A ROUGH SET APPROACH TO
FACE RECOGNITION

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULLFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
IN COMPUTER SCIENCE
UNIVERSITY OF REGINA

By

Xuguang Chen

Regina, Saskatchewan

April 2014

© Copyright 2014: Xuguang Chen
Xuguang Chen, candidate for the degree of Doctor of Philosophy in Computer Science, has presented a thesis titled, *A Rough Set Approach to Face Recognition*, in an oral examination held on February 14, 2014. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner: *Dr. James F. Peters, University of Manitoba*

Co-Supervisor: Dr. Xue Dong Yang, Department of Computer Science

Co-Supervisor: Dr. Wojciech Ziarko, Adjunct

Committee Member: Dr. Bruce Gilligan, Department of Mathematics & Statistics

Committee Member: Dr. Yiyu Yao, Department of Computer Science

Committee Member: Dr. Cortney J. Butz, Department of Computer Science

Chair of Defense: Dr. Dongyan Blachford, Faculty of Graduate Studies & Research

*via teleconference*
Abstract

Rough decision tables were originally proposed by Pawlak. The hierarchies of probabilistic decision tables (the hierarchies), defined in the context of Variable Precision Rough Set Theory (VPRS model), generalized this notion. In recent years, the hierarchies have been successfully applied to many research areas. Compared to other facial representation and recognition techniques, the hierarchies have several unique advantages. As a classifier, the hierarchies can update their structures dynamically and meanwhile maintain prior knowledge when new face images appear. In addition, the most relevant features of facial representation and recognition can be selected heuristically, and the structure of the template images representing each category is extremely simple.

The objective of this research is to explore the feasibility of applying the hierarchies of probabilistic decision tables to facial representation and recognition, under the framework of the VPRS model. As a result, a new facial representation and classification methodology is proposed in this thesis called soft-cut and probabilistic distance-based classifier (soft-cut classifier).

The fundamental contribution of this research is that, for the first time, the hierarchies were applied to face recognition and representation. More specifically, the primary contributions can be summarized as follows.

Firstly, a set of approaches was proposed for the application of the hierarchies,
including how to represent each facial image with wavelet coefficients and heuristically reduce the redundancy in the coefficients. Secondly, a new method of how to discretize real-valued face features (named *soft-cut discretization*) was developed. Thirdly, a new method of heuristically selecting the most relevant features as the conditional attributes for each decision table of the hierarchies was proposed. Fourthly, a technique of pruning the hierarchies to simplify their structures was suggested and applied in the presented methodology. Fifthly, a new hybrid matching method called *probabilistic distance-based matching method* was proposed, classifying the unknown images into each decision category after the hierarchies have been formed. Finally, it was shown how to modify the structure of the hierarchies for the purpose of incremental learning and, for the first time, implement it in the context of the VPRS Model.

In order to determine whether the hierarchies can be successfully applied, various experiments have been conducted. Based on the experimental results and a comparison to other existing methods, if the hierarchies are formed by a training set with a small number of participants, the presented methodology has a higher accuracy rate (greater than 90%). During the process of incremental learning, the hierarchies can always learn from new images and meanwhile retain prior knowledge. Following incremental learning, the performance of the hierarchies can always be improved upon by making the training set either larger or smaller.
Acknowledgements

I sincerely appreciate the support of my supervisor, Dr. Wojciech Ziarko, who has always had confidence in me regardless of the circumstances. I would like to express my gratitude to my co-supervisor, Dr. Xuedong Yang, for giving me the opportunity to teach and for his time and efforts for reading my thesis and providing valuable comments and suggestions. I am greatly thankful to the members of my thesis committee Dr. Cory J. Butz, Dr. Bruce. C. Gilligan and Dr. Yiyu Yao. Their valuable feedback and constructive criticism helped me to improve my thesis in many ways.

Special appreciation is extended to the Department of Computer Science and the Faculty of Graduate Studies and Research for their financial support given at various stages of this research. I would also like to acknowledge financial support from the Natural Sciences and Engineering Research Council via research grants to Dr. Wojciech Ziarko.

Thank you to my parents, friends, and colleagues for their unrelenting encouragement.
Post Defense Acknowledgements

I am greatly thankful to my external examiner Dr. James F. Peters, internal examiners Dr. Cory J. Butz, Dr. Bruce. C. Gilligan and Dr. Yiyu Yao and chair Dr. Dongyan Blachford. Their valuable feedback and constructive criticism helped me to improve the thesis in many ways.
# Table of Contents

Abstract ......................................................................................................................... i

Acknowledgements ........................................................................................................ iii

Post Defense Acknowledgements ........................................................................ iv

Table of Contents ......................................................................................................... v

List of Tables .............................................................................................................. viii

List of Figures ........................................................................................................... xi

List of Acronyms .................................................................................................... xiii

List of Symbols ....................................................................................................... xiv

1. Introduction ............................................................................................................ 1

   1.1 Motivation ....................................................................................................... 2

   1.2 Contributions ............................................................................................... 5

   1.3 Outline of the Dissertation ........................................................................ 9

2. Background and Literature Review ........................................................................ 10

   2.1 Review of Previous Work ........................................................................... 10

   2.2 Face Recognition System and Face Recognition Application Scenarios ...... 17

   2.3 Classification of Face Recognition Methods .............................................. 19

   2.4 Face Representation Methods ...................................................................... 21

   2.5 Wavelet Transformation and Principal Component Analysis ............... 27

   2.6 Variable Precision Rough Set Theory ....................................................... 32
2.7 Hierarchies of Probabilistic Decision Tables .................................................41
2.8 Overview of Incremental Learning .................................................................55

3. Facial Feature Representation ...........................................................................58
   3.1 Reason for Selection of Wavelet Transform for Face Representation ...........58
   3.2 Representation of Facial Images with Haar Wavelet Coefficients ...............60
   3.3 Heuristically Removing Redundant Coefficients .......................................63

4. Facial Feature Extraction and Discretization ..................................................68
   4.1 Steps for Applying PCA to Extract Facial Features ....................................68
   4.2 Discretizing PCA-Based Features with Soft-Cut Discretization Technique ..77
   4.3 Theoretical Aspects of Soft-Cut Discretization ............................................83

5. Forming Hierarchies of Probabilistic Decision Tables ....................................96
   5.1 Selecting Facial Features for Probabilistic Decision Tables .........................97
   5.2 Forming and Pruning Hierarchies of Probabilistic Decision Tables ............102
   5.3 Classifying new Images using Hierarchies ................................................108

6. Updating Decision Table Hierarchies by Incremental Learning ....................116
   6.1 Incremental Learning and Rough Decision Tables .....................................116
   6.2 Theoretical Aspects of the Incremental Learning Algorithm .......................119
   6.3 Incremental Learning with Hierarchies of Probabilistic Decision Tables ...127

7. Experiments and Discussions ........................................................................131
   7.1 Photos for Experiments and Experimental Datasets .................................131
   7.2 Cross-Validation .........................................................................................136
7.3 K-fold Cross-Validation Experiments .......................................................... 137
7.4 Experiments with Balanced and Unbalanced Datasets ................................ 144
7.5 Incremental Learning Experiments .............................................................. 162

8. Conclusions ........................................................................................................... 180

8.1 Research Summary ...................................................................................... 180
8.2 Research Contributions ................................................................................ 183
8.3 Future Work ................................................................................................. 186

Appendix A Ethical Clearance ................................................................................ 187
Appendix B Fundamentals of Harr Wavelet Transformation .............................. 188
Appendix C Overview of Relevant Statistical Concepts ..................................... 195
Appendix D Overview of Related Rough Set Models ......................................... 198
Appendix E Concept of Reducts .......................................................................... 205
Appendix F Example of Incremental Learning with Hierarchies ....................... 209

Bibliography ............................................................................................................. 224
List of Tables

Table 2.1 Comparison between soft-cut classifier and other rough set based methods ... 14

Table 2.2 An review of other recent methods ............................................................... 15

Table 2.3 Comparison between the results of soft-cut classifier and other methods .... 16

Table 2.4 Sample of classification table .................................................................... 43

Table 2.5 Sample of probabilistic decision table ....................................................... 43

Table 2.6 A sample of an incomplete probabilistic decision table ................................ 45

Table 3.1 Wavelet Coefficients at Level 2 and Level 3 ............................................. 62

Table 3.2 Algorithm for Heuristically Reducing Wavelet Coefficients .................... 67

Table 4.1 Algorithm of soft-cut discretization ......................................................... 80

Table 4.2 Determine the initial value of threshold $\tau$ .................................................. 94

Table 4.3 Data distribution and discretization rate ................................................... 95

Table 5.1 HDTML-M [244] .................................................................................... 103

Table 5.2 A sample hierarchy of decision tables ....................................................... 105

Table 5.3 A sample of pruned hierarchy of decision tables ....................................... 108

Table 5.4 A sample decision table at 1st layer .......................................................... 113

Table 6.1 Algorithm implementing three strategies[244] ......................................... 120

Table 6.2 Algorithm of re-computing decision tables for all affected subordinate layers .................................................................................................................. 121

Table 6.3 Algorithm for creating a new elementary set in POS or NEG .................. 121
Table 7.1 Sample results of the 6th original training set in two series of experiments... 139
Table 7.2 Averaged experimental results................................................................. 140
Table 7.3 Sample results of the 6th original training set in two series................. 143
Table 7.4 Averaged experimental results................................................................. 143
Table 7.5 Photos selected for each experimental set............................................. 148
Table 7.6 Averaged results in 4 series of experiments with 4-person datasets....... 149
Table 7.7 Photos selected for each experimental set and average accuracy rate...... 152
Table 7.8 Facial expressions for training sets and test sets in each group .......... 156
Table 7.9 Experiments with datasets having 4-participants .................................. 157
Table 7.10 Experiments with datasets having 6-participants ............................... 157
Table 7.11 Experiments with datasets having 8-participants .............................. 157
Table 7.12 Experiments with datasets having 10-participants ............................. 158
Table 7.13 Averaged accuracy rate of experiments with datasets having different sizes and different facial expressions ................................................................. 159
Table 7.14 The number of experimental sets, participants and photos ............... 160
Table 7.15 Average number of photos correctly classified and average accuracy rate of various groups ............................................................................................ 160
Table 7.16 Experimental sets for various experiments............................................ 165
Table 7.17 Averaged experimental results in each serials...................................... 166
Table 7.18 Details of each type of experimental set............................................. 171
Table 7.19 Results of experiments with datasets having specific facial expressions... 172
Table 7.20 Results of experiments with datasets having unbalanced gender ratio ....... 174

Table B.1 Algorithm for one-dimensional Haar Wavelet Transformation ................. 189

Table B.2 Standard decomposition ................................................................................. 191

Table B.3 Non-standard decomposition ......................................................................... 193

Table F.1 The 1\textsuperscript{st} hierarchy ...................................................................... 210

Table F.2 The 2\textsuperscript{nd} hierarchy ...................................................................... 211

Table F.3 The 3\textsuperscript{rd} hierarchy ...................................................................... 212

Table F.4 The 4\textsuperscript{th} hierarchy ...................................................................... 213

Table F.5 The 1\textsuperscript{st} hierarchy after adding two images .................................. 215

Table F.6 The 1\textsuperscript{st} hierarchy after complete reconstruction ....................... 216

Table F.7 The 1\textsuperscript{st} hierarchy after re-computing decision tables ................... 218

Table F.8 The 1\textsuperscript{st} hierarchy after all new images have been added .............. 219

Table F.9 The 2\textsuperscript{nd} hierarchy after all images have been added ..................... 220

Table F.10 The 3\textsuperscript{rd} hierarchy after all images have been added .................... 221

Table F.11 The 4\textsuperscript{th} hierarchy after all images have been added .................... 222
List of Figures

Figure 2.1 A labeled graph cited from [111]................................................................. 25
Figure 2.2 Jet cited from [210]....................................................................................... 26
Figure 2.3 Face Bunch Graph cited from [111][210]..................................................... 26
Figure 2.4 A sample of tree-structured decision tables cited from [248]......................... 48
Figure 2.5 A Sample of linear-hierarchy-structured decision tables cited from [248].... 49
Figure 3.1 Sample facial images .................................................................................. 60
Figure 4.1 An “average” image cited from [197]............................................................. 70
Figure 4.2 Eigenfaces cited from [197] ......................................................................... 74
Figure 4.3 Samples of soft-cut discretization ................................................................ 81
Figure 7.1 Steps of processing photo manually ............................................................. 135
Figure 7.2 Accuracy rate (%) of each training set in the 1st series............................... 150
Figure 7.3 Accuracy rate (%) of each training set in the 2nd series............................... 150
Figure 7.4 Accuracy rate (%) of each training set in the 3rd series............................... 151
Figure 7.5 Accuracy rate (%) of each training set in the 4th series............................... 151
Figure 7.6 Accuracy rate (%) of experiments with 6-person datasets ......................... 153
Figure 7.7 Accuracy rate (%) of experiments with 8-person datasets ......................... 154
Figure 7.8 Accuracy rate (%) of experiments with 10-person datasets ....................... 154
Figure 7.9 Chart of averaged accuracy rate based on Table 7.13................................. 159
Figure 7.10 Photos correctly classified in each experiment of serial 1 of 4-person cases
Figure 7.11 Photos correctly classified in each experiment of series 2 of 4-person cases

Figure 7.12 Photos correctly classified in each experiment of series 3 of 4-person cases

Figure 7.13 Photos correctly classified in each experiment with 6-person cases, 8-person cases, and 10-person cases

Figure 7.14 Example of learning new photos with a balanced dataset

Figure 7.15 Example of learning each new photo with an unbalanced dataset only having specific facial expressions

Figure 7.16 Example of learning new photos with an unbalanced gender ratio

Figure B.1 Decomposing an image using Standard Decomposition [187]

Figure B.2 Decomposing an image using Non-standard Decomposition

Figure F.1 $\lambda$ – dependency of the 1st hierarchy after adding new images

Figure F.2 $\lambda$ – dependency of the 2nd, 3rd, and 4th hierarchies after adding new images
# List of Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFGR</td>
<td>International Conference on Automatic Face and Gesture Recognition</td>
</tr>
<tr>
<td>1D</td>
<td>1-dimensional</td>
</tr>
<tr>
<td>2D</td>
<td>2-dimensional</td>
</tr>
<tr>
<td>3D</td>
<td>3-dimensional</td>
</tr>
<tr>
<td>AVPBA</td>
<td>International Conference on Audio and Video-Based Authentication</td>
</tr>
<tr>
<td>BND</td>
<td>boundary region</td>
</tr>
<tr>
<td>EBGM</td>
<td>Elastic Bunch Graph Matching</td>
</tr>
<tr>
<td>HMM</td>
<td>Hidden Markov Models</td>
</tr>
<tr>
<td>ICA</td>
<td>independent component analysis</td>
</tr>
<tr>
<td>ILTC</td>
<td>Incremental Learning of Text Classification</td>
</tr>
<tr>
<td>LDA</td>
<td>Linear Discriminant Analysis</td>
</tr>
<tr>
<td>MLP</td>
<td>multi-layer perceptron</td>
</tr>
<tr>
<td>NEG</td>
<td>negative region</td>
</tr>
<tr>
<td>PCA</td>
<td>principal component analysis</td>
</tr>
<tr>
<td>POS</td>
<td>positive region</td>
</tr>
<tr>
<td>RBF</td>
<td>radial basis function</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
</tr>
<tr>
<td>SVM</td>
<td>support vector machine</td>
</tr>
<tr>
<td>VPRS</td>
<td>Variable Precision Rough Set</td>
</tr>
</tbody>
</table>
List of Symbols

$\text{arg min}$ the argument of the minimum (in Section 2.4 Page 23)

$\text{arg max}$ the argument of the maximum (in Section 2.4 Page 24)

$\text{att}_i^E$ the $i^{th}$ binary condition attribute of an elementary set (in Section 5.3 Page 112)

$\text{Ave}$ the vector the components of which is the average of each column of the matrix containing the wavelet coefficients of all images in the training set (in Section 3.3 Page 60)

$\text{Ave}_i$ the average of each column of the matrix containing the wavelet coefficients of all images in the training set (in Section 3.3 Page 65)

$\text{BND}_{i,u}(X)$ boundary region in VPRS model (in Section 2.6 Page 36)

$\text{BND}^*(X)$ boundary region in Bayesian rough set model (in Appendix D Page 203)

$\text{BND}^*(X, \neg X)$ absolute boundary region of the target set $X$ (in Section 2.6 Page 38)

$\text{BND}_{\alpha, \beta}(X)$ boundary region in the decision-theoretic rough set model (in Appendix D Page 203)

$c_i$ arithmetic average of the selected principal component of all images in the training set (in Section 4.2 Page 80)

$\text{cov}(X,Y)$ covariance (in Appendix C Page 196)
\( \text{cov}_A^{U'}(DT) \)  
domain coverage (in Section 2.7 Page 46)

\( \text{cov}_T^{U'}(DT) \)  
theoretical domain coverage (in Section 2.7 Page 46)

\( d(x_i, E) \)  
distance function (in Section 5.3 Page 107)

\( dabs(X, Y) \)  
absolute dependency gain between set \( X \) and set \( Y \) (in Section 2.6 Page 37)

\( \text{Distance} \)  
the distance between a new image and each image in the training set (in Section 2.4 Page 23)

\( DT_i \)  
the \( i^{th} \) decision table in the hierarchy (in Section 6.2 Page 119)

\( egabs(X \mid C) \)  
expected gain function (in Section 2.6 Page 40)

\( E' \)  
the finite subsets of elementary set \( E \) (in Section 2.6 Page 34)

\( f(x) \)  
sigmoid function (in Section 4.2 Page 79)

\( gabs(X \mid Y) \)  
absolute certainty gain between set \( X \) and set \( Y \) (in Section 2.6 Page 36)

\( I \)  
an image set (in Section 2.4 Page 21)

\( I_i \)  
an image in a set (in Section 2.4 Page 22)

\( I_{Haar} \)  
an image represented by a group of selected Haar wavelet coefficients (in Section 3.3 Page 66)

\( I_{pca} \)  
an image in a set that is represented by a group of PCA-based features (in Section 4.1 Page 73)

\( \text{IND}_r \)  
indiscernibility relation on the universe \( U \) (in Section 2.6 Page 33)
NEG\(_i\)\( (X) \) negative region in VPRS model (in Section 2.6 Page 35)

NEG\(^*\)\( (X) \) negative region in Bayesian rough set model (in Appendix D Page 203), also called absolute negative region of the target set \( X \) (in Section 2.6 Page 39)

NEG\(_{\alpha,\beta}\)\( (X) \) negative region in the decision-theoretic rough set model (in Appendix D Page 203)

\( P \) the conditional probability for classifying an unseen image \( Y \) with each image \( I_i \) when its value contains the maximum (in Section 2.4 Page 24)

\( P(E) \) probability associated with an equivalence class \( E \) (in Section 2.6 Page 34)

POS\(_u\)\( (X) \) positive region in VPRS model (in Section 2.6 Page 36)

POS\(^*\)\( (X) \) positive region in Bayesian rough set model (in Appendix D Page 203), also called absolute positive region of the target set \( X \) (in Section 2.6 Page 39)

\( P(X|E) \) conditional probability of finding the equivalence class containing the element \( x \) (in Section 2.6 Page 34)

POS\(^*\)\( (X,\neg X) \) absolute positive region (in Section 2.6 Page 39)

POS\(_{\alpha,\beta}\)\( (X) \) positive region in the decision-theoretic rough set model (in Appendix D Page 203)
\( R(a_i | E) \)  expected loss associated with each action (in Appendix D Page 201)

\( RT_{for} \) running time of the “for” loop (in Section 6.2 Page 120)

\( RT_{incr.} \) in the worst case, the complexity of the incremental learning algorithm in Table 6.1 (in Section 6.2 Page 126)

\( RT_{while} \) running time for the “while” loop (in Section 6.2 Page 122)

\( S \) standard deviation (in Appendix C Page 195)

\( S^2 \) variance (in Appendix C Page 195)

\( S_{ft} \) soft cut (in Section 4.2 Page 84)

\( T_{train} \) a training set represented by a matrix (in Section 2.4 in Page 22)

\( u_i \) Vector determining the linear combination of images in the training set to form the eigenfaces (in Section 4.1 Page 72)

\( U \) a infinite and non-empty set name d the universe (in Section 2.6 Page 33)

\( U' \) finite subset of the universe \( U \) (in Section 2.6 Page 33)

\( \overline{U} \) the eigenvectors of the covariance matrix of the images in the training set (in Section 4.1 Page 71)

\( V^j \) vector space (in Appendix B Page 190)

\( W^j \) vector space (in Appendix B Page 190)

\( X' \) finite subset of the target set \( X \) (in Section 2.6 Page 34)
$X_{Haar}$ matrix having $N$ images from $K$ people in the training set are used to build the probabilistic decision table in the hierarchy (in Section 3.3 Page 64)

$X'_{Haar}$ a set of images, each of which is represented by a group of selected Haar wavelet coefficients (in Section 3.3 Page 66)

$X^{pca}$ a set of images, each of which is represented by a group of PCA-based features (in Section 4.1 Page 73)

$X^{pca}_{discr}$ a set of images, each of which is represented by a group of discretized PCA-based features (in Section 4.2 Page 82)

$X_{simp}$ a set of images, each of which is represented by the most appropriate discretized PCA-based features (in Section 5.1 Page 99)

$X_{target}$ a set containing all the images in the target set $X$ (in Section 4.3 Page 87)

$\overline{X}_{target}$ a set containing all of images of the complementary set of the target set $X$ (in Section 4.3 Page 87)

$X_{1D}$ a 1D array containing the elements of the corresponding column in $X_{target}$ (in Section 4.3 Page 88)

$\overline{X}$ mean of a set of data (in Appendix C Page 195)

$\overline{X}_{1D}$ a 1-D array containing the elements of the corresponding column in
$\overline{X}_{target}$ (in Section 4.3 Page 88)

$\varepsilon_k$ the threshold value used by PCA to determine which category a unknown image should be classified (in Section 4.1 Page 77)

$\phi_i'(x)$ scaling function of space $V^i$ (in Appendix B Page 190)

$\lambda$ eigenvalue (in Appendix C Page 197)

$\lambda_k$ one of eigenvalues of the covariance matrix of the images in the training set as defined in (in Section 4.1 Page 71)

$\lambda(a_i \mid w_j)$ a set of loss functions (in Appendix D Page 199)

$\lambda(X \mid C)$ $\lambda – dependency$ of the decision table (in Section 2.7 Page 53)

$\lambda_{i,a}(X \mid C)$ $\lambda – dependency$ in the context of the VPRS model (in Section 2.7 Page 54)

$\lambda_{i,a}(X \mid R)$ $\lambda – dependency$ of the hierarchy (in Section 2.7 Page 55)

$\lambda_{POS^+}(X \mid C)$ $\lambda – dependency$ of $POS^+(X)$ (in Section 2.7 Page 54)

$\gamma_{i,a}(X \mid C)$ $\gamma – dependency$ in the context of the VPRS model (in Section 2.7 Page 51)

$\gamma_{i,a}^{U}(X \mid R)$ global $\gamma – dependency$ in the context of the VPRS model (in Section 2.7 Page 52)

$\tau$ threshold value (in Section 4.2 Page 79)

$\tau_j$ threshold value in the $j^{th}$ dimension of the feature space (in Section 4.3 Page 84)
\( \psi^j(x) \)  
wavelet function of space \( W^j \) (in Appendix B Page 190)

\( \Phi \Phi(x,y) \)   
scaling function of 2D Haar Wavelet Transformation (in Appendix B Page 194)

\( \Phi \Psi(x,y) \)   
wavelet function of 2D Haar Wavelet Transformation (in Appendix B Page 194)

\( \Gamma_i \)   
the vector representing an image when the image is processed by PCA (in Section 4.1 Page 70)

\( \Omega_{\text{new}} \)   
a unknown image represented as a vector (in Section 4.1 Page 76)

\( \Psi \)   
"average" face in PCA (in Section 4.1 Page 70)

\( \Psi \Phi(x,y) \)   
wavelet function of 2D Haar Wavelet Transformation (in Appendix B Page 194)

\( \Psi \Psi(x,y) \)   
wavelet function of 2D Haar Wavelet Transformation (in Appendix B Page 194)

\( \| \cdot \| \)   
Euclidean distance (in Section 2.4 Page 23)
1. Introduction

The objective of this research is the investigation of the methodology of applying the hierarchy of probabilistic decision tables to the research area of face recognition in the context of Variable Precision Rough Set Theory (VPRS model) [88][243]. A new facial representation and recognition methodology called “soft-cut and probabilistic distance-based classifier” (soft-cut classifier) has been proposed. An approach using the hierarchy of probabilistic decision tables as a classifier for the purpose of facial representation and recognition has been developed, investigated, and tested.

