1999)

[64] Mewes, H.W., Hani, J., Pfeiffer, F., Frishman, D.: Mips: a database for protein sequences and complete genomes. Nucleic Acids Research **26(1)**, 33–37 (1998)

[65] Narayanan, A., Keedwell, E.C., Olsson, B.: Artificial intelligence techniques for bioinformatics. Applied Bioinformatics **1(4)**, 191–222 (2002)

[66] Obozinski, G., Lanckriet, G., Grant, C., Jordan, M., Noble, W.S.: Consistent probabilistic outputs for protein function prediction. Genome Biology **9(s6)**, 1–8 (2008)

[67] Petrovskiy, M.: Paired comparisons method for solving multi-label learning problem. In: Proc. of the Sixth International Conference on Hybrid Intelligent Systems (HIS'06) (Rio de Janeiro, Brazil), pp. 42–48 (2006)

[68] Petrowski, A.: A clearing procedure as a niching method for genetic algorithms. In: Proc. of the IEEE International Conference on Evolutionary Computation (Nagoya, Japan), pp. 798–803 (1996)

[69] Platt, J.C.: Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In: Advances in Large Margin Classifiers, pp. 61–74. The MIT Press, Cambridge, Massachusetts London, England (1999)

[70] Punera, K., Rajan, S.: Improved multi label classification in hierarchical taxonomies. In: Proc. of the IEEE International Conference on Data Mining Workshops (ICDMW'09)(Miami, USA), pp. 388–393 (2009)

[71] Rojo-Alvarez, J., Martinez-Ramon, M., Prado-Cumplido, M., Artes-Rodriguez, A., Figueiras-Vidal, A.: Support vector method for robust ARMA system identification. IEEE Transactions on Signal Processing **52(1)**, 155–164 (2004)

[72] Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Learning hierarchical multi-category text classification models. In: Proc. of the the 22nd international conference on Machine learning (Bonn, Germany), pp. 744–751 (2005)

[73] Rousu, J., Saunders, C., Szedmak, S., Shawe-Taylor, J.: Kernel-based learning of hierarchical multilabel classification models. The Journal of Machine Learning Research **7**, 1601–1626 (2006)

[74] Ruepp, A., Zollner, A., Maier, D., Albermann, K., Hani, J., Mokrejs, M., Tetko, I., Guldener, U., Mannhaupt, G., Munsterkotter, M., Mewes, H.: The FunCat, a functional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acids Research **32**, 5539–5545 (2004)

[75] Santana, R., Larranaga, P., Lozano, J.A.: Protein folding in simplified models with estimation of distribution algorithms. IEEE Transactions on Evolutionary Computation **12(4)**, 418–438 (2008)

[76] Sareni, B., Krahenbuhl, L.: Fitness sharing and niching methods revisited. IEEE Transactions on Evolutionary Computation **2(3)**, 97–106 (1998)

[77] Schapire, R.E., Singer, Y.: Boostexter: A boosting-based system for text categorization. Machine Learning **39(2-3)**, 135–168 (2000)

[78] Schietgat, L., Vens, C., Struyf, J., Blockeel, H., Kocev, D., Dzeroski, S.: Predicting gene function using hierarchical multi-label decision tree ensembles. BMC Bioinformatics **11(2)**, 1–14 (2010)

[79] She, Q., Su, H., Dong, L., Chu, J.: Support vector machine with adaptive parameters in image coding. International Journal of Innovative Computing, Information and Control **4(2)**, 359–367 (2008)

[80] Shu, P., Xu, J.: A multi-label classification algorithm based on triple class support vector machine. In: Proc. of the International Conference on In Wavelet Analysis and Pattern Recognition (ICWAPR'07) (Beijing, China), pp. 1447–1452 (2007)

[81] Smith, R.E., Forrest, S., Perelson, A.S.: Searching for diverse, cooperative populations with genetic algorithms. Evolutionary Computation **1(2)**, 127–149 (1993)

[82] Song, J., Cheng, J., Zheng, T., Mao, J.: A novel genetic algorithm for HP model protein folding. In: Proc. of the 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2005) (Dalian, China), pp. 935–937 (2005)

[83] Stenger, B., Thayananthan, A., Torr, P., Cipolla, R.: Estimating 3D hand pose using hierarchical multi-label classification. Image and Vision Computing **5(12)**, 1885–1894 (2007)

[84] Streichert, F., Stein, G., Ulmer, H., Zell, A.: A clustering based niching ea for multimodal search spaces. In: P.L. et al. (ed.) Lecture Notes in Computer Science (LNCS 2936), pp. 293–305. Springer-Verlag Berlin Heidelberg, Germany (2004)

[85] Suykens, J., Gestel, T., Brabanter, J., Moor, B., J.Vandewalle: Least Squares Support Vector Machines. World Scientific, Singapore (2002)

[86] Szafranski, M., Grandvalet, Y., Rakotomamonjy, A.: Composite kernel learning. Machine Learning **79(1-2)**, 73–103 (2010)

[87] Tao, Q., Wu, G., Wang, F., Wang, J.: Posterior probability support vector machines for unbalanced data. IEEE Transactions on Neural Networks **16(6)**, 1561–1573 (2005)

[88] Tsoumakas, G., Katakis, I.: Multi label classification: An overview. International Journal of Data Warehousing and Mining **3(3)**, 1–13 (2007)

[89] Unger, R., Moult, J.: Genetic algorithms for protein folding simulations. Journal of Molecular Biology **231(1)**, 75–81 (1993)

[90] Valentini, G.: True path rule hierarchical ensembles for genome-wide gene function prediction. IEEE ACM Transactions on Computational Biology and Bioinformatics (2010)

[91] Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Verlag Berlin (1999)

[92] Vens, C., Struyf, J., Schietgat, L., Dzeroski, S., Blockeel, H.: Decision trees for hierarchical multi-label classification. Machine Learning **73(2)**, 185–214 (2008)

[93] Wang, L., Chang, M., Feng, J.: Parallel and sequential support vector machines for multi-label classification. International Journal of Information Technology **11(9)**, 11–18 (2005)

[94] Wu, H., Hsu, C., Lee, T., Fang, F.: Improved SVM and ANN in incipient fault diagnosis of power transformers using clonal selection algorithms. International Journal of Innovative Computing, Information and Control **5(7)**, 1959–1974 (2009)

[95] Yang, Y.: An evaluation of statistical approaches to text categorization. Journal of Information Retrieval **1**, 67–88 (1997)

[96] Yang, Y.: A study of thresholding strategies for text categorization. In: Proc. of the 24th annual international ACM SIGIR conference on Research and development in information retrieval (New Orleans, USA), pp. 137–145 (2001)

[97] Yao, X., Liu, Y., Lin, G.: Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation **3(2)**, 82–102 (1999)

[98] Zhang, M., Zhou, Z.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Transaction on Knowledge and Data Engineering **18(10)**, 1338–1351 (2006)

[99] Zhang, Q., Sun, J., Tsang, E.: An evolutionary algorithm with guided mutation for the maximum clique problem. IEEE Transaction on Evolutionary Computation **9(2)**, 192–200 (2005)

[100] Zhang, Y., Rajapakse, J.C.: Machine Learning in Bioinformatics. John Wiley and Sons, Chichester and New York (2008)