A Gaussian Mixture-Based Approach to Synthesizing Nonlinear Feature Functions for Automated Object Detection

Pei Fang Guo

A Thesis

In

The Department

Of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montréal, Québec, Canada

August 2010

Your file Votre référence
ISBN: 978-0-494-67351-5
Our file Notre référence
ISBN: 978-0-494-67351-5

ABSTRACT

A Gaussian Mixture-Based Approach to Synthesizing Nonlinear Feature Functions for Automated Object Detection

Pei Fang Guo

Feature design is an important part to identify objects of interest into a known number of categories or classes in object detection. Based on the depth-first search for higher order feature functions, the technique of automated feature synthesis is generally considered to be a process of creating more effective features from raw feature data during the run of the algorithms. This dynamic synthesis of nonlinear feature functions is a challenging problem in object detection. This thesis presents a combinatorial approach of genetic programming and the expectation maximization algorithm (GP-EM) to synthesize nonlinear feature functions automatically in order to solve the given tasks of object detection. The EM algorithm investigates the use of Gaussian mixture which is able to model the behaviour of the training samples during an optimal GP search strategy. Based on the Gaussian probability assumption, the GP-EM method is capable of performing simultaneously dynamic feature synthesis and model-based generalization. The EM part of the approach leads to the application of the maximum likelihood (ML) operation that provides protection against inter-cluster data separation and thus exhibits improved convergence. Additionally, with the GP-EM method, an innovative technique, called the histogram region of interest by thresholds (HROIBT), is introduced for diagnosing protein conformation defects (PCD) from microscopic imagery. The experimental results show that the proposed approach improves the detection accuracy and efficiency of pattern object discovery, as compared to single GP-based feature synthesis methods and also a number of other object detection systems. The GP-EM method projects the hyperspace of the raw data onto lower-dimensional spaces efficiently, resulting in faster computational classification processes.

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude and respect to my co-supervisor, Dr. Prabir Bhattacharya, for his time, help and guidance in the research and writing process with invaluable comments and advice. It has been a wonderful academic experience under Dr. Bhattacharya's supervision after I met him over two years ago. I also want to thank my co-supervisor, Dr. Nawwaf Kharma, for all his help during my graduate studies at Concordia University.

I would like to thank the thesis committee members (Dr. Omair Ahmad, Dr. Peter Grogono, Dr. Wei-ping Zhu and the external examiner Dr. Otman Basir from the University of Waterloo) for their many useful comments and helpful advice about my thesis. Special thanks also go to Dr. Peter Grogono for his suggestions regarding LISP and LISP-like expressions.

I would like to thank the members of the Student Learning Service at Concordia University for their efforts. They would never grow tired of answering my endless stream of questions regarding academic writing.

Finally, thanks to my family for their patience and understanding. They were always there for me throughout the period of my studies.

This research is supported by grants from the Canada Research Chair Foundation and NSERC.

CONTENTS

List of Figures ix

List of Tables xi

List of Abbreviations xii

1 Introduction & Literature Review **1**

1.1 Literature Review 3

1.1.1 Feature Synthesis Applications 3

1.1.2 Genetic Programming (GP) Applications 6

1.1.3 Expectation Maximization (EM) Applications 8

1.2 Objectives 10

1.3 Organization of the Thesis 10

2 Background Information **12**

2.1 Object Features 12

2.1.1 Feature Vectors and Spaces 12

2.1.2 Concepts of Feature Extraction and Selection, in General 13

2.1.3 Target of Feature Synthesis 14

2.2 Classification 14

2.2.1 Definitions of Object Classification, or Detection 15

2.2.1.1 Supervised and Unsupervised Object Classification 16

2.2.2 Minimum Distance Classifier (MDC) 16

2.2.3 K Nearest Neighbors (KNN) Classifier 17

2.3 Genetic Programming (GP) 18

2.3.1 Evolutionary GP Process 18

2.3.2 Generative Graphic Representations 18

2.3.3 Genetic Operations 19

2.3.4 Fitness Measure 20
2.4 Expectation Maximization (EM) Algorithm 21
2.4.1 Estimation of the Means of the k Gaussians 21
2.4.2 Statement of the EM Algorithm 22

3 The Hybrid GP-EM Approach 23
3.1 Framework Overview 23
3.2 GP-EM Tree Representations 25
3.2.1 Function and Terminal Sets 25
3.2.2 An Example of a GP-EM Tree Representation 26
3.3 The k-means Problem 27
3.4 Fitness Function, the J Value 30
3.5 Pseudo-Code for the GP-EM Implementation 33
3.6 Validation 32
3.6.1 N - fold Cross-Validation Approach 32
3.6.2 Receiver Operating Characteristic (ROC) 32
3.7 Implementation 33

4 Applications 34
4.1 Detecting Protein Conformation Defects from Microscopic Imagery
4.1.1 Introduction 35
4.1.2 The Dataset of CellsDB 35
4.1.3 HROIBT-Based Texture Data Preparation 35
4.1.3.1 Designs of HROIBT 35
4.1.3.2 Basic Texture Feature Extraction from HROIBT 35
4.1.4 Experimental Results 35
4.1.4.1 Convergence Results 35
4.1.4.2 Detection Results 35
4.1.5 Further Comparisons 35
4.2 Detecting Breast Cancer Disease 35

4.2.1 Introduction 35
4.2.2 The Dataset of WDBC..... 35
4.2.3 Experimental Results 35
4.2.3.1 Convergence Results 35
4.2.3.2 Feature Synthesized Results..... 35
4.2.3.3 Training Data Representation Results..... 35
4.2.3.4 Detection Results 35
4.2.4 Further Comparisons 35
4.2.4.1 Comparisons with the Single GP-Based Feature Generators. 35
4.2.4.2 Comparisons with Other Detection Systems..... 35
4.3 Supporting Identification in a Hand-Based Biometric System 34
4.3.1 Introduction 35
4.3.2 Previous Feature Selection Work on the Hand Image Dataset.... 35
4.3.3 The GP-EM-MSE Supporting Biometric Identification System
4.3.3.1 Mean Square Error (MSE) Fitness..... 35
4.3.4 Experimental Results 35
4.3.4.1 Convergence Results 35
4.3.4.2 Detection Results 35
4.3.5 Further Comparisons 35
4.4 Detecting Parkinson's Disease 34
4.4.1 Introduction..... 35
4.4.2 The Dataset of OPDD 35
4.4.3 Experimental Results 35
4.4.3.1 Convergence Results 35
4.4.3.2 Detection Results 35
4.4.4 Further Comparisons 35

5 Conclusions 60

5.1 Discussion and Conclusions 60
5.2 Summary of Main Contributions..... 61
5.3 Other Possible Applications 63

5.3.1 Power Quality Monitoring System 63
5.3.1.1 Primitive Feature Extraction from Vibration Signals 64
5.4 Refereed Publications Based on the Research 65
5.5 Final Remarks and Future Work..... 65

Bibliography **67**

List of Figures

Figure 2.1 A 3-D hyperspace for a function $z = xy\,(x^3 - y^2) / (x^2 + y^2)$..... 14

Figure 2.2 Flowchart of evolutionary GP..... 19

Figure 2.3 A expression of $z\{x + sin(y)\}$ in a GP graphic representation 20

Figure 2.4 GP crossover..... 21

Figure 2.5 GP mutation..... 21

Figure 2.6 The samples, shown by the points along the *s*-axis, are mapped onto a mixture of three Gaussian distributions..... 23

Figure 3.1 The process of the GP-EM feature generator..... 25

Figure 3.2 An example of a GP-EM tree representation..... 29

Figure 4.1 Four cell images, healthy and sick samples, from the CellsDB..... 41

Figure 4.2 Procedures of the basic texture feature preparation..... 42

Figure 4.3 The upper threshold of the HROIBT, $\mathbf{T}_U = 200$, defined by the histogram measure for all image samples in the binary classes of the healthy and sick conditions of the OPMD..... 43

Figure 4.4 The lower threshold of the HROIBT, $\mathbf{T}_L = 50$, defined by $\mathbf{V}a$..... 43

Figure 4.5 Solution of the HROIBT in the bin range of (50, 200), with the grayscale histogram measure of the healthy and sick samples in Figure 7.2..... 43

Figure 4.6 Resulting **H**f-processed data representation after 350 iterations..... 45

Figure 4.7 ROC analysis based on the distance proportions for the synthesized feature **H**f (thick line), the 17 basic features (thin line), D1-D17, and the 11 basic features (dashed line), D7 - D17..... 47

Figure 4.8 Convergence results from one of 10 runs on WDBC..... 49

Figure 4.9 The feature synthesized results for a single run on WDBC..... 49

Figure 4.10 The GP-EM training data representation results from the example of the 10 runs on WDBC..... 50

Figure 4.11 Detection accuracy (DA) of the combined training sets and test sets against the index of 10 independent runs on WDBC..... 51

Figure 4.12 A sample of a hand image..... 55

Figure 4.13 The GP-EM-MSE hand-based biometric identification system..... 56

Figure 4.14 Convergence results for the features produced by GP-EM-MSE 58

Figure 4.15 The convergence results for the created feature function, $\mathbf{Pa}_{\_detection}$..... 61

Figure 5.1 Procedures for preparing primitive features for a power quality monitoring system..... 68

Figure 5.2 Phenomena for the power quality problem; the horizontal axis presents the time in second and the vertical axis presents the magnitude in its four power conditions..... 68

List of Tables

Table 1.1 Applications of feature synthesis..... 6

Table 1.2 Applications of GP..... 8

Table 1.3 Applications of the EM algorithm..... 10

Table 3.1 Function set 28

Table 3.2 The common GP parameter values..... 38

Table 4.1 Comparisons of the accuracy using the synthesized feature and a certain number of basic features with the classifier KNN..... 46

Table 4.2 The detection accuracy (DA) over 10 runs on WDBC..... 52

Table 4.3 Comparisons of the GP-EM approach and the single GP-Based feature generators on WDBC..... 53

Table 4.4 Comparisons of the proposed GP-EM Algorithm and other object detection systems on WDBC. The number of features is the mean of the features used in detection..... 54

Table 4.5 Identification accuracy (%) using two features, P_feature 1 and P_feature 2, produced by GP-EM-MSE with MDC..... 58

Table 4.6 The comparision between the features produced by GP-EM-MSE / MDC and 41 primitive features selected by [Gu03],[KS05] / KNN, in terms of the detection accuracy (%)..... 59

Table 4.7 Descriptions of the features of the dataset OPDD..... 60

Table 4.8 Detection accuracy (DA) for 10 runs on OPDD..... 62

Table 4.9 Comparisions of the GP-EM detector and other methods in terms of the detection accuracy (DA) on OPDD..... 62

List of Abbreviations

ANN	artificial neural network
CV	cross validation
CellsDB	a database of cell images regarding the disease of OPMD
EC	evolutionary computation
EM	expectation maximization
EP	evolutionary programming
GA	genetic algorithm
GP	genetic programming
HROIBT	histogram region of interest by thresholds
HMM	hidden Markov model
INIs	intranuclear inclusions
JPEG	joint photographic experts group
KNN	K nearest neighbors
LGP	linear genetic programming
LISP	LISt Processing
OPMD	oculopharyngeal muscular dystrophy
PCA	principal component analysis
PCD	protein conformation defects
MDC	minimum distance classifier
ML	maximum likelihood
NN	neural networks

SVM	support vector machine
SOM	self organizing map
RGB	red-green-blue color space
ROC	receiver operating characteristic
WWW	world wide web

Chapter 1

Introduction & Literature Review

In object detection techniques, the objects are generally represented by vectors of feature values. The concept of detection can be expressed in terms of the partitions of feature spaces (i.e., mappings from feature spaces to decision spaces). It is not known in advance which features will provide the best discrimination among the classes. A common approach is to let human experts provide as many features as possible that are readily measurable and likely to be related to the detection categories. However, the computational complexity of classification grows with each additional feature.

In defining features, feature selection is a breadth-first search for many simple features. The search begins with a large pool of original features and outputs a population of feature subsets in which the feature number and combination are adapted to seek decision boundaries [MN09], [MP06]. Obviously, the selection of feature subsets can improve classification speed and efficiency [KS05], [OL04].

Nowadays, object detection problems often involve large feature datasets. In high-dimensional problems, feature subsets are selected to establish complex relationships within large datasets where the mapping from data to class labels is often obscure or difficult to identify. The process of finding the appropriate feature subsets would be a time-consuming task, and considerable efforts have gone into automating the process of

the feature synthesis [GJ05], [RZ02], [Ko94].

Based on the depth-first search for higher order feature functions, the technique of dynamic feature synthesis is generally considered to be a process of creating more effective features from raw features. The purpose is to employ relatively fewer synthesized features to perform functional mapping for the data representation without a decline in the discriminative capability. Researchers have used various approaches for finding good algorithms for the feature synthesis, such as neural networks (NN) [WJ06], [GK96], fuzzy systems [YM01] and evolutionary algorithms [Lo07], [GN06], [GJ05], [KB05], [CT01], [YB06].

Evolutionary algorithms (EA) are randomized search and optimization techniques guided by the principles of evolution and natural genetics [LR05]. Evolutionary genetic programming (GP) [Ko94] is a variant of genetic algorithm (GA) [Go89] and evolutionary computation [FO66], in which the GP hypotheses being manipulated are computer programs, rather than bit strings [ES03].

To perform feature synthesis, a number of researchers have used the single EA-based methods, such as the GP-based method [Lo07], [GJ05], [KB05], [GN06], [YB06] and the GA-based method [CT01]. In these methods, the researchers choose specific fitness functions, operation selections, and other details. The disadvantages of the single EA-based feature generators are:

(1) they typically use only one type of processing components regardless of the problem domain;

(2) they require a large amount of computation time for convergence;

(3) they lack computer models to interpret the visual training data representations in the

computational evolutionary process.

In order to overcome these limitations, an alternative approach is to choose an integrated algorithm to constitute an efficient mechanism in the evolutionary computation. This thesis proposes to integrate GP with EM (we call it the "GP-EM") to synthesize nonlinear feature functions dynamically from primitive features, to achieve mapping from the hyperspace of raw data to lower dimensional spaces. During the runs of the GP-EM algorithm, the system is allowed to alter dynamically the structures of synthesized feature functions via GP tree representations, which are evolved by undergoing adaptive combination of primitive feature vectors.

1.1 Literature Review

1.1.1 Feature Synthesis Applications

Many researchers experimented with various problems for feature systhesis [GJ05], [Lo07], [YB06], [WJ06], [KB05], [HG02], [CT01]. The GP-based method for synthesizing new features from raw vibration data recorded from a rotating machine was proposed in [GJ05] to detect bearing faults in six different conditions. The experimental results showed that the classification performance changed drastically with the variation of classification methods and data, while the features synthesized by GP maintained a constantly high level of performance. Based on the Fisher criterion (a two-class based discriminant), the fitness function proposed in [GJ05] gave the synthesized features more chance to survive for closest classes, although a compensation was given in order to achieve the best overall performance.

The work based on GP in [Lo07] presented a methodology for recognizing epileptic patterns in human electroencephalogram (EEG) signals recorded from the scalp of real patients. Prior to the application of GP, the feature extraction was done as a preprocessing step. The computation time for the classifiers was reasonably fast for the binary classes of conditions, but more time was needed to produce the resulting synthesized features. A genetically inspired learning method for facial expression recognition (FER) in [YB06] was proposed to synthesize features automatically under a GP-based approach; the FER applied the Gabor wavelet representation for primitive features and linear/nonlinear operators for synthesizing new features.

A method of machine condition monitoring (MCM) using vibration signals was proposed in [WJ06] to produce new features based on higher-order statistics of the power spectral density. The MCM used a modified self organising map (SOM) via artificial neural network (ANN). Using real-world vibration data sets, the MCM achieved higher detection accuracy. A slightly lower accuracy in recognizing the normal class was also reported in [WJ06]. However, this could be improved when a more representative data set was used for training, or including a pre-training outlier removal process.

In [KB05], a feature synthesis algorithm was proposed to detect objects from the synthetic aperture radar (SAR) images without or with little human intervention,. In the approach, a variety of Linear Genetic Programming (LGP) procedures were encoded in fixed-length sequences of bytes and applied to both the images and scalar data. Since this was done once per recognition system, a more sophisticated and more time-consuming for training and/or querying classifier was needed for computation.