Presently, in the research area of face recognition, a typical application scenario of the presented methodology would be focusing on facial identification [111]. In such a scenario, a set of template images, each of which has a unique identity, has been initially constructed based on the digital images in the training set. The identity of each template image should be the information uniquely identifying a single person in the training set. Then, an unknown digital image appears. The unknown image will be compared to all of the template images to determine its identity. At the end of the process, the unknown image will be assigned the same identity as a template image with the highest similarity.

Section 1.1 describes the advantages of the hierarchies when they are applied to face recognition. Section 1.2 is about the contribution and Section 1.3 pertains to the remainder of the thesis.
1.1 Motivation

Rough set theory was developed by Pawlak [147-149] [151-153][155] and extended by many researchers [59-60][104-105][150][183][213-214][221][232]. Rough set theory extended by Ziarko [88][243] was presented as the Variable Precision Rough Set Theory (VPRS model). Rough set theory is a useful tool with regard to knowledge discovery in databases and in processing experimental data sets. Given an information system, rough set theory can be applied to determine the most appropriate attributes from the point of view of their classificatory power. During the past twenty years, it has been an active area of research including the area of face recognition [190-192], in which, when given a facial image, rough set theory can be very useful in finding the most relevant features for face recognition.

Decision tables are a very special class of knowledge representation systems and are important for many applications, for instance software engineering and electronic circuit design. Decision tables that were originally proposed by Pawlak in the framework of rough set theory [147][151] are referred to rough decision tables. Different from decision tables used in other areas, rough decision tables are learned and then created from data. The probabilistic decision tables and their hierarchies [248-249] were proposed by Ziarko, which were defined in the context of a VPRS model [243]. They generalized the notion of rough decision tables. In recent years, the hierarchy of probabilistic decision tables has been applied to many research areas, such as the control rules acquisition in artificial
intelligence-related research [156] and the control algorithm acquisition systems for vehicle control [171]. Hierarchies are applied to face recognition because the hierarchy of probabilistic decision tables has several unique advantages when compared to other techniques.

Selecting the most relevant features that can distinguish one image from the others is always vital, whether this is performed with holistic matching methods (which use the whole image as the input) or feature-based matching methods (which only use facial features as the input). The human face is a very complex pattern [220], and therefore it is not easy to distinguish between the relevant features. It becomes more difficult if there are many images in the training set. The hierarchies of the probabilistic decision tables are formed from a dataset based on a VPRS model. As described in Chapter 5, a probabilistic decision table in the hierarchy usually has a boundary region (BND) containing elements that cannot be classified into a positive region (POS) or negative region (NEG) according to the conditions of the present probabilistic decision table. The elements can be used to build the subsequent decision table in the hierarchy based on a group of new conditions, and this time some elements can be classified into a POS or NEG. Thus, when the hierarchies are applied to face recognition, several groups of features can be selected instead of just one, each of which only needs to work well on some of the images rather than all of the images. If an image cannot be classified into a POS or NEG, it will be classified into a BND automatically and processed by another group of features.
When the input is an unknown image, it is compared to each template image to
determine its identity. The structure of the template images will have a significant impact
on the number of calculations. For example, the method proposed in [197] contains 115
images in the training set are transformed using principal component analysis (PCA), so
40 eigenfaces are used to describe the set of images, which means each template image is
represented by 40 features. Similarly, when Elastic Bunch Graph Matching (EBGM)
[166][210] is applied to face recognition, the identity of the unknown face has to be
identified by comparing similarities between the new face and all the faces stored in the
face bunch graph. When the hierarchies of probabilistic decision tables are applied to face
recognition, each template image is represented by an elementary set according to the
conditions of a decision table. Degrees of similarity between the unknown face and each
elementary set are compared, and assigned an unknown image to the identity of an
elementary set that generates the highest similarity. In practice, the number of conditions
from each decision table is restricted to less than six. Based on the experimental results, if
the number of images in the training set is less than 200, two or three conditions will
suffice. Hence, when a probabilistic decision table is used to represent template images,
the structure of the template images can be very simple because each template image can
be represented by only a few features.

Template images are formed according to the images currently available in the
training set but their representations are not sufficient. For example, adding new images
to the training set may reduce the representativeness of the template images. If this is the
case, the template images will have to be re-generated so as to reflect the presence of the new images. However, this extended process is time-consuming and knowledge acquired from the old data can be lost. Due to these restraints, it would be preferable if a face recognition classifier could continuously improve the performance by learning additional information from the newly added images and retain prior knowledge. Unfortunately, many techniques, when applied to face recognition, cannot improve performance with incremental learning. For instance, common neural network paradigms such as Multi-layer Perceptron (MLP) and Radial Basic Function (RBF) network [161] can only learn newly added objects by totally re-training the classifier using all the accumulated data so far. On the other hand, the hierarchies are very suited to incremental learning. Various strategies describing how to modify the hierarchies in the framework of classic rough set theory have been proposed in [244]. When new faces are available, the hierarchies can effectively improve their performance by updating their structures following the modification strategies rather than total re-generation.

1.2 Contributions

The fundamental contribution of this thesis is the proposed new methodology in which the hierarchies are, for the first time, applied to face recognition and representation. During the research, various problems occurred because the hierarchies had never been applied to this area. A set of methods and techniques were proposed as solutions to the problems, and more specifically, the primary contributions are summarized below.
1. As described in Section 3.2 and Section 3.3, a method was proposed with which to represent each facial image with several groups of wavelet coefficients using the technique heuristically to reduce the redundancy of the coefficients. Wavelet transformation has been widely used in face recognition [31][49][54][82][190-192], but wavelet coefficients from different sub-bands are usually used as one single set of features for facial representation [192]. In this research, as explained in Section 3.3, each face is represented with several groups of coefficients, each of which is from a specific sub-band. Wavelet transformation can reduce the size of each image considerably, but each group of coefficients still has redundant information. Therefore, a technique was proposed to heuristically remove the redundancy from each group of coefficients to reduce the number of computations.

2. A new method of discretizing real-valued face features, named soft-cut discretization, was proposed in Section 4.2. After being processed by PCA, the features representing each face are transformed from wavelet-based into PCA-based. The value of each condition attribute of a probabilistic decision table must be binary-valued. Because the PCA-based features are real-valued, they must be discretized. A traditional and widely-used approach is crisp cut discretization [115], but usually it cannot deal appropriately with situations where a feature value is close to the cut point and possibly affected by random noise. In such a case, a wrong value is often assigned to the feature. In order to overcome this type of limitation, as described in Section 4.2, a new method is proposed. This new
method would allow for accommodation of the aforementioned situations during the discretization and recognition stages.

3. In Section 5.1, a new method of heuristically selecting the most relevant features as the conditional attributes for each decision table was proposed and applied in the presented methodology. The best method for feature selection is to try every combination of the features and compare their performance. However, this is unfeasible in practice. Assuming a probabilistic decision table can have 2, 3, 4 or 5 conditional attributes, there are 24 features as the candidates. If every combination of the features were tried, the whole process of feature selection would need about two hundred thousand calculations to complete. As introduced in Section 5.1, a method is proposed which heuristically selects the most relevant conditional attributes for each decision table by evaluating the value of \( \lambda - \text{dependency} \). The \( \lambda - \text{dependency} \) is a very important parameter as it describes the quality of the hierarchies. By employing the method, the aforementioned process can be accomplished hundreds of times.

4. In Section 5.2, a technique was proposed to prune the hierarchies to simplify their structure. Typically, a decision table contains elementary sets, classified into its positive, negative, and boundary regions [147]. When the hierarchies are applied to face recognition, the negative regions of the generated decision tables are pruned, and in some cases, many decision tables can be completely removed. The advantages of this technique are that the size of the decision table and the
hierarchies can be reduced, and the unknown images only need to be matched to an elementary set in the positive region or boundary region. These advantages will speed up the classifying of new objects resulting in an improvement.

5. In Section 5.3, a new hybrid matching method called probabilistic distance-based matching method was proposed. This method classifies the unknown images into each category after the hierarchies have been formed. A person’s face often can have some variations such as the viewing angle, illumination, facial expression, and noise. The distance-based classification methods are widely used, but they usually cannot deal appropriately with the variances [19]. Due to these issues, a hybrid matching method was proposed that is a combination of distance-based and probabilistic-based methods. Different from standard distance-based classification methods involving an exact match between the feature vectors representing images and elementary sets, the presented method can “soften” the matching procedure in order to ignore small differences between compared patterns that are possibly caused by noise and would result in many unclassified test objects. Varying from traditional probabilistic-based methods only using conditional probability as a parameter, the presented method selects different probabilities based upon where (positive region or boundary region) each elementary set is located. The values of the probabilities are compared, and the unknown image is assigned the identity of the decision class of a hierarchy with the highest probability.

6. The algorithm proposed in [244] by Ziarko was originally defined in the context of
classic rough set theory. For the first time, it was implemented in the context of the VPRS Model as presented in Section 6.3.

1.3 Outline of the Dissertation

The background information and relevant literature review are presented in Chapter 2, focusing on the various face recognition methods and the techniques applied to the presented methodology. Chapter 3 introduces the method with which to represent each facial image using wavelet coefficients and how to preliminarily remove their redundancy. Chapter 4 introduces a way to transform the facial images by PCA. Especially, a new discretization method, named *soft-cut discretization*, is explained in this chapter. Chapter 5 is concerned with how to heuristically select features for the conditional attributes, form the hierarchies, and classify an unknown image with a probabilistic distance-based matching method. Chapter 6 introduces a method to update the structures of the hierarchies when new images are available for incremental learning. Experiments are described and the corresponding analyses are included in Chapter 7. Finally, the conclusion and future work are presented in Chapter 8.

In Appendix B, the principal concepts of the wavelet transform, in particular Haar Wavelet, are introduced. The statistical concepts as required by the presented methodology are covered by Appendix C. Three probabilistic rough set models are reviewed in Appendix D and the concept of reducts is in Appendix E. Appendix F further explained the working process of incremental learning with a complete example.
2. Background and Literature Review

Background information and previous works related to face recognition are introduced in Chapter 2. The previous research with similarities to the presented methodology is reviewed in Section 2.1. The face recognition system and application scenarios are covered in Section 2.2. The classifications of face image representation and recognition are introduced in Section 2.3 and Section 2.4. The application of wavelet transformation and principal component analysis (PCA) to face recognition is described in Section 2.5. In Section 2.6, the principal concepts of Variable Precision Rough Set model (VPRS model) as required in the presented methodology are covered. Section 2.7 describes the decision tables, including their classifications, a typical example being probabilistic decision tables and the hierarchy of probabilistic decision tables. An overview of incremental learning is in Section 2.8.

2.1 Review of Previous Work

Subsequent to many years of research, a variety of techniques and methods have been proposed and successfully applied to face recognition. In this Section, some of face recognition methods based on the application of rough set theory are briefly introduced and compared with the presented methodology. After that, these methods and other recent face recognition methods are summarized in Table 2.1 and Table 2.2. Finally, the comparison between the results of the presented methodology and some of these methods
are listed in Table 2.3.

Swiniarski applied data mining to face recognition [192]. A Haar Wavelet Transformation extracted features from each face photo and the number of coefficients were heuristically reduced before they were processed with principal component analysis (PCA). Rough set theory was applied to the coefficients for feature reduction and selection. Different from the presented methodology in this thesis, only the coefficients from the first level transform were selected for face representation, and the classifier was an error back-propagation neural network. Although a decision table was involved in this method, its primary function was to determine the most appropriate reduct (a selected minimum set of features used for face recognition). Thirteen classes of facial photos, with 30 different instances for each class, were used for experimental measurements. Seventy percent of them were used for the training set, 15% for the validation set and the final 15% for the test set. The resulting accuracy for the test set was about 99.24% [192].

Swiniarski and Skowron described two hybrid methods for face recognition which were based on independent component analysis (ICA), PCA and rough set theory [191]. The methods were similar to the method in [192]. ICA or PCA were used for feature extraction and reduction and then the selected set of features were processed according to rough set theory for the purposes of feature selection and data reduction. The features, selected by rough set theory, were processed by the rough set rule based classifier for the purpose of recognition. As opposed to [192] and the presented methodology, face photos in [191] are not transformed by a Haar Wavelet Transformation. Each face photo is
represented by pixels and processed by ICA or PCA for feature extraction and reduction. Forty classes of 112 x 92 pixel gray scale images with 10 instances per class were used in the experiments. The classification accuracy of the test set is about 88.75% using the ICA-based method and about 86.25% for the PCA-based method.

Swiniarski and Skowron proposed a method based on the application of rough set methods for feature selection during pattern recognition which can be applied to face recognition [190]. PCA was used for feature projection and reduction, and rough set theory was used for feature selection as in [192]. Instead of Haar Wavelet Transformation [192], each face photo was processed by singular value decomposition (SVD) [53] which is a matrix processing technique for face feature extraction before processing with PCA. This method was similar to the technique in [192] and used a decision table similar to the one used in the presented methodology. But, its primary function was to determine the most appropriate reduct (a selected set of features used for face recognition) rather than act as the classifier. Before the rough set theory could be applied, the PCA features were discretized as in the presented methodology, but it differed from the soft-cut classifier as each attribute value range was divided into 10 evenly spaced bins rather than binary bins. An error back propagation neural network and the rough set rule based classifier were selected as the classifier as in [192] and, as in the presented methodology, statistical methods such as the cross validation technique were used to estimate the robustness of the classifier. The neural network provided an accuracy of 75.5% to 99% on various test sets and the rough set rule based classifier provided an accuracy of 94.5% to 100% on
various test sets.

A procedure for reducing the size of the input feature vector using rough set theory for human face recognition was proposed by Bhattacharjee and other researchers [13]. Its idea was similar to [190], but each face photo represented by pixels was directly processed by PCA. They deployed photos from the Cambridge (ORL) database as in [190-192], but they also deployed experiments based on the photo database created by their own laboratory. One purpose of the experiments was to study whether recognition performance increased with an increase in the number of eigenvectors. Based on their results, a general trend could be observed. This trend showed significant improvement in the performance with an increasing number of dimensions until a threshold in the number of dimensions had been reached, after which the further improvements became insignificant.

Rough set theory can be applied to both holistic-based matching and feature-based matching methods. Singh and Raghuwanshi introduced an approach to enhance the performance of face recognition systems based on feature-based matching methods using rough set theory [175-176]. In their approach, nose length, nose width and the distance between eyeballs were selected as the facial features. These elements were the basis for the construction of the decision table.

Rough set theory was applied to the decision table to generate classification rules. Derived equivalence classes were used as input to approximation neurons that carried out computations in a rough membership neural network. The values generated by a layer of
approximation neurons were used to construct a condition vector having values of attributes for a new face. This vector was used as input for a rough set based expert network for face recognition. Their experiments also used photos from the Cambridge (ORL) database and the average accuracy was about 96.5%.

<table>
<thead>
<tr>
<th>Method Proposed by</th>
<th>Feature Representation</th>
<th>Feature Selection and Reduction</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft-cut classifier [27-28][30]</td>
<td>Haar Wavelet Coefficients from several levels</td>
<td>PCA and Rough Set</td>
<td>Hierarchies of rough set based decision tables</td>
</tr>
<tr>
<td>Swiniarski and Skowron [191]</td>
<td>Pixel values</td>
<td>ICA/PCA and Rough Set</td>
<td>Rough set rule based classifier</td>
</tr>
<tr>
<td>Swiniarski and Skowron [190]</td>
<td>SVD matrix</td>
<td>PCA and Rough Set</td>
<td>Neural network and rough set rule based classifier</td>
</tr>
<tr>
<td>Bhattacharjee and others [13]</td>
<td>Pixel values</td>
<td>PCA and Rough Set</td>
<td>Neural network</td>
</tr>
<tr>
<td>Singh and Raghuwanshi [175-176]</td>
<td>Nose length, nose width, and distance between eyeballs as features</td>
<td>N/A</td>
<td>Rough neural network</td>
</tr>
<tr>
<td>Swiniarski [192]</td>
<td>1st Level Haar Wavelet Coefficients</td>
<td>PCA and Rough Set</td>
<td>Error back-propagation neural network</td>
</tr>
<tr>
<td>Kandepet and Swiniarski [86]</td>
<td>Face components such as eyes, eyebrows, lips and nose</td>
<td>Support vector machine and PCA</td>
<td>Error back-propagation neural network</td>
</tr>
</tbody>
</table>

In [86], Kandepet and Swiniarski introduced a component based approach to face recognition too, the recognition rates of which were up to 92%. Their approach was implemented as two systems: a face detection system and a face recognition system. The face detection system took a given image as the input, and then located the face on the
input image. After that, the face was extracted using 2D Haar wavelets and support vector machines. The extracted face region was used to locate and extract the individual components of the face, such as eyes, eyebrows, lips and nose. These components were processed by the PCA to reduce their size, and then used as the input of the face recognition system that was implemented as an error back-propagation neural networks.

**Table 2.2** An review of other recent methods

<table>
<thead>
<tr>
<th>Method Proposed by</th>
<th>Feature Representation</th>
<th>Feature Selection and Reduction</th>
<th>Classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chien and Wu [31]</td>
<td>Wavelet Coefficients</td>
<td>Linear Discriminant Analysis</td>
<td>Nearest neighbor and nearest feature line classifiers</td>
</tr>
<tr>
<td>Zhang, Ma, and Wu [235]</td>
<td>Wavelet Coefficients</td>
<td>Image Wavelet Analysis</td>
<td>Neural Network</td>
</tr>
<tr>
<td>Bartlett, Movellan, and Sejnowski [5]</td>
<td>Pixel values</td>
<td>ICA</td>
<td>Nearest neighbor</td>
</tr>
<tr>
<td>Buddharaju, Pavlidis, and Kakadiaris [18]</td>
<td>Pixel value processed by Fuzzy-based segmentation</td>
<td>Gabor Wavelet filter</td>
<td>Bayesian classifier</td>
</tr>
<tr>
<td>Cevikalp, Neamtu, Wilkes, and Barkana, [23]</td>
<td>Pixel values</td>
<td>Discriminative Common Vector</td>
<td>A variation of Fisher’s Linear Discriminant Analysis</td>
</tr>
<tr>
<td>Kouzani, He, and Sammut [98]</td>
<td>Pixel values</td>
<td>Fractal code</td>
<td>Neural Network</td>
</tr>
<tr>
<td>Pan, Healey, Prasad, and Tromberg [142]</td>
<td>Near infrared hyperspectral images</td>
<td>Spectral reflectance vector</td>
<td>Cumulative match score based on Mahalanobis distance between the test image and the images in the gallery set</td>
</tr>
<tr>
<td>Wiskott, Fellous, Kruger, and Malsburg [210]</td>
<td>Facial features represented as jets</td>
<td>Gabor Wavelet transformation</td>
<td>Compare the test image with Elastic Bunch Graph based on the graph similarity function</td>
</tr>
</tbody>
</table>
Table 2.3 Comparison between the results of *soft-cut classifier* and other methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy Rate (%)</th>
<th>Without Incremental Learning</th>
<th>After Incremental Learning</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Balanced dataset</td>
<td>Unbalanced expression dataset</td>
</tr>
<tr>
<td><strong>Soft-cut classifier</strong></td>
<td></td>
<td>4-person</td>
<td>6-person</td>
</tr>
<tr>
<td>[27-28][30]</td>
<td></td>
<td>91.73</td>
<td>92.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>87.28</td>
<td>86.51</td>
</tr>
<tr>
<td></td>
<td></td>
<td>80.05</td>
<td>84.55</td>
</tr>
<tr>
<td></td>
<td></td>
<td>73.69</td>
<td>79.97</td>
</tr>
<tr>
<td><strong>Swiniarski et al.</strong> [191]</td>
<td></td>
<td>ICA-based experiments: 88.75</td>
<td>PCA-based experiments: 86.25</td>
</tr>
<tr>
<td><strong>Swiniarski et al.</strong> [190]</td>
<td></td>
<td>neural network as classifier: 75.50 to 99</td>
<td>rough set rule as classifier: 94.50 to 100</td>
</tr>
<tr>
<td><strong>Singh and et al.</strong> [176]</td>
<td></td>
<td>96.50</td>
<td></td>
</tr>
<tr>
<td><strong>Swiniarski</strong> [192]</td>
<td></td>
<td>99.24</td>
<td></td>
</tr>
<tr>
<td><strong>Kandepet et al.</strong> [86]</td>
<td></td>
<td>92%</td>
<td></td>
</tr>
<tr>
<td><strong>Chien et al.</strong> [31]</td>
<td></td>
<td>From 75.40 to 96.00</td>
<td></td>
</tr>
<tr>
<td><strong>Bartlett et al.</strong> [5]</td>
<td></td>
<td>89.00</td>
<td></td>
</tr>
<tr>
<td><strong>Belhumeur et al.</strong> [6]</td>
<td></td>
<td>99.60</td>
<td></td>
</tr>
<tr>
<td><strong>Buddharaju et al.</strong> [18]</td>
<td></td>
<td>86.80</td>
<td></td>
</tr>
<tr>
<td><strong>Cevikalp et al.</strong> [23]</td>
<td></td>
<td>Images from Yale 97.33</td>
<td>Images from AR database 99.35</td>
</tr>
<tr>
<td><strong>Kouzani et al.</strong> [98]</td>
<td></td>
<td>100</td>
<td></td>
</tr>
<tr>
<td><strong>Pan et al.</strong> [142]</td>
<td></td>
<td>92.00</td>
<td></td>
</tr>
<tr>
<td><strong>Wiskott et al.</strong> [210]</td>
<td></td>
<td>Images from FERET 80.00</td>
<td>Images from Property DB 90.00</td>
</tr>
</tbody>
</table>
2.2 Face Recognition System and Face Recognition Application Scenarios

A face recognition system can always be divided into three logical sub-systems; the face detection sub-system, the feature extraction sub-system and the face recognition sub-system. Correspondingly, a face recognition system can have three functions, which are the face detection function, the feature selection and extraction function and the face recognition function.