In [KO99], a feature systhesis method based on GP was proposed to improve detection accuracy using the K nearest neighbor classifier. The experiments were performed on both synthetic and real time datasets, such as the acoustic diagnosis of compressors. However, the proposed method had to select in advance the number of chromosomes and the penalty coefficient p for the fitness function by trials. A technique for electromyography (EMG) application was proposed in [HG02] to detect the intention of paraplegic persons when standing up or sitting down with an electrical stimulation orthosis. Using genetic algorithm (GA), EMG signals were then used to extract learning features for both standing and sitting positions. Since the selection of the number of samples was important in the detection process, there was a tradeoff to achieve a reasonable delay time and low probability of false alarms.

The feature function of symptom parameters synthesized using GA was proposed in [CT01] for diagnosing machinery faults of rolling bearing to discriminate two states-normal and abnormal conditions. Using statistical theory, a distinction index (DI) has been defined to evaluate the goodness of symptom parameters (SP) to form a new SP, called GA-SP, in order to ensure high accuracy in the fault diagnosis. The results demonstrated that DI can be used to evaluate the sensitivity of SP.

A summary of the reviewed work on the applications of feature synthesis is given in Table 1.1.

Table 1.1. Applications of feature synthesis.

papers	methods	applications	fitness functions	limitations
[GJ05]	GP	machinery fault detection	the Fisher criterion	fitness was biased on closest classes
[Lo07]	GP	human epileptic pattern recognition	the fitness_cases" and "hits"	time-consuming
[KB05]	LGP	object recognition for 3-D synthetic aperture radar (SAR) imagery	the classifiers SVM / C4.5	time-consuming / more sophisticated
[YB06]	GP	facial expression recognition (FER)	a voting strategy	classification accuracy was depended on primitive features chosen
[WJ06]	ANN	machine condition monitoring (MCM)	the modified Self Organising Map (SOM)	slightly lower accuracy in recognizing / needed pre-training outliers removal
[HG02]	GA	detection of the intention of a paraplegic person to stand up or to sit down	a neuro-fuzzy classifier	a tradeoff problem for the number of samples N chosed
[CT01]	GA	machinery fault detection	a distinction index (DI) of statistical theory	computational complexity

1.1.2 Genetic Programming (GP) Applications

Genetic programming (GP) extends genetic algorithms (GAs) to the evolution of complete computer programs. GP has been demonstrated to produce intriguing results in a number of applications [CP10], [PH09], [ZG07], [VL06], [JH06], [MF07], [KL07], [CW06]. A GP based approach, named the $(\mu+\lambda)$ GP, was introduced in [CP10] to solve the software quality classification problems. The $(\mu+\lambda)$ GP algorithm applied boosting techniques to improve the performance based on characteristics of the testing activity of GP. The results showed that the proposed method of $(\mu+\lambda)$ GP was an excellent, computationally less expensive technique to model software reliability.

The study in [PH09] presented a majority voting genetic programming classifier (MVGPC) to find the possible biomarkers of cancer diseases. MVGPC evolved multiple rules in different runs of GP, and then applied them one by one to a test sample. The

sample was assigned to the class by the majority voting. Conducting on four cancer data sets, including two multiclass data sets, the results showed that the proposed MVGPC method was suitable for detecting the labels of test samples.

In [ZG07], a more intelligent crossover operator based on GP was developed in the applications of various object detection problems. The classification accuracy of training sets was employed as a fitness function. The results showed that this approach outperformed the standard GP crossover operator in terms of detection accuracy.

[VL06] investigated the time series analysis of the demand and price of an electric power supply system by modeling them as the output of a low dimensional chaotic dynamical system. The histogram of the results predicted that the model, more or less like a Gaussian distribution with a peak close to 0, was correct when compared to other predicting systems. In [JH06], GP involving different control parameters and fitness functions was applied to a low-pass filter synthesis problem, using restricted component values to investigate how they affected the performance. In the case where the component values were restricted, the method demonstrated that GP can effectively find solutions by means of circuit topologies, but the computation time was two hours.

A new genetic representation, named analog genetic encoding (AGE), was proposed in [MF07] for the evolutionary synthesis of analog networks. AGE was designed to search for the optimal representation of the topologies and sizes of analog networks. The connection between the evolved network and pre-assigned external devices was obtained by associating the sequences of characters with the external devices that might be connected to the evolving networks. The results in [MF07] showed that AGE performed well when compared with other algorithms reported in the literature.

The study in [KL07] presented the GP-based decision tree model that facilitated a multi-objective optimization to solve software quality problems. In that approach, the set of Pareto-optima was employed to present a multi-objective classification model, and the experimental results were promising, and demonstrated the effectiveness of the proposed model. Applied to landmark detection problems, a method was described in [CW06] which involved GP and particle swarm optimization where a window of pixels was selected as an input, and then slided over all pixel positions in the region in order to find landmark locations. In addition, the work presented a quantitative comparison of search space size and landscape using the evolutionary GP and parameter optimization.

A summary of the reviewed work on the applications of GP is given in Table 1.2.

Table 1.2. Applications of GP.

Papers	applications	fitness functions	data sets
[CP10]	software reliability classification	the average error	failure datasets
[PH09]	prediction of cancer diseases	the majority voting	microarray datasets
[ZG07]	various image object detection	detection accuracy	images data sets (Shape, Coins and Texture)
[VL06]	time series of power pool demand and price	the prediction MSE	time series data
[JH06]	low-pass filter synthesis	estimation of error values	restricted component values
[MF07]	synthesis of analog electronic circuits	the desired output of voltage and the target output of current	the parameters number of the device set
[KL07]	software quality classification	modified expected cost of misclassification (MECM)	two large windows with the embedded system
[CW06]	object detection	detection accuracy	landmark images

1.1.3 Expectation Maximization (EM) Applications

Generally, the expectation maximization (EM) algorithm can be applied in many settings where we wish to estimate some parameters that describe an underlying probability distribution [Bi06]. An iterative sequence detector based on the expectation-maximization

(EM) algorithm was proposed in [CU10] for terrestrial optical wireless (OW) systems. The complexity of the proposed algorithm was considerably less than a direct evaluation of the log-likelihood function that was independent of the channel's fading statistics. The results demonstrate that the EM-based algorithm outperforms the symbol-by-symbol decoder and achieves a high detection rate.

In [HY09], the EM algorithm was employed to estimate parameters of models for applications to speech recognition. While estimating the hidden Markov model (HMM) parameters by taking the benefit of hybridization, a staged-fusion approach maintained the global sampling capabilities of evolutionary algorithm (EA), and met the population-diversity requirement for optimization of high dimensional objective functions. A hierarchical and spatially constrained mixture model was proposed in [NG07] for image segmentation. This model took into account spatial information by imposing distinct smoothness priors on the probabilities of each cluster and pixel neighborhoods. Experimental results showed that the approach improved significantly not only standard segmentation but also its spatially variant version. Moreover, the number of iterations of the EM algorithm was reduced significantly.

In [Lu06], the EM algorithm was used to model a set of observation data, which contained a finite number of components. Using penalized minimum matching distance, the mixture model fitting technique via the EM algorithm was proposed to find the number of mixture components. In [GÜ05], under the Gaussian probability assumption via the EM algorithm. the approach was capable of providing an effective training mechanism for the mixture of experts (ME), a modular neural network architecture. The learning process was

thus decoupled in a manner that the data were fitted well under the modular structure for diagnosing the disease.

The integration of the EM algorithm and rough set initialization was proposed to perform a non-convex clustering for real life data sets [MP03]. The number of clusters in the algorithm was specified by users. In the approach, EM provided the statistical model of the data to handle the associated uncertainties, in which the data was modeled under the Gaussian mixture in the local level, while the Gaussians were partitioned globally using graph-theoretic technique; thereby enabling the efficient detection of the non-convex clusters for the data. Using the EM algorithm, a feature selection method was proposed by [PN95] to estimate the unknown distributions for a product type of the data. The results showed that the proposed method was able to perform the feature selection when the class conditional distributions and functional forms of data were unknown, and further demonstrated that the approach with a multivariate normal model could be superior to other feature selection algorithms.

Up to now, to our knowledge, the EM algorithm has not yet been applied in the area of feature synthesized for automatic object recognition.

A summary of the reviewed works on the applications of EM is given in Table 1.3.

Table 1.3. Applications of the EM algorithm.

papers	applications	methodologies & algorithms
[CU10]	terrestrial optical wireless (OW) systems	the EM algorithm
[HY09]	speech recognition	integrating the EM and a constraint-based evolutionary algorithm
[NG07]	image segmentation	the EM algorithm
[Lu06]	evaluation of the number of mixture components	the EM algorithm
[GÜ05]	prediction of diseases in the supervised learning	integrating the EM and a modular neural network
[MP03]	non-convex clustering on real life data sets	integrating the EM and rough set theory
[PN95]	feature selection by estimating the density function	the EM with unknown mixture distributions

1.2 Objectives

In object detection systems, the general techniques of feature extraction and selection involve linear transformations from primitive feature vectors to obtain new vectors of lower dimensionality. At times, the newly extracted features might be linear combinations of some primitive features that are not able to provide a better classification accuracy. However, feature synthesis is a process that automatically synthesizes higher order functions based on the given primitive feature set. Clearly, more efficiency can be gained by automatically synthesizing non-linear feature functions while the algorithms are searching for the optimal solutions for the problems.

The aim of this thesis is to investigate a method for using GP and the EM algorithm for the task of feature synthesis. The proposed method should allow the synthesis of feature functions with the minimal user interaction. The strategies for integrating GP and EM in this thesis are: (1) GP tree structures are conducive to a global search of non-linear forms of primitive feature vectors in an optimal control environment; (2) the EM algorithm consists of hypothesis and modeling; using EM, the learning task then changes to output a hypothesis that estimates the means of each of the k Gaussians, thus making it easy to find solutions.

The main objectives of the thesis are:

- to increase the contribution that the overall performance of the detection would make toward a higher accuracy rate by the improved feature representations.
- to reduce the feature dimensionality of the transformed data, while maximizing classification accuracy and efficiency.

- to compare the proposed approach with single GP-based feature synthesis methods reported in the literature in terms of detection accuracy for the same problems.
- to demonstrate the potential benefits of the proposed approach being applicable to a range of engineering domains.

1.3 Organization of the Thesis

The remaining chapters are organized as follows. Chapter 2 is a description of the theoretical background related to this study, which contains the concept of object features, the classification definitions, and the basis for the GP and the parameter EM algorithm. Chapter 3 presents the GP-EM method for feature synthesis based on primitive feature vectors. In particular, we explain the k-mean problem via the EM algorithm in the estimation of the Gaussian parameters. The validation stage employs the methods of N folds cross validation and receiver operating characteristic (ROC).

Chapter 4 describes applications of GP-EM for (i) detecting protein conformation defects (PCD) from microscopic imagery, (ii) diagnosing breast cancer disease, (iii) supporting identification in a hand-based biometric system, and (iv) detecting Parkinson's disease. For each application mentioned above, we present our experimental results and comparison results with some other algorithms reported in the literature, in terms of the detection accuracy.

Finally in Chapter 5, we give discussions, conclusions, and also include an example of a potential application - a power quality monitoring system.

Chapter 2

Background Information

2.1 Object Features

For object detection techniques, objects are generally represented by a vector of feature values. Each object is represented as a point in a measurement space, in which the coordinates are the values of features. In order to manage the data involved in the object, we define feature values as the set of measurements, which can be interpreted as a *multidimensional vector* in hyperspace. In this way, each sample is assigned one axis of the object hyperspace in a well-defined array of measurements.

Whether of one homogeneous object variable or a number of them, feature values are ordered in such a way that a particular element in the array has a fixed relationship to the sampling process; there must always be a fixed number of feature values in a fixed order.

2.1.1 Feature Vectors and Spaces

For a given problem, certain types of features defined can be arranged as an ordered set. Such a set is called a *feature vector*. Feature vectors define a multidimensional space in the object feature space, where each feature is laid out along one axis.

Since features must be ordered in fixed number, thus giving rise to another pattern object space—*feature space*. In this space, each object, encountered in object detection procedures, is represented by a single point in each hyperspace, referred as simply *feature spaces* hereafter.

An example of a 'three dimensional hyperspace' is shown in Figure 2.1.

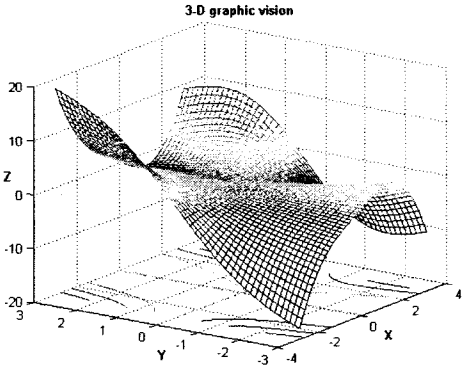


Figure 2.1 A 3-D hyperspace for a function $z = xy\,(x^3 - y^2) / (x^2 + y^2)$

2.1.2 Concepts of Feature Extraction and Selection, in General

The first stage in any object detection task is usually referred to as primitive feature extraction. Feature extraction changes the data into a form that is simpler and easier for a system to detect objects. The advantage of performing primitive feature extraction is that it is a preprocessing process and works on data independently for learning algorithms so that one can still use any learning algorithm one prefers, but has its performance boosted.

The technique of feature selection is a breadth-first search for many primitive features. It begins with a large pool of primitive features and outputs a population of feature subsets in which the feature number and combination are adapted to seek decision boundaries. The objective of feature selection is to characterize objects, and further to reduce the

dimensionality of measurement spaces. Obviously, the selection of feature subsets can improve classification speed and efficiency for certain feature dimensional problems.

2.1.3 Target of Feature Synthesis

Based on the depth-first search for higher order feature functions, the technique of feature systhesis refers to the study of creating new features dynamically in a nonlinear fashion, which projects raw data from primitive feature space onto a lower dimensional synthesized feature space. The purpose is to employ fewer efficient features to represent objects without a decline in the discriminative capability of object detection.

Feature synthesis is a process that automatically synthesizes higher order functions based on the given primitive feature set. The aim to synthesize features is to reduce the dimensionality of the pattern object space by using such new created features. While in the general case synthesized features are functions of the primitive feature set in measurement spaces, it will be possible to reduce the feature dimensionality or eliminate redundancy of primitive features in the process of feature creation.

2.2 Classification

In the discipline of computer vision, the classification (or detection) is the process of grouping objects of interest into a known number of categories or classes, given a set of observations. The goal of object classification is that the objects or events with some similar properties are grouped into a class.

The categories of objects to be classified, or to be detected could be sets of complex patterns in image analysis areas, or sets of subtle defect states in medical fields, or sets of multifaceted machine fault phenomena in engineering domains.

2.2.1 Definitions of Object Classification, or Detection

The concept of the object classification, or the object detection, can be expressed in terms of the partition of feature space (or a mapping from feature spaces to decision spaces). Suppose that N features are to be measured from each input object. Each set of d features can be considered as a vector, $\mathbf{\Omega}_d$, called a d-dimensional feature (measurement) vector.

The problem of classification is to assign each sample to a proper object class by the means of feature vectors. This can be interpreted as a partition of the feature space into mutually exclusive regions and each region will correspond to a particular object class. Mathematically, the problem of classification can be formulated in terms of "discriminant function", denoted as the D function. Let $\omega_1, \omega_2, \ldots, \omega_k$ defined as the k classes, and let $\mathbf{\Omega}_d = \{\phi_1, \ldots, \phi_d\}$ be the d-dimensional feature (measurement) vector where ϕ_l represents the lth feature measurement. Then the discriminant function $D_j(\mathbf{\Omega}_d)$ associated with the feature vector $\mathbf{\Omega}_d$ ($l = 1, \ldots, d$) and object class ω_j ($j = 1, \ldots, k$), is such that if the input object represented by the d-dimensional feature vector $\mathbf{\Omega}_d$ is in class ω_j, denoted as $\mathbf{\Omega}_d \sim \omega_j$, then the value of $D_j(\mathbf{\Omega}_d)$ must be the largest, i.e., for all $\mathbf{\Omega}_d \sim \omega_j$, it must be satisfied with the expression of $D_j(\mathbf{\Omega}_d) > D_n(\mathbf{\Omega}_d)$ in which $n, j = 1, \ldots, k$ and $n \neq j$.