When a still image or video clip is provided, a face recognition system begins by identifying whether a face is available on the image or in the clip. Once a face is detected, the face recognition system will find and extract the features which best represent faces in the image or clip. The system will then use the features to recognize the face.

The sub-systems can work simultaneously, so these three functions cannot be totally separated. For example, facial features (eyes, mouth or nose) can often be used for face detection and recognition. Once a face is detected, the details of its features can be extracted instantly. In other words, face detection and feature extraction can be completed simultaneously [237].

Although a face recognition system should consist of all three functions, it is important to focus on each single sub-system separately during development. It is common for a researcher to only focus on one or two sub-systems rather than the entire system. Over the past thirty years, substantial research has been conducted on each
function or specific sub-system producing many publications. According to the specific application, face recognition scenarios can be classified into three classes [111]: face verification, face identification and watch list.

A typical application of face verification is a one-to-one match. A template face image is constructed according to a number of face images and its identity is assigned. Then, an unknown face image, claiming the same identity as the template face image, appeared. To identify if the unknown image is from an imposter, the unknown image will be compared to the template face image.

Face identification is usually applied with the one-to-many match process. An unknown image is compared to all of the template images available in a database to determine its identity. The unknown image is assigned the same identity as the template image exhibiting the highest similarity. Face identification can be considered the process of “closed” recognition, because an unknown image may have the same identity as one of the template images in the database.

Similar to face identification, a watch list compares an unknown image to all of the template images in a database, but it is not known if the unknown image has the same identity as the template image. Thus, it is possible that an unknown image may have an identity different from all of the template images. In this possible scenario, the unknown image will be compared to each template image and the corresponding level of similarity will be scored. If the highest recorded score is larger than the threshold value, the unknown image will be deemed an image having the same identity as the template image.
2.3 Classification of Face Recognition Methods

During the past thirty years, many methods of face recognition have been proposed. Most methods are based on more than one technique, so it is not easy to clearly classify them. In addition, a method can be classified into different categories based on various criteria. In this Section, two typical classification methods are introduced.

2.3.1 Still-Image-based Methods versus Video-based Methods

Face recognition methods can be classified into different categories. If a method uses still-images, it can be considered a still-image-based method. If a face recognition method uses video clips, it can be considered a video-based method. Usually, a video-based face recognition system can automatically detect face regions, select and extract facial features from the video and complete the face recognition. True video-based methods did not appear until a few years ago [237] and currently present many challenges. For example, compared to still images, the size of face images on a video can be small, and the quality of the video files can be poor. It is common for video files to be created under bad lighting conditions. This leaves face images on the video over expressed or having significant pose variations.

Historically, video-based face recognition was considered an extension of still-image-based face recognition. After a face has been detected and extracted from the video file, still-image-based face recognition techniques [104][197] can be applied with
some modifications. One example is Eigen-pictures [197].

More advanced approaches use multimodal cues in which facial images and other information such as from voice and/or fingerprints are collaboratively used to achieve identification [32]. Newer methods have been proposed which exploit spatial and temporal information at the same time. These approaches exploit information in each frame of a video file including trajectories of facial features and combining them [103].

2.3.2 Holistic Matching Methods versus Feature-based Matching Methods

Face recognition methods using still images (for example, photos taken by a digital camera) can be further classified into different categories, holistic matching methods and feature-based matching methods. A holistic matching method usually accepts the entire digital image as the input. Then, statistical techniques are applied to significantly reduce the data and build an efficient and effective representation (a classifier). When analyzing a test image, the similarity between the test image and the derived classifier will be determined to correctly classify the test image.

Feature-based matching methods focus on facial features such as the eyes, nose and mouth rather than the whole face. For face recognition, the methods extract facial features from digital images and use the features and information in relations to their locations and local statistics (geometric appearance).

The history of feature-based matching methods is more extensive than holistic
matching methods. Almost all early face recognition methods are feature-based matching methods. For example, in [84], the corners of the eyes and nostrils are localized in the frontal view of each image. The parameters based on the features are used for face recognition. More recently, a method called “elastic bunch graph matching” has been developed [210]. This method represents a face as a graph, with nodes positioned at locations such as the eyes, nodes by sets of wavelet coefficients [210] and edges labeled with two-dimensional distance vectors. The details of this method will be covered in the next Section.

2.4 Face Representation Methods

The face data (for example digital facial images) collected for face recognition can be represented with various representational methods. In this Section, the procedures used for representing a digital facial image with holistic matching method and feature-based matching methods are described.

2.4.1 Representing a Facial Image with Holistic Matching Methods

Holistic matching methods usually consider an image (having \( n \) pixels) as a point in an \( n \)-dimensional space. Moreover, an image \( I \) can be more conveniently represented as a 1-dimensional column vector of concatenated rows of pixels represented in (2.1):

\[
I = [x_1, x_2, \ldots, x_i, \ldots, x_n]^T, \quad (2.1)
\]
where $n$ is the total number of pixels in the image and $x_i$ is the $i^{th}$ pixel value in the image. Holistic matching methods often need to work with a set of images (usually called a training set $T_{rain}$), so the images in the set $T_{rain}$ can be represented as a matrix. Each row of the matrix contains pixel values for each image as shown in (2.2):

$$\begin{pmatrix}
x_{11} & \cdots & x_{1n} \\
\vdots & \ddots & \vdots \\
x_{m1} & \cdots & x_{mn}
\end{pmatrix},$$

(2.2)

where $n$ is the number of pixels in each image and $m$ is the total number of images in the training set. Alternatively, (2.2) can be expressed as a series of vectors in (2.3). Each image $I_i$ in the training set $T_{rain}$ is represented as a vector:

$$T_{rain} = [I_1, I_2, \ldots, I_j, \ldots, I_m]^T.$$

(2.3)

An image can be represented in different ways rather than pixel values. Conversely, for a specific face recognition classifier, only one or some of the representational methods are appropriate. Hence, given a specific representation, usually efficiency and discriminating power [19] can be used to determine the appropriate representation and recognition classifier. For instance, if efficiency is the most important factor, a classifier based on the feature-based matching method is often a better choice.

Holistic-based matching methods require more calculations than feature-based methods because whole facial images are used as input. A facial image can often contain a large amount of redundant information [192]. This means that holistic-based matching methods often apply some statistical techniques to improve their efficiency by reducing the size of an image. Currently, one of the most widely used statistical techniques is
principal component analysis (PCA). PCA can accurately identify patterns in data including the image and express the data so as to highlight their similarities and differences. This method can significantly reduce the number of data dimensions without vital data loss.

Other techniques such as Fisherfaces [6], support vector machine [160] and independent component analysis (ICA) are also applied in order to reduce the size of images. Instead of directly applying one technique to face recognition, several techniques can be applied at the same time, such as in [190-192].

Often a representation can be very efficient but can demonstrate weak discriminating power. This weakness means it is rather poor at correctly classifying an unknown image into a correct classification, so after reducing image size, holistic-based matching methods often apply other techniques (for example rough sets theory [190-192]) to improve performance. This extracts information that is highly useful for face recognition within the image to enhance its discriminating power.

A very common approach used to classify a new image with holistic-matching methods is called “distance-based classification”. A training set having $n$ images, the new image $Y$ will be matched to each image in the training set, and assigned the same identity as a training image $I_i$ if formula (2.4) is satisfied:

$$Distance = \text{arg min} \|Y - I_i\|,$$  

(2.4)

where $\|\cdot\|$ is the Euclidean distance between image $Y$ and image $I_i$ and $1 \leq i \leq n$. $\text{arg min}$ is defined as the argument of the minimum and includes the images in the training set for
which the value of the $||.||$ attains its minimum.

The face of a given person can have some variations due to the viewing angle, illumination, facial expression and noise. Often distance-based classification cannot decipher variations to correctly classify an unknown image. In contrast, when identifying a new image, another method called “maximum likelihood-based classification” is widely used. The conditional probability for classifying an unseen image $Y$ with each image in the training is calculated first and then $Y$ is classified into the same category as a training image $I_i$ if (2.5) can be satisfied, where $1 \leq i \leq n$. In formula (2.5), $\arg \max$ represents the argument of the maximum and includes the set of the images in the training set for which the value of $P(Y|I_i)$ attains its maximum value:

$$P = \arg \max P(Y|I_i). \quad (2.5)$$

### 2.4.2 Representing a Facial Image in Feature-Based Methods

In the past few years, many feature-based approaches have been proposed. Primarily, the methods are based on the geometry of local features [85] and Hidden Markov Models (HMM) [165-166]. One of the most successful methods is Elastic Bunch Graph Matching (EBGM) using Gabor wavelets as a building block [166][210]. EBGM is briefly introduced in this Section as an example of feature-based face representation methods.

The basic idea of EBGM is based on a graph called a Bunch Graph. A general in-class recognition method to classify members of a known class of objects is proposed in [210] under the assumption the topological structures of all human faces are similar to
each other. Faces are represented as graphs on which nodes are marked at fiducial points such as the eyes, the tip of the nose and other contour points. The edges are labeled with 2-dimensional distance vectors as cited from [111] and shown in Figure 2.1.

Each node on a Bunch Graph has a set of 40 complex Gabor wavelet coefficients that include phase and magnitude. It is called a jet in Figure 2.2 which was cited from [210]. The wavelet coefficients are extracted using a family of Gabor kernels that have 5 different frequencies and 8 orientations. For the sake of clarity, a jet is represented with only 3 frequencies and 4 orientations in Figure 2.1.

![Figure 2.1 A labeled graph cited from [111]](image)

Face recognition is based on labeled graphs, each of which is a set of nodes labeled with jets and connected with edges. As shown in Figure 2.1, the geometry of a face is encoded with edges, while the gray value distribution is encoded with jets. A Face Bunch Graph uses a structure similar to a stack of combined labeled graphs of all the available sample faces to represent a class having many faces. All individual graphs have the same structures and the nodes refer to the same fiducial points. Jets referring to the same fiducial point are bundled together in a bunch, from which any jet can be selected as an alternative description. For example, in Figure 2.3, the left figure represents a Face
Bunch Graph [210] where each node is labeled with a bunch of 6 jets. As represented by the right figure, the left-eye bunch may contain a male eye, a female eye, a closed male eye, an open female eye. Each fiducial point is represented by a set of alternatives and any jet can be independently selected from each bunch. Thus, this is a face bunch graph based on 6 sample faces, but it can potentially represent $6^9 = 10,077,695$ different faces.

**Figure 2.2** Jet cited from [210]

When a new face is available, EBGM locates the fiducial points and extracts the points to form a graph maximizing the graph similarity function. After the grid has been well positioned on the new face, the identity of the new face is identified by comparing the similarities between the new face and every face stored in the face bunch graph. The new face is assigned the same identity as the face in the face bunch graph producing the highest similarity.

**Figure 2.3** Face Bunch Graph cited from [111][210]
2.5 Wavelet Transformation and Principal Component Analysis

In the presented methodology, the wavelet transformation was selected as the technique for face representation. Principal component analysis (PCA) was selected to reduce the number of calculations when processing the face images by extracting the most appropriate facial features from each face. This Section reviews the application of wavelet transformation and principal component analysis to face recognition.

2.5.1 Application of Wavelet Transformation to Face Recognition

Initially, a wavelet transform had been a useful tool for approximation theory and signal processing [113], but recently it has been applied to many other problems such as image editing [7], image compression [39], surface reconstruction [119], animation [107], and face recognition [31][49][54][190-192][235]. Wavelets can be classified into various forms such as Gabor wavelets [210] and Haar Wavelets [187-188].

There are several ways that wavelet transformation can be applied to face recognition. The simplest application is to use wavelet coefficients directly. The basic idea is to first transform a face image using the standard or non-standard decomposition as introduced in Appendix B and then use the coefficients as features. According to formula (B.3), three types of wavelet basic functions can be obtained after completion of the transformation.
These functions can encode information about an image [145], including differences in the average intensity along a vertical border, a horizontal border and diagonal boundaries. Because basic functions can encode differences in three directions, wavelet coefficients, based on the basic functions, can always ascertain the visual features of a facial image. In essence, a wavelet transform can be used to locally detect the edges of facial images which are very useful in face recognition. If the value of a coefficient is large, this indicates a boundary is available and its location can be obtained from the image. A uniform area is represented by coefficients having values close to zero.

For example, a “wavelet face” was proposed in [31]. The basic premise of a wavelet face is to use a wavelet transform to decompose each face image into four sub-images. Then, a low spatial-frequency sub-image, which refers to the coefficients of a scaling basic function, is selected for further decomposition. Assuming the size of each image is \( n \times n \), a three-level lowest spatial-frequency sub-image, the size of which is \( (n/2^3) \times (n/2^3) \), will be extracted as the feature vector called a wavelet face. The sub-image presents the basic figure of a face and has the highest discriminating power but is not as sensitive to varying images.

Wavelet coefficients can be combined with statistical measures such as discriminative features (mean and variance), to assist in finding the most powerful features. Each face is described by a subset of sub-band filtered images containing wavelet coefficients after a two-level wavelet packet transform [54]. Then, a set of simple statistical measures is used to reduce the number of dimensions of each face so a set of
compact and meaningful feature vectors can be formed. After all the images in the training set are processed, only the components having a mean value above a predefined threshold value, are considered for feature vector formation. Finally, the Bhattacharya distance between two feature vectors are computed on a component-pair basis to classify the face feature vector into appropriate person classes [82].

The third way of applying wavelet coefficients is to transform wavelet coefficients using traditional statistical methods. Wavelet coefficients can represent facial features, but they are not significantly discriminative features. Thus, as previously preformed, the wavelet transform is only used to pre-process face images with respect to the formation of preliminary face features. After that, the facial features are processed by other methods, such as principal component analysis (PCA), linear discriminant analysis (LDA), independent component analysis (ICA) or neural networks. The methods are performed on one or several special spatial-frequency sub-bands of wavelets which are chosen according to certain criteria.

For instance, PCA is applied to wavelet sub-bands for face recognition [49][190-192]. A wavelet sub-band approach, using PCA for human face recognition, has been proposed in [49]. A three-level wavelet transform was selected to decompose each facial image into different sub-bands with different frequency components. Subsequently, a sub-band with a midrange frequency was transformed and represented with a PCA presentation. Based on experimental results, their method was found to have low computation and higher accuracy compared to directly transforming original images with PCA.
In [192], the first level of 2-dimensional Haar wavelet coefficients of each face image was selected. After that, the coefficients formed a concatenation of matrix rows (i.e., the coefficients of each image occupied one row of the matrix). Next, a row of the coefficient matrix was heuristically reduced by removing \( r \) trailing elements from each row of the matrix. Finally, PCA was applied to transform the reduced matrix and reduce its size.

In addition, a wavelet transform decomposes a face image into different frequency ranges, which can accurately isolate the frequency components introduced by illumination effects into specific subspaces. Hence, when employing face recognition, the subspaces can be neglected. This can avoid the influence of changes in illumination. A facial compensation approach, based on a wavelet and neural network and combining multi-resolution analysis of a wavelet and self adaptation, was proposed in [235]. A linear illumination compensation, which could reduce the learning pressure for the neural network, accelerated the rate of convergence, improved the learning accuracy and improved the extensibility of the neural network, was performed on the lower frequency features of each given face image.

### 2.5.2 Application of Principal Component Analysis to Face Recognition

The purpose of applying PCA to face recognition is to reduce the number of calculations when processing the images by extracting the most appropriate facial
features from each face. Each facial image is initially represented by its pixels or wavelet coefficients. They are each processed with PCA and represented by a small group of PCA-based features. The extracted features can be directly used for face recognition. Also, if possible, they can be further processed to either select more relevant features for face recognition or reduce the number of facial features – in turn reducing the number of calculations.

Over the past thirty years, many PCA-based methods have been proposed. Examples include the Karhunen-Loeve Procedure for the Characterization of Human Faces [92], Eigenfaces[197] and density estimation in a high-dimensional space using eigenspace decomposition[122].

When applying PCA to face recognition, each image in the training set is considered a point in a high-dimensional space called face image space. Assuming an image has \( n \) pixels, it is considered an \( n \)-dimensional point in the face image space. If a training set has \( m \) images, it is considered as \( m \) points in this huge space. Although faces differ from each other, their overall configurations are similar. So, in this \( n \)-dimensional space, the images represented as \( m \) points will not be randomly distributed. It is possible and feasible to describe the images in a relatively low subspace. The objective of applying PCA to the face recognition is to find the vectors which best account for the distribution of the face images in the whole face image space. The vectors are used to define the subspace of face images, which is called “face space” [197]. The vectors are \( n \)-dimensional vectors, and are a linear combination of the original face images in the
training set. When applying PCA, the vectors are selected from the eigenvectors of the covariance matrix and are the original face images in the training set. The vectors appear face-like, especially when the faces are represented by pixels, so they are called “eigenfaces”.

### 2.6 Variable Precision Rough Set Theory

In this Section, the principal concepts of Variable Precision Rough Set Theory (VPRS model) [2][88][214][243][245][253] as required in the presented methodology are covered.

Rough set theory was introduced by Pawlak [147-149][151-155] with respect to the study of intelligent systems characterized by insufficient and incomplete information. During the past twenty years, many extended rough set models have been proposed, such as [88][181][224][232]. Based on whether statistical information is used, the existing models can be classified into two categories [225], which are the algebraic rough set models (including the model proposed by Pawlak [147-148][154] and its extensions [228]) and probabilistic rough set models.

The probabilistic rough set models were proposed twenty-seven years ago [150][212][224][232]. Until now, many methods have been proposed in which probabilistic approaches have been incorporated into the rough set theory in different ways [221][231], such as rough set based probabilistic classification [214], 0.5 probabilistic rough set model [150][212], decision-theoretic rough set model [221]
When the traditional rough set model is applied to partial classifications; the classifications must be fully correct or certain in the positive and negative regions. Otherwise, it will not work well. In order to overcome the limitations, a probabilistic extension, referred to as Variable Precision Rough Set model (VPRS model), was proposed by Ziarko [88][214][243][245][251-253]. The concern of the VPRS model is the recognition and modeling of set overlap-based relationships among sets. Such relationships can be used to construct parameter-controlled approximations of undefinable sets.

The VPRS model is developed in the framework of a set of objects of interest (named the universe $U$). The universe $U$ is often assumed to be infinite, so only a finite subset $U' \subseteq U$ can be accessed. In addition, it also assumes the knowledge of an equivalence relation $R$, which is called an indiscernibility relation on the universe $U$ [147] and denoted as $IND_R$ [251], with finite number of equivalence classes called elementary sets $E$. The indiscernibility relation $IND_R$ represents a pre-existing classification knowledge of the universe $U$. It is defined as $IND_R \subseteq U \otimes U$, where the operator $\otimes$ denotes Cartesian product operator [245]. The pair $(U, IND_R)$ is called the approximation space. The collection of the elementary sets $E$ forms a partition of the universe $U$ and can be denoted as $U/R$, i.e. $E \in U/R$. A subset of the universe $U$ is definable if it can be expressed as a union of some elementary sets $E$. Otherwise, it is not
definable, or rough [147]. In the context of the rough set theory, an arbitrary subset \( X \) of the universe \( U \), referred to as the target set, is usually undefinable, but an approximate definition of the target set \( X \) can be formed in terms of some definable sets [251].

In probabilistic approaches to the rough set model, a subset \( X \) of the universe \( U \) under consideration is measurable by its prior probability \( P(X) \). Each equivalence class \( E \) is associated with two measures that are the probability \( P(E) \) and the conditional probability \( P(X \mid E) \). The probability \( P(E) \) is a measure of the relative size of the class \( E \) in the universe \( U \), and the conditional probability \( P(X \mid E) \) is a measure of overlap with the target set \( X \).

In practice, because it is often unfeasible to directly compute \( P(E) \) and \( P(X \mid E) \) based on the whole universe \( U \) of interest, they are computed on the basis of available sample data. Based on the sample data, if each of the finite subsets of the domain is defined as \( E \subseteq E \), \( X \subseteq X \) and \( U \subseteq U \) respectively, the probability \( P(E) \) and the conditional probability \( P(X \mid E) \) can be approximated by [252]:

\[
P(E) \approx \frac{\text{card}(E')}{\text{card}(U')},
\]

and

\[
P(X \mid E) \approx \frac{\text{card}(X' \cap E')}{\text{card}(E')},
\]

where \( U' \) and \( E' \) are nonempty, and \( \text{card} \) means set cardinality.

In the VPRS model, the probability estimates associated with elementary sets are used to construct rough approximations of the target set \( X \). The defining criteria can be
expressed in terms of the conditional probabilities $P(X \mid E)$ and the prior probability $P(X)$ of the target set $X$. Two precision control parameters \cite{88} called lower limit $l$ and upper limit $u$ are used in the lower approximation of the target set $X$ or its complement. Lower limit $l$ presents the highest acceptable degree of conditional probability $P(X \mid E)$ to include the elementary set $E$ in the negative region of the target set $X$. Its value should satisfy $0 \leq l < P(X) < 1$. The upper limit $u$ defines the positive region of the target set $X$, and its value should satisfy $0 < P(X) < u \leq 1$. It describes the least acceptable degree of the conditional probability $P(X \mid E)$ to include the elementary set $E$ in the positive region.

**Positive region $POS_u$, Negative region $NEG_l$, and Boundary region $BND_{l,u}$**

In order to define the approximations in terms of unions of some elementary sets, lower limit $l$ and upper limit $u$ are used, as introduced above. Each elementary set is classified into a positive region $POS_u$, a negative region $NEG_l$, or a boundary region $BND_{l,u}$, based on the values of the two parameters, as defined by the followings \cite{246} [251-252]:

**Definition 2.1** The negative region of the target set $X$ denoted as $NEG_l$ is a collection of objects for which the probability of membership in the target set $X$ is significantly lower than the prior probability $P(X)$, that is

$$NEG_l(X) = \cup \{E : P(X \mid E) \leq l\}. \quad (2.8)$$

**Definition 2.2** The positive region of the target set $X$ denoted as $POS_u$ is a collection of objects for which the probability of membership in the target set $X$ is significantly greater than the prior probability $P(X)$, that is
\[ P_{\text{POS}_{n}}(X) = \cup \{ E : P(X \mid E) \geq u \}. \quad (2.9) \]

**Definition 2.3** The boundary region denoted as \( BND_{I,u} \) is a collection of objects for which the probability of membership in the target set \( X \) is not significantly different from the prior probability \( P(X) \), that is

\[ BND_{I,u}(X) = \cup \{ E : l < P(X \mid E) < u \}. \quad (2.10) \]

In Definitions 2.1-2.3, the prior probability \( P(X) \) plays an essential role. It represents the likelihood that a random element of the universe \( U \) is a member of the target set \( X \), if no classification knowledge about the element is available. On the other hand, if the classification knowledge is available, as represented by the equivalence relation \( I_{\text{IND}} \), the likelihood of membership in the target set \( X \) of elements that are belonging to different elementary sets \( E \) can change: increase, decrease or stay approximately the same as \( P(X) \). These variations in the target set \( X \) membership likelihood across different elementary sets are reflected in Definitions 2.1-2.3.

**Absolute certainty gain between set \( X \) and set \( Y \)**

According to [245], absolute certainty gain is proposed to evaluate the degree of dependency between any two sets. Given two arbitrary measurable subsets \( X \) and \( Y \) of the universe \( U \), it is a one-way dependency measure representing the degree of change of the certainty of prediction of the set \( X \) as a result of the occurrence of the set \( Y \). Absolute certainty gain can be defined as [245]:

\[ g_{\text{abs}}(X \mid Y) = | P(X \mid Y) - P(X) |, \quad (2.11) \]

where \(|.|\) is the absolute value function.
If the subset $Y$ is definable in the approximation space $(U, IND_R)$, it can be expressed as a union of some elementary sets $E \in U/R$, i.e. $Y = \bigcup_{E \in Y} E$. By definition, $P(X \mid Y)$ can be represented as $P(X \mid Y) = P(X \cap Y) / P(Y)$. Therefore, in the approximation space $(U, IND_R)$, $P(Y)$, $P(X \cap Y)$ and $P(X \mid Y)$ can be expressed as $P(Y) = \Sigma_{E \subseteq Y} P(E)$, $P(X \cap Y) = \Sigma_{E \subseteq Y} P(X \cap E)$ and $P(X \mid Y) = \Sigma_{E \subseteq Y} P(X \cap E) / \Sigma_{E \subseteq Y} P(E)$. In addition, $P(X \cap E)$ can be $P(X \cap E) = P(E)P(X \mid E)$, so $P(X \mid Y)$ in (2.11) can be expressed as $P(X \mid Y) = \Sigma_{E \subseteq Y} P(E)P(X \mid E) / \Sigma_{E \subseteq Y} P(E)$. Thus, in the approximation space $(U, IND_R)$, the absolute certainty gain between the subsets $X$ and $Y$ can be calculated directly from the available probabilistic knowledge based on the formula below [245]:

$$gabs(X \mid Y) = \frac{\left| \sum_{E \subseteq Y} P(E)P(X \mid E) - P(X)\sum_{E \subseteq Y} P(E) \right|}{\sum_{E \subseteq Y} P(E)},$$

where

1. $P(E)$ is the probabilistic measure of each equivalence class $E$ of the indiscernibility relation $IND_R$.
2. $P(X \mid E)$ is the conditional probability of the subset $X$ for each elementary set $E$.

**Absolute dependency gain between set $X$ and set $Y$**

According to [245], absolute dependency gain is a two-way dependency measure to evaluate the degree of the two-way connection between any two sets. Given two arbitrary subsets $X$ and $Y$ of the universe $U$, the absolute dependency gain can be defined as [245]:

$$dabs(X, Y) = | P(X \cap Y) - P(X)P(Y) |.$$

(2.13)
That is, absolute dependency gain reflects the degree of probabilistic dependency between the sets \( X \) and \( Y \) by quantifying the amount of deviation of \( P(X \cap Y) \) from probabilistic independence between sets \( X \) and \( Y \), as expressed by \( P(X)P(Y) \).

Similar to the absolute certainty gain, in the approximation space \( (U, \text{IND}_r) \), if the subset \( Y \) is definable, then the absolute dependency gain between the subsets \( X \) and \( Y \) can be computed directly from the available probabilistic knowledge based on the formula below [245]:

\[
dabs(X,Y) = \sum_{E \in \U} P(E)P(X \mid E) - P(X) \sum_{E \in \U} P(E),
\]

(2.14)

where each elementary set \( E \) is belonging to the partition \( U/R \), i.e. \( E \in U/R \) and \( P(X) \) is the prior probability of the subset \( X \).

**Absolute boundary region, absolute positive region, and absolute negative region**

According to the definition from [245][253], the absolute boundary region of the target set \( X \) is a definable region of the universe \( U \) characterized by the total lack of relationships with the target set \( X \subseteq U \). In the absolute boundary region, each elementary set \( E \) is probabilistically independent from the target set \( X \) (i.e. \( P(X \mid E) = P(X) \)). It can be defined by the absolute dependency gain as:

\[
\text{BND}^\wedge(X, -X) = \bigcup \{ E : \text{dabs}(X \mid E) = 0 \},
\]

(2.15)

where \( -X \) is the complementary set of the target set \( X \).

The region of the universe \( U \) that is characterized by at least some probabilistic connection with the target set \( X \subseteq U \) is a definable region denoted as \( \text{POS}^\wedge(X, -X) \) and named the absolute positive region of the classification \( (X, -X) \). It can be defined by the
absolute dependency gain as:

\[ \text{POS}^*(X, \neg X) = \cup \{ E : \text{dabs}(X \mid E) > 0 \} . \]  \tag{2.16} 

When \( \text{dabs}(X \mid E) \) is greater than zero, it means \( P(X \mid E) > P(X) \) or \( P(X \mid E) < P(X) \). In other words, \( \text{POS}^*(X, \neg X) \) can be considered a combination of two regions which are the absolute positive region \( \text{POS}^*(X) \) of the target set \( X \) defined as:

\[ \text{POS}^*(X) = \cup \{ E : P(X \mid E) > P(X) \} , \]  \tag{2.17} 

and the absolute negative region \( \text{NEG}^*(X) \) of the target set \( X \) defined as:

\[ \text{NEG}^*(X) = \cup \{ E : P(X \mid E) < P(X) \} . \]  \tag{2.18} 

In comparison to the prior probability, the absolute positive region \( \text{POS}^*(X) \) is a definable region of the universe \( U \) characterized by an increased probabilistic connection with the target set \( X \). The absolute negative region \( \text{NEG}^*(X) \) is a definable region characterized by reduced likelihood of the target set \( X \) occurrence. Moreover, comparatively, the formulas (2.17) and (2.18) to the Definition D.1 in Appendix D are the same. The definitions above constitute the basis of another probabilistic rough set model named Bayesian rough set model. As described in Appendix D, the set approximation in the Bayesian rough set model is defined using the prior probability \( P(X) \) as a reference.

**Expected gain function**

In many applications, the knowledge about the objects of the universe \( U \) can be expressed in terms of values of some selected attributes (features) [251]. Such a knowledge representation system can be defined as follows:

**Definition 2.4** [147] Knowledge representation system is a pair
\[ S = (U, A), \quad (2.19) \]

where \( U \) is the universe of objects of interest and \( A \) is a finite set of attributes. Each attribute \( a \in A \) is a function \( a: U \rightarrow V_a \), where \( V_a \) is the set of values of \( a \), called the domain of attribute \( a \).

Given an attribute \( a \in A \), it can define a classification of the universe \( U \) into elementary sets according to its various attribute values. Similarly, a subset of \( A \) can define a classification of \( U \) according to the combinations of values of the attributes in the subset. If it is empty, the subset is considered a trivial attribute, i.e. an attribute having only one attribute value corresponding to \( U \). If there are non-trivial classifications for the universe \( U \), the degree of the decision certainty may be improved. According to [245][251], the degree of the improvement can be quantified by the expected gain function.

Assuming that a subset \( A' \subseteq A \) is given, the indiscernibility relation corresponding to \( A' \) is denoted as \( IND_{A'} \) and the collection of the elementary sets \( E \) induced by the subset \( A' \) forms a partition denoted as \( U/A' \), i.e. \( E \in U/A' \). The expected gain function can be defined as [251]:

\[
\text{egabs}(X \mid A') = \sum_{E \in U/A'} P(E) | P(X \mid E) - P(X) |. \quad (2.20)
\]

As a result of presence of the classification knowledge that is expressed by equivalence classes of the indiscernibility relation \( IND_{A'} \) and the associated probabilities, the expected gain function measures the average degree of increase of the occurrence
probability of the target set $X$ or its complement $-X$, relative to its prior probability $P(X)$.

By definition, $P(X \cap E) = P(E)P(X \mid E)$. Therefore, the expected gain function can be defined in terms of the absolute dependency gain as:

$$egabs(X \mid A') = \sum_{E \in \mathcal{U} / X} |P(X \cap E) - P(X)P(E)| = \sum_{E \in \mathcal{U} / A} dabs(X, E).$$

That is, the expected gain function can also be considered the measure of the degree of probabilistic dependency between the classification that is represented by the indiscernibility relation $IND_{A'}$ and the partition of the universe $U$ corresponding to the set $X$ or its complement $-X$.

### 2.7 Hierarchies of Probabilistic Decision Tables

Over the past twenty years, rough set theory has been an active area of research. The decision tables, their derivations and the corresponding algorithms have played an essential role in researches. In this Section, decision tables, including their classifications, a typical example being probabilistic decision table and the hierarchy of probabilistic decision table, are covered.

#### 2.7.1 Decision Tables and Probabilistic Decision Tables

According to [150], a knowledge representation system can be perceived as a decision table. In such a table, the values of the condition attributes stipulate the conditions of a particular decision that is specified by the values of the decision attributes.
Thus, the decision table can be defined in terms of knowledge representation systems as follows:

**Definition 2.5** [147] Let $S = (U, A)$ be a knowledge representation system and let $C, D \subseteq A$ be two subsets of attributes, called condition and decision attributes respectively. The knowledge representation system with distinguished condition and decision attributes is called a decision table.

According to [249], decision tables can be classified into three categories, which are deterministic decision tables, non-deterministic decision tables and probabilistic decision tables.

Deterministic decision tables describe the functional relation between a set of observations and the corresponding decisions. Usually, based on the requirements of the designer, it is designed manually. Deterministic decision tables are useful tools applied to simple problems. In practice, deterministic decision knowledge is not always available, so a proper function describing the relationship between table conditions (observations) and decisions (predication or identities) is often unavailable. The designer often constructs a decision table based on the observation data rather than his/her knowledge. This being the case, a decision table can be thought of as non-deterministic decision tables.

Compared to the first types of decision tables, a probabilistic decision table is extracted from data based on the VPRS model, in which one observation can have more than one decision. The VPRS model contains additional information that can be used as a
probabilistic measure to identify the correctness of decisions made with such a table. A sample probabilistic decision table has been listed in Table 2.5. The definition of probabilistic decision tables and the details regarding the construction of the table are covered below.

**Table 2.4 Sample of classification table**

| E_i | A_1 | A_2 | A_3 | P(E_i) | P(X|E_i) |
|-----|-----|-----|-----|--------|---------|
| E_0 | 1   | 1   | 1   | 0.0520 | 0.7000  |
| E_1 | 1   | 1   | 0   | 0.1354 | 0.0     |
| E_2 | 1   | 0   | 1   | 0.1562 | 1.0     |
| E_3 | 1   | 0   | 0   | 0.1562 | 0.3666  |
| E_4 | 0   | 1   | 1   | 0.1406 | 0.0     |
| E_5 | 0   | 1   | 0   | 0.1093 | 0.0     |
| E_6 | 0   | 0   | 1   | 0.1562 | 0.0     |
| E_7 | 0   | 0   | 0   | 0.0937 | 0.0     |

**Table 2.5 Sample of probabilistic decision table**

| E_i | A_1 | A_2 | A_3 | P(E_i) | P(X|E_i) | Region |
|-----|-----|-----|-----|--------|---------|--------|
| E_0 | 1   | 1   | 1   | 0.0520 | 0.7000  | BND    |
| E_1 | 1   | 1   | 0   | 0.1354 | 0.0     | NEG    |
| E_2 | 1   | 0   | 1   | 0.1562 | 1.0     | POS    |
| E_3 | 1   | 0   | 0   | 0.1562 | 0.3666  | BND    |
| E_4 | 0   | 1   | 1   | 0.1406 | 0.0     | NEG    |
| E_5 | 0   | 1   | 0   | 0.1093 | 0.0     | NEG    |
| E_6 | 0   | 0   | 1   | 0.1562 | 0.0     | NEG    |
| E_7 | 0   | 0   | 0   | 0.0937 | 0.0     | NEG    |

For a probabilistic decision table, the collection of the elementary sets E induced by the condition attributes C forms a partition of the universe U and can be denoted as U/C, i.e. $E \in U/C$. The pair $(U, U/C)$ is the approximation space induced by C. Similarly, the partition and approximation space of the decision attributes D are denoted as U/D and
The elementary sets induced by \( D \) are called decision categories. If the set of decision attributes only has one binary-valued decision attribute, the decision attribute \( D \) classifies all objects in the university \( U \) into two disjoint sets, the target set \( X \) and its complement \( \neg X \), that is, \( U/D = \{ X, \neg X \} \).

When constructing a probabilistic decision table, based on available data, over and above identifying all feasible combinations of condition attribute values in the domain, close estimates of \( P(E) \) and \( P(X|E) \) are computed. Such a table is called a classification table as listed in Table 2.4. In such a table, the classification of the universe \( U \) is illustrated in terms of the elementary sets induced by the conditional attributes \( C = \{ A_1, A_2, A_3 \} \) plus the conditional probabilities with respect to a target set \( X \), specified by providing an estimate of \( P(X|E) \) for each elementary set \( E \). The information contained in Table 2.4 can be used to build rough approximations of the target set \( X \subseteq U \).

Once the rough region of each elementary set \( E \) is identified, a unique designation with respect to the target set \( X \) is matched to each elementary set \( E \) in Table 2.4. Hence, the classification table becomes a probabilistic decision table that is fully deterministic with respect to the new decision attribute “Region”. The decision attribute “Region” has three values: “POS”, “NEG” and “BND”. For example, the decision attribute value of elementary set \( E_0 \) is “BND”. Hence, for the given target set \( X \) and the values for the lower limit \( l \) and the upper limit \( u \), the probabilistic decision table can be defined as a mapping derived from the classification table as follows:

**Definition 2.6** [247][249-250] The probabilistic decision table can be defined as a
mapping associating each combination of condition attributes $c \in C$ with a pair of values representing (1) the unique destination of the rough approximation region (positive, negative, or boundary region) the element set $E$ belongs to and (2) the estimated values of the elementary set probability $P(E)$ and the conditional probability $P(X \mid E)$.

**Table 2.6** A sample of an incomplete probabilistic decision table

<table>
<thead>
<tr>
<th>$E_i$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>$P(E_i)$</th>
<th>$P(X \mid E_i)$</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.0520</td>
<td>0.7000</td>
<td>BND</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.1354</td>
<td>0.0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.1562</td>
<td>1.0</td>
<td>POS</td>
</tr>
<tr>
<td>$E_3$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.1562</td>
<td>0.3666</td>
<td>BND</td>
</tr>
<tr>
<td>$E_4$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.1406</td>
<td>0.0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_5$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.1093</td>
<td>0.0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_6$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.2499</td>
<td>0.0</td>
<td>NEG</td>
</tr>
</tbody>
</table>

If a new case, matching the combination $A_1=1, A_2=1$ and $A_3=1$, materializes, it cannot be classified as a member of the target set $X$ or its complement set. Table 2.5 has provided a complete representation of the elementary sets in the universe, but in practice a probabilistic decision table is often not complete. In Table 2.6, if a new case matching the combination $A_1=0, A_2=0$ and $A_3=0$ occurs, it cannot be matched to any row of the table because no elementary set in the table contains the combination $A_1=0, A_2=0$ and $A_3=0$. So, when applying probabilistic decision tables to new cases, the completeness is an important factor. Formally, the completeness and coverage measures can be evaluated using two parameters called “actually domain coverage” and “theoretical domain coverage” [247].

Given a decision table denoted as $DT$, the actually domain coverage $cov^U_A(DT)$
can be represented as the ratio of the number of elementary sets, determined by the condition attributes $C$ and represented by the decision table, to the actual number of elementary sets determined by the condition attributes $C$ in the universe $U$:

$$
\text{cov}^V_{a}(DT) = \frac{\text{card}(U' \cap C)}{\text{card}(U / C)},
$$

(2.22)

where \text{card} means cardinality, $U' \subseteq U$ is a sample set of objects in the universe $U$, $U'/C$ is the collection of the elementary sets represented by the decision table and $U/C$ is the collection of the elementary sets in the universe $U$.

Because $\text{card}(U/C)$ in (2.22) is often unknown, the theoretical domain coverage has been introduced, which is an estimate of the percentage coverage of the universe by the decision table. According to [247], its formula is defined as:

$$
\text{cov}^V_T(DT) = \frac{\text{card}(U' \cap C)}{\prod_{a \in C} \text{card}(V_a)},
$$

(2.23)

where $V_a$ is a finite set of values called the domain of the attribute $a \in C$ and $\prod_{a \in C} \text{card}(V_a)$ is a product of cardinalities of all its attribute domains.

### 2.7.2 Reducing the Size of the Boundary Region

A probabilistic decision table usually has a boundary region. When constructing a probabilistic decision table, a problem to be addressed is the problem of narrowing down its boundary region. Generally, there are three possible solutions that can be selected.

**Solution 1. Increase the number of attributes or possible attribute values of each or**
some of the attributes

The simplest method is to add more conditional attributes or increase the precision of existing attributes by providing more cut-points in the data discretization process before constructing the decision table. This is simple, but there is a very serious limitation: the number of feasible value combinations of the attributes increase exponentially. For example, Table 2.5 has three conditional attributes $C=\{A_1, A_2, A_3\}$ and each attribute has two possible values $\{0, 1\}$. The feasible combinations of attribute values are $2^3 = 8$. If two more attributes are added, and their possible values are also $\{0, 1\}$, then the combinations becomes $2^5 = 32$. The decision table may become very complex due to this method and the relative domain coverage of the known combinations will decrease.

Solution 2. Form a tree-structured hierarchy of decision tables

The basic idea behind this method is to form another layer of decision tables. A new layer of decision tables would take into consideration the fact each sub-domain of samples, matching the description of an elementary set of the boundary region of the initial decision table, is a new domain by itself. Such a sub-domain is used to form a decision table for the subset of the target set only within the sub-domain. When applying this method, the problem of attributes used in the present layer should be considered because they must have sufficient precision to be divided into several subclasses that may vary from those used in the initial decision table. If this method is applied recursively, a hierarchy of tree-structured decision tables can be formed as cited from [248] and shown
in Figure 2.4.

**Figure 2.4** A sample of tree-structured decision tables cited from [248]

Finally, the boundary region of each decision table, as a leaf in the hierarchy, can disappear or at worst, its expansion will be stopped because the new attributes cannot reduce the boundary region of several layers. A serious limitation of this method is the number of possible paths in the tree of decision tables can grow exponentially with the number of layers. Eventually, the whole structure becomes too complex to handle.

**Solution 3. Form a linear hierarchy of decision tables**

This method is to form another layer of decision tables in a way different from that of Solution 2. The original domain of the initial decision table can be divided into three sub-domains that are corresponding to the whole positive region, negative region, and boundary region of the initial decision table. These sub-domains can be determined by the conditional attribute values of each elementary set belonging to each region. Differing from the aforementioned method, this method treats the sub-domain corresponding to the whole boundary region as a new domain. When building the decision table, the attributes should be different from those used in the initial decision table. By using this method recursively, a hierarchy of linear structure of the decision tables can be constructed as
illustrated in Figure 2.5 which was cited from [248].

![Figure 2.5](image)

**Figure 2.5** A Sample of linear-hierarchy-structured decision tables cited from [248]

The basic idea behind this method is similar to the previous method, but the intermediate decision table is built based on the boundary region (as a whole) of the previous decision table. Consequently, the hierarchy is linear but not tree-structured. This type of a linear hierarchy can expand until there is no boundary region in the last node, no new condition attributes available for the next decision table, or new condition attributes cannot reduce the boundary region of the current decision table. The main advantage is the size of the hierarchy can be well controlled. Unlike the previous method, the size of the hierarchy is not increased exponentially.

### 2.7.3 Dependencies between the Partitions in Hierarchies of Probabilistic Decision Tables

In this Section, two parameters, $\gamma$ – *dependency* and $\lambda$ – *dependency*, are reviewed. They can reflect the statistical connection among the attributes (conditional attributes and decision attributes). In the presented approach, they are used as the primary parameters help to evaluate the quality of a decision table or the hierarchy of probabilistic decision tables during the machine learning process. The probabilistic decision table represents the
classification of the sample objects from the universe of interest into individual discernible categories (i.e. elementary sets) based on the values of the attributes. Moreover, the probabilistic decision table can also depict the dependencies between attributes, so the accuracy of the predictions of the decision attribute values can be identified from the known conditional attributes. Similar to a single probabilistic decision table, the quality of a hierarchy of decision tables also requires evaluation so that it can be determined if it is a suitable classifier for a new case.

Dependencies between attributes can be defined in different ways. Functional dependencies and partially functional dependencies were explored in [147] and were referred to as $\gamma$-dependency [246]. They can explain the quality of approximation of the target category in terms of the elementary sets of the approximation space. For a decision table, the $\gamma$-dependency can represent the degree of its determinism, as acquired from a dataset. It can be described in terms of the probability of the positive region of the partition determined by the decision attributes and define decision categories.

It is assumed the collection of the elementary sets $E$ induced by the condition attributes $C$ forms the partition $U/C$ (i.e. $E \in U/C$), and the pair $(U, U/C)$ is the corresponding approximation space. The collection of the decision categories $X$ induced by the decision attributes $D$ forms the partition $U/D$ (i.e. $X \in U/D$). According to [147][250], the original $\gamma$-dependency is defined as:

$$\gamma(D \mid C) = P(\text{POS}(D \mid C)),$$

(2.24)

where $\text{POS}(D \mid C)$ is the positive region of the partition $U/D$ in the approximation space.
\[(U, U/C)\) and can be defined as:

\[\text{POS}(D \mid C) = \cup\{E : P(X \mid E) = 1\}. \tag{2.25}\]

The \(\gamma\) – dependency can be extended within the framework of the VPRS model. In the binary case of two decision categories, it means the partition \(U/D = \{X, \neg X\}\).

According to (2.8)-(2.10), the positive and negative regions of the two class partitions corresponding to \(X\) and \(\neg X\) in the approximation space \((U, U/C)\) are defined as:

\[\text{POS}_u(X \mid C) = \cup\{E : P(X \mid E) \geq u\}, \tag{2.26}\]

and:

\[\text{NEG}_l(X \mid C) = \cup\{E : P(X \mid E) \leq l\}, \tag{2.27}\]

where \(C\) is the set of condition attributes, and parameters \(l\) and \(u\) are the lower and upper limits in VPRS model.

With \(\text{POS}_u(X \mid C)\) and \(\text{NEG}_l(X \mid C)\), the \(\gamma\) – dependency in the VPRS model can be defined as the relative “size” of the positive region of the two class partitions corresponding to \(X\) and \(\neg X\) in the approximation space induced by the conditional attributes \(C\). In other words, the combined probability of the \(u\)-positive and \(l\)-negative regions below is:

\[\gamma_{l,u}(X \mid C) = P(\text{POS}_u(X \mid C) \cup \text{NEG}_l(X \mid C)). \tag{2.28}\]

It can reflect a proportion of objects in the universe of interest \(U\) that can be classified as a member of the target set \(X\) or a complement of the target set \(X\) with sufficiently acceptable certainty.

If the approximation spaces are obtained via the hierarchical classification process,
the $\gamma$-dependency between the hierarchical partition $R$ and the partition $(X, -X)$ can be calculated in two ways. The first method is to directly compute the value of $\gamma$-dependency by analyzing all classes of the hierarchical partition. The second method is to compute recursively [246]. That is, starting from the leaf table of the hierarchy and climbing to the root table as per the following formula:

$$
\gamma_{i,u}^{U}(X \mid R) = \gamma_{i,u}^{U}(X \mid C) + P(U')\gamma_{i,u}^{U'}(X \mid R')
$$

(2.29)

must be applied recursively to produce the $\gamma$-dependency of the whole structure of probabilistic decision tables at the end of the process.