2.2.1.1 Supervised and Unsupervised Object Classification

In the problems of object classification, if there exists some set of objects, the individual category of which is already known, then it is called a problem of *supervised object classification.* If the classes of all of the available objects are unknown, and perhaps even the number of these categories is unknown, then it is called a problem of *unsupervised object classification* or *clustering* [Bi06].

2.2.2 Minimum Distance Classifier (MDC)

Mathematically, the problem of classification can be formulated in terms of a predetermined measure [Fu90]. For the minimum distance classifier (MDC), the distance is defined as an index of similarity so that the minimum distance is identical to the maximum similarity. The MDC has been used as an important object detection tool. In such method, the MDC uses the distances between the input object and a set of reference vectors or prototype points in the feature space as the detection criterion.

Suppose that *n* reference vectors R_1, R_2, ..., R_n are given in which R_j is associated with the object class ω_j. A minimum-distance classification scheme with respect to R_1, R_2, ..., R_n is to classify the input X as from class ω_j, i.e., $X \sim \omega_j$ if $| X - R_j |$ is the minimum. The expression of $| X - R_j |$ is the distance between X and R_j.

2.2.3 K Nearest Neighbors (KNN) Classifier

The aim of the K nearest neighbors (KNN) method is to find the nearest neighbors of an unidentified test object within a hyper-sphere of pre-defined radius in order to determine its true class. The traditional KNN rule has been described as follows [Bi06]:

- Out of N training vectors, identify the K nearest neighbors, irrespective of class label. K is chosen to be odd.
- Out of these K neighbors, identify the number of vectors, K_i, that belong to class ω_i, i = 1, 2,..., M (obviously, $\sum_i K_i = K$).
- Assign x to the class ω_i with the maximum number K_i of samples.

KNN can detect a single or multiple number of nearest neighbors. A single nearest-neighbor method is primarily suited to detect objects where we have sufficient confidence that class distributions are non-overlapping and the features are discriminatory.

In the KNN methods, the first assumption requires that feature vectors for classes are discriminatory. This means that feature vectors are different among various classes as to ensure that classes are surrounded by their true samples. The second assumption requires that the unique characteristic of an object that defines its signature, and ultimately its class, is not significantly dependent on the interaction among various features.

In other words, the KNN method works better with data where features are statistically independent. This is because KNN is based on some form of distance measure, not depending on their feature interaction.

2.3 Genetic Programming (GP)

2.3.1 Evolutionary GP Process

The first step of GP is to randomly create initial populations of computer programs, including the parameters for controlling the run. After that, there are two major tasks processed in the GP loop [Ko94], as shown in Figure 2.2:

1) The evaluation of each program is done by using a *fitness function*, a user defined function that will determine how suitable the programs are for the environment;

2) The genetic operations, such as the reproduction, crossover, and mutation, create new populations based on their fitness values. Selection by themselves allows for the identification of the best individuals in a probabilistic way.

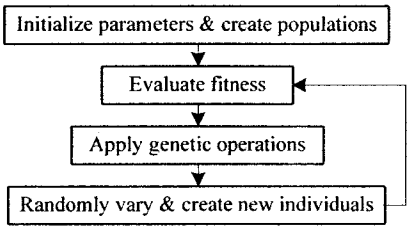


Figure 2.2. Flowchart of evolutionary GP.

The cycle ends when a program either reaches some predetermined fitness, or reaches some other termination criteria such as having evaluated a certain number of generations.

2.3.2 Generative Graphic Representations

One of the key features of GP is that it uses tree structure representations to solve problems. GP considers the tree structure representations as computer programs that are able to take various sizes and shapes, as well as use different functions to find solutions.

An example of a tree representation for the function of z{x + *sin* (y)} is illustrated in Figure 2.3.

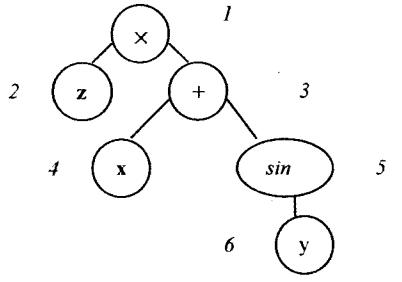


Figure 2.3. An expression of z{x + *sin*(y)} in a GP graphic representation

Generally, the evaluation of genetic programming is done in a recursive, depth-first way, starting from the left. The three internal points on the tree are labeled with unctions, such as "+", "×" and "*sin*". The three external points /nodes on the tree are labeled with the arguments (the variables of x, y and z).

2.3.3 Genetic Operations

The operations of reproduction, crossover and mutation in GP are performed on copies of the selected individuals. The selected individuals remain unchanged in the population until the end of the current generation [Ko94].

Reproduction

The principle of reproduction is used to create a new offspring population of an individual computer program from the current population of programs. The reproduction operation, with its probability of P_r, involves selecting the best individuals on the basis of their fitness by copying them into the new population.

Crossover

With the crossover probability of P_c, the sub-tree crossover occurs by selecting two parents and then choosing random points in their program trees to be swapped over. This creates two new children. Figure 2.4 illustrates a typical sub-tree crossover operation.

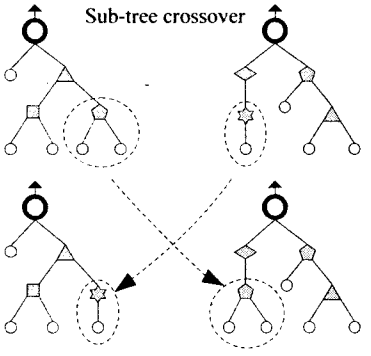


Figure 2.4. GP crossover.

Because the programs are selected with the reproduction probability based on their fitness to participate in the crossover operation, which allocates future trials of the search for a solution to the problem whose programs contain parts from promising programs.

Mutation

The mutation is performed by a single node exchanged against a random node of the same class. A random index number is generated to indicate the point where the mutation will happen. The basic idea of this structure-preserving GP mutation for terminators and operators is that any noninvariant point anywhere in the overall program is randomly chosen, without any restriction, with the mutation probability of P_m, shown in Figure 2.5.

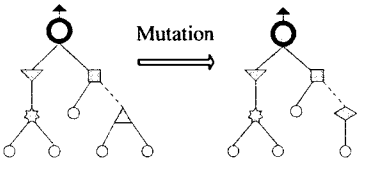


Figure 2.5. GP mutation.

2.3.4 Fitness Measure

The evolutionary GP process is driven by a fitness measure that evaluates how well each individual computer program in a population performs in its problem environment. The basic mechanism in GP is Darwinian evolution: bad traits are eliminated from the population; and good traits survive and are mixed by recombination (mating) to form better individuals.

The design of fitness function is to ensure that diversity is not lost in the population, so GP can continue to explore. GP allows each computer program to evolve under specified selection rules to a state that maximizes fitness values. The fitness measure must satisfy the requirement of being fully defined in the sense that it is capable of evaluating each computer program that it encounters in any generation of the population.

2.4 Expectation Maximization (EM) Algorithms

The goal of EM is to find the maximum likelihood solutions for models having hidden variables [Bi06]. Note that the unobserved data **Z** are treated as hidden variables whose probability distributions depend on the parameter $\boldsymbol{\theta}$ and the observed data **S**. The EM algorithm uses its current hypothesis $\boldsymbol{h}$ in place of the parameter $\boldsymbol{\theta}$ to estimate the distributions governing the complete data **Y**, a combination of the observed data $s_i \in \mathbf{S}$ and unobserved data $z_{ij} \in \mathbf{Z}$.

2.4.1 Estimation of the Means of the *k* Gaussians

Suppose **S** is a set of m independently drawn samples that are mapped by a mixture of the

k distinct Gaussian distributions. Figure 2.6 illustrates an extended example with $k = 3$ from [Mi97] where the samples are the points shown along the s-axis. Since we cannot observe which samples are generated by which distribution, we have a prototypical example of a problem involving the hidden variables z_{ij}. Taking the example of Figure 2.6, the full description of the ith sample can be written as the expression of (s_i, z_{ij}), where z_{ij} indicates which of the j Gaussian distributions, $j = 1, 2, 3$, is being used to generate the value s_i.

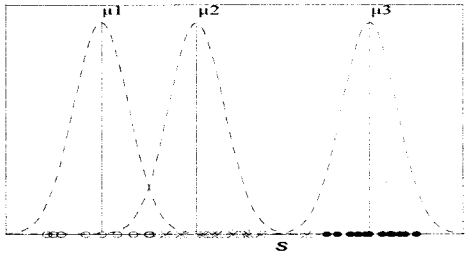


Figure 2.6. The samples, shown by the points along the s-axis, are mapped onto a mixture of three Gaussian distributions.

2.4.2 Statement of the EM Algorithm

The term "incomplete data" in its general form implies the existence of two sample spaces **S** and **Y** and a many-one mapping from **Y** to **S**. The observed s_i is a realization from **S**. The corresponding y_i in **Y** could not observed directly, but only through s_i indirectly, and that y_i is known only to lie in $\mathbf{Y}(s_i)$ [Bi06].

Since the complete data **Y** are a combination of the observed data **S** and unobserved data **Z**, we must average over the possible values of the unobserved **Z** according to the probability. In other words, we take the expected value $E[\ln P(\mathbf{Y}/\, h')]$ over the probability distribution to govern the random variable **Y**, with h' denoted the revised hypothesis that is estimated on each iteration of the EM algorithm. Therefore, the EM algorithm uses its

current hypothesis h in place of the actual parameter θ to estimate the distribution which governs **Y**.

In the estimation of the k-means problem, the EM algorithm consists of the following steps [Mi97], [DH01], here we use the notation introduced at the beginning of Section 2.4 and Subsection 2.4.1:

***E*-step**: Define the Q function to estimate the expected value $E[z_{ij}]$ that the sample s_i is generated by the jth Gaussian distribution, given its current hypothesis $h = (\mu_1,\ldots, \mu_k)$:

$$Q(h'/h) = E[\ln P(\mathbf{Y} / h')/h, \mathbf{S}]. \qquad (2.1)$$

***M*-step**: Replace the hypothesis h with the revised hypothesis $h' = (\mu_1',\ldots, \mu_k')$ that maximizes this Q function:

$$h \leftarrow \text{argmax}\, Q(h'/h). \qquad (2.2)$$

Here, the Q function in the form $Q(h' / h)$ indicates that it is defined partially with the assumption that the current hypothesis h is equal to θ. The complete data **Y** represents a combination of the observed data $s_i \in \mathbf{S}$ and unobserved data $z_{ij} \in \mathbf{Z}$, h is the current hypothesis, h' is the revised hypothesis and $E[z_{ij}]$ is the probability that the sample s_i is generated by the jth Gaussian distribution.

Chapter 3

The Hybrid GP-EM Approach

3.1 Framework Overview

The design of the GP-EM algorithm involves the optimization of the parameters, i.e., the means of the k Gaussian distributions, which evaluate the performance of the feature synthesis. Figure 3.1 is a flowchart of the process. In Subsections 3.2 -3.5, we shall explain

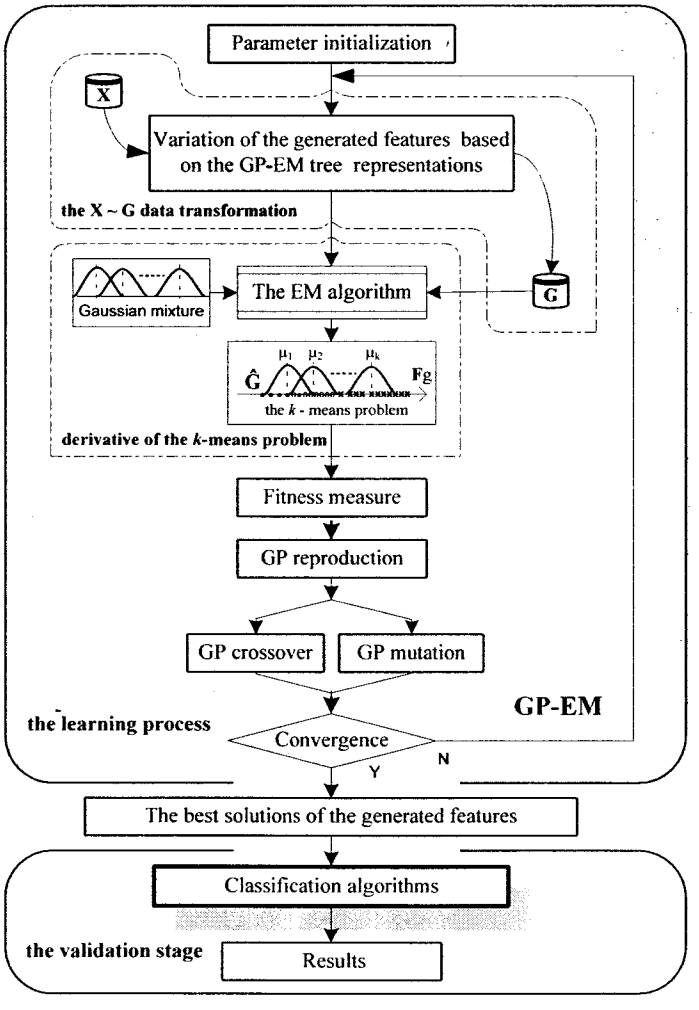


Figure 3.1. The process of the GP-EM feature generator.

in detail the various steps shown in Figure 3.1.

Consider a problem in which the raw data **X** is a set of m instances, that belong to the k known classes, with a d-dimensional primitive feature set. Let **G** denote the entire generated data, the input of the EM algorithm. After going through the **X~G** data transformation when the synthesized data **G** are mapped from the raw data **X,** based on the variation of the GP-EM tree representations, the question remains how to determine the probability density function of **G**, which are the input data of EM (see Figure 3.1). The sum of a sufficiently large number of independent, identically distributed random variables itself obeys a Gaussian distribution, regardless of the distributions of the individual variables [Mi97]. This implies that the value of g_i, $g_i \in \mathbf{G}$, generated by the sum of a large set of independent but identically distributed factors will itself be Gaussian distributed. Since the generated value g_i is a Gaussian signal, the parametric EM algorithm assumes that its input data **G** follow the Gaussian distribution for each class in the estimation of the parameters $\theta = \{\pi, \mu, \Sigma\}$. For this independently drawn Gaussian signal g_i, the selection of each of the k Gaussians is based on choosing one with the equal prior probability, $1/k$, in the mixing probability vector π [RD03]. Each of the k Gaussian distributions has the same variance, which means that the covariance matrix Σ contains only one component of the variance σ^2 [Bi06]. Thus, the mean vector, the values of the k-means, becomes the only unknown component in the parameters $\theta = \{\mu\}$. Further detail of the k-means problem via EM is presented in Subsection 3.3.

As the generalization of the k-means problem, there are two steps in the GP-EM learning process. First, one of the k Gaussian distributions is selected with the equal prior probability $1/k$ for all the k categories [Mi97]. Second, the generated value g_i is mapped

onto this selected distribution as a sample $\hat{g}_i$. This process is repeated to generate the set of the sample points, $\hat{g}_i \in \hat{\mathbf{G}}$, under the Gaussian mixture along the synthesized feature **Fg**-space on each iteration. Once done, the data $\hat{\mathbf{G}}$ under the Gaussian mixture in the hypothesis of the *k*-means problem goes through a typical GP process with the following steps:

(a) the computation of the fitness measure (to be explained in Subsection 3.4);

(b) the application of reproduction, crossover and mutation to generate the next population (as explained earlier in Subsection 2.3.2);

(c) the convergence testing (to decide whether to stop the process or not).

This process is iterated until the convergence test is satisfied and then the process is stopped. At the end of the learning process, the GP-EM system returns the best solutions, see Figure 3.1, as the input into the validation stage.

The following subsections give a detailed description of the various aspects of the proposed GP-EM algorithm that changes the learning task of the feature synthesis into the *k*-means problem (see Subsection 3.3). This is followed by an explanation of each step of the approach, including the GP-EM tree representations (see Subsection 3.2), which are the sequences of applications of the function set to the terminal set, the fitness measure (see Subsection 3.4) and the validation stage (see Subsection 3.5).