In (2.29), $\gamma_{i,u}^{U}(X \mid R)$ is the dependency of the parent table in the hierarchical approximation space $(U, R)$, obtained by nesting partitions produced by conditional attributes on each level of the hierarchy. $\gamma_{i,u}^{U'}(X \mid R')$ is the dependency of a child level decision table in the sub-approximation space $(U', R')$, where $U' = BND_{i,u}(X)$. The parameter $C$ is a collection of condition attributes inducing the approximation space $U$ and $U' = BND_{i,u}(X)$. $\gamma_{i,u}^{U}(X \mid R)$ measures the proportion of objects that can be classified into the target set $X$ or its complement with sufficiently acceptable certainty using decision tables included in a hierarchy. When applying recursively, the dependency of the whole structure of decision tables, called global $\gamma$-dependency, is the last dependency computed by (2.29).

For a probabilistic decision table, the other parameter called $\lambda$-dependency is a dependency that can reflect the average degree of improvement of the certainty of occurrence of the target set $X$ and its complement $-X$, relative to the prior probability.


\( P(X) \). According to [245][246][250], \( \lambda - \text{dependency} \) of the decision table can be defined as follows:

**Definition 2.7** \( \lambda - \text{dependency} \) of the decision table is defined as the normalized expected degree of deviation of the conditional probability \( P(X|E) \) from the prior probability \( P(X) \) in the approximation space \((U, U/C)\). That is:

\[
\lambda(X | C) = \frac{\sum_{E \in U/C} P(E)\left|P(X | E) - P(X)\right|}{2P(X)(1 - P(X))},
\]

where \( 2P(X)(1 - P(X)) \) is used for the purpose of normalization, \( U/C \) is the partition induced by the condition attributes \( C \) in the universe \( U \) and the elementary sets \( E \) are belonging to the partition \( U/C \), i.e. \( E \in U/C \).

Absolute positive region \( POS^*(X) \) is defined in formula (2.17). For the probabilistic decision table, the absolute positive region of the two class partitions corresponding to \( X \) and \( \neg X \) in the approximation space \((U, U/C)\) can be defined as:

\[
POS^*(X | C) = \cup\{E : P(X | E) > P(X)\},
\]

where \( C \) is the set of condition attributes, \( E \in U/C \) and the target set \( X \in U/D \).

Correspondingly, \( \lambda - \text{dependency} \) for \( POS^*(X | C) \) can be defined as:

**Definition 2.8** If \( POS^*(X | C) > 0 \), the \( \lambda - \text{dependency} \) for \( POS^*(X | C) \) is defined as the normalized expected degree of deviation of the conditional probability \( P(X|E) \) from the prior probability \( P(X) \) in the absolute positive region \( POS^*(X | C) \) of the target set \( X \) in the approximation space \((U, U/C)\). Otherwise, its value is zero. That is:
\[
\lambda_{POS}^\gamma(X \mid C) = \begin{cases} 
\frac{\sum_{E \in POS^\gamma(X \mid C)} P(E) |P(X \mid E) - P(X)|}{P(POS^\gamma(X \mid C))(1 - P(X))}, & \text{if } POS^\gamma(X \mid C) > 0, \\
0, & \text{if } POS^\gamma(X \mid C) = 0,
\end{cases}
\] (2.32)

where \( P(POS^\gamma(X \mid C))(1 - P(X)) \) is used for the purpose of normalization.

In the context of the VPRS model, \( \lambda \)-dependency can be defined as the parametric \( \lambda \)-dependency. The definition is:

**Definition 2.9** The parametric \( \lambda \)-dependency is defined as the normalized expected degree of deviation of the conditional probability \( P(X \mid E) \) from the prior probability \( P(X) \) in the union of the positive and negative regions of the target set \( X \) in the approximation space \((U, U/C)\). That is:

\[
\lambda_{\gamma,u}(X \mid C) = \frac{\sum_{E \in POS_u(X \mid C) \cup NEG_u(X \mid C)} P(E) |P(X \mid E) - P(X)|}{2P(X)(1 - P(X))},
\] (2.33)

where \( POS_u(X \mid C) \) and \( NEG_u(X \mid C) \) are the positive and negative regions of the two class partitions corresponding to \( X \) and \( \neg X \) in \((U, U/C)\), and the parameters \( l \) and \( u \) are the lower limit and the upper limit used in the VPRS model.

The higher the deviation, the stronger the probabilistic correlation between conditional attributes \( C \) and the decision partition \((X, \neg X)\), with the total probabilistic independence at \( \lambda(X \mid C) = 0 \). The parametric \( \lambda \)-dependency can be calculated directly by analyzing all classes of the hierarchical partitions of the universe \( U \). Moreover, similar to \( \gamma \)-dependency, in a case where the approximation space is formed through hierarchical classification, the parametric \( \lambda \)-dependency between the partition \( R \) and the target category \( X \) can be computed via a recursive process [246]. In this case, the
following parametric $\lambda - \text{dependency}$ calculation formula must be applied recursively, starting with the leaf table of the hierarchy and continuing up to the root level table:

$$\lambda_{i,u}(X \mid R) = \lambda_{i,u}(X \mid C) + P(U')\lambda_{i,u}(X \mid R'),$$

(2.34)

where $R$ and $R'$ are “parent” and “child” level nested hierarchical partitions, $\lambda_{i,u}(X \mid R)$ and $\lambda_{i,u}(X \mid R')$ are the parametric $\lambda - \text{dependency}$ in the respective “parent” and “child” approximation spaces, where $U' = BND_{i,u}(X)$.

### 2.8 Overview of Incremental Learning

An important issue when building a pattern classifier in machine learning is the representativeness of the training samples. Unfortunately, such a dataset is not always available. For example, faces of the same person can change greatly due to such factors as expressions, make-up use, and illumination [89]. When constructing a face recognition system, it is often difficult to collect all variations of face images for each participant. This makes it necessary for a system to continuously improve the performance by learning additional information via incremental learning, and newly added objects (facial images) while retaining knowledge acquired from old data.

Generally speaking, incremental learning can be considered a process of accumulating and managing knowledge over time [16][25][198]. The incremental learning model, which can be adopted as a classifier, includes such as neural networks [21][140-141][217], support vector machines [22][40][71][164][207], decision tables [244], and decision trees [209]. According to [161], an incremental learning algorithm
should satisfy the following four conditions:

1. Additional information should be gained from new data
2. The original data does not necessarily have to be accessed in order to re-train the existing classifier
3. Previously acquired knowledge should be preserved
4. If new data introduce new classes, the algorithm should be able to accommodate the classes.

As per the aforementioned descriptions, an incremental learning technique must be able to learn new information from new data and still remember previously acquired knowledge. But, in practice, this is not always achievable because a problem called stability-plasticity dilemma [61] often arises in the incremental learning process. In order to learn new information, some acquired information may have to be lost as learning new patterns tends to overwrite formerly acquired knowledge. A completely stable classifier can securely preserve existing knowledge, but it cannot accommodate any new information, whereas, a completely plastic classifier can adeptly learn new information but it cannot conserve prior knowledge.

Thus, a large number of algorithms can still be considered incremental learning algorithms although they only partially meet these conditions. When applied to incremental learning, the algorithms typically fall somewhere in between the stability-plasticity spectrum. To clarify, the algorithms are to combine two operations: updating the classifier using only the new data in the incremental learning process and
when applicable, total re-generation of the classifier using all the data accumulated so far.

In fact, incremental learning is a term that has been widely and loosely used to describe various algorithms and models such as incremental networking growth and pruning, on-line learning, or relearning instances formerly misclassified [242-245]. Thus, other terms such as constructive learning, lifelong learning, and evolutionary learning are partially interchangeable with incremental learning.

Furthermore, methods of employing the incremental learning process can be different too. Most algorithms assume only one object at a time is added to the training set prior to the classifier being updated, but new features and new classes can also be learned from a batch of new data via the classifier. For instance, an incremental learning framework, ILTC (Incremental Learning of Text Classification), was proposed in [25]. It did not update the system knowledge while receiving an individual training object during its operation. In contrast, it would learn new knowledge from a new batch of training examples without referring to the formerly used training data [25].
3. Facial Feature Representation

This Chapter describes how to represent a facial image in the presented methodology. Pixel values are often used to represent facial images, but when applied in the presented methodology, it has some limitations. Section 3.1 explains why wavelet coefficients are selected for facial representation instead of using pixel values. Section 3.2 covers the proposed method to represent face images with several groups of wavelet coefficients. Compared with presenting an image by its pixel values, wavelet transformation can reduce the size of the image considerably, but each group of coefficients still has redundant information and correspondingly the amount of computation is still huge. Thus, a technique was proposed in Section 3.3, which can heuristically remove the redundancy from each group of coefficients to reduce the number of calculations.

3.1 Reason for Selection of Wavelet Transform for Face Representation

There are three reasons why wavelet coefficients have been selected over the alternative of using pixel values. Firstly, when constructing the hierarchies, the conditional attributes for different decision tables are selected from different groups of features representing facial images. The conditional attributes used for a “child” decision table must be different from those attributes used for the “parent” decision table [246]. That is to say, before the hierarchies can be formed, several sets of features must be
prepared. Each set of features can represent the same image independently in different decision tables, if necessary. However, this is not always easy to achieve. For instance, if a facial image is represented by its pixel values, it will be hard to tell, as a general rule, how to divide the pixel values of one image into several sets. Also, it may be difficult to determine which set of pixel values is best for each decision table.

Likewise, wavelet coefficients can work well in this situation. If the dimension of an image is 64 x 64, after it has been transformed by Haar Wavelet Transformation based on the algorithm listed in Table B.3 in Appendix B, this can utilize wavelet coefficients from up to 5 different levels. When employing the same image with a hierarchy of probabilistic decision tables, the image can be represented by coefficients from various levels. For example, coefficients from level 2 can be used as features when constructing the second decision table in the hierarchy and coefficients from level 3 can be used as features for the third decision table.

Secondly, representing facial images with wavelet coefficients can work well for incremental learning, which requires the extraction of new information from new data while still retaining the previously acquired knowledge [161]. When a new facial image is added, some statistical methods, such as PCA, will have to regenerate the presentation of each face image. The previously acquired knowledge will be deleted after regenerating the presentation. Wavelet transformation always extracts features of each facial image independently without adding new images.

Thirdly, wavelet transformation can significantly reduce the size of each image
lowering the number of computations while retaining sufficient information for recognition. The images directly represented by their pixel values contain too much redundant information, increasing their size. In other words, even if a general rule were available to divide the pixel values of one image into several sets, the number of computations would be too large.

Based on the aforementioned reasons, wavelet transformation is selected to process each face image and wavelet coefficients as facial representations.

### 3.2 Representation of Facial Images with Haar Wavelet Coefficients

In this Section, the method used for representing facial images with Haar wavelet coefficients is presented. Before facial images are transformed with Haar Wavelet Transformations, they must be pre-processed, which may include resizing all images and centering the faces. Facial images, collected for recognition, must be of the same size. If they are different, they must be re-sized. The face on each image must be centered, and their sizes should be similar to each other, as in the images in Figure 3.1.

![Figure 3.1 Sample facial images](image-url)

After pre-processing, the facial images are transformed with the non-standard
decomposition Haar Wavelet Transformation algorithm, listed in Table B.3 in Appendix B. A non-standard decomposition is chosen because the coefficients from every portion, of each level, presented a “face image” with relatively low resolution. This image has a coarse overall shape compared to the original image, but can sufficiently describe the features of the entire face. After transformation the image is represented by a set of Haar wavelet coefficients chosen from different levels. These coefficients can be used as the features representing each facial image when each decision table in the hierarchy is created.

Although coefficients from any level can be used as the potential features, coefficients at mid-range levels usually perform better [49]. Thus, only coefficients from levels 2, 3 and 4 are selected as features to represent each image. The coefficients of each image are divided into several groups, and each group is used to represent the image when it is used to build a decision table. If the wavelet coefficients are from different parts or different levels, then, as a general rule, it will be difficult to tell from which level or which part the coefficients are taken. Therefore, instead of using coefficients from different parts or levels, each image in the training set is represented by a group of coefficients from the scale, horizontal, vertical, or diagonal parts of levels 2, 3, or 4 only. Each decision table is built with coefficients from one specific part of level 2, 3, or 4. For example, if the dimension of each image is 64x64, and all of its pixels are used to represent the face, its size will be 4,096 pixels. Theoretically, the decision table formed with this facial presentation method is able to have up to 4,096 conditional attributes. If
coefficients from the scale part of level 3 are used to represent each image, a single image can be represented by 8x8 wavelet coefficients. A decision table formed for each group of coefficients can only have a maximum of 64 conditional attributes.

Following a Haar Wavelet Transformation, the size of each image has been significantly reduced and the size of the decision table has become smaller. But, the number of wavelet coefficients is still too large to build the decision table. In addition, there is a large amount of redundant information still included in the coefficients, especially the coefficients from the vertical, horizontal and diagonal parts.

**Table 3.1 Wavelet Coefficients at Level 2 and Level 3**

<table>
<thead>
<tr>
<th>Vertical part of level 2</th>
<th>Vertical part of level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>26.38 22.37 23.32 27.21</td>
<td>0 -0.54 -0.49 0.07 0.26 0.28 0.39 0.01</td>
</tr>
<tr>
<td>22.13 23.84 23.99 21.38</td>
<td>-1.63 0.18 -0.14 0.26 -0.06 -0.13 0.46 1.44</td>
</tr>
<tr>
<td>27.04 23.89 24.47 27.51</td>
<td>0.31 0.01 -0.05 -0.05 -0.07 -0.05 -0.59 0.17</td>
</tr>
<tr>
<td>29.39 23.49 23.87 29.72</td>
<td>-0.35 -0.09 -0.15 0.25 -0.33 0.12 -0.54 1.15</td>
</tr>
<tr>
<td>-4.16 -2.29 -2.94 -3.58</td>
<td>-0.76 0.44 -0.62 1.33</td>
</tr>
<tr>
<td>0.43 -0.13 -0.49 0.65</td>
<td>-0.36 0.5 0.01 0.46</td>
</tr>
<tr>
<td>1.4 0.13 0.56 1.45</td>
<td>-0.72 -0.38 0.42 1.16</td>
</tr>
<tr>
<td>-0.06 0.22 -0.11 -0.29</td>
<td>-0.09 0.2 0.02 0.29</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Horizontal part of level 2</th>
<th>Vertical part of level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 -1.09 -2.81 -2.92</td>
<td>0 -0.54 -0.18 -0.02 0.14 0.27 0.37 0.01</td>
</tr>
<tr>
<td>-1.61 0.45 0.88 0.99</td>
<td>0.14 0.3 0.01 0.18 -0.06 -0.07 -0.71 0.73</td>
</tr>
<tr>
<td>0.11 0.23 0.22 0.2</td>
<td>0.1 -0.01 -0.03 -0.03 0.02 0.03 0.04 -0.09</td>
</tr>
<tr>
<td>0.54 -0.01 -0.73 -0.18</td>
<td>-0.29 0.25 -0.33 0.38 -0.27 0.24 0.35 0.58</td>
</tr>
<tr>
<td>0.43 0.12 0.44 0.13</td>
<td>0.07 0.12 0.17 -0.09 0.24 -0.14 -0.03 -0.07</td>
</tr>
<tr>
<td>0.64 -0.04 -0.23 -0.51</td>
<td>0.09 0.02 0.01 -0.03 0.08 0.06 0.36 -0.36</td>
</tr>
<tr>
<td>0.02 0.27 0.11 0.13</td>
<td>0.01 -0.47 0.07 -0.03 -0.01 -0.12 0.11 0</td>
</tr>
<tr>
<td>0 -0.37 -0.05 -0.09</td>
<td>0 0.01 -0.11 -0.03 -0.04 0.07 0.12 0</td>
</tr>
</tbody>
</table>

| ----------Diagonal part of level 3--------|
| 0 -0.37 -0.05 -0.09 | 0 -0.2 0.06 -0.31 0 |
Following a transformation, the majority of the “detail” coefficients appear very small in magnitude, such as the example in Table 3.1 where the coefficients are selected from levels 2 and level 3 of a face image. If the coefficients are large, this indicates a boundary is available in that location of the image and the boundary can potentially be used as a valuable feature for face recognition. If the coefficients are close to zero, this indicates a uniform area in the image. Correspondingly, their contribution to the study of face recognition is very small [90][192]. Nevertheless, before using wavelet coefficients as conditional attributes, another transformation is deployed, as described in Section 3.3.

3.3 Heuristically Removing Redundant Coefficients

In this Section, the proposed method for heuristically reducing the number of Haar wavelet coefficients representing each facial image is introduced. In order to remove redundant information, a method is proposed to heuristically reduce the number of wavelet coefficients representing each face. A transformation further reduces the number of coefficients which are close to zero.

If wavelet coefficients are used for face recognition, the ability to completely reduce redundancy will become important because the redundancy is useless for recognition and can increase computational demands. In the presented methodology, wavelet coefficients are not used for face recognition directly. They are used to pre-process face images with respect to forming preliminary facial features that will be processed later by other methods. Unlike the methods in [90], the presented method does not attempt to
completely locate and reduce redundant coefficients, but does attempt to partially reduce the number of wavelet coefficients. This reduces the number of calculations and retains sufficient information from each image to ensure principal component analysis (PCA) and rough set theory can process them successfully in the future. The present method can be summarized as an algorithm having four steps, as listed in Table 3.2, and the details of the steps are described below.

**Step 1 Form a coefficient matrix of the images**

Assuming each image $I$ is represented by $l \times l = m$ Haar-based wavelet coefficients, and those coefficients can be from the scale, horizontal, vertical or diagonal parts of levels 2, 3 or 4, the coefficients can eventually be used to form a probabilistic decision table in the hierarchy at a certain layer. The definition of each image $I$ is listed below:

$$I = (x_1, x_2, \ldots, x_i, \ldots, x_m),$$

(3.1)

where each element $x_i$ is one Haar-based wavelet coefficient representing image $I$.

If $N$ images from $K$ people in the training set are used to build the probabilistic decision table in the hierarchy, the images can form an $N \times m$ matrix as defined in (3.2):

$$X_{\text{Haar}} = \begin{pmatrix}
    x_{11} & \cdots & x_{1m} \\
    \vdots & \ddots & \vdots \\
    x_{N1} & \cdots & x_{Nm}
\end{pmatrix},$$

(3.2)

or defined as:

$$X_{\text{Haar}} = [I_1, I_2, \ldots, I_i, \ldots, I_N]^T,$$

(3.3)

where $I_i$ is the $i^{th}$ image represented by Haar-based wavelet coefficients.
Step 2 Calculate the average value of each column

Once matrix $X_{Haar}$ has been built, the average value of each column denoted as $Ave_i$ in the matrix is calculated. The corresponding formula is defined in (3.4):

$$Ave_i = \frac{\sum_{j=1}^{N} x_{ij}}{N}.$$  \hspace{1cm} (3.4)

After the average value of each column is completed, the values form a vector as defined below:

$$Ave = [Ave_1, Ave_2, \ldots, Ave_i, \ldots, Ave_m]^T.$$  \hspace{1cm} (3.5)

The elements of the vector $Ave$ are sorted and ordered from the maximum to the minimum and denoted as $A'$.

Step 3 Identify which column in $X_{Haar}$ should be removed

The decision as to which column to remove from matrix $X_{Haar}$ is based upon the sorted average values in $A'$. Only columns having the average values of the first $n$ elements in vector $A'$ are selected. The remainders are rejected. The value of parameter $n$ can be determined heuristically, according to two rules:

1. If the wavelet coefficients originate from the scale part, they usually contain a large amount of useful information. Their values are usually much larger than zero. If the wavelet coefficients originate from the scale part of any level, all the coefficients are retained. The value of parameter $n$ equals the number of wavelet coefficients from each image. For instance, if the coefficients from the scale part in Table 3.1 are used, the value of parameter $n$ is 16.
Wavelet coefficients from the remaining parts contain a great deal of useless information with regard to face recognition because their values are usually much closer to zero. Hence, if wavelet coefficients are not from the scale part of any level, then approximately 30% to 60% of the coefficients are retained. The inequality is listed below:

$$0.3m < n < 0.6m,$$

where $m$ is the total number of wavelet coefficients from this part. For example, if coefficients of the vertical part of level 3 in Table 3.1 are used, the value of $n$ is 32.

**Step 4 Form a simplified coefficient matrix of all images**

After sorting and truncation, the dimension of each image becomes $n$. Each image originally represented by $m$ coefficients is now represented by $n$ coefficients, as defined in (3.7):

$$I_{\text{Haar}}'=(x_1^{\text{Haar}}, x_2^{\text{Haar}}, \ldots, x_i^{\text{Haar}}, \ldots, x_n^{\text{Haar}}),$$

where each element $x_i^{\text{Haar}}$ is one selected Haar wavelet coefficient. Correspondingly, the dimension of matrix $X_{\text{Haar}}'$ is reduced into $N\times n$, as defined in (3.8):

$$X_{\text{Haar}}' = \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{Nn} \end{pmatrix},$$

or

$$X_{\text{Haar}}' = [I_1', I_2', \ldots, I_s', \ldots, I_N']^T.$$
recognition as they are usually not significant discriminative features. Therefore, $X$ will be further processed by PCA and rough set theory. The overall goal is a probabilistic decision table having only a few conditional attributes with the most relevant features for face recognition can eventually be built.

**Table 3.2** Algorithm for Heuristically Reducing Wavelet Coefficients

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1    | **Input:** a matrix $X$ having $N$ images represented by $m$ wavelet coefficients<br>$X = \begin{pmatrix} x_{11} & \cdots & x_{1m} \\
\vdots & \ddots & \vdots \\
 x_{N1} & \cdots & x_{Nm} \end{pmatrix}$<br>2 | define a 1-D array $A$
| 3    | for ($i = 0, i < m, i++$)<br>for ($j = 0, j < N, j++$)<br>$A[i] = \frac{\sum_{j=1}^{N} x_{ij}}{N}$<br>end for<br>end for | sort array $A$ and arrange them from max to min:<br>$A = [A_{\text{max}}, \ldots, A_{\text{min}}]$<br>4 | if $X$ is from scale part<br>return $X$
| 5    | else<br>sort $X$ based on values of $A$
remove last $(m-n)$ columns from $X$
return $X$ |
4. Facial Feature Extraction and Discretization

Chapter 4 outlines how facial features may be extracted from images and discretized as conditional attributes for probabilistic decision tables. Crisp cut [115] is a widely-used approach for discretizing the features from real-valued numbers into binary-valued numbers. But, it cannot appropriately deal with the situations if a feature value is very close to the cut point and possibly affected by random noise. In such a case, the crisp cut often assigns a wrong value to the feature. In order to overcome this type of limitation, a new method of discretizing real-valued facial features, named soft-cut discretization, was proposed. It would allow for accommodation of the aforementioned situations during the discretization and recognition stages. In Section 4.1, the steps for applying principal component analysis (PCA) to extract facial features are covered. The application of soft-cut discretization to discretize the facial features is introduced in Section 4.2. The theoretical aspects of soft-cut discretization are described in Section 4.3.