3.2 GP-EM Tree Representations

3.2.1 Function and Terminal Sets

The function and terminal sets are the elements from which the GP-EM approach attempts to build the tree representations for the synthesized feature functions.

Constructed from the raw data **X**, the terminal set chosen in this study comes from a set of d raw features that are replaced with the primitive features of each experimental dataset.

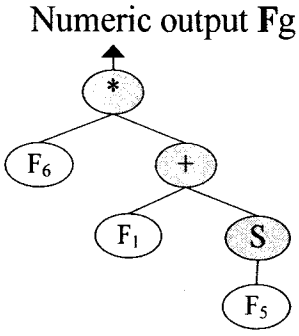
Table 3.1 lists the mathematical elementary functions designed in this study to fit the terminal set, which is replaced with the primitive feature set. The division and square root operators in the function set are protected against divisions by zero and negative values, respectively.

Table 3.1. Function set.

Signs	Elementary functions	Terminators
+, -	addition, subtraction	2
*, /	multiplication, division	2
A, N	absolute, negative value	1
S, C, T	sine, cosine, tangent functions	1
P, R, E	square, square root, exponential	1

3.2.2 An Example of a GP-EM Tree Representation

The programs of the LISP programming language are S-expressions, in effect, the parse tree of the program [Ko94]. The outputs of the resulting synthesized features in this study are presented in parenthesized prefix-terminator expressions, named LISP-like expressions. Its main characteristic, in comparison to LISP, is that the terminators in each level are enclosed by a pair of parentheses following each function symbol. However, in LISP the functions are inside the parentheses. Therefore, the synthesized feature functions in LISP-like expressions take the form of the GP tree structures. From our LISP-like expressions, each individual member can be written into a rational expression, an algebraic expression composed of elementary functions and primitive features.



(a) A GP-EM tree representation in LISP-like expression $\mathbf{Fg}^{[\text{LISP-like}]}$ = *(F6 +(F1 S(F5))).

F1	F5	F6	Fg
0.1	90°	0.1	0.11
0.2	90°	0.1	0.12
0.3	90°	0.1	0.13
0.4	90°	0.1	0.14

(b) The values of the terminators / the output for the corresponding rational expression $\mathbf{Fg}^{[\text{rational}]}$ = F6 X {F1+ sin (F5)}.

Figure 3.2. An example of a GP-EM tree representation.

Figure 3.2 illustrates a single GP-EM tree representation in the LISP-like expression $\mathbf{Fg}^{[\text{LISP-like}]}$ = *(F6 +(F1 S(F5))) and rational expression $\mathbf{Fg}^{[\text{rational}]}$ = F6 X {F1 + sin (F5)}. On the tree, there are two kinds of sets: function and terminal. The functions are internal nodes (the gray nodes), which represent elementary operators such as addition, multiplication and sine function. The terminators are leaf nodes (the white nodes) that receive the values of the primitive features, F1, F5 and F6, from the experimental data environment. The root is the output of the GP-EM tree for the synthesized feature **Fg**, which is determined by executing the terminal arguments Fl, l = 1, 5, 6, to the elementary functions with the signs of [+, *, S] (see Table 3.1).

3.3 The *k*-means Problem

For a given data set **G** with m instances, the probability density function of **G**, which follows the k-component finite mixture distributions evaluated at g_i, can be written as [DH01]:

$$p(g_i / \theta) = \sum_{j=1}^{k} \pi_j f_j(g_i / \theta_j), \quad j = 1,...,k, \qquad (3.1)$$

$$\text{with} \quad \sum_{j=1}^{k} \pi_j = 1, \quad \pi_j \geq 0, \quad j = 1,...,k, \qquad (3.2)$$

in which $f_j(g_i/\theta_j)$ represents the component density functions modeling the points of the jth distribution, and π_j is the mixing coefficient. The multivariate Gaussian mixture with the mean vector μ and the covariance matrix Σ_j is given by [Lu06]:

$$f_j(g_i / \theta_j) = \frac{\exp(-\frac{1}{2}(g_i - \mu_j)^T (\Sigma_j)^{-1}(g_i - \mu_j))}{\sqrt{(2\pi)^d |\Sigma_j|}}. \qquad (3.3)$$

The parameters $\theta = \{\pi, \mu, \Sigma\}$ in (3.1) are determined to measure how well the corresponding mixture model fits the data **G.** We use the term $P(\mathbf{G}/\pi, \mu, \Sigma)$ to denote the the likelihood of the data given the model, expressed in (3.1) – (3.3).

In many cases the correlations between the variables are the same within each group and this can be used to simplify the EM algorithm that the covariance matrix Σ contains only one component – the variance σ^2 [Bi06]. The selection of the prior probabilities for the various categories has been the subject of a substantial body of literature [Mi97]. One of the most common methods is to estimate the relative frequency for each class from the training data and use these values for the prior probabilities. An alternate method is to assume equal prior probabilities, $1/k$, for all the k categories [RD03].

Therefore, the learning task is reduced to a hypothesis $\boldsymbol{h} = \boldsymbol{\theta} = \{\mu\}$ that describes the means of each of the k distributions. In that case, the probability of a single instance y_i of the complete data **Y**, the combination of the data **G** and **Z,** for a mixture of the k Gaussian distributions is [Bi06], [Mi97]:

$$p(\mathbf{y}_i / \boldsymbol{h}') = p(g_i, z_{i1}, \ldots, z_{ij}, \ldots, z_{ik} / \boldsymbol{h}')$$
$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(g_i - \mu_j')^2\right). \quad (3.4)$$

For each instance $\boldsymbol{y}_i = (g_i, z_{i1}, \ldots, z_{ij} \ldots, z_{ik})$, g_i is the observed value of the ith instance. z_{ij} is a binary k-dimensional vector that has the value *1* if g_i is created by the j distribution and the value *0* otherwise. From (6), the log-likelihood of the complete data **Y** is:

$$\ln P(\mathbf{Y} / \boldsymbol{h}') = \sum_{i=1}^{m} \left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(g_i - \mu_j')^2\right). \quad (3.5)$$

Note that the above expression of ln $P(\mathbf{Y}/\boldsymbol{h}')$ is a linear function of z_{ij}. In general, for any function L(z) that is a linear function of z, we have that the expectation E is such that E [L(z)] = L(E [z]). So, the Q function defined by (2.1) for the k-means problem is:

$$Q(\boldsymbol{h}'/\boldsymbol{h}) = E[\ln P(\mathbf{Y} / \boldsymbol{h}')/\boldsymbol{h}, \mathbf{G}]$$
$$= \sum_{i=1}^{m}\left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}](g_i - \mu_j')^2\right) \quad (3.6)$$

When applying the maximum likelihood (ML) method, the first term in (3.6) is a constant that is independent of $\boldsymbol{h}$, and can therefore be discarded. Since maximizing the negative quantity is equivalent to minimizing the corresponding positive quantity, we have:

$$\arg\max_{h'} Q(\boldsymbol{h}' / \boldsymbol{h}) = \arg\max_{h'} \sum_{i=1}^{m}\left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{\sum_{j=1}^{k} E[z_{ij}](g_i - \mu_j')^2}{2\sigma^2}\right)$$
$$\equiv \arg\min_{h'} \sum_{i=1}^{m}\sum_{j=1}^{k} E[z_{ij}](g_i - \mu_j')^2. \quad (3.7)$$

In (3.7), the ML estimator minimizes a weighted sum of squared errors over the m instances to their means, which are weighted by $E[z_{ij}]$. $E[z_{ij}]$ is the probability that the instance $\hat{g}_i$ is generated by the j Gaussian distribution, defined as the "E-step" [Mi97]:

$$E[z_{ij}] = \exp\left(-\frac{1}{2\sigma^2}(\hat{g}_i - \mu_j)^2\right) \Big/ \sum_{n=1}^{k} \exp\left(-\frac{1}{2\sigma^2}(\hat{g}_i - \mu_n)^2\right). \quad (3.8)$$

We use $E[z_{ij}]$ to obtain a new maximum likelihood hypothesis $\boldsymbol{h}' = (\mu_1', \ldots, \mu_k')$, defined as

the "*M*-step" [Mi97]:

$$\mu_j' \leftarrow \sum_{i=j}^{m} E[z_{ij}]\hat{g}_i / \sum_{i=1}^{m} E[z_{ij}], \mathrm{j}=1,\ldots, k. \quad (3.9)$$

In practice, (3.8) and (3.9) establish the EM steps in the implementation of the GP-EM approach for devising the mixture model of the k Gaussians.

3.4 Fitness Function, the *J* Value

The nature of the fitness measure varies with the problems. In the GP-EM approach, one important consideration for the fitness measure is to ensure that the measurement procedure gives the detection evaluation for the samples, $\hat{g}_i \in \hat{\mathbf{G}}$, regardless of the coordinate system of these generated samples. In addition, there exists in the design the process of the *k*-means assignment via EM in the iteration of the GP application.

We wish to see whether the k classes could be separated and whether the homogeneous instances could be tightly closed by using few resulting synthesized features. The technique to find the appropriate fitness measure is to determine the *separability* of the k classes by a scattering criterion function and the *tightness* of the homogeneous instances by a within-class measurement.

In this study, the fitness measure is performed by checking how far separable the various spectral k classes remain and how much close the homogeneous instances stay together. For a lower synthesized feature space, we use a scalar measure of the "size" of the scatter values, instead of matrices.

Suppose there is a set of m instances in the data $\hat{\mathbf{G}}$ divided into the k subsets $\hat{\mathbf{G}}_1,\ldots,\hat{\mathbf{G}}_k$. The total mean value is [Fu90]:

$$\mu_T = \frac{1}{m}\sum_{i=1}^{m} \hat{g}_i. \qquad (3.10)$$

The scatter value for the *j*th class is given by [DH01], [Th89]:

$$S_j = \sum_{\hat{g}_i \in \hat{G}_j} (\hat{g}_i - \mu_j)(\hat{g}_i - \mu_j)^T, \qquad (3.11)$$

and the within-class scatter value is given by:

$$S_W = \sum_{j=1}^{k} S_j \qquad (3.12)$$

Further, the between-class scatter value is given by:

$$S_B = \sum_{j=1}^{k} n_j(\mu_j - \mu_T)(\mu_j - \mu_T)^T \qquad (3.13)$$

where $\hat{\mathbf{G}}_j$ is the *j*th subset of $\hat{\mathbf{G}}$, n_j is the number of samples in the *j*th class, and μ_j is replaced with the mean of the *j* Gaussian distribution in the GP-EM approach, instead of the mean of the *j*th subset $\hat{\mathbf{G}}_j$.

There is an exchange between the within-class and the between-class scatter values, one goes up as the other goes down. Since maximizing the logarithm of a quantity also maximizes that quantity, we use an optimal partition measurement as the fitness function by using the *J* value, which is given by [Fu90]:

$$J = \ln | S_B / S_W |. \qquad (3.14)$$

Therefore, in the assignments of the means of the *k* Gaussians for the generated data $\hat{\mathbf{G}}$, the higher the homogeneity within the class region, the higher would be the *J* value, or the higher the separate measurement among the *k* class categories, the higher would be the *J* value.

3.5 Pseudo-Code for the GP-EM Implementation

We use the following running notation:

t: the index (superscript)_of the maximal generations, N.
p: the index of the population size, P.
i: the index of the training samples, *m*.
j: the index of the classes, *k*.
d: the number of the primitive features obtained from a dataset.

Initialize: the *k* mean values, $\mu_j^{[0]}(p)$, $\forall j \in (1,\ldots, k)$, $\forall p \in (1,\ldots, P_s)$.

***Input*:** the raw data $\mathbf{X}$, $x_i^{(d)} \in \mathbf{X}$ with a *d*-dimensional feature set, $\forall\, i \in (1,\ldots, m)$, replaced individually with each dataset.

1. **while** (Increment in the iteration of generations if $t \leq N^{[t]}$) **do** {
2. **for** $p = 0$: P_s populations {
3. synthesize the feature functions based on the GP-EM tree representations (see Subsection 3.2);
4. **for** $i = 0$: *m* samples {
5. perform the X ~ G data transformation based on the variation of the GP-EM tree structures (see Subsection 3.2):
6. calculate the expected value, "*E*-step", as expressed in equation (3.8) (see Subsection 3.3):

$$E[z_{ij}]^{[t]}(p) = \frac{\exp[-\frac{1}{2[\sigma^{[t]}(p)]^2}[\hat{g}_i^{[t]}(p) - \mu_j^{[t]}(p)]^2]}{\sum_{n=1}^{k} \exp[-\frac{1}{2[\sigma^{[t]}(p)]^2}[\hat{g}_i^{[t]}(p) - \mu_n^{[t]}(p)]^2]}, \quad \forall j, n \in (1,\ldots, k);$$

7. **end for** (*i*)
8. derive the revised *k* mean values, "*M*-step" , as expressed in equation (3.9) (see

Subsection 3.3):

$$\mu_j^{[t+1]}(p) \leftarrow \sum_{i=1}^{m}\{E[z_{ij}]^{[t]}(p)\hat{g}_i^{[t]}(p)\}\Big/\sum_{i=1}^{m} E[z_{ij}]^{[t]}(p), \quad \forall j \in (1,\ldots k);$$

9. replace the k mean values with the new revised ones:
$t \leftarrow t+1$;

10. calculate the within-class scatter value, as expressed in equation (3.12) (see Subsection 3.4):

$$S_W^{[t]}(p) = \sum_{j=1}^{k} \sum_{\hat{g}_i^{[t]}(p) \in G_j^{[t]}(p)} [\hat{g}_i^{[t]}(p) - \mu_j^{[t]}(p)][\hat{g}_i^{[t]}(p) - \mu_j^{[t]}(p)]^T, \quad \forall j \in (1,\ldots k);$$

11. calculate the between-class scatter value, as expressed in equation (3.13) (see Subsection 3.4):

$$S_B^{[t]}(p) = \sum_{j=1}^{k} n_j [\mu_j^{[t]}(p) - \mu_T^{[t]}(p)][\mu_j^{[t]}(p) - \mu_T^{[t]}(p)]^T, \quad \forall j \in (1,\ldots k);$$

12. calculate the fitness function using the logarithm of the J values, as expressed in equation (3.14) (see Subsection 3.4):

$$J^{[t]}(p) = \ln | S_B^{[t]}(p)/S_W^{[t]}(p) |; \}$$

13. **end for** (p)

14. select {the best tree representations}$^{[t]}$ according to the fitness measure, the J values, as expressed in equation (3.14) (see Subsection 3.4);

15. apply { the reproduction}$^{[t]}$ with its probability P_r;

16. apply { the crossover}$^{[t]}$ with its probability P_c;

17. apply { the mutation}$^{[t]}$ with its probability P_m;

18. create and vary new_populations : = { survivors}$^{[t, t+1]}$; }

19. **end while** $^{[t]}$

Complexity theory deals with how algorithms scale with an increase in the input size. In general, the time required to solve a problem is calculated as a function of the size of

the instance. If the input size is n, the time taken can be expressed as a time complexity function of n, T(n), which is equal to the maximum number of basic operations that the algorithm performs on the input of length n [AB09]. From the pseudo-code above, the process of the proposed GP-EM algorithm is evolved in a loop structure, i.e., the number of N cycles (see the step 1 in the pseudo-code). It is clear that the proposed GP-EM algorithm has a polynomial time complexity because the run time of the loop is a linear function of the input size of N cycles, which can be expressed as the function of T(N). Thus, GP-EM does not have an exponential complexity and so it is computationally efficient.

3.6 Validation

3.6.1 *N*-fold Cross Validation Approach

Cross validation, sometimes called *rotation estimation* is the statistical practice of partitioning a sample of data into subsets to estimate the true performance of a classifier. The idea is to randomly divide the data into N mutually exclusive partitions (folds), keeping one fold for testing and the rest for training. Then another fold will be selected for testing and all the remaining folds for training. The cross validation process is then repeated N times (the *folds*), with each of the N subsets used exactly once as the validation data. The N results from the folds then can be averaged (or otherwise combined) to produce a single estimation.

We employ the 10-fold Cross Validation approach in the applications to detecting breast cancer disease (see Subsection 4.2) and detecting Parkinson's disease (see Subsection 4.3).

3.6.2 Receiver Operating Characteristic (ROC)

A receiver operating characteristic (ROC) graph is a two-dimensional depiction of the detection performance in estimating a curve of the hit rate (sensitivity) vs. the false positive rate (1–specificity). The two axes represent tradeoffs between benefits (the hit rate) and errors (the false positive rate) that a detection system makes between classes. The ROC curves are obtained by computing threshold values within a given range, from which the hit rate is the proportion of true positives and the false positive rate is the proportion of false positives [Fa06]. The threshold range is a function of the fragment length. The areas below the ROC curves indicate the discrimination capabilities of approaches, and thus, as a function of desirable goals, the curves of ROC provide convenient means of evaluating the performance for the designs of systems.

We employ the ROC analysis in the application of detecting protein conformation defects (see Subsection 4.1).

3.7 Implementation

Table 3.2 lists the common GP parameter settings used for all the experiments. Depending on the *k*-means problem needed to be solved, we choose a medium population size and relatively large number of generations to help the algorithm adequately sample the search surface to find better solutions of synthesized features. The rationale for this is that once the search surface is adequately explored, the algorithm would work with a subset of the best samples. In addition, the crossover and mutation increase the sampling range of the algorithm.

Table 3.2. The common GP parameter values.

Parameter names	Values
Population size, P_s	32
No. of generations, N'	400
Max. of tree levels	5
No. of elementary functions	12
Reproduction probability, P_r	0.125
Crossover probability, P_c	0.50
Mutation probability, P_m	0.375
Function crossover rate	0.75
Terminal crossover rate	0.25
Function mutation rate	0.75
Terminal mutation rate	0.25

GP allocates to every individual some chance of being selected to participate in the operations of reproduction, crossover and mutation. In each generation, we take these three genetic operations with the probability condition: $P_r + P_c + P_m = 1$.

Chapter 4

Applications

4.1 Detecting Protein Conformation Defects from Microscopic Imagery

4.1.1 Introduction

Protein conformational diseases (PCDs) are pathologically diverse disorders in which specific proteins accumulate in certain organs. Each proteopathy of PCDs is characterized by a disease-specific buildup of aggregated proteins within cells of the body. Under the electron microscope, such deposits of abnormal protein aggregation often have a fibrillar appearance associated with at least 40 human diseases, including oculopharyngeal muscular dystrophy (OPMD), Alzheimer's disease, Parkinson's disease, Huntington's disease and prion diseases [NCBI]. Patients with PCDs may benefit from early diagnoses in order to receive primary care or undergo surgical procedures to address their progressive symptoms [WM05]. Our work is aimed at providing medical decision criteria to the physicians, specifically for detecting the OPMD, and may also help them in identifying the difficult cases.

The OPMD involves a cognitive decline and psychotic manifestations. There is

currently no cure for the OPMD [BB09]. In most cases, the microscopic imagery of patients with the OPMD have certain muscular intranuclear inclusions (INIs) that are thought to be the hallmark of this disease. The detection of the OPMD is usually established by the presence or absence of the INIs in the microscopic imagery [KM07]. When examining the OPMD microscopic imagery, it is often difficult to analyze the complex structures of the images due to their irregular sizes and shapes with localized background structures. Possible approaches could use the existing background extraction techniques, such as the threshold segmentation [GW07], or the mathematical morphology [TM96], but such methods require complex preprocessing steps.

To simplify the process, we use a bin-threshold-based technique to filter the image backgrounds into a histogram margin that is independent of the image size and shape; then, we perform the texture extraction in a histogram region of interest by thresholds, named HROIBT that encompasses the color information of the INIs.

4.1.2 The Dataset of CellsDB

Collected using microscopic imagery, the CellsDB is a database of pre-segmented cell images in the binary classes of the healthy and sick conditions of the OPMD [KM07]. Each image contains a single cell on a white background at an optimal level of RGB values. Usually, the healthy cell images have a dark green color while the sick ones contain the lighter green color of INIs with a bumpy appearance. The presence of INIs in patients' cell images leads to the suggestion that protein aggregation is a critical molecular component of the OPMD disease. Figure 4.1. shows four sample images, healthy cells and sick cells, from CellsDB.

Figure 4.1. Four cell images, healthy and sick samples, from the CellsDB.

4.1.3 HROIBT-Based Texture Data Preparation

4.1.3.1 Designs of HROIBT

Conversion of a color image to grayscale is very common in digital image processing [GW07]. To be able to capture the inclusion of color information, we begin our color analysis by converting the cell images into grayscale. We find that: (a) the image backgrounds are filtered to the margin of the histogram, regardless of differences in size and shape; (b) the lighter INIs of the OPMD images reside in the middle region of the grayscale histogram. Accordingly, if we take into consideration the entire region of the grayscale histogram, we might make errors to compute the intensities of the INIs that possess discriminant texture features. In order to capture the color information of the INIs, we use two thresholds to limit the interval of their gray values, and name the interval the *Histogram Region Of Interest By Thresholds* (HROIBT). Figure 4.2 illustrates the procedures for preparing the basic features from HROIBT, using the database CellsDB.

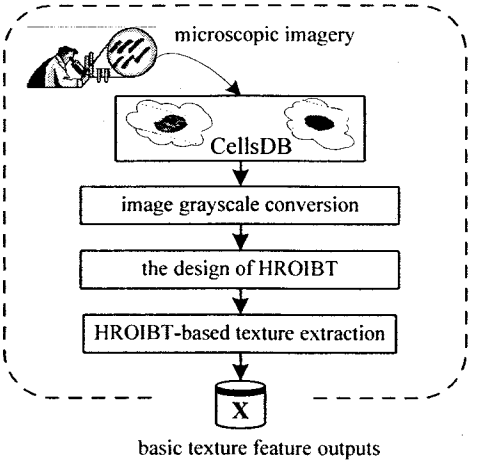


Figure 4.2. Procedures of the basic texture feature preparation.

Figure 4.3 presents the output of the grayscale histograms for all the sample images with two classes, the healthy and sick conditions of the OPMD. From Figure 4.3, we see that there is a near-zero intensity region centered at the bin of 200. The upper threshold, $\mathbf{T}_U$, of the HROIBT is then set at 200. On the other side of the histograms, there is an overlapping bin region between the healthy and sick cells in the intensity boundary range (25, 75). We use the average variances ($\mathbf{V}a$) to measure the degree of spread of the texture-based features (the details of the categories will be described in the following subsection) to determine the lower threshold, $\mathbf{T}_L$. A feature set with a good stability and reliability should exhibit a small $\mathbf{V}a$ among the samples [Fu90]. As shown in Figure 4.4, the optimal value of the bin should be at the lowest $\mathbf{V}a$, so we set the lower threshold, $\mathbf{T}_L$, at the bin of 50.

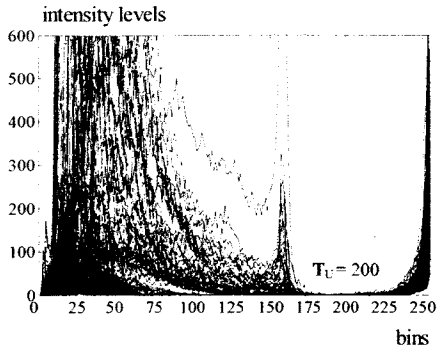


Figure 4.3. The upper threshold of the HROIBT, $\mathbf{T}_U = 200$, defined by the histogram measure for all image samples in the binary classes of the healthy and sick conditions of the OPMD.

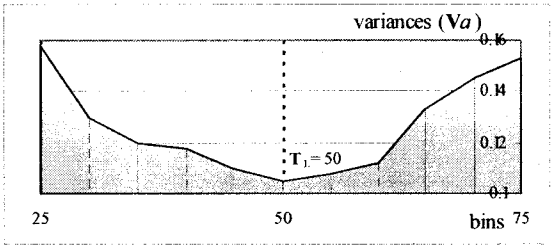


Figure 4.4. The lower threshold of the HROIBT, $\mathbf{T}_L = 50$, defined by $\mathbf{V}a$.

Figure 4.5 illustrates the constraint solution of the HROIBT in the bin range of (50, 200), together with the grayscale histograms of four sample images in Figure 4.1.

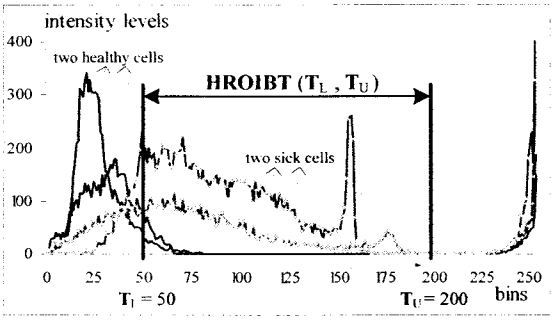


Figure 4.5. Solution of the HROIBT in the bin range of (50, 200), with the grayscale histogram measure of the healthy and sick samples in Figure 4.1.

4.1.3.2 Basic Texture Feature Extraction from HROIBT

After the presence of the OPMD inclusions is expressed in the HROIBT, 17 basic features,

D1-D17, are subsequently extracted from the HROIBT as described below [GW07].

Fourier Descriptors (D1-D6): the *l*th harmonic amplitude of Fourier descriptors [GW07].

$$Dl = \sqrt{a_l^2 + b_l^2}, \quad l = 1,...,6 \tag{4.1}$$

$$\text{where} \quad a_l = \frac{2}{N_{HROIBT}} \sum_{n=1}^{N_{HROIBT}} p(z_n) \times \cos(2\pi l \times n / N_{HROIBT})$$

$$\text{and} \quad b_l = \frac{2}{N_{HROIBT}} \sum_{n=1}^{N_{HROIBT}} p(z_n) \times \cos(2\pi l \times n / N_{HROIBT}). \tag{4.2}$$

Mean (D7): the average intensity [GW07].

$$D7 = \sum_{n=1}^{N_{HROIBT}} z_n p(z_n). \tag{4.3}$$

Moments (D8~D12): the five (second to sixth order) moments [GW07].

$$Dl = \sum_{n=1}^{N_{HROIBT}} z_i^q p(z_i). \quad q = 2,...,6 \quad \text{and} \quad l = 8....,12 \tag{4.4}$$

Entropy (D13): the homogeneity of the histogram distribution [GW07].

$$D13 = -\sum_{n=1}^{N_{HROIBT}} p(z_n) \log_2 p(z_n). \tag{4.5}$$

Angular Second Momentum (D14): the uniformity [GW07].

$$D14 = \sum_{n=1}^{N_{HROIBT}} p^2(z_n). \tag{4.6}$$

Peak Density (D15): the strength of the local dominant peak [GW07].

$$D15 = \max_{N_{HROIBT}} \{p(z_n)\}. \tag{4.7}$$

Range of the Densities (D16): the range of the intensity [GW07].

$$D16 = \text{range}_{N_{HROIBT}} \{p(z_n)\}. \tag{4.8}$$

Median Density (D17): the median value of the intensity [GW07].

$$D17 = \text{median}_{N_{HROIBT}} \{p(z_n)\}. \tag{4.9}$$

Here, z_n indicates the intensity, $p(z_n)$ is the histogram of the intensity levels, N_{HROIBT} is the total number of intensity levels within the HROIBT, and n is the index of N_{HROIBT}.

4.1.4 Experimental Results

A collection of 500 microscopic images is selected from the CellsDB database [KM07] at random for the training (with 300 images) and evaluation (with 200 images). Using a total of 17 basic features as the input, the HROIBT-GP-EM training process is evolved in the medium population size of 32 with the maximum tree depth of 5 in order to avoid any unnecessary computation. the GP terminal set consists of 17 basic features.

4.1.4.1 Convergence Results

After evolving 350 iterations, the best resulting feature synthesized by GP-EM based on HROIBT texture is given by:

$$\mathbf{Hf} = \cos\{-[\cos(D13 - D16) - (D3 - 0.9961)] + |D0 \times D5 \times (D2 - D12)| \times \frac{e^{(D2 \times D14)} + e^{(-D8/0.9961)}}{(D7 - D0)/(D11 + D13)}\}$$

where Dl, l = 0,…, 16, represents the (l+1)th basic feature; Figure 4.6 shows the resulting **Hf**-processed data that are normalized in the range of (-1, 1). There are altogether 300 images from the binary classes of the healthy and sick conditions of the OPMD, with 150 images for each class.

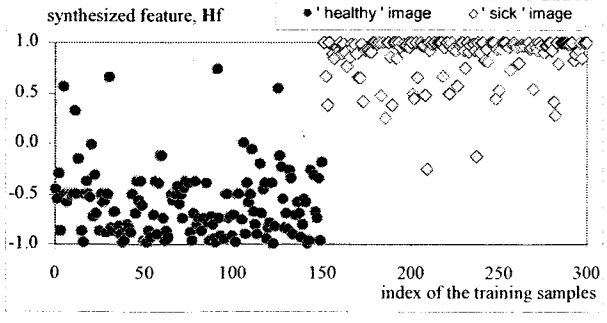


Figure 4.6. Resulting **Hf**-processed data representation after 350 iterations.

We can see from Figure 4.6 that the training data between the healthy and sick classes

are clearly separated by the resulting feature **Hf** that is synthesized without any prior knowledge about the description of the training examples.

4.1.4.2 Detection Results

Table 4.1 shows the detection results using the synthesized feature **Hf** and a certain number of basic features with the classifier K nearest neighbors (KNN) [Fu90] on the test set of 200 images. The synthesized feature **Hf** yields at best 92.5% healthy/sick detection accuracy, as compared to the accuracies of 84.5% and 88.0% obtained by using the 11 basic features, D7-D17, and the 17 basic features, D1-D17, respectively.

Table 4.1. Comparisons of the accuracy using the synthesized feature and a certain number of basic features with the classifier KNN.

Accuracy	Healthy (%)	Sick (%)	Total (%)
11 basic features, D7-D17	78.0 (78/100)	91.0 (91/100)	84.5
17 basic features, D1-D17	84.0 (84/100)	92.0 (92/100)	88.0
synthesized feature, **Hf**	90.0 (90/100)	95.0 (95/100)	92.5

4.1.5 Further Comparisons

The performance of the design systems is further validated by means of the receiver operating characteristic (ROC) [Fa06], as shown in Figure 4.7. The areas below the ROC curves indicate the discrimination capabilities of the design systems. The distance proportions in the range of the normalized features serve as the intervals; for example, the end-points of (-1.0) and (1.0) in the synthesized feature **Hf**-dimension (see Figure 4.6) produce the points (0, 0) and (1, 1) in the ROC curve.

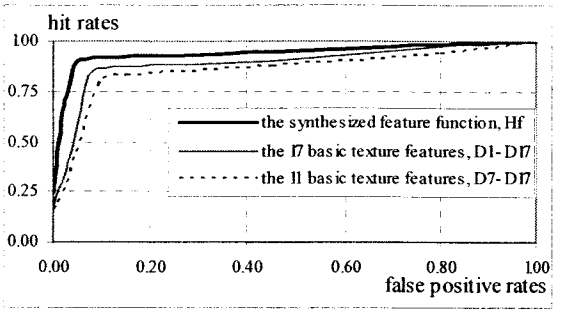


Figure 4.7. ROC analysis based on the distance proportions for the synthesized feature **Hf** (thick line), the 17 basic features (thin line), D1-D17, and the 11 basic features (dashed line), D7 - D17.

For each interval, we record the number of samples that match the classes. In Figure 4.7, the performance of **Hf** achieves a sensitivity of 0.9 at a specificity of 0.95 with an area under the ROC curve of 0.963 (thick line) whereas the areas below the curves for the performance of the 17 basic features is 0.938 (thin line) and for the performance of the 11 basic features is 0.882 (dashed line).

4.2 Detecting Breast Cancer Disease

4.2.1 Introduction

Cancer research has led to real progress in the prevention, detection, and treatment of breast cancer. Breast cancers can be classified by different schemata. Although every aspect influences treatment response and prognosis, doctors recommend that treatment is more likely to work well when breast cancer is detected early [NCBI]. Research studies worldwide are designing many types of detection systems in an attempt to achieve an earlier diagnosis [GN06], [MP06], [OL04], [BK98], [WS95].