4.1 Steps for Applying PCA to Extract Facial Features

In the presented methodology, when applying PCA to extract facial features, the steps proposed by Turk and Pentland [197] were followed. The thorough discussion on the implementation of the steps in the context of the presented methodology is introduced in this Section. The purpose of applying PCA in the presented methodology is not to
directly build a PCA-based classifier, but extract facial features from each image by representing each image using eigenvectors. Therefore, it did not totally implement the step of classifying a new facial image using eigenfaces based on “Euclidian distance-based classification” as introduced in [197].

Step 1 Prepare facial images as the training set

Before being processed by PCA, the face images must be centered and be of the same size if they are represented by pixels. In the presented methodology, because each facial image has been processed by wavelet transformation, according to formula (3.7), each image \( I \) is directly represented as Haar wavelet coefficients as listed below:

\[
I_{Haar}^i = (x_{1, Haar}^i, x_{2, Haar}^i, \ldots, x_{n, Haar}^i, \ldots, x_{n, Haar}^i),
\]

where \( x_i \) is a Haar wavelet coefficient, and there are a total of \( n \) coefficients for each image.

According to formula (3.8), the total set of facial images as the training set is represented by a matrix \( X_{Haar} \) as listed below:

\[
X_{Haar} = \begin{pmatrix}
x_{11} & \cdots & x_{1n} \\
\vdots & \ddots & \vdots \\
x_{N1} & \cdots & x_{Nn}
\end{pmatrix},
\]

where \( N \) is the total number of images in the training set.

Step 2 Represent each image in the training set as a vector

Each image \( I_i \) is defined as a vector \( I_i \) as shown in (4.1) and the dimension of such a vector is \( n \)-dimension. The components of \( I_i \) are Haar wavelet coefficients representing image \( I_i \):
Step 3 Calculate the “average” face vector $\Psi$

The elements of the “average” face vector $\Psi$ can be calculated based on formula (C.1) defined in Appendix C, and can be defined as:

$$\Psi = \frac{1}{N} \sum_{i=1}^{N} \Gamma_i.$$  \hspace{1cm} (4.2)

Each component of $\Psi$ defined as $\psi_j$, can be calculated using the formula below:

$$\psi_j = \frac{1}{N} \sum_{i=1}^{N} x_{ij}^{Haar},$$  \hspace{1cm} (4.3)

where $x_{ij}^{Haar}$ is the $j^{th}$ Haar coefficient of image $I_i$.

Figure 4.1 An “average” image cited from [197]

If each image is represented by its pixel values, such an “average” face will be a face image that is similar to, but different from, all facial images in the training set, such as the sample “average” image from [197] listed in Figure 4.1. The presented methodology represented each face by a small group of wavelet coefficients, so the resolution of the “average” face is too low to construct a facial image similar to the image from [197].

Step 4 Subtract the mean face

Each vector $\Gamma_i$ will be subtracted by a mean vector defined as $\Phi_i$:

$$\Phi_i = \Gamma_i - \Psi.$$  \hspace{1cm} (4.4)

That is, each component of $\Phi_j$ is defined as:
\[ \phi_j = \phi_{ij}^H - \psi_j. \]  

(4.5)

Then, this set of vectors \{\Phi_1, \Phi_2, \ldots, \Phi_i, \ldots, \Phi_{N-1}, \Phi_N\} is used to seek a set of \(N\) orthonormal vectors that can best describe the distribution of the data. If such a set is denoted as \(\overline{U}\), its vectors can be selected such that

\[ \lambda_k = \frac{1}{N} \sum_{i=1}^{N} (\overline{U}_k \Phi_i)^2, \]  

(4.6)

is a maximum, subject to

\[ \overline{U}_i^T \overline{U}_k = \delta_{ik}, \quad \begin{cases} 1 & i = k, \\ 0 & \text{otherwise}. \end{cases} \]  

(4.7)

Here, the vectors \(\overline{U}_k\) and scalars \(\lambda_k\) are the eigenvectors and eigenvalues of the covariance matrix of the images in the training set as defined in (4.8).

**Step 5 Compute the covariance matrix \(C\) of the training set**

The covariance matrix can be defined as

\[ C_{mn} = \frac{1}{N} \sum_{i=1}^{N} \Phi_i \Phi_i^T = AA^T, \]  

(4.8)

where the matrix \(A = [\Phi_1, \Phi_2, \ldots, \Phi_i, \ldots, \Phi_{N-1}, \Phi_N]\). However, if formula (4.8) is applied, this means the level of computation can still be very significant in some cases.

If an image is \(n\)-dimensional, the covariance matrix in (4.8) will be an \(n \times n\) matrix. If the images are presented by their pixel values, (128x128 of each image), the size of the covariance matrix will be 16384x16384. Even if each image is represented by the wavelet coefficients such as the coefficients from the scale part at level 4, the size of an image is 256 coefficients, and the covariance matrix \(C\) will be a 256 x 256 matrix. Thus, the
method in [197] was implemented to calculate the covariance matrix and its basic idea was described in the following paragraph.

Firstly, a matrix $A^T A$ is constructed, and its eigenvector, defined as $v_i$, will be computed based on the formula below:

$$A^T A v_i = \mu_i v_i.$$  \hspace{1cm} (4.9)

Then, if pre-multiplying both sides of (4.9) by $A$:

$$A A^T A v_i = \mu_i A v_i.$$ \hspace{1cm} (4.10)

According to formula (4.8):

$$C^{nnn} = A A^T.$$ 

$A v_i$ are in fact the eigenvectors and $\mu_i$ are the eigenvalues of the covariance matrix $C$.

Hence, $N \times N$ matrix denoted as $L^{ NN} = A^T A$ can be constructed, where $L_{ij} = \Phi_i^T \Phi_j$ and the $N$ eigenvectors are denoted as $v_i$ of $L^{ NN}$. The vectors can determine the linear combination of $N$ images in the training set to form the eigenfaces:

$$u_i = \sum_{k=1}^{N} v_{ik} \Phi_k,$$ \hspace{1cm} (4.11)

where $l = 1, \cdots, N$, and $v_{ik}$ is the $k^{th}$ value of the $l^{th}$ eigenvector $v_i$. Compared to the size of an image, the number of images in a training set are usually smaller (i.e. $N \ll n$), so the computation time can be reduced. Once eigenvectors are available, they are ranked based on the values of the corresponding eigenvalues in order to characterize the variations among the facial images. In practice, only $r$ eigenvectors that correspond to the $r$ largest
eigenvalues should be kept, where \( r \leq N \). If images are represented by pixels, after transformation, they will appear to be the sample images cited from [197], as listed in Figure 4.2. Similarly, the presented methodology represented each face by a small group of wavelet coefficients, so the resolution of the eigenfaces is too low to construct a facial image similar to the image from [197].

**Step 6 Represent facial images using eigenvectors**

Each facial image denoted as \( \Phi_i \) in the training set can be represented as a linear combination of the \( r \) eigenvectors:

\[
\Phi_i = \sum_{j=1}^{r} w_j \mu_j,
\]

(4.12)

where \( w_j = \mu_j^T \Phi_i \).

The total training set can be represented by a PCA-based matrix denoted as \( X^{pca} \) in (4.13):

\[
X^{pca} = \begin{pmatrix}
  w_{11}^{pca} & \cdots & w_{1r}^{pca} \\
  \vdots & \ddots & \vdots \\
  w_{N1}^{pca} & \cdots & w_{Nr}^{pca}
\end{pmatrix}.
\]

(4.13)

If each facial image is represented by a vector \( I_i^{pca} \) as shown in (4.14):

\[
I_i^{pca} = (w_{i1}^{pca}, w_{i2}^{pca}, \ldots, w_{ij}^{pca}, \ldots, w_{ir}^{pca}),
\]

(4.14)

where \( i = 1, 2, \ldots, N \), each components of \( I_i^{pca} \) will be one PCA-based feature of that facial image denoted as \( w_{ij}^{pca} \).
Theoretically, the number of PCA-based features $r$ can be the same as the number of facial images $N$ in the training set (i.e. let $r = N$). However, because the purpose for processing each facial image using PCA is to extract a small group of facial features useful in face recognition, its size must be limited to a reasonable number during the transformation. Methods of deciding how many of the eigenvector to retain are many and usually can be classified in three categories. The first of these categories removes the last 40% of the eigenvectors [124][218], which is a heuristic threshold primarily selected by experience. The second uses a threshold, typical 0.9, to decide the minimum number of eigenvectors [92][97][218]. The first $i$ eigenvectors will be kept if the ratio of the sum of all eigenvalues up to and including $i$ over the sum of all the eigenvalues is greater than 0.9. The third of these categories depends upon the ratio of the $i^{th}$ eigenvalue $\lambda_i$ over the largest eigenvalue $\lambda_1$ [92][218]. The first $i$ eigenvectors will be retained if the ratio is greater 0.01. Furthermore, in [177], Sirovich and Kirby reported that, based on the procedure proposed in [197], 40 eigenvectors would be sufficient for a very good description of a training set containing 115 facial images of Caucasian males. Similarly,
Turk and Pentland reported in many of their test cases, if a training set had 16 face images, the number of eigenvectors would be 7 [197].

All in all, taking all factors above into consideration, the number of eigenvectors in the presented methodology is primarily decided based upon the number of images used to build a probabilistic decision table in a hierarchy, heuristically. Initially, the number of eigenvectors, $r$, are specified and must be less than 50 percent of the total number of facial images. Next, based on a specific number of facial images $N$, in a training set, when building a decision table, the number of eigenvectors $r$ will be set according to the following rules:

1. if $N \geq 100$, $r$ will be set as 30,
2. if $75 \leq N < 100$, $r$ will be set as 24,
3. if $50 \leq N < 75$, $r$ will be set as 18,
4. if $30 \leq N < 50$, $r$ will be set as 16,
5. if $N < 30$, $r$ will be set as 10.

For example, when 200 images are used as a training set, according to the description in Step 5, the number of Eigenvectors is 200. According to the aforementioned rules, the number of eigenvectors $r$ will be set as 24, and correspondingly, after the transformation, each image will be represented by 24 PCA-based features when used to build a decision table in the hierarchy.

In other words, if the dimension is limited to $p (p < r)$, then each image represented by $p$ PCA-based features is denoted as:
\begin{align}
I_i^{pca} &= [w_{i1}^{pca}, \ldots, w_{ij}^{pca}, \ldots, w_{ip}^{pca}]^T, \\
\text{and the matrix } X^{pca} \text{ will become:}
\begin{align*}
X^{pca} &= \begin{pmatrix}
    w_{11}^{pca} & \cdots & w_{ip}^{pca} \\
    \vdots & \ddots & \vdots \\
    w_{N1}^{pca} & \cdots & w_{Np}^{pca}
\end{pmatrix}.
\end{align*}
\end{align}

\textbf{Step 7 Use eigenfaces to represent an unknown face image}

Once the facial images in a training set are processed and each one is represented by a vector as shown in (4.12), the eigenfaces can be used to classify an unknown facial image as in [197]. This step was not entirely implemented in the presented methodology, but methods in (4.17), (4.18) and (4.19) were used to pre-process the unknown image. Nevertheless, it is described here for the sake of completeness.

Given an unknown image \( \Gamma_{\text{new}} \) that has been centered and of the same size \( n \) pixels as the training images, it will be subtracted by mean vector first, as shown in (4.3):
\begin{equation}
\Phi_{\text{new}} = \Gamma_{\text{new}} - \Psi. 
\end{equation}

Then, \( \Phi_{\text{new}} \) is projected in the eigenspace as shown in (4.18):
\begin{equation}
\Phi_{\text{new}} = \sum_{j=1}^{p} w_j u_j, 
\end{equation}

where \( w_j = u_j^T \Phi_{\text{new}} \). Thus, \( \Phi_{\text{new}} \) is represented as a vector as shown in (4.19):
\begin{equation}
\Omega_{\text{new}} = [w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{ip}]^T. 
\end{equation}

Finally, Euclidean distance between the unknown image and each image in the training set will be calculated. The new image will be classified into the same class with the image in the training set having a minimum distance, if such a distance is less than a
4.2 Discretizing PCA-Based Features with Soft-Cut Discretization Technique

In this Section, a method to discretize PCA-based features is introduced. After Steps 1 to 6 described in Section 4.1 have been applied to each facial image, originally represented by a group of selected Haar wavelet coefficients in the training sets, each facial image is represented by a group of $p$ PCA-based features as shown by (4.15) in Section 4.1. A group of $p$ PCA-based features can be considered as a small group of extracted facial features that can be potentially used to form a decision table in the hierarchy.

Although the number of features representing each facial image has been reduced, and the extracted facial features are more appropriate, the features are still not useful in forming a decision table. Following transformation by PCA, its elements are real-valued, and the conditional attribute values of a probabilistic decision table in a hierarchy must be binary-valued. This means that the extracted facial features must be discretized into binary-valued first.

The techniques for discretization are many, and a traditional and widely-used approach is crisp cut. Although it is easy to implement, usually crisp cut cannot deal well with situations where a feature value is close to the cut point and possibly affected by
random noise. In such a case, a traditional crisp cut method can often assign a wrong value to that feature. In order to accommodate such situations, a novel discretization method called *soft-cut discretization* is proposed and applied in the presented methodology. The basic idea is similar to the idea in [47]. On the other hand, unlike the support vector machine [20], the method does not attempt to find a threshold value so that all principal components can be discretized because, in practice, this is not always easily achieved. A worst case scenario might be when a threshold value is completely unavailable. Thus, as described in the next Section, initially, *soft-cut discretization* splits the real axis of each dimension and only those principal components, which can be classified into two specific intervals, are then discretized. Compared to traditional crisp cut, this method is able to allow for accommodation of the above noted situations during the discretization and recognition stages. When the feature value is close to the cut point and possibly affected by random noise, often a feature cannot be discretized by the method. If this is the case, the image will be classified into the boundary region of the decision table, and will be discretized again when working on a decision table of the next layer in the hierarchy.

More specifically, *soft-cut discretization* can be summarized as the algorithm listed in Table 4.1. Its theoretical aspects are introduced in the next Section. The procedure of discretizing each image in the matrix $X^{\text{pca}}$ having PCA-based features from $N$ people can be described as:
Step 1. Define a threshold \( \tau \)

The first step in discretization is to define a threshold \( \tau \). Its value should satisfy \( 0 < \tau < 0.5 \) and can be identified in different ways, the detail of which is introduced in the next section. The threshold \( \tau \) is used to assist in deciding the size of each interval on each real axis of each dimension for the purpose of discretization. Although its value is not fixed, based on the experimental results, if the threshold value is close to 0, the quality of the corresponding decision table improves.

Step 2 Transform each PCA-based feature with a sigmoid function

After the threshold value was defined, each PCA-based feature \( x_{ij}^{pca} \) of each image \( I_{ij}^{pca} \) is transformed according to the following sigmoid function, as shown in (4.21):

\[
f(x) = \frac{1}{1 + e^{a(x-c)}}.
\]

(4.21)

When function (4.21) is applied for discretization, \( x \) is a PCA-based feature (i.e. \( w_{ij}^{pca} \) of each image \( I_{ij}^{pca} \)), \( a \) is a parameter, the value of which can be identified heuristically. Based on the experimental results [27], its value has no significant impact on the quality of the decision table, so it is randomly defined as \( a=2 \). Parameter \( c \) is the arithmetic average of the selected PCA-based feature of all images. If the \( i^{th} \) PCA-based feature of the \( j^{th} \) image in the training set is represented as \( w_{ij}^{pca} \) and there are \( N \) images, the value of parameter \( c \) is the arithmetic average of the selected principal component of all images in the training set and can be identified by the following formula:
\[
    c_i = \frac{\sum_{j=1}^{N} w_{ij}^{pca}}{N}.
\]

(4.22)

Before being transformed by the sigmoid function, the value of each PCA-based feature \( w_{ij}^{pca} \) is real-valued. This satisfies \(-\infty < w_{ij}^{pca} < +\infty\). Following the transformation, the value of each PCA-based feature, defined as \( f_{ij}(x) \), becomes real-valued, but satisfies \( 0 < f_{ij}(x) < 1 \), as shown in Figure 4.1.

**Table 4.1 Algorithm of soft-cut discretization**

1. defining a threshold \( \tau \) that can satisfy \( 0 < \tau < 0.5 \)
2. input matrix \( X^{pca} \) having PCA-based features from \( N \) people
   \[
   X^{pca} = \begin{pmatrix}
   w_{11}^{pca} & \ldots & w_{p}^{pca} \\
   \vdots & \ddots & \vdots \\
   w_{N}^{pca} & \ldots & w_{Np}^{pca}
   \end{pmatrix}
   \]
3. calculate the arithmetic average \( c_i \) of each principal component of all images
   for \( i=0, i<N, i++ \)
   
   \[ c_i = \frac{\sum_{j=1}^{N} w_{ij}^{pca}}{N} \]
   end for
4. for \( j=0, j<N, j++ \)
   for \( i=0, i<p, i++ \)
   transform each element \( w_{ij}^{pca} \) in \( X^{pca} \) by sigmoid function
   \[
   f(x) = \frac{1}{1 + e^{\alpha(c-x)}}
   \]
   define transformed \( w_{ij}^{pca} \) as \( f_{ij}(x) \)
   define discretized \( f_{ij}(x) \) as \( f(x)_{ij}^{discr} \)
   if \( f(x)_{ij} \geq 0.5 + \tau \)
   then \( f(x)_{ij}^{discr} = 1 \)
   else if \( f(x)_{ij} \leq 0.5 - \tau \)
   then \( f(x)_{ij}^{discr} = 0 \)
   else if \( 0.5 - \tau < f(x)_{ij} < 0.5 + \tau \)
   then no assignment is made
   end if
   if \( f(x)_{ij}^{discr} \neq 1 \) AND \( f(x)_{ij}^{discr} \neq 0 \)
   then remove that image from \( X^{pca} \)
   end if
   end for
   end for
5. return \( X^{pca} \)
Step 3 Discretizing transformed PCA-based features

After each PCA-based feature is transformed by function (4.21), it is discretized based on the comparison result between \( f(x) \) and the threshold \( \tau \), defined in Step 1, according to the rules below:

1. if \( f(x) \geq 0.5 + \tau \), then the selected PCA-feature is assigned 1 as its value.
2. if \( f(x) \leq 0.5 - \tau \), then 0 is assigned as its value.
3. if \( 0.5 - \tau < f(x) < 0.5 + \tau \), then no assignment is made, which means a PCA-based feature cannot be discretized.

According to the above discretization strategy, the choice of the reference point 0.5 is due to the fact that \( f(c) = 0.5 \), where \( c \) is given by formula (4.22). If applying the aforementioned rules, it is obvious some PCA-based features cannot be discretized. Only those PCA-based features classified into two specific intervals, \( f(x) \geq 0.5 + \tau \) or \( f(x) \leq 0.5 - \tau \), can be discretized. Correspondingly, if a facial image has one, or more than one PCA-based feature that cannot be discretized, it cannot be used as the image in the training set for constructing the decision table in the hierarchy.

Figure 4.3 Samples of soft-cut discretization
Therefore, after discretization, the entire training set is observed and the facial images are classified into two sets, a discretized set and a non-discretized set. Only those images classified into the discretized set are kept and used to build the probabilistic decision table. If there are $N$ facial images in the training set initially, each image is represented by $p$ PCA-based features, and $Q$ images can be totally discretized. Then the images classified into the discretized set can be represented by a $Q \times p$ matrix denoted as:

$$
X^{\text{pca discr}} = \begin{pmatrix}
 x'_{11} & \ldots & x'_{1p} \\
 \vdots & \ddots & \vdots \\
 x'_{Q1} & \ldots & x'_{Qp}
\end{pmatrix},
$$

(4.23)

where $x'_{ji} = 0$ or $1$. Then, the discretized PCA-based features of the $Q$ images will be evaluated to select the most appropriate features with which to build the decision table.

The images classified into the non-discretized set will not be evaluated, which means they will not be involved in forming the probabilistic decision table at the current layer. However, the images are still useful when building the decision table at the next layer. As described in Section 3.2, when forming a decision table in the hierarchy, the features representing each facial image are a group of wavelet coefficients from the scale part, vertical part, or horizontal part of a certain level. Different groups of the facial features will be used to construct the decision table at different layers. Although an image cannot be discretized when constructing the decision table at the current layer, it is very possible it can be completely discretized when constructing the decision table at the next layer.
because, in that case, the same image will be represented by another group of facial features. Thus, after the decision table at the current layer has been constructed, images in the non-discretized set are automatically classified into the boundary region of the decision table. They are considered again when working on the images in the boundary region in order to build the next probabilistic decision table in the hierarchy.

4.3 Theoretical Aspects of Soft-Cut Discretization

In this Section, the theoretical aspects of soft-cut discretization are investigated, including its formal definitions, how to identify the value of threshold $\tau$ in each dimension and their properties.

4.3.1 Formal Definition and Properties

Assuming there is a set of images denoted as $T_{rain} = \{I_1, \ldots, I_i, \ldots, I_N\}$ and each image is represented by a set of $p$ real-valued features (for example, PCA-based features) denoted as $x_1, \ldots, x_j, \ldots, x_p$, such that $-\infty < x_j < +\infty$, then the features form a $p$-dimensional feature space and each image $I_i$ can be considered a point in such a feature space. Based on the aforementioned assumption, the following theorems can be formulated:

**Theorem 4.1** For any $x_j$ such that $-\infty < x_j < +\infty$, $0 < f(x_j) < 1$. Also, when $x_j = c_j$, $f(x_j) = 0.5$, where $c_j$ is the arithmetic average of that feature of all images in set $T$.

**Proof:** The domain of $x_j$ is $-\infty < x_j < +\infty$, so
(1) when \( x_j \) is approaching \(+\infty\), \( \lim_{x \to +\infty} f(x_j) = \lim_{x \to +\infty} \frac{1}{1 + e^{a(c_j-x_j)}} = 1 \),

(2) when \( x_j \) is approaching \(-\infty\), \( \lim_{x \to -\infty} f(x_j) = \lim_{x \to -\infty} \frac{1}{1 + e^{a(c_j-x_j)}} = 0 \),

(3) if \( x_j = c_j \); \( f(x_j) = \frac{1}{1 + e^{a(c_j-x_j)}} = \frac{1}{1 + e^{0}} = 0.5 \).

Therefore, following transformation by the sigmoid function as defined in (4.21), the range of the feature \( x_j \) is always \( 0 < f(x_j) < 1 \), and \( f(x_j) = 0.5 \), if \( x_j = c_j \). □

Based on Theorem 4.1, the soft-cut discretization can be defined as:

**Definition 4.1** A soft cut is a pair \( Sft = < f(x_j), \tau_j > \), where \( x_j \) is a real-valued feature, \( f(x_j) \) is the value of \( x_j \) transformed by the sigmoid function, and \( \tau_j \) satisfying \( 0 < \tau_j < 0.5 \) is a parameter used to define an interval \( (0.5-\tau_j, 0.5+\tau_j) \) called the uncertain interval. The soft-cut discretization function is defined as a function \( D(x_j) \) such that:

\[
D(x_j) = \begin{cases} 
1, & \text{if } f(x_j) \geq 0.5 + \tau_j, \\
0, & \text{if } f(x_j) \leq 0.5 + \tau_j.
\end{cases}
\]  

(4.24)

A real-valued feature \( x_j \) is said to be discretizable into a binary-valued feature denoted as \( D(x_j) \) by soft-cut discretization function if \( f(x_j) \) is not in the uncertain interval.