In order to improve the detection accuracy, genetic programming was employed to generate nonlinear features based on three fitness measures: original Fisher criterion (F-

GP), alternative Fisher criterion (AF-GP) and modified Fisher criterion (MF-GP); these methods were compared with the SVM classifier which used the entire raw feature set [GN06]. Muni *et al.* [MP06] introduced an online feature selection using multitree genetic programming (GP_{mtfs}) with two modified crossover operations for multicategory detection problems. In hybrid genetic algorithms (HGAs) [OL04], chromosomes were improved by local search operations to enhance the fine-tuning capability of simple GAs in searching for the feature subsets applied to the problem of detecting the disease.

4.2.2 The Dataset of WDBC

The Wisconsin Diagnostic Breast Cancer (WDBC) dataset [BK98] were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass, which described characteristics of the cell nuclei present in the image. All feature values were recorded with four significant digits.

The WDBC contains 569 instances, divided into two cases with 357 benign and 212 malignant. The mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features.

Class distribution:	357 benign, 212 malignant
Number of instances:	569
Number of attributes:	32 (ID, diagnosis, 30 real-valued input features)

4.2.3 Experimental Results

4.2.3.1 Convergence Results

The convergence results from one of 10 runs on WDBC are displayed in Figure 4.8 in which the fitness values are normalized in the range of (0, 1). When the GP-EM system proceeds into iteration, the fitness values of the best synthesized features from the

examples of the experiments begin to increase. This indicates that the surfaces are being successfully explored and that optimization is proceeding. However, there comes a point where repeated applications of the operations yield little or no effect on the fitness values associated with the most elite solutions. This is a termination criterion for the evolutionary algorithms. From that point, we terminate at a maximum of 400 generations when the maximum fitness value converges to less than 0.0001 from the variance prior in each of the training datasets.

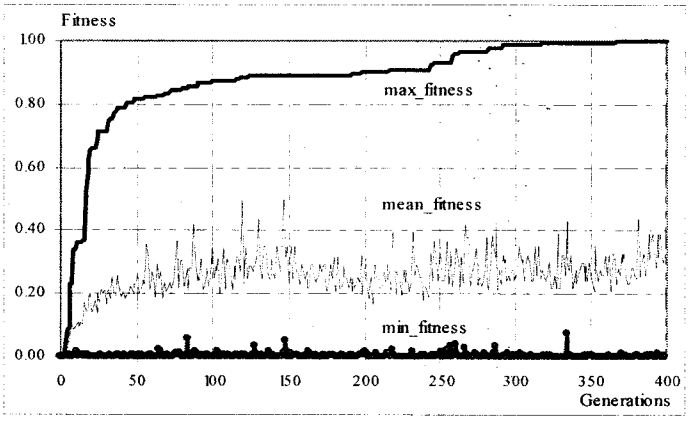


Figure 4.8. Convergence results from one of 10 runs on WDBC.

4.2.3.2 Feature Synthesized Results

The output results of the synthesized functions in this study are LISP-like expressions, composed of the signs of elementary functions and 30 features of the WDBC data, which can be written as rational expressions.

The generated feature in LISP-like expression:

$\mathbf{F_g}\|_{WDBC}$ = P(-(-(C(*(-(d22 d16)E(/(d14 d2))))E(*(*(d19 d17)*(d12 d29))))
+(/(+(d23 d25)C(*(d17 d23)))*(-d10 d17)S(*(d20 d1))))));

written as the rational expression:

$$\mathbf{Fg}\|_{WDBC} = \{\cos((d22 - d16)\times e^{d14/d2}) - e^{(d19\times d17\times d12\times d29)} - [(d23+d25)/\cos(d17\times d23)+(d10-d17)\times\sin(d20\times d1)]\}^2$$

Figure 4.9. The feature synthesized results for a single run on WDBC.

Figure 4.9 shows the best of the synthesized functions for a single experiment on WDBC. The signs of 'S', 'C', 'E' and 'P' represent sine, cosine, exponential and square functions and d*i* (i = 0,…, 29) refers to the (i+1)th primitive feature of the WDBC data.

4.2.3.3 Training Data Representation Results

Figure 4.10 illustrates the behavior of the training samples, 313 benign and 199 malignant, on WDBC from an example of a GP-EM learning experiment. Modeled as the Gaussian mixture with the common variance σ^2, the generated data $\hat{\mathbf{G}}$ is normalized in the range of (-1, 1) along the 1-dimensional $\mathbf{Fg}\|_{\text{WDBC}}$-space in certain selected generations.

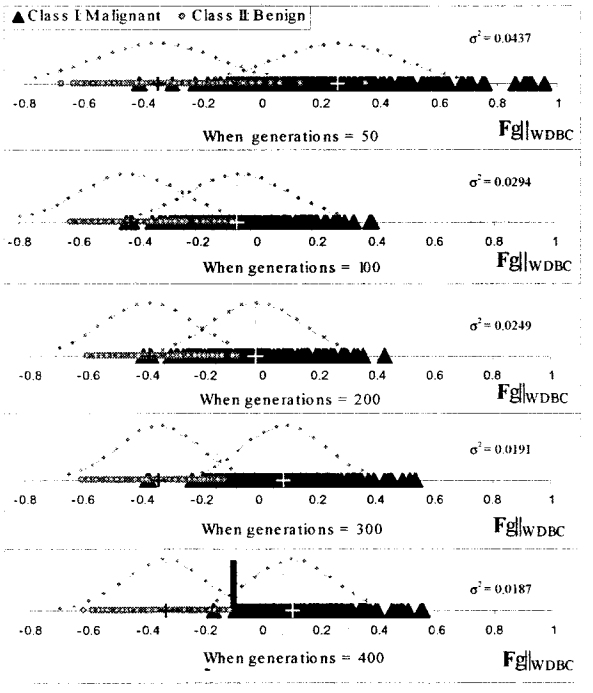


Figure 4.10. The GP-EM training data representation results from the example of the 10 runs on WDBC.

Initialized with unit variance, the variance σ^2 is calculated experimentally in each generation to illustrate the evolutionary learning process. In Figure 4.10, the data clouds along the $\mathbf{Fg}\|_{\text{WDBC}}$-space persist in moving to symmetrically surround their centers (the

signs of '+') that define the means of the Gaussian distributions through the generations. The bulk of the improvement occurs before the first 100 generations in which the difference of the variance σ^2 (e.g. 0.0143 decrements between the 50th ~ 100th generations) is larger than those decrements in each subsequent 100 generation interval. This fact is clear in the responses to the fitness curve on WDBC in Figure 4.10 where a big jump in performance occurs during the first 100 learning cycles.

When reaching 400 generations, we can see that the interface, as indicated by the solid vertical line in Figure 4.10, between the class data sets is better defined. Clearly, the overlap is smaller and the cluster centers are decoupled in a manner that fits well with the modular structure of the Gaussian mixture in the GP-EM training process.

4.2.3.4 Detection Results

The detection accuracy (DA) rates of the training accuracy from the training sets and the target recognition accuracy from the test sets over 10 runs are presented in Figure 4.11; the statistic analysis for 10 runs on the combined training and test sets is illustrated in Table 4.2. The classifier MDC is employed to evaluate the performance of the GP-EM method on each given object vector of their training and test sets.

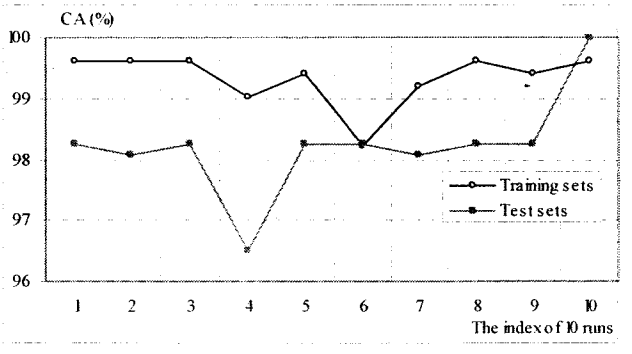


Figure 4.11. Detection accuracy (DA) of the combined training sets and test sets against the index of 10 independent runs on WDBC.

We achieve the average training score of 99.32% accuracy. With only one synthesized feature, the average performance over 10 runs on WDMC shows that 98.21% of the test data can be classified correctly.

Table 4.2. The detection accuracy (DA) over 10 runs on WDBC.

Statistics	Maximum	Minimum	Average	Std
Training sets (CA%)	99.61	98.23	99.32	0.437
Test sets (CA%)	100	96.49	98.21	1.230

4.2.4 Further Comparisons

To determine whether the proposed GP-EM system is potentially a good automatic feature generator, we further compare it with the single GP-based feature generators and some other well-known object detection algorithms reported in the literature on the WDBC dataset [GN06], [MP06], [OL04].

Researchers have worked on WDBC [BK98] for detecting breast cancer disease using the method of the feature synthesis based on GP [GN06], and feature selection algorithms based on GP and hybrid GAs [MP06], [OL04]. In the following sections of the comparisons, the average CA rates, including the GP-EM study, are all recorded over 10 runs, except for the rates recorded by [OL04] over 5 runs.

4.2.4.1 Comparisons with the Single GP-Based Feature Generators

Using one synthesized feature as the input to the same classifier MDC, the average CA rates are reported in Table 4.3 for our GP-EM feature generator and are compared to the performances achieved by the single GP-based feature generators [GN06]. The results were all documented on the 10-folds of cross validation method using each algorithm. In

accuracy comparisons, the standard deviation (Std) is needed to determine whether any difference in accuracy is significant.

Table 4.3. Comparisons of the GP-EM approach and the single GP-Based feature generators on WDBC.

Algorithms	Average CA (%)	No. of features	Std. (%)
F-GP [GN06]	97.40	1	1.60
AF-GP [GN06]	97.36	1	1.39
MF-GP [GN06]	97.47	1	1.56
GP-EM (this study)	98.21	1	1.23

Notably, for the test CA, the GP-EM algorithm achieves a 98.21% average performance rate, the highest of all the single GP-based feature generators. It also yields the lowest standard deviation of 1.23% on detection, suggesting that the hybrid GP-EM algorithm correctly labeled target objects with a high level of consistence and confidence, which is important in real-time applications.

4.2.4.2 Comparisons with Other Detection Systems

In Table 4.4, we show the comparison results with some other published recognition systems [GN06], [MP06], [OL04] on WDBC. As shown in Table 4.4, the difference of the features in the range of (1, 30) used by the algorithms is particularly large, compared with the performances ranged (94.27%, 98.21%). Significantly, the GP-EM approach achieves the highest average test score of 98.21% accuracy by using the synthesized feature $\mathbf{Fg}\|_{\text{WDBC}}$, compared with the recognition systems [GN06], [MP06], [OL04] using multiple or all 30 primitive features.

Table 4.4. Comparisons of the proposed GP-EM Algorithm and other object detection systems on WDBC. The number of features is the mean of the features used in detection. Note: the run time is the average computation time (hours: minutes: seconds)

Algorithms	Average DA (%)	# Features	Run time
SVM [GN06]	96.32	30	~
HGAs [OL04]	94.27	12	~
GP_{mtfs} [2]	96.31	6.72	0: 04: 10
GP-EM (this study)	98.21	1	0: 06: 40

4.3 Supporting Identification in a Hand-Based Biometric System

4.3.1 Introduction

Biometrics-based identification is a verification approach using biological features in each individual. Hand features have been widely used in designing a biometric identification system and the challenge has been established [ZW07], [ZK03], [KS05].

In object detection problems, the analytical selection of features and the automatic synthesis of features provide two distinct approaches, because they rely on different sources of information [RZ02]. It would be useful to explore new biometric identification systems that combine both methods of the feature selection and synthesis.

The method of cooperative coevolutionary clustering algorithm (CCCA) on a hand image dataset can be categorized as the unsupervised feature selection for clustering hand images [Gu03], [KS05]. The main tool for accomplishing this was the GA. The CCCA was designed to search for a proper number (without prior knowledge of it) of clusters of hand images, and simultaneously to achieve the goal of feature selection.

4.3.2 Previous Feature Selection Work on the Hand Image Dataset

The hand image dataset consists of 1000 hand images, from which about 83% showed 'good' contour output, see a sample of a hand image in Figure 4.12. The problems related to the rest of hand images were predominantly the results of poor image enrolment system [KS05]. Therefore, positive identification, based on this hand image dataset, is a challenging biometric identification problem in its own right.

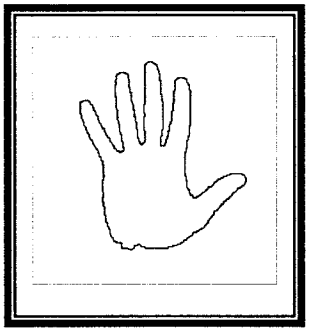


Figure 4.12. A sample of a hand image.

In the previous work [Gu03], [KS05], the CCCA method attempted to find classes of hand image objects with similar properties. The CCCA was implemented using 100 hand images as the test set. The experimental results showed that the dimensionality of the clustering space was reduced from 84 original biometric features to 41 selected features (11 geometric and 30 statistical features), with 4 clusters produced. At the end, the output clusters were labeled with the number of input hand image objects per class, assigned to each cluster.

4.3.3 The GP-EM-MSE Supporting Biometric Identification System

The pre-requisite for the technique of the feature synthesis is the preparation of primitive feature sets. Using our previous method in [Gu03], [KS05] on a dataset of 200 hand

images, we cluster the 200 images into 4 categories, with a total number of 41 features selected as the primitive feature set for the current study of the feature synthesis.

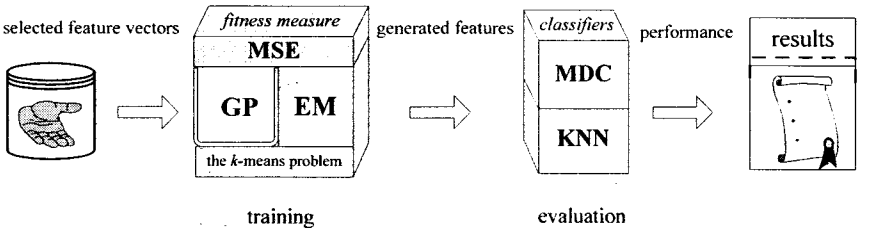


Figure 4.13. The GP-EM-MSE hand-based biometric identification system.

Figure 4.13 presents the method of the feature synthesis by GP-EM with the MSE fitness indicator (GP-EM-MSE). As an extension of the method of CCCA for the feature selection in [Gu03], [KS05], the purpose of GP-EM-MSE in this work is to find the improved feature representations in an optimal control environment in order to further minimize identification errors for the hand-based biometric system.

The previous work of the CCCA method [Gu03], [KS05] was evolved without supervision in such a way that the selected feature sets operated globally to cluster hand images. Consequently, the current study of GP-EM-MSE for the feature synthesis is a supervised learning algorithm, via the results of CCCA.

4.3.3.1 Mean Square Error Fitness (MSE)

The MSE measure is well known in the function approximation and learning system theories when the number of classes is assumed to be a prior known. In this supervised problem of the feature synthesis, we employ the MSE measure as the fitness which is given by [Le02]:

$$\text{the MSE fitness} = \sum_{j=1}^{k}\sum_{i=1}^{m} D(c_j, r_i^j), \qquad (4.10)$$

where D is the Euclidean distance between the instance r_i and its mean center c_j.

4.3.4 Experimental Results

In the implementation, we divided the 200 images into the training and testing sets, each with 100 images. Using the 41 primitive features as the input, the GP-EM-MSE training is evolved with the maximum tree depth of 5 and the population size of 32.