Based on Definition 4.1, the soft cut discretization has the following properties:

**Theorem 4.2** Soft-cut discretization splits the real axis of each dimension of the feature space into five intervals which are \((-\infty, 0], (0, 0.5-\tau_j], (0.5-\tau_j, 0.5+\tau_j), [0.5+\tau_j, 1), \) and \([1, +\infty)\).

**Proof:** (1) Based on Theorem 4.1, after being transformed by the sigmoid function (4.21),
the range of a feature $x_j$ is $0 < f(x_j) < 1$, which means that, after transformation, each feature $x_j$ of every image in each dimension is only in the interval $(0, 1)$. Thus, the real axis of each dimension in the feature space is divided into three intervals: $(-\infty, 0]$, $(0, 1)$, and $[1, +\infty)$ based on $f(x_j)$.

(2) According to Definition 4.1, if feature $x_j$ can be discretized by \textit{soft cut discretization}, $f(x_j)$ must be $f(x_j) \leq 0.5 - \tau_j$ or $f(x_j) \geq 0.5 + \tau_j$. That is, $f(x_j)$ must be in either the interval $(0, 0.5 - \tau_j]$ or the interval $[0.5 + \tau_j, 1)$. Hence, based on whether a feature can be discretized by \textit{soft-cut discretization}, the interval $(0, 1)$ is divided into three intervals, which are $(0, 0.5 - \tau_j)$, $(0.5 - \tau_j, 0.5 + \tau_j)$, and $[0.5 + \tau_j, 1)$.

In summation, when \textit{soft-cut discretization} is applied, it splits the real axis of each dimension of the feature space into five intervals. □

\textbf{Theorem 4.3} Given a real-valued feature $x_j$ satisfying $-\infty < x_j < +\infty$ and having been transformed by the sigmoid function as defined in (4.21).

If $f(x_j)$ can be classified into one of two specific intervals $(0, 0.5 - \tau_j]$ or $[0.5 + \tau_j, 1)$ of the real axis of that dimension in the feature space, then the feature $x_j$ can be discretized into a binary-valued feature by \textit{soft-cut discretization} function as defined in (4.24).

\textbf{Proof:} According to the definition of \textit{soft-cut discretization}, a real-valued feature $x_j$ can be discretized by (4.24) if $f(x_j) \leq 0.5 - \tau_j$ or $f(x_j) \geq 0.5 + \tau_j$. Based on Theorem 4.1, the range of $f(x_j)$ is $0 < f(x_j) < 1$. If $f(x_j)$ is in one of two specific intervals, either $(0, 0.5 - \tau_j]$ or $[0.5 + \tau_j, 1)$, the feature $x_j$ can be discretized into a binary-valued feature by (4.24). □
Theorem 4.4 Given a real-valued feature $x_j$.

If $f(x_j) = 0.5$, then it can never be discretized into a binary-valued feature by soft-cut discretization.

Proof: According to the definition of soft-cut discretization, because parameter $\tau_j$ is $0 < \tau_j < 0.5$, the inequality $0.5 - \tau_j < 0.5 < 0.5 + \tau_j$ always can hold. Thus, no matter how small the value of $\tau_j$ is, $f(x_j) = 0.5$ is always a point in the interval $(0.5 - \tau_j, 0.5 + \tau_j)$. Hence, it can never be discretized. \( \square \)

4.3.2 Selecting the Value of Threshold $\tau$

According to the definition, when soft-cut discretization is applied, $\tau_j$ is the threshold that determines the size of the uncertain interval and thereby determines if a feature $x_j$ can be discretized in that dimension of the feature space. If it is located in the uncertain interval $(0.5 - \tau_j, 0.5 + \tau_j)$, the feature $x_j$ will be considered as a feature that is unable to be discretized. Its value is close to the crisp cut point 0.5 and has probably been affected by noise.

When a set of images is discretized, the value of $\tau_j$ can be identified in different ways. For example, its value can totally be identified heuristically. Randomly selected values will be tried first, and then the value which can generate the best result (for instance, the maximum number of images that can be discretized) will be selected as the value of $\tau_j$. This method is easy to implement, but it is time-consuming and difficult to estimate how many images can be discretized before implementation.
In this section, a method of estimating the value of $\tau_j$, based on the value of $f(x_j)$ of each image, is introduced and its procedure is described as:

**Step 1 Pre-process the images**

When given a set of $N$ images, denoted as $I = \{I_1, \ldots, I_i, \ldots, I_N\}$, represented by a set of $p$ real-valued features $\{x_1, \ldots, x_j, \ldots, x_p\}$, they are classified into two groups denoted as $X_{\text{target}}$ and $\overline{X}_{\text{target}}$. The group $X_{\text{target}}$ contains all images of the target set $X$, and the group $\overline{X}_{\text{target}}$ contains all images of the complementary set of the target set $X$. Then, the images in $X_{\text{target}}$ and $\overline{X}_{\text{target}}$ are transformed by the sigmoid function $f(x_j)$, as defined in (4.21). The images can be presented by the two following matrices, assuming there are $n$ images in $X_{\text{target}}$ and $m$ images in $\overline{X}_{\text{target}}$.

$$X_{\text{target}} = \begin{bmatrix} f_{i1}(x) & \ldots & f_{ij}(x) & \ldots & f_{ip}(x) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{ni}(x) & \ldots & f_{nj}(x) & \ldots & f_{np}(x) \end{bmatrix} \quad (4.25)$$

$$\overline{X}_{\text{target}} = \begin{bmatrix} f_{i1}(\overline{x}) & \ldots & f_{ij}(\overline{x}) & \ldots & f_{ip}(\overline{x}) \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ f_{mi}(\overline{x}) & \ldots & f_{mj}(\overline{x}) & \ldots & f_{mp}(\overline{x}) \end{bmatrix} \quad (4.26)$$

**Step 2 Decide the initial value of $\tau$ in each dimension**

When deciding the value of $\tau$ in each dimension, the elements of the corresponding column in $X_{\text{target}}$ and $\overline{X}_{\text{target}}$ will be selected and added to two 1-dimensional arrays denoted
as $X_{1D}$ and $\overline{X}_{1D}$. That is, when deciding the value of $\tau_j$ in the $j^{th}$ dimension, the elements in the $j^{th}$ column of $X_{target}$ and $\overline{X}_{target}$ will be selected. After that, the elements in $X_{1D}$ and $\overline{X}_{1D}$ are sorted so that the first element is the minimum and the last element is the maximum in each array. Next, the initial value of $\tau_j$ will be decided according to the value of each element in $X_{1D}$ and $\overline{X}_{1D}$. Such an initial value must ensure that all images in $(0, 0.5-\tau_j]$ or $[0.5+\tau_j, 1)$ have the same identity. For example, all images in $(0, 0.5-\tau_j]$ are from the target set $X$, and all images in $[0.5+\tau_j, 1)$ are from the complementary set.

Based on the data distribution of $X_{1D}$ and $\overline{X}_{1D}$ in the $j^{th}$ dimension of the feature space, as listed in Tables 4.2 and 4.3, an initial value will be assigned to $\tau_j$. In this case, if all the elements of $X_{1D}$ are in $(0, 0.5)$, and all elements of $\overline{X}_{1D}$ are in $(0, 1)$, some of the elements of $\overline{X}_{1D}$ can be in $(0, 0.5]$ and the remaining elements of $\overline{X}_{1D}$ can be in $(0.5, 1)$. As shown in Case 3 of Table 4.2, if $X_{1D}[0] < \overline{X}_{1D}[0]$, the initial value of $\tau_j$ can be $\tau_j = 0.5 - \overline{X}_{1D}[0]$. According to the definition of soft-cut discretization, the interval $(0.5 - \tau_j, \tau_j + 0.5)$ is the uncertain interval, so if $\tau_j = 0.5 - \overline{X}_{1D}[0]$, the uncertain interval becomes $(\overline{X}_{1D}[0], -\overline{X}_{1D}[0])$. That is, an element of $X_{1D}$ or $\overline{X}_{1D}$ can be discretized if it is not in the interval $(\overline{X}_{1D}[0], -\overline{X}_{1D}[0])$.

**Step 3 Decide the final value of the parameter $\tau$ in each dimension**

When Variable Precision Rough Set Theory is applied, as described in Section 2.6, the objects in each elementary set can have different decision attribute values. If the feature $x_j$ is selected as one of the conditional attributes of the probabilistic decision table,
the images in \((0, 0.5-\tau_j)\) or \([0.5+\tau_j, 1)\) of that dimension can be from both the target set \(X\) and the complementary set. Hence, after the initial value of \(\tau_j\) is determined, its value will be modified further.

The basic idea of such a modification is to heuristically reduce the initial value of \(\tau_j\) so that some of the images initially in \((0.5-\tau_j, 0.5+\tau_j)\) can be moved into \((0, 0.5-\tau_j]\) or \([0.5+\tau_j, 1)\) so as to be discretized in that dimension. In practice, the final value of \(\tau_j\) will primarily be selected based on three factors. Firstly, if a feature is very close to the crisp cut point 0.5, the possibility that its value has been affected by noise is relatively high. The final value of \(\tau_j\) should ensure that images having the feature very close to 0.5 are not discretized. Secondly, if the number of images of the target set \(X\) or the complementary set in \((0, 0.5-\tau_j]\) or \([0.5+\tau_j, 1)\) is relatively low in that dimension, \(\tau_j\) should be assigned a final value that can assist in increasing the presence of that category in that dimension, i.e., increase the number of images of that category in the interval \((0, 0.5-\tau_j]\) or the interval \([0.5+\tau_j, 1)\). Thirdly, if the initial value of \(\tau_j\) is very close to 0.5, its value should be reduced very carefully to avoid a case wherein some of features which have been affected by noise are moved into \((0, 0.5-\tau_j]\) or \([0.5+\tau_j, 1)\).

**Step 4 Estimate the discretization rate in each dimension**

In the \(j^{th}\) dimension of the feature space, the discretization rate can be defined as:

\[
Discr_j = 1 - \frac{l+k}{n+m},
\]

where \(n\) is the total number of elements in \(X_{1D}\), \(l\) is the total number of elements in \(X_{1D}\).
that cannot be discretized, \( m \) is the total number of elements in \( \overline{X}_{1D} \), and \( k \) is the number of elements in \( \overline{X}_{1D} \) that cannot be discretized.

When the presented method is applied, the discretization rate \( Discr_j \) in the \( j^{th} \) dimension of the feature space can be estimated according to Theorem 4.5:

**Theorem 4.5** Given two 1-dimensional arrays denoted as \( X_{1D} \) and \( \overline{X}_{1D} \) containing the \( n \) sorted elements of the \( j^{th} \) column of \( X_{\text{target}} \) and the \( m \) sorted elements of the \( j^{th} \) column of \( \overline{X}_{\text{target}} \).

Based on the value of \( \tau_j \) and the data distribution of \( X_{1D} \) and \( \overline{X}_{1D} \) in interval \((0, 1)\), the maximum discretization rate in the \( j^{th} \) dimension of the feature space can be 1 and the minimum discretization rate can be estimated in different ways as shown below:

(1). when both \( X_{1D} \) and \( \overline{X}_{1D} \) are in the interval \((0, 0.5)\), if \( X_{1D}[0] < \overline{X}_{1D}[0] \) then

\[
Discr_j \geq 1 - \frac{l + m}{n + m}, \quad \text{otherwise} \quad Discr_j \geq 1 - \frac{n + k}{n + m};
\]

(2). when both \( X_{1D} \) and \( \overline{X}_{1D} \) are in the interval \((0.5, 1)\), if \( X_{1D}[n-1] > \overline{X}_{1D}[m-1] \) then

\[
Discr_j \geq 1 - \frac{l + m}{n + m}, \quad \text{otherwise} \quad Discr_j \geq 1 - \frac{n + k}{n + m};
\]

(3). when \( X_{1D} \) is in the interval \((0, 0.5)\) and \( \overline{X}_{1D} \) is in the interval \((0, 1)\) then

if \( X_{1D}[0] < \overline{X}_{1D}[0] \), \( Discr_j \geq 1 - \frac{l + k}{n + m} \), otherwise \( Discr_j \geq 1 - \frac{n + k}{n + m} \);

(4). when \( X_{1D} \) is the interval in \((0, 1)\) and \( \overline{X}_{1D} \) is the interval \((0.5, 1)\) then

if \( X_{1D}[n-1] < \overline{X}_{1D}[m-1] \), \( Discr_j \geq 1 - \frac{l + k}{n + m} \), otherwise \( Discr_j \geq 1 - \frac{l + m}{n + m} \);

(5). when both \( X_{1D} \) and \( \overline{X}_{1D} \) are in the interval \((0, 1)\) then

if \( X_{1D}[0] > \overline{X}_{1D}[0] \) and \( X_{1D}[n-1] > \overline{X}_{1D}[m-1] \) or
if $X_{1D}[0] < \overline{X_{1D}}[0]$ and $X_{1D}[n-1] < \overline{X_{1D}}[m-1]$, $Discr_j \geq 1 - \frac{l+k}{n+m}$;

if $X_{1D}[0] > \overline{X_{1D}}[0]$ and $X_{1D}[n-1] < \overline{X_{1D}}[m-1]$, $Discr_j \geq 1 - \frac{n+k}{n+m}$;

otherwise $Discr_j \geq 1 - \frac{l+m}{n+m}$;

(6). when $X_{1D}$ is in the interval $(0, 0.5)$ and $\overline{X_{1D}}$ is in the interval $(0.5, 1)$ or $X_{1D}$ is in the interval $(0.5, 1)$ and $\overline{X_{1D}}$ is in the interval $(0, 0.5)$, $Discr_j = 100\%$.

**Proof**: The data distribution shown in Case 3 is selected as an example to prove Theorem 4.5. With respect to other types of data distribution, Theorem 4.5 can be proven in the same way.

In Case 3, all the elements of $X_{1D}$ are in $(0, 0.5)$, some elements of $\overline{X_{1D}}$ are in $(0, 0.5]$, and the remaining elements of $\overline{X_{1D}}$ are in $(0.5, 1)$.

If $X_{1D}[0] \geq \overline{X_{1D}}[0]$, the initial value of $\tau_j$ can be $\tau_j = 0.5 - X_{1D}[0]$ from Table 4.2. All elements of $X_{1D}$ cannot be discretized because they are in the uncertain interval. Thus, the total number of elements in $X_{1D}$ that cannot be discretized is the same as the total number of elements in $X_{1D}$, which means $l = n$. On the other hand, an element in $\overline{X_{1D}}$ can be discretized if it is greater than $X_{1D}[0]$.

If assuming that $k$ images of $\overline{X_{1D}}$ are in the uncertain interval, according to (4.27), the discretization rate is:

$$Discr_j = 1 - \frac{n+k}{n+m}.$$  (4.28)

When deciding the final value of $\tau_j$, the initial value of $\tau_j$ is reduced heuristically and the number of elements in the uncertain interval can be changed in two different ways.

(1) If the number of elements is reduced, the discretization rate is increased. When
the final value of $\tau_j$ is determined, if the uncertain interval becomes empty, the value of the discretization rate $Discr_j$ will have $Discr_j = 1$. Otherwise, its value can always have:

$$Discr_j > 1 - \frac{n + k}{n + m}. \quad (4.29)$$

(2) If number of elements is not changed, it means that, when the final value of $\tau_j$ is determined, the value of discretization rate $Discr_j$ will be the same as that of (4.28).

Therefore, the discretization rate can always have:

$$1 - \frac{n + k}{n + m} \leq Discr_j \leq 1. \quad (4.30)$$

Similarly, if $X_{1D}[0] < X_{1D}^0[0]$, the initial value of $\tau_j$ can be $\tau_j = 0.5 - X_{1D}[0]$ from Table 4.2. The elements of $X_{1D}$ and $X_{1D}^0$ located in the uncertain interval cannot be discretized. If the total number of elements in $X_{1D}$ that cannot be discretized is $l$ and that in $X_{1D}^0$ is $k$, according to (4.27), its initial discretization rate is:

$$Discr_j = 1 - \frac{l + k}{n + m}. \quad (4.31)$$

When deciding the final value of $\tau_j$, if the number of elements is reduced, the discretization rate is increased. When the final value of $\tau_j$ is determined, if the uncertain interval becomes empty, the value of the discretization rate $Discr_j$ will have $Discr_j = 1$. Otherwise, its value can always have:

$$Discr_j > 1 - \frac{l + k}{n + m}. \quad (4.32)$$

Also, if the number of elements is not changed, it means that the value of discretization rate $Discr_j$ is not changed. The discretization rate can always have:

$$1 - \frac{l + k}{n + m} \leq Discr_j \leq 1. \quad (4.33)$$
Step 5 Estimate the total number of images that can be discretized in all dimensions

After the discretization rate in each dimension has been identified, the number of discretizable images in all dimensions can be determined according to the Theorem 4.6:

**Theorem 4.6** The set of discretizable images in all dimensions is an intersection of the set of discretizable images in each dimension.

**Proof:** The intersection set is a set containing elements common to two or more sets, but having no other elements. If an image can be discretized in all dimensions, it must be able to be discretized in each dimension. That is, an image is in matrix $X_{\text{discr}}^{\text{eca}}$ as shown in (4.23), it must be an image common to the set of discretizable images in each dimension.

□
Table 4.2 Determine the initial value of threshold $\tau$

1. **input** $X_{1D}$ and $\overline{X}_{1D}$

Define $TempX$ and $\overline{TempX}$ as two variables, the data type of which is real number

2. sort $X_{1D}$ and $\overline{X}_{1D}$ $X_{1D}[0]=$ min, $X_{1D}[n-1]=$ max, $\overline{X}_{1D}[0]=$ min, and $\overline{X}_{1D}[m-1]=$ max

3. **Case 1** ($X_{1D}[n-1]<0.5$ and $\overline{X}_{1D}[m-1]<0.5$)
   
   if ($0.5-X_{1D}[0])>(0.5-\overline{X}_{1D}[0])$ then $\tau \leq 0.5-\overline{X}_{1D}[0]$ else $\tau \leq 0.5-X_{1D}[0]$

4. **Case 2** ($X_{1D}[0]>0.5$) and ($\overline{X}_{1D}[0]>0.5$)
   
   if ($X_{1D}[n-1]-0.5)>($0.5-\overline{X}_{1D}[0])$ then $\tau \leq X_{1D}[n-1]-0.5$ else $\tau \leq X_{1D}[0]-0.5$

5. **Case 3** ($X_{1D}[n-1]<0.5$) and ($\overline{X}_{1D}[0]<\overline{X}_{1D}[i]<\overline{X}_{1D}[i+1]<\overline{X}_{1D}[m-1]$)
   
   if ($X_{1D}[0]<\overline{X}_{1D}[0]$) then $\tau \leq 0.5-\overline{X}_{1D}[0]$ else $\tau \leq 0.5-X_{1D}[0]$

6. **Case 4** ($X_{1D}[0]<\overline{X}_{1D}[i]<\overline{X}_{1D}[i+1]<\overline{X}_{1D}[m-1]$) and ($0.5-X_{1D}[0])$
   
   if ($X_{1D}[n-1]<\overline{X}_{1D}[m-1]$) then $\tau \leq X_{1D}[n-1]-0.5$ else $\tau \leq \overline{X}_{1D}[m-1]-0.5$

7. **Case 5** ($X_{1D}[0]<\overline{X}_{1D}[i]<\overline{X}_{1D}[i+1]<\overline{X}_{1D}[m-1]$) and
   
   ($\overline{X}_{1D}[0]<\overline{X}_{1D}[i]<\overline{X}_{1D}[i+1]<\overline{X}_{1D}[m-1]$)
   
   if ($X_{1D}[0]>X_{1D}[0]$ and $X_{1D}[n-1]>X_{1D}[m-1]$) or ($X_{1D}[0]<\overline{X}_{1D}[0]$ and $X_{1D}[n-1]<X_{1D}[m-1]$)
   
   if ($X_{1D}[n-1]-0.5)<(0.5-X_{1D}[n-1])$ then $TempX = X_{1D}[n-1]-0.5$
   else $TempX = 0.5-X_{1D}[n-1]$
   
   $\text{else if } X_{1D}[0]>\overline{X}_{1D}[0]$ and $X_{1D}[n-1]<\overline{X}_{1D}[m-1]$ then
   
   $\text{if } X_{1D}[0]<\overline{X}_{1D}[0]$ then $\tau \leq X_{1D}[0]-0.5$ else $\tau \leq 0.5-\overline{X}_{1D}[0]$
   
    $\text{else if } X_{1D}[m-1]<0.5<\overline{X}_{1D}[0]$ then
   
    $\text{if } X_{1D}[0]<\overline{X}_{1D}[0]$ then $\tau \leq 0.5-\overline{X}_{1D}[0]$ else $\tau \leq 0.5-\overline{X}_{1D}[0]$

8. **Case 6** ($X_{1D}[n-1]<0.5<\overline{X}_{1D}[0]$)
   
   if ($0.5-X_{1D}[n-1]) < ($\overline{X}_{1D}[0]-0.5)$ then $\tau = 0.5 - X_{1D}[n-1]$ else $\tau = \overline{X}_{1D}[0] - 0.5$
   
   $\text{else if } X_{1D}[m-1]<0.5<\overline{X}_{1D}[0]$ then
   
   $\text{if } X_{1D}[0]<\overline{X}_{1D}[0]$ then $\tau = \overline{X}_{1D}[0] - 0.5$ else $\tau = 0.5 - \overline{X}_{1D}[m-1]$

9. return $\tau$
Table 4.3 Data distribution and discretization rate

<table>
<thead>
<tr>
<th>Case</th>
<th>Image Distribution in dimension $j$</th>
<th>Minimum Number of Non-discretized Images</th>
<th>Discretization Rate $Discr_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Both $X_{1D}$ and $\bar{X}_{1D}$ are in $(0, 0.5)$</td>
<td>if $(0.5 - X_{1D}[0]) &gt; (0.5 - \bar{X}<em>{1D}[0])$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $(0.5 - X_{1D}[0]) \leq (0.5 - \bar{X}<em>{1D}[0])$ $X</em>{target} = n$, $\bar{X}_{target} = k$</td>
<td>$\geq 1 - \frac{n+k}{n+m}$</td>
</tr>
<tr>
<td>2</td>
<td>Both $X_{1D}$ and $\bar{X}_{1D}$ are in $(0.5, 1)$</td>
<td>if $(X_{1D}[n-1]-0.5) &gt; (\bar{X}<em>{1D}[m-1]-0.5)$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $(X_{1D}[n-1]-0.5) \leq (\bar{X}<em>{1D}[m-1]-0.5)$ $X</em>{target} = n$, $\bar{X}_{target} = k$</td>
<td>$\geq 1 - \frac{n+k}{n+m}$</td>
</tr>
<tr>
<td>3</td>
<td>$X_{1D}$ is in $(0, 0.5)$ $\bar{X}_{1D}$ is in $(0, 1)$</td>
<td>if $(X_{1D}[0]) &lt; (\bar{X}<em>{1D}[0])$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $(X_{1D}[0]) \geq (\bar{X}<em>{1D}[0])$ $X</em>{target} = n$, $\bar{X}_{target} = k$</td>
<td>$\geq 1 - \frac{n+k}{n+m}$</td>
</tr>
<tr>
<td>4</td>
<td>$X_{1D}$ is in $(0, 1)$ $\bar{X}_{1D}$ is in $(0.5, 1)$</td>
<td>if $(X_{1D}[n-1]) &lt; (\bar{X}<em>{1D}[m-1])$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $(X_{1D}[n-1]) \geq (\bar{X}<em>{1D}[m-1])$ $X</em>{target} = n$, $\bar{X}_{target} = k$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td>5</td>
<td>Both $X_{1D}$ and $\bar{X}_{1D}$ are in $(0, 1)$</td>
<td>if $(X_{1D}[0]) &gt; (\bar{X}<em>{1D}[0])$ and $(X</em>{1D}[n-1]) &gt; (\bar{X}<em>{1D}[m-1])$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or if $(X_{1D}[0]) &lt; (\bar{X}<em>{1D}[0])$ and $(X</em>{1D}[n-1]) &lt; (\bar{X}<em>{1D}[m-1])$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $(X_{1D}[0]) &gt; (\bar{X}<em>{1D}[0])$ and $(X</em>{1D}[n-1]) &lt; (\bar{X}<em>{1D}[m-1])$ $X</em>{target} = n$, $\bar{X}_{target} = k$</td>
<td>$\geq 1 - \frac{n+k}{n+m}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>if $(X_{1D}[0]) &lt; (\bar{X}<em>{1D}[0])$ and $(X</em>{1D}[n-1]) &gt; (\bar{X}<em>{1D}[m-1])$ $X</em>{target} = l$, $\bar{X}_{target} = m$</td>
<td>$\geq 1 - \frac{l+m}{n+m}$</td>
</tr>
<tr>
<td>6</td>
<td>$X_{1D}$ is in $(0, 0.5)$ $\bar{X}<em>{1D}$ is in $(0.5, 1)$ $X</em>{1D}$ is in $(0, 1)$ $\bar{X}_{1D}$ is in $(0.5, 0.5)$</td>
<td>$X_{target} = 0$, $\bar{X}_{target} = 0$</td>
<td>$\leq 100%$</td>
</tr>
</tbody>
</table>

Note: Assuming that the number of images that can not be discretized in $X_{target}$ is $l$ and that in $\bar{X}_{target}$ is $k$, where $l \leq n$ and $k \leq m$. 
5. Forming Hierarchies of Probabilistic Decision Tables

This chapter introduces how to build the hierarchies of probabilistic decision tables as a classifier for the purpose of facial representation and recognition. Although the best method for feature selection is to try every combination of the features and then compare their performance, it is often unfeasible due to the amount of computation. To overcome this type of limitation, a new method was proposed in Section 5.1. It heuristically selects the most relevant features as the conditional attributes for each decision table in the hierarchies so that the number of calculations can be reduced considerably.