4.3.4.1 Convergence Results

We ended the GP-EM-MSE training after running 400 iterations. The total time for computation was about three minutes on a Pentium 4 at 1.60 GHz. The results for the two features, P_feature 1 and P_feature 2, produced by GP-EM-MSE are as follows:

$$\begin{aligned} \text{P\_feature 1} = \cos\{&\sin[h0 \times h39 \times tg(h22 \times h21)] \times tg[|\, h32 \times h0\,| + h39 \times h27] \\ &+ |\sin[(h30/h7)^2 - e^{h2 \times h19}] - (h12 \times h31 + h14 \times h26)^2\,|\} \end{aligned} \qquad \textit{and}$$

$$\text{P\_feature 2} = \cos\left(\frac{\cos[-(h32 + h25) \times (h9 + h28)]}{\left| \frac{h25}{h23} + \cos\left(\frac{h9}{h8}\right) \right| \times \sqrt{\frac{h8}{h22 \times \sin(h25 + h32)} + \frac{h10}{h8} + \cos(h35 \times h20)}} \right),$$

where hd is the (d+1)th feature of 41 primitive features, and the convergence results are presented in Figure 4.14; in the sequel, the resulting synthesized features will be employed to identify hand images.

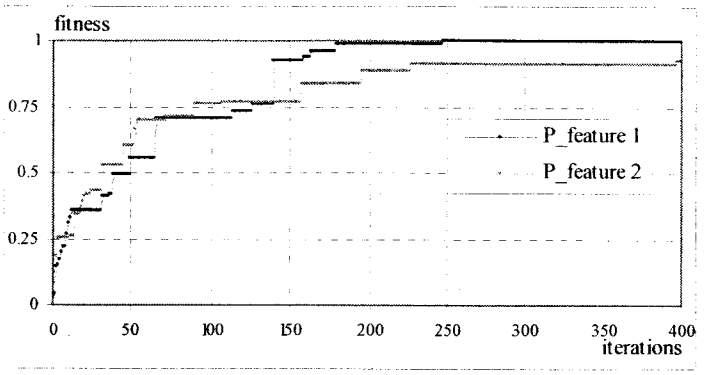


Figure 4.14. Convergence results for the features produced by GP-EM-MSE.

4.3.4.2 Detection Results

The goal of the identification is to classify hand images into a known number of 4 categories or classes according to the decision space of the synthesized features. With the classifier MDC, Table 4.5 shows the confusion matrix of the identification performance for four classes on the test set, using two features, P_feature 1 and P_feature 2, produced by GP-EM-MSE. Each row of the table represents the identification performance for a given category, while each column represents a percentage of the number of samples in an actual class. It can be observed, from Table 4.5, that the difference of the related identification accuracy among the classes is small, ranging from 95.00 % to 96.96 %.

Table 4.5. Identification accuracy (%) using two features, P_feature 1 and P_feature 2, produced by GP-EM-MSE with MDC.

categories	class I	class II	class III	class IV
class I (33 samples)	96.96	3.04	0	0
class II (25 samples)	0	96.00	0	4.00
class III (22 samples)	4.55	0	95.45	0
class IV (20 samples)	0	5.00	0	95.00

4.3.5 Further Comparisons

The classifiers MDC and KNN are utilized in order to assess the capability of different feature sets over different detection systems. In terms of the detection accuracy, Table 4.6 shows the comparison results between the features produced by GP-EM-MSE with MDC

and 41 primitive features selected by the method [Gu03],[KS05] with KNN in which the value of K is tested by the input of 41 primitive features (K = 41).

Table 4.6. The comparision between the features produced by GP-EM-MSE / MDC and 41 primitive features selected by [Gu03],[KS05] / KNN, in terms of the detection accuracy (%).

feature types / classifier	no. of features	accuracy (%)
the primitive features selected by [Gu03], [KS05] / KNN	41	93.0
P_feature 1 / MDC	1	92.0
P_feature 1, P_feature 2 / MDC	2	96.0

It can be seen from Table 4.6 that the detection accuracy using one feature, P_feature 1, is slightly lower than when using 41 primitive features. However, the combination of two features, P_feature 1 and P_feature 2, produced by GP-EM-MSE achieves the best detection performance with an accuracy rate at 96.0%.

4.4 Detecting Parkinson's Disease

4.4.1 Introduction

Symptoms of Parkinson's disease (PD) include muscle rigidity, tremors, and change in speech and gait. The causes of PD are currently unknown. There is no cure for PD and the prognosis depends on the patient's age and symptoms [ET03]. Research has shown that the typical Parkinsonian movement disorders (i.e. tremor at rest, rigidity, akynesia and postural instability) considerably reduce when medication is available offering clinical intervention to alleviate symptoms at the onset of the illness [RC07]. To this aim, studies in medical biometrics on detecting PD in the early stage are under way and have drawn a lot of attention from the biometrics community in recent years [LM98], [RG09], [CS08], [SS07]. Little *et. al* [LM98] applied a feature pre-selection filter that removed redundant

measures, and then using an SVM classifier, an exhaustive search was executed by testing all possible subsets of features in order to discriminate healthy from disordered PD subjects. In [RG09], features sets were analyzed using the rough set approach that mapped feature vectors associated with objects onto the medical decision support system for detecting the PD objects.

4.4.2 The Dataset of OPDD

The OPDD dataset contained 195 biomedical voice samples recorded from 31 people, in which 23 were diagnosed with PD. Each datum is 22-dimensional and consisted of certain types of voices ranging from 1 to 36 seconds in length, listed in Table 4.7. The dataset is divided into two classes according to its "status" column which is set to 0 for healthy subjects and 1 for those with PD. The data set used for this implementation are described in detail in [LM09], and at the UCI website [BK98].

Table 4.7 Descriptions of the features of the dataset OPDD.

No	Features	Descriptions
1-3	MDVP (Hz)	ave./max. /min. vocal fundamental frequency
4	MDVP (%)	MDVP jitter as a percentage
5	MDVP (abs)	MDVP absolute jitter in microseconds
6-8	RAP/ PPQ/ DDP	3 measures of variation in frequency
9-10	shimmer / (dB)	2 MDVP local shimmer
11-12	shimmer: APQ	3- /5- point Amplitude Perturbation Quotient
13	MDVP: APQ	11- point Amplitude Perturbation Quotient
14	shimmer: DDA	a measure of variation in amplitude
15-16	NHR / HNR	2 ratios of the noise to tonal components
17-18	RPDE / D2	2 nonlinear dynamical complexity measures
19	DFA	signal fractal scaling exponent
20-21	spread 1 - 2	2 measures of frequency variation
22	PPE	pitch period entropy

4.4.3 Experimental Results

Using the classifier MDC, we employ the 10-fold cross validation approach to assess the capabilities of the GP-EM detector for the implementation of Oxford Parkinson's Disease Database (OPDD) [BK98].

4.4.3.1 Convergence Results

In the evolutionary explorations, the specification of a termination criterion is usually required to terminate runs. In this application, we ended the evolutionary GP-EM training process when a certain number of iterations are reached, as the result of a run. After running 350 iterations, the resulting created feature function from one of the 10 runs is given by:

$$\begin{aligned}\mathbf{Pa}_{\_\text{detection}} = \cos\{\cos\{-[(f4-f5)/(f4+f16)^2]\times \text{tg}[f2+f7-(f5\times f18)^2]\}\\ \times[f6+f3-(f18/f21)+\text{tg}(f15^2)/(f4-f0)]\},\end{aligned}$$

where fd, d = 0,..., 21, is the (d+1)th ordinary feature of OPDD listed in Table 4.7; normalized in the range of (0, 1), its fitness values is presented in Figure 4.15.

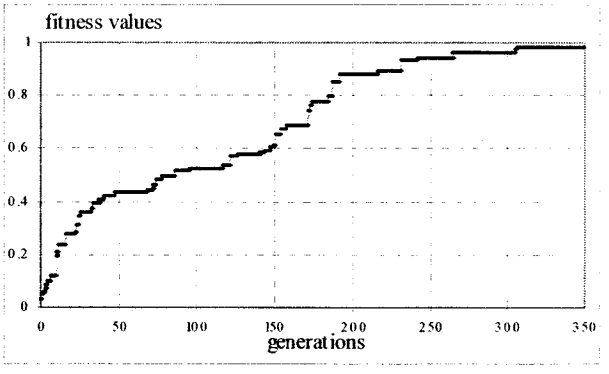


Figure 4.15. The convergence results for the created feature function, $\mathbf{Pa}_{\_\text{detection}}$.

4.4.3.2 Detection Results

In the detection experiments, the resulting feature function, **Pa**_detection, is employed as the medical decision space to diagnose PD. By using the classifier MDC, the detection performance for 10 runs on the combined training and test sets is illustrated in Table 4.8. We achieve the average training score of 95.06 % detection accuracy (DA); with only one created function **Pa**_detection, the average performance over 10 runs shows that 93.12 % of the test data can be classified correctly.

Table 4.8. Detection accuracy (DA) for 10 runs on OPDD.

Performance	The training (DA)	The test (DA)
maximum (%)	97.71	95.00
minimum (%)	91.42	80.00
average (%)	95.06	93.12
Std (%)	1.42	2.86

4.4.4 Further Comparisons

To assess whether a single feature function created by the GP-EM detector would be sufficient for the identification of the Parkinsonian subjects, we compare our algorithm with other methods existing in the literature on OPDD [BK98]. The comparison works with [LM98], [RG09] are given in Table 4.9.

Table 4.9. Comparisions of the GP-EM detector and other methods in terms of the detection accuracy (DA) on OPDD; note: the sign '~' means 'not specified'.

Methods	DA (%)	No. of features	Time
rough set [RG09]	100	22	~
feature preselection with an exhaustive search [LM98]	91.40	10	~
the GP-EM detector	93.12	1	5 minutes

Little *et. al* [LM98] applied exhaustive search by testing all possible subsets of features; the subset, consisted of 10 ordinary features, was thus selected that produced the

best detection performance of 91.4%. Using all of 22 ordinary features, the approach using the rough set approach [RG09] yielded the highest recognition rate of 100% from one example of 10 runs.

As shown in Table 4.9, by using the single created function, **P**a_detection, the GP-EM detector achieves the average detection accuracy of 93.12% over 10 runs, indicating the advantages of the reliability and efficiency for the discrimination of healthy subjects from those with PD.

Chapter 5

Conclusions

5.1 Discussion and Conclusions

The performance of automated feature synthesis depends upon the designed approaches. The method should allow the system to synthesize the feature functions in an optimal control environment without any or with minimal user interaction. We use the EM algorithm involved in an evolutionary GP to simultaneously achieve the dynamic feature synthesis and the model-based generalization without user influence for the object detection. In the GP-EM systems, the process is driven by a fitness measure, the J value, accounting for the inter-class separability and intra-class homogeneity. As the generalization of the k-means problem via EM, the GP-EM approach projects the hyperspace of each dataset onto its lower dimensional synthesized feature space with the competitive correct recognition rates.

One function of the EM algorithm involved in this study is its ability to perform the optimization of parameters $\theta = \{\pi, \mu, \Sigma\}$. Direct optimization of the multivariate Gaussian mixture $P(\mathbf{G}/\pi, \mu, \Sigma)$ is difficult, as the covariance matrix Σ cannot be inverted. However, the optimization of the univariate Gaussian mixture is significantly easier when we assume the independence among all the k classes. In the approach, the synthesized feature functions are independent of each other, and the generated data $\mathbf{G}$ is mapped onto

lower dimensional feature spaces in which the covariance matrix contains only one component of the variance. Nevertheless, this is one reason why we could reduce the feature dimensions from the hyperspace of primitive features to a lower synthesized feature space, while improving the detection accuracy; the results consistently validate the initial assumption of reducing the covariance matrix to an identical variance.

The search space in GP-EM is the space of all possible computer programs composed of the designing mathematical elementary functions and the primitive features. A pre-condition for solving the problem with GP in this study is that the sets of functions (elementary functions) and terminals (the primitive features) satisfy the requirement in a conceptually straightforward way. Together they are capable of expressing a solution to the problem of the nonlinear feature function synthesis. In addition, each complete EM cycle, the "*E*-step" and the "*M*-step", of the GP-EM approach turns out to be simple to implement by performing a complete, rather than partial, optimization of the means of the k Gaussians.

5.2 Summary of Main Contributions

The main contributions of the thesis are:

- When a designed approach has a streamlined hierarchy and a parallel processing structure based on multifaceted components, the capacity of data processing will increase.

  Unlike single GP-based feature synthesis methods existing in the literature, the hybrid GP-EM approach is composed of many different components that apply to the problem of feature synthesis. This flexibility in GP-EM allows the system to have

multifaceted components, i.e., some components dynamically synthesize features from the reduced data flow; some will combine the features to perform data modeling under the Gaussian mixture; and others are going to act as a device for accelerating the convergence, just as a multifunctional machine would solve the different tasks associated with specific components of the complete system.

For example, acting as an acceleration device to help GP, the EM with the operation of ML keep the generated values near their means, see equation (3.7), thereby the convergence of GP-EM is assured.

- The innovative histogram-based technique of HROIBT helps to constrain the problem space within the region of the color information, i. e., the lighter INIs is indicative of the OPMD, regardless of image sizes, shapes and background structures. A combination of methods based on the bin-threshold HROIBT and feature-synthesis GP-EM, as a whole, not only shows the applicability to irregular images, but also exhibits a good detection strategy for PCD in the development of medical decision systems.
- Since many practical problems have a class-conditional density that is approximately Gaussian, we assume that the generated data with the k known classes are modeled under the mixture of the k Gaussians, from which the task of feature synthesis, via EM, results in searching for the optimal means of the k Gaussians only. In other words, it dramatically changes the task into a simpler k-means problem; generally, the number of Gaussian distributions, k, is much less than the total number of data points, m, and therefore the hybrid GP-EM approach exhibits the computational efficiency of the technique.

5.3 Other Possible Applications

Due to the wide ranges of examples that fall under the umbrella of the Gaussian distribution, a number of problem domains are good candidates for successful applications, including electronic systems for monitoring and controlling machine faults [KT07], bioinformatics for predicting protein-protein interactions [NCBI], telecommunication for detecting fraud cases in handling both voice and data type of data record [MB09]. In addition, finding more sophisticated circumstances in engineering domains that make use of the approach is an interesting direction for future works. Next, an example of a potential application for monitoring the power quality is described in the following subsection.

5.3.1 Power Quality Monitoring System

In recent years, concern over the quality of electrical power in manufacture industries has been increasing rapidly since poor electrical power quality causes many problems for the affected loads, such as malfunctions, instabilities, short life time and so on. Poor quality of electrical power is normally caused by power line disturbances, such as impulses, notches, glitches, momentary interruptions, wave-faults, over-voltages, under-voltages, and harmonic distortion [KT07].

In order to determine the causes and sources of disturbances, we must have the capability to detect those disturbances and further to improve electrical power quality. To this aim, the procedures for preparing the primitive feature set for a power monitoring system is given in Figure 5.1. In the following subsection we explain the steps in detail.

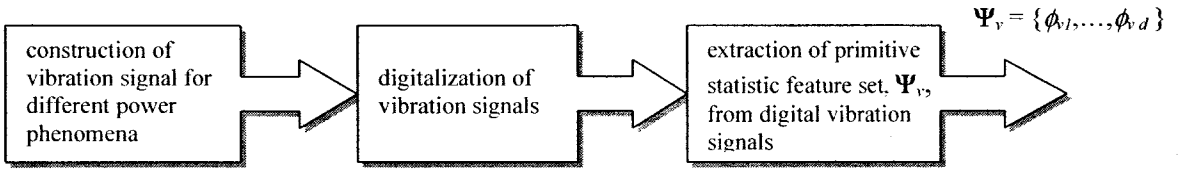
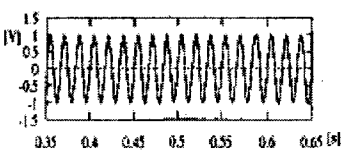


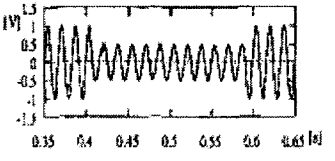
Figure 5.1. Procedures for preparing primitive features for a power quality monitoring system.

5.3.1.1 Primitive Feature Extraction from Vibration Signals

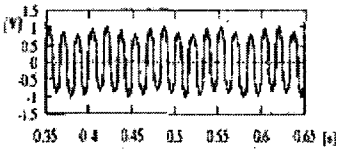
Figure 5.2 illustrates phenomena for the power quality problem. It is known that vibration signals depend mainly on the resonant frequencies of different parts of the machine. If the machine condition varies due to wear or damage, the resonant frequencies and the vibrations will change. It is generally not possible to classify the condition based upon an individual sample of the vibration, thus, some transformation of the recorded vibration time-series is required to extract time invariant features [KT07].



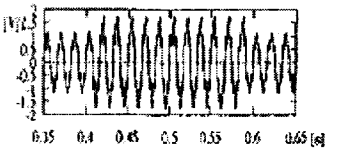
(a) A pure sine wave



(b) A sag distorted signal



(c) A swell distorted signal



(d) A harmonic distorted signal

Figure 5.2. Phenomena for the power quality problem; the horizontal axis presents the time in second and the vertical axis presents the magnitude in its four power conditions.

For example, in the case of machine condition deteriorates, the energy (mean square value) in vibration signals is expected to increase. A number of different statistical features (prepared for automated synthesizing nonlinear feature functions) can be extracted from the vibration data by using statistical measures and cumulants, such as central moments, Fourier descriptors and Zernike moments.

5.4 Refereed Publications Based on the Research

P.-F. Guo, P. Bhattacharya and N. Kharma, "Advances in detecting Parkinson's disease," in *Internat. Conf. Med. Biometrics* (ICMB 2010), Hong Kong, July, 2010. Proceedings appeared in the series *Lecture Notes in Computer Science* (LNCS), Springer, vol. no, 6165, pp. 306-314 (2010).

P.-F. Guo, P. Bhattacharya and N. Kharma, "On supporting identification in a hand-based biometric framework," in *Internat. Conf. Image Signal Process.* (ICISP 2010), Quebec, June, 2010. Proceedings appeared in the series *Lecture Notes in Computer Science* (LNCS), Springer, vol. no, 6134, pp. 210-217 (2010).

P.-F. Guo, P. Bhattacharya and N. Kharma, "Automated synthesis of feature functions for pattern detection," in *IEEE Canadian Conf. Electr. Comput. Eng.* (CCECE 2010), Calgary, pp.105-108, May, 2010.

P.-F. Guo, P. Bhattacharya and N. Kharma, "An efficient image pattern recognition system using an evolutionary search strategy," in *IEEE Internat. Conf. Syst., Man, Cybernetics* (IEEE SMC 2009), San Antonio, pp. 605-610, October, 2009.

P.-F. Guo and P. Bhattacharya, "An evolutionary approach to feature function generation in application to biomedical image patterns," *Internat. Conf. Genetic Evol. Comput.* (GECCO 2009), Montreal, pp. 1883-1884, July, 2009.

5.5 Final Remarks and Future Work

For multiclass problems, finding the appropriate boundary values, which are mostly driven by fitness measures, is difficult in evolutionary computation. The proposed GP-EM algorithm offers the framework to perform the pattern classification and visual training data representations to improve the pattern discovery and decision support in computational evolutionary processes. Our future work will be dedicated to assessing further the practical components of the proposed algorithm.

Bibliography

[AB09] S. Arora and B. Barak, *Computational Complexity: A Modern Approach.* Cambridge, New York, 2009.

[BB09] S. C. Blumen, J. -P. Bouchard, B. Brais, R. L. Carasso, D. Paleacu, V. E. Drory, S. Chantal, N. Blumen and I. Braverman, "Cognitive impairment and reduced life span of oculopharyngeal muscular dystrophy homozygotes," *J. Neurology*, pp. 596-601, 2009.

[Bi06] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, New York, 2006.

[BK98] C. Blake, E. Keogh and C. J. Merz. (1998) UCI repository of machine learning databases, Dept. Inform. Comput. Sci., Univ. California, Irvine, CA. [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

[CP10] E. O. Costa, A. T. R. Pozo, and S. R. Ve, "A genetic programming approach for software reliability modeling," *IEEE Trans. Reliability*, vol. 59, pp. 222–230, 2010.

[CS08] L. Cnockaert, J. Schoentgen, P. Auzou, C. Ozsancak, L. Defebvre, and F. Grenez, "Low-frequency vocal modulations in vowels produced by Parkinsonian subjects," *Speech. Commun.*, vol. 50, pp. 288–300, 2008.

[CT01] P. Chen, T. Toyota and Z. He, "Automated function generation of symptom parameters and application to fault diagnosis of machinery under variable operating conditions," *IEEE Trans. Syst., Man, Cybern., Part A,* vol. 31, pp. 775-781, 2001.

[CU10] N.D. Chatzidiamantis, M. Uysal and T.A. Tsiftsis, "Iterative Near Maximum-Likelihood Sequence Detection for MIMO Optical Wireless Systems," *J. Lightwave Technology*, vol. 28, pp. 1064–1070, 2010.

[CW06] V. Ciesielski, G. Wijesinghe, A. Innes and S. John, "Analysis of the superiority of parameter optimization over genetic programming for a difficult object detection problem," *IEEE Congress on Evol. Comput*, pp. 1264-1271, 2006.

[DH01] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification.* Wiley, New York, 2001, 2nd. ed.

[ET03] S.K.V.D. Eeden, C.M. Tanner, A. L. Bernstein, R.D. Fross, A. Leimpeter, D. A. Bloch, and L. M. Nelson, "Incidence of Parkinson's disease: Variation by age, gender, and race/ethnicity," *Amer. J. Epidemiol.*, vol. 157, pp. 1015–1022, 2003.

[ES03] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing.* Springer, New York, 2003.

[FO66] L. J. Fogel, A. J. Owens and M. J. Walsh, *Artificial Intelligence Through Simulated Evolution.* Wiley, New York, 1966.

[Fa06] T. Fawcett, "An introduction to ROC analysis," *Pattern Recog. Lett.*, vol. 27, pp. 861-874, 2006.

[Fu90] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic, Boston, 1990, 2nd ed.

[GJ05] H. Guo, L. B. Jack, and A. K. Nandi, "Feature generation using genetic programming with application to fault classification," *IEEE Trans. Syst., Man, Cybern., Part B,* vol. 35, pp. 89-99, 2005.

[GK96] P. D. Gader and M. A, Khabou, "Automatic feature generation for handwritten digit recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, pp. 1256-1261, 1996.

[GN06] H. Guo and A. K. Nandi, "Breast cancer diagnosis using genetic programming generated feature," *Pattern Recog.*, vol. 39, pp. 980-987, 2006.

[Go89] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Wesley, New York, 1989.

[GS99] A.M. Gaouda, M.M.A. Salaina, M.K. Sultan and A.Y. Cliildiaoi, "Power quality detection and classification," *IEEE Trans. on Power Delivery*, vol. 14, pp. 1469-1476, 1999.

[Gu03] P. -F. Guo, "Palmprints: a cooperative co-evolutionary clustering algorithm for hand-based biometric identification." M.A.Sc. thesis, Concordia University, Montreal, Canada, 2003.

[GÜ05] I. Güler and E.D. Übeyli, "A mixture of experts network structure for modeling Doppler ultrasound blood flow signals," *Computers in Biology and Medicine*, vol. 35, pp. 565-582, 2005.

[GW07] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, MA, 3rd ed., 2007.

[HG02] S. E. Hussein and M. H. Granat, "Intention detection using a neuro-fuzzy EMG classifier", *IEEE Engineering in Medicine and Biology*, vol. 2, pp. 123-129, 2002.

[HY09] S. Huda, J. Yearwood and R. Togneri, "A constraint-based evolutionary learning approach to the expectation maximization for optimal estimation of the hidden

Markov model for speech signal modeling," *IEEE Trans. Syst., Man, Cybern., Part B,* vol. 39, pp. 182-197, 2009.

[JH06] S.Chang, H. S. Hou and Y. K. Su, "Automated passive filter synthesis using a novel tree representation and genetic programming," *IEEE Trans. Evol. Comput.*, vol. 10, pp. 93-100, 2006.

[KB05] K. Krawiec and B. Bhanu, "Visual learning by coevolutionary feature Synthesis," *IEEE Trans. Syst., Man, Cybern., Part B*, vol. 35, pp. 405-425, 2005.

[KL07] T. M. Khoshgoftaar and Y. Liu, "A multi-objective software quality classification model using genetic programming," *IEEE Trans. Reliability*, vol. 56, pp. 237-245, 2007.

[KM07] N. Kharma, H. Moghnieh, J. Yao, Y. Guo, A. Abu-Baker, J. Laganiere, G. Rouleau and M. Cheriet, "Automatic segmentation of cells from microscopic imagery using ellipse detection," *IET Image Processing*, vol. 1, pp. 39-47, 2007.

[Ko94] J. R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Pr., Cambridge, MA, 1994.

[KO99] M. Kotani, S. Ozawa, M. Nakai and K. Akazawa, "Emergence of feature function using genetic programming", in *Proc. 3rd Int. Conf. Knowledge-Based Intelli. Inform. Eng. Sys.*,pp. 149-152, 1999.

[KS05] N. Kharma, C. Y. Suen and P.-F. Guo, "Palmprints: a cooperative co-evolutionary algorithm for clustering hand images," *Internat. J. Image Graphics*, vol. 5, pp. 595-616, 2005.

[KS03] N. Kharma, C. Y. Suen and P.-F. Guo, "PalmPrints: a novel co-evolutionary algorithm for clustering finger images," *Internat. Conf. Genetic Evol. Comput.*

(GECCO 2003), Chicago, July, 2003. Proceedings appeared in the series *Lecture Notes in Computer Science* (LNCS), Springer, vol. no, 2723, pp. 322-331 (2003).

[KT07] A. Kusko and M. T. Thompson, *Power Quality In Electrical Systems*. McGraw-Hill, New York, 2007.

[Le02] C.-Y. Lee, "Efficient automatic engineering design synthesis via evolutionary exploration." Ph.D thesis, California Institute of Technology, Pasadena, California, 2002.

[LM09] M. A. Little, P. E. McSharry, E. J. Hunter, J. Spielman and L. O. Ramig, "Suitability of dysphonia measurements for telemonitoring of Parkinson's disease," *IEEE Trans. Biomedical Engineering*, vol. 56, pp. 1015-1022, 2009.

[Lo07] H. S. Lopes, "Genetic programming for epileptic pattern recognition in electroencephalographic signals," *Applied Soft Computing*, vol. 7, pp. 343-352, 2007.

[LR05] S. M. Lucas and T. J. Reynolds, "Learning deterministic finite automata with a smart state labeling evolutionary algorithm," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, pp. 1063-1074, 2005.

[Lu06] W. Luo, "Penalized minimum matching distance-guided EM algorithm," *Int. J. Electronics and Communi.*, vol. 60, pp. 235-239, 2006.

[MB09] A. Mohamed, A. F. M. Bandi, A. R. Tamrin, M. D. Jaafar, S. Hasan and F. Jusof, "Telecommunication fraud prediction using backpropagation neural network," *Int. Conf. Soft Comput. And Pattern Recog.*, pp. 259-264, 2009.

[MF07] C. Mattiussi and D. Floreano, "Analog genetic encoding for the evolution of circuits and networks," *IEEE Trans. Evol. Comput.*, vol. 11, pp. 596-607, 2007.

[MN09] T. Mu and A. K. Nandi, "Multiclass classification based on extended support vector data description," *IEEE Trans. Syst., Man, Cybern., Part B,* vol. 39, pp. 1206-1216, 2009.

[Mi97] T. M. Mitchell, *Machine Learning*. McGraw-Hill, New York, 1997.

[MP03] P. Mitra, S. K. Pal and M. A. Siddiqi, "Non-convex clustering using expectation maximization algorithm with rough set initialization," *Pattern Recog. Letters,* vol. 24, pp. 863-873, 2003.

[MP06] D. P. Muni, N. R. Pal and J. Das, "Genetic programming for simultaneous feature selection and classifier design," *IEEE Trans. Syst., Man, Cybern., Part B,* vol. 36, pp. 106-117, 2006.

[NCBI] The National Center for Biotechnology Information (NCBI) (1988), National Institutes of Health, Maryland. [online]. Available: http://www.ncbi.nlm.nih.gov/bookshelf/br.fcgi?book=gene&part=opmd

[NG07] C. Nikou, N. P. Galatsanos and A. C. Likas, "A class-adaptive spatially variant mixture model for image segmentation," *IEEE Trans. Image Processing.*, vol. 16, pp. 1121-1131, 2007.

[OL04] II-S. Oh, J.-S. Lee and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, pp. 1424-1437, 2004.

[PH09] T. K. Paul and H. Iba, "Prediction of cancer class with majority voting genetic programming classifier using gene expression data," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 6, pp. 353-367, 2009.

[PN95] P. Pudil, J. Novoviová, N. Choakjarernwanit and J. Kittler, "Feature selection based on the approximation of class densities by finite mixtures of special type," *Pattern Recog.*, vol. 28, pp. 1389-1398, 1995.

[RC07] D. A. Rahn, M. Chou, J. J. Jiang, and Y. Zhang, "Phonatory impairment in Parkinson's disease: Evidence from nonlinear dynamic analysis and perturbation analysis," *J. Voice*, vol. 21, pp. 64–71, 2007.

[RD03] M. L. Raymer, T. E. Doom, L. A. Kuhn and W. F. Punch, "Knowledge discovery in medical and biological datasets using a hybrid Bayes classifier/evolutionary algorithm," *IEEE Trans. Syst., Man, Cybern., Part B,* vol. 33, pp. 802-813, 2003.

[RG09] K. Revett, F. Gorunescu and A. M. Salem, "Feature selection in Parkinson's disease: a rough sets approach," in *Proc. Int. Multiconference on Comput. Science and Information Technology*, Mragowo, Poland, 2009, pp.425-428.

[RZ02] M. M. Rizki, M. A. Zmuda and L. A. Tamburino, "Evolving pattern recognition systems," *IEEE Trans. Evol. Comput.*, vol. 6, pp. 594-609, 2002.

[SS07] S. Sapir, J. L. Spielman, L. O. Ramig, B. H. Story, and C. Fox, "Effects of intensive voice treatment (the Lee Silverman voice treatment [LSVT]) on vowel articulation in dysarthric individuals with idiopathic Parkinson disease: Acoustic and perceptual findings," *J. Speech. Lang. Hear. Res.*, vol. 50, pp. 899–912, 2007.

[Te89] C. W. Therrien, *Decision Estimation and Classification.* Wiley, New York, 1989.

[TM96] J.-P. Thiran and B. Macq, "Morphological feature extraction for the classification of digital images of cancerous tissues," *IEEE Trans. Biomed. Eng.*, vol. 43, pp. 1011-1020, 1996.

[VL06] V.Varadan, H.Leung and E. Bosse, "Dynamical model reconstruction and accurate prediction of power-pool time series," *IEEE Trans. Instrumentation and Measurement,* vol. 55, no. 1, pp. 327-336, 2006.

[WJ06] M. L. D. Wong, L. B. Jack and A. K. Nandi, "Modified self-organising map for automated novelty detection applied to vibration signal monitoring," *Mechanical Syst. and Signal Processing*, vol. 20, pp. 593-610, 2006.

[WM05] S. D. Westerheide and R. I. Morimoto, "Heat shock response modulators as therapeutic tools for diseases of protein conformation," *J. Biol. Chem.*, pp. 33097-33100, 2005.

[WS95] W.H. Wolberg, W.N. Street, D.M. Heisey, and O.L. Mangasarian, "Computer-derived nuclear features distinguish malignant from benign breast cytology," *Human Pathology*, vol. 26, pp.792-796, 1995.

[YB06] J. G. Yu and B. Bhanu, "Evolutionary feature synthesis for facial expression recognition," *Pattern Recog. Lett.*, vol. 27, pp. 1289-1298, 2006.

[YM01] G. G. Yen and P. Meesad, "An effective neuro-fuzzy paradigm for machinery condition health monitoring," *IEEE Trans. Syst., Man, Cybern., Part B*, vol. 31, pp. 523–536, 2001.

[ZG07] M. Zhang, X. Gao and W. Lou, "A new crossover operator in genetic programming for object classification," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 37, pp. 1332-1343,2007.

[ZK03] D. Zhang, W. Kong and J. You, "Online palmprint identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, pp. 1041-1050, 2003.

[ZW07] G. Zheng, C.-J. Wang and T. E. Boult, "Application of projective invariants in hand geometry biometrics," *IEEE Trans. Information Forensics and Security*. vol. 2, pp. 758-768, 2007.