A decision table in the hierarchies usually has elementary sets, classified into its positive, negative, and boundary regions [147]. When applied to face recognition, some elementary sets can be redundant. Section 5.2 described a new technique to prune the hierarchies to simplify their structure. When it is employed, the negative regions of the generated decision tables in the hierarchies are pruned, and in some cases, many decision tables can be completely removed. Because the size of the decision tables and the hierarchies is reduced, the test images only need to be matched to each elementary set in the positive region or boundary region, and correspondingly the classification process can speed up.

A person’s face can have some variations such as the viewing angle, illumination, facial expression, and noise. The distance-based classification methods often cannot
properly deal with them [19]. In Section 5.3, a new hybrid matching method was presented, which is a combination of distance-based and probabilistic-based classification methods. The advantage is that it can ignore small differences between compared patterns that are possibly caused by noise and would result in many unclassified test cases. In addition, the traditional probabilistic-based methods only use the conditional probability as the parameter for each unknown image. In contrast, the proposed method would select the best probability from different probabilities for each test image based upon where (positive region or boundary region) it is located.

5.1 Selecting Facial Features for Probabilistic Decision Tables

In this Section, the proposed method to heuristically apply Variable Precision Rough Set model (VPRS model) to select the most relevant PCA-based features is described. Principal component analysis (PCA) can extract facial features for face representation, but it cannot decipher which PCA-based feature or which group of PCA-based features is the best for face recognition. Therefore, the most relevant feature(s) using other techniques must be selected, which is not always easy to achieve. To precisely identify which feature or which group of features are the most relevant, every combination of these features must be tried for face recognition. Next in the process, one must select the most relevant feature(s) by comparing their performance, such as comparing their accuracy recognition rates. Specifically, assuming that each facial image is represented
by 24 discretized PCA-based features, in trying every combination of these features for face recognition, the whole process would need about two hundred thousand calculations to complete. Obviously, this means a huge amount of calculation is required, and correspondingly, in practice, it is almost unfeasible.

Therefore, a method to heuristically find the most relevant feature(s) in the context of the VPRS model is proposed. First, the $\lambda$–dependency of the probabilistic decision table defined in (2.30) is used as the parameter to select the most appropriate PCA-based features. Only those features that can generate the highest value are chosen. Then, the selected PCA-based features are used to create a new matrix defined as $X_{\text{simp}}$. After $X_{\text{simp}}$ is formed, another two parameters are used, which are the $\lambda$–dependency of the absolute positive region $\lambda_{\text{POS}}(X \mid C)$ defined in (2.32) and the $\lambda$–dependency in the VPRS model $\lambda_{\text{cl}}(X \mid C)$ defined in (2.33). They assist in determining the number of selected columns for matrix $X_{\text{simp}}$ which in turn is used to construct a decision table for the current level in the hierarchy.

Here, it is assumed $Q$ images can be totally discretized in the training set, each image is represented by $p$ PCA-based features, and they are represented by a $Q \times p$ matrix denoted as $X_{\text{discr}}^{\text{pca}}$. The procedure of evaluating $X_{\text{simp}}$ consists of the following three steps described below:
Step 1. Search $X_{\text{discr}}^{\text{pca}}$ and select the most appropriate column as the first column of $X_{\text{simp}}$.

In the first step, the first column of matrix $X_{\text{discr}}^{\text{pca}}$ is selected as the first column of $X_{\text{simp}}$. After that, the matrix $X_{\text{simp}}$ is used to construct a probabilistic decision table denoted as $DT_1$ in the current level of the hierarchy. This type of decision table is used as a trial in order to help identify if the first column of matrix $X_{\text{discr}}^{\text{pca}}$ is the most adequate column. Once the decision table $DT_1$ has been constructed, the value of the $\lambda$–dependency for the probabilistic decision table is calculated according to (2.30) and denoted as $\lambda_1$.

Following this step, the second column of the matrix $X_{\text{discr}}^{\text{pca}}$ is used as the first column of $X_{\text{simp}}$, assuming $X_{\text{simp}}$ is a matrix with only one column and its elements are from the second column of $X_{\text{discr}}^{\text{pca}}$. Similar to what was conducted previously, $X_{\text{simp}}$ is used to construct a probabilistic decision table denoted as $DT_2$, and then the $\lambda$–dependency is calculated according to (2.30) and denoted as $\lambda_2$.

In an effort to find the most adequate column from the matrix $X_{\text{discr}}^{\text{pca}}$, the process above is repeated until all columns from $X_{\text{discr}}^{\text{pca}}$ have been utilized. Finally, the values of the $\lambda$–dependency are compared to each other and the column generating the highest value is selected as the first column of $X_{\text{simp}}$. The selected column of $X_{\text{discr}}^{\text{pca}}$ is permanently removed from $X_{\text{discr}}^{\text{pca}}$ and added to $X_{\text{simp}}$. Now, $X_{\text{discr}}^{\text{pca}}$ becomes a $Q \times (p - 1)$ matrix and $X_{\text{simp}}$ becomes a $Q \times 1$ matrix.
Step 2. Select the remaining columns for $X_{simp}$ from $X_{discr}^{pca}$

The process of identifying the remaining columns for $X_{simp}$ from $X_{discr}^{pca}$ is similar to step 1. The only difference is when building a trial decision table, all columns in $X_{simp}$ and each column of the matrix $X_{discr}^{pca}$ are used together. Each time, one column of $X_{discr}^{pca}$ and all columns of $X_{simp}$ are used together to construct a trial decision table within the current level of the hierarchy, and the corresponding $\lambda$–dependency is computed. Once all columns from $X_{discr}^{pca}$ have been utilized, the column of $X_{discr}^{pca}$ generating the highest value is permanently removed from $X_{discr}^{pca}$ and added to $X_{simp}$.

Step 3. Decide the maximum number of columns in $X_{simp}$

Theoretically, the above process can be repeated until all columns of $X_{discr}^{pca}$ have been added to $X_{simp}$. However, another problem should be considered in practice, which is the dimension of $X_{simp}$. If the dimension of $X_{simp}$ is the same as that of $X_{discr}^{pca}$, the above noted procedure will not have any impact with regard to selecting the most appropriate features. The matrix $X_{simp}$ is actually the same as $X_{discr}^{pca}$ with the only difference being columns in $X_{discr}^{pca}$ are re-arranged to form $X_{simp}$. In order to select the most appropriate features as the conditional attributes, a stopping criterion must be created in order to decide the maximum number of columns in $X_{simp}$.

Identifying the maximum number of columns is also a heuristic process, and based on the experimental results, 6 columns should be enough to build a high quality decision table if the total number of images is less than 1500. In other words, 6 columns are
selected as a threshold value. Once $X_{simp}$ has 6 columns, the process in steps 1 and 2 is stopped immediately. If a decision table has 6 columns and if the most relevant features are to be found for the columns from 24 PCA-based features, only 129 attempts are required. In order to select the best feature for the first column, all 24 PCA-based features must be attempted. When selecting the best feature for the second column, only 23 PCA-based features must be attempted. Similarly, when selecting the best feature for the sixth column, only 18 PCA-based features are attempted.

Although $X_{simp}$ has only 6 columns, its columns are still too many in most cases. Before constructing the decision table, it is still possible to further modify $X_{simp}$ so as to reduce its columns, by considering the following two factors:

1. **Number of images available to build the decision table**

   If the number of images $Q$ is small, the matrix $X_{simp}$ having less than 6 columns can work very well. Based on the experimental results, if less than 200 images are used to build the decision table, only using the first one, two, or three columns of $X_{simp}$ are enough to build a decision table well classifying the images into $POS$, $NEG$ and $BND$.

2. **Value of $POS_u(X)$, value of $\lambda - dependency$ for positive region $\lambda_{u,\mu}(X \mid C)$ and value of $\lambda - dependency$ for absolute positive region $\lambda_{pos}^+(X \mid C)$**

   The value of $POS_u(X)$, $\lambda_{pos}^+(X \mid C)$ or $\lambda_{u,\mu}(X \mid C)$ can assist in determining the number of selected columns for matrix $X_{simp}$. More specifically:

   (1) If $POS_u(X) = 0$, then $\lambda_{pos}^+(X \mid C)$ is used as a parameter to determine how many columns are to be selected.
(2) If \( \text{POS}_u(X) > 0 \), \( \lambda_{\ell,u}(X | C) \) is used.

The first \((r+1)\) columns of \( X_{\text{simp}} \) rather than the first \( r \) columns of \( X_{\text{simp}} \) will be selected only if \( \lambda_{\text{POS},u}(X | C) \) or \( \lambda_{\ell,u}(X | C) \) of the decision table constructed by the \((r+1)\) columns is significantly greater. Based on the experimental results, 30 percent was selected as the threshold value. The first \((r+1)\) columns of \( X_{\text{simp}} \) will be selected if the value of \( \lambda_{\text{POS},u}(X | C) \) or \( \lambda_{\ell,u}(X | C) \) is 30 percent higher than that of the decision table having the first \( r \) columns of \( X_{\text{simp}} \).

After the number of columns of \( X_{\text{simp}} \) (denoted as \( k \) and \( k \leq 6 \)) has been decided, if \( k < 6 \), then the first \( k \) columns of \( X_{\text{simp}} \) will be retained, and the rest will be permanently removed from \( X_{\text{simp}} \). Eventually, the facial images in the training set are represented by a matrix \( X_{\text{simp}} \) having 6 or less than 6 columns, and each image is represented by 6 or less than 6 PCA-based features. Until now, all steps required before forming a decision table in the hierarchy have been finished, and the matrix \( X_{\text{simp}} \) is ready for the decision table.

### 5.2 Forming and Pruning Hierarchies of Probabilistic Decision Tables

In this Section, how to form the hierarchies of probabilistic decision tables as the classifier and the method proposed to prune the decision tables are introduced. In [244], an algorithm, called \( \text{HDTL-M} \), which implemented the Solution 3 of Section 2.7.2 was proposed and has been listed in Table 5.1.
In this thesis, when forming the hierarchies of probabilistic decision tables, the presented methodology followed this algorithm, while considering other factors. For instance, the possibility of simplifying the structure of the hierarchy and removing
redundant information of the hierarchy was addressed. More specifically, after $X_{simp}$ has been constructed, rough set theory is applied so the probabilistic decision table of the current hierarchy can be built. In such a decision table, images are classified into elementary sets and then the elementary sets are assigned to rough approximation regions: the positive region, the negative region, and the boundary region as based on formulas (2.8)-(2.10).

In this algorithm, the parameter $\alpha$ satisfying $0 < \alpha \leq 1$ is the desired degree of the global dependency of the whole structure of the decision tables. It is used as a stop condition. The parameters $\gamma^U_{i,\mu}(X | R)$ and $\gamma^C_{i,\mu}(X | C)$ are used to evaluate the quality of each decision table and the hierarchy of probabilistic decision tables. Their definitions, defined as formula (2.28) and (2.29), are introduced in Chapter 2.

Once the probabilistic decision table of the current layer has been completed, the process, as described in Chapter 3, can be recursively repeated on images classified into the boundary region in order to build the probabilistic decision table of the next layer in the hierarchy. When working on images in the boundary region, the images should be initially represented by another group of Haar wavelet coefficients.

For example, if coefficients from the scale part of level 2 are used in building the current decision table, then coefficients from the vertical part of level 2 can be used when using images in the boundary region in order to build the next decision table and so on. The process described above is continued until all images are classified into either the positive region or the negative region of a probabilistic decision table or all Haar-wavelet
coefficients have been utilized. The end-result of this process is the hierarchy of probabilistic decision tables are based on images from the training set. A sample hierarchy of decision tables has been listed in Table 5.2.

**Table 5.2** A sample hierarchy of decision tables

| E_i | A_{29} | A_{38} | P(E_i) | P(X | E_i) | Region |
|-----|--------|--------|--------|----------|--------|
| E_0 | 1      | 1      | 0.1542 | 1.0000   | POS    |
| E_1 | 1      | 0      | 0.3457 | 0.2769   | BND    |
| E_2 | 0      | 1      | 0.2500 | 0        | NEG    |
| E_3 | 0      | 0      | 0.2500 | 0        | NEG    |

| E_i | A_{17} | A_{16} | P(E_i) | P(X | E_i) | Region |
|-----|--------|--------|--------|----------|--------|
| E_0 | 1      | 1      | 0.2461 | 0        | NEG    |
| E_1 | 1      | 0      | 0.2615 | 0        | NEG    |
| E_2 | 0      | 1      | 0.2153 | 0        | NEG    |
| E_3 | 0      | 0      | 0.2769 | 1.0000   | POS    |

After the hierarchy of probabilistic decision tables is built and before working on classifying test images, a modification is required if applicable, which is pruning hierarchies of probabilistic decision tables. The purpose of the modifications is to simplify the structure of the hierarchy and remove redundant information, if available, so as to improve its performance. Essentially, the simplification prunes the negative regions of the generated decision tables, to replace matching new objects with negative region elementary sets during classification by assignment of properly selected default values of output probabilities. When applicable, some decision tables will be removed from the hierarchy as well. The advantages of this technique are in the ability to reduce the size of the hierarchy and speed-up the classification of new objects. With such a hierarchy, an
test image only has to be matched to the elementary sets in the positive region and boundary region of each decision table. The working process of the modification has two steps, as described below:

**Step 1.** Each decision table is searched to find the elementary sets classified in the negative region of each decision table. Next, the elementary sets are removed, and only the elementary sets classified in the positive region or boundary region are retained. Only the elementary sets which belong to the upper approximation of the target set $X$ are retained. In the context of Pawlak's model, it means retaining the elementary sets whose elements possibly belong to the target set $X$. In the context of VPRS model, the elementary sets whose elements have sufficient (or acceptable, depending on the application) likelihood of belonging to the target set $X$, as defined by the lower limit $l$ described in Section 2.6, are retained.

**Step 2.** All modified decision tables in the hierarchy are scanned one by one to identify which decision table can be selected as the last decision table or how to modify the last decision table of the hierarchy. This step is completed according to the following rules:

**Rule 1.** If a decision table in the hierarchy has no boundary region, it means all elementary sets have been classified into the positive region. In this case, such a decision table is selected as the last decision table in the hierarchy and all decision tables below it, if existing, are removed from the hierarchy.

**Rule 2.** As described in Chapter 3, the attributes of the probabilistic decision tables in the hierarchy are formed with coefficients from several groups of 2-dimensional Haar
Wavelet Transformation. So, if a decision table is built by the last available group of Haar wavelet coefficients, and its boundary region is not empty, its elementary sets, classified into the positive region, are retained. Absolute positive region $POS^*(X)$ is defined in formula (2.17). The elementary sets, classified into the boundary region and which can satisfy $POS^*(X)$ are also retained and others are removed.

**Step 3.** If a decision table has an empty positive region, which means it has no elementary sets satisfying $POS_u(X)$, it will be selected as the last decision table in the hierarchy and all decision tables below it, if existing, are removed from the hierarchy. Then, its elementary sets, having been classified into the boundary region and which can satisfy $POS^*(X)$, are also retained and others are removed.

Once the above noted steps have been completed, a hierarchy is considered to be successfully built. As an example, the hierarchy in Table 5.2, after being pruned, has been listed in Table 5.3. In this sample, all elementary sets classified in the negative region of each table have been removed. The second table has no boundary region, so it is selected as the last decision table for the hierarchy based on Rule 1 above.

If the images used to build the hierarchy of probabilistic decision tables are from $K$ people, then $K$ hierarchies based on the steps described in Chapters 3, 4 and 5 are built. Each hierarchy uses one person’s images as the elements for its target set $X$. For example, if images from four people are indexed as $P1$, $P2$, $P3$ and $P4$, four hierarchies indexed as $H1$, $H2$, $H3$ and $H4$ are built. Images from $P1$ are used as elements of set $X$ in $H1$, images from $P2$ are used as elements of the target set $X$ in $H2$, and so forth.
Table 5.3 A sample of pruned hierarchy of decision tables

| Region | Ei | A28 | A28 | P(Ei) | P(X| Ei) |
|--------|----|-----|-----|-------|--------|
| POS    | E0 | 1   | 1   | 0.1542| 1.0    |
| BND    | E1 | 1   | 0   | 0.3457| 0.2769 |

Decision Table at Layer 1

| Region | Ei | A17 | A16 | P(Ei) | P(X| Ei) |
|--------|----|-----|-----|-------|--------|
| POS    | E3 | 0   | 0   | 0.2769| 1.0    |

Decision Table at Layer 2

5.3 Classifying new Images using Hierarchies of Probabilistic Decision Tables

In this Section, how to classify test facial images is introduced. The primary focus is the method proposed to classify test images based on “probabilistic distance matching”. Subsequent to a set of $K$ hierarchies having been successfully constructed and based on a training set containing images from $K$ participants, having followed all the aforesaid steps, the hierarchies of decision tables can be used as a classifier to classify test (unknown) facial images for the purpose of recognition. (That is, an unknown image is known as an image having the same identity as one of the template images.) The test image is compared to all of the template images in order to determine its identity. The unknown image will be assigned the same identity as the template images having the highest similarity to it.

A very common approach used to classify a test image by holistic-matching methods is “distance-based classification” as described in Chapter 2. The test image will be
assigned the same identity as a template image that has the minimum Euclidean distance to the test image. But, this method frequently cannot work well due to variations such as: the viewing angle, illumination, facial expression and noise [19]. Hence, instead of “distance-based classification”, a method called probabilistic distance-based method was proposed which can be considered a hybrid of “distance-based classification” and “maximum likelihood-based classification”. Initially, a distance-based technique described in Step 2 is applied, which can be considered a departure from the standard distance-based techniques. It does not perform an exact match between the feature vectors representing the test images and the elementary sets. Next, as described in Steps 2 and 3, it is determined if an image can be classified into a decision class by assigning a proper selected default value as the probability of classifying the image into that class. Different from traditional maximum likelihood-based classification methods, such a value can be from different probabilities rather than conditional probability only, based on which region (positive or negative region) the elementary set is located. Then, the values of the probabilities are compared, and the image is classified into the decision class having the maximum value.

The proposed method can classify images into appropriate categories in the following three steps:

**Step 1. Pre-process the test image using information collected in the training phase**

When a test image (an unknown image) $I_{test}$ is taken as input, it should be pre-processed in the same way as the images in the training set. The working process is
the same as the process described in Chapters 3, 4, and 5.

At first, the test image should be modified to be the same size as the images in the training set with the face on the test image being centered; as well the size of the face should be similar to that of the images in the training set.

Then, the test image is transformed by Haar Wavelet Transformation so it can be represented by several groups of wavelet coefficients. The order of selecting the wavelet coefficients must be in the same order as for the images in the training set. For example, when constructing the first decision table of the hierarchy, images are represented by wavelet coefficients from the vertical part of level 3. When matching the test image to each elementary set of the first decision table, the test image must be represented by wavelet coefficients from the vertical part of level 3. In addition, the coefficients of the test image must be retained or removed via the same method as the images in the training set which were used to build the first decision table. According to the description in Chapter 3, some wavelet coefficients are too small to make a contribution to face recognition, so they can be removed. Assuming the coefficients, the index of which is 1, 3, 5, 7, and 9, are kept and the remainder are removed, then those coefficients of the test image, index of which is 1, 3, 5, 7, and 9, must be retained and the remainder rejected.

After that, the test image will be processed by PCA so that the useful face recognition features can be extracted according to (4.17), (4.18) and (4.19) of Section 4.1. When being transformed, the “average” image, Eigenvectors and so on are those generated when the images in the initial training set are processed.
Finally, the PCA-based features of the test image selected as the conditional attributes of the decision table will be retained and the rest will be removed. For example, if PCA-based features, the index of which is 1, 9, and 10, are selected to build the first decision table of the hierarchy, the features of the test image having the same index number (index 1, index 2, and index 9) will be kept and the rest will be removed. Then, each of the selected PCA-based feature (denoted as $x_i$) will be processed by formula (4.21), with the result denoted as $f(x_i)$. The value of $f(x_i)$ is less than 1 but greater than 0. Unlike processing the images in the training set, selected PCA-based features of the test image are not discretized. They will be used directly in Step 2.

Step 2. Evaluate the distance between the test image and each elementary set in each decision table of each hierarchy

The technique used in Step 2 is a departure from the standard technique involving an exact match between the feature vectors representing images and elementary sets. The approach is motivated by the need to "soften" the matching procedure in order to ignore small differences between compared patterns, possibly caused by noise which would result in many unclassified test objects.

As described in Section 2.4, an image can be more conveniently represented as a vector. Hence, after PCA-based features have been processed in step 1, the test image $I_{test}$ is represented by a real-valued vector:

$$I_{test} = [f(x_1), ..., f(x_i), ..., f(x_n)]^T,$$

(5.1)
where \(0 < f(x_i) < 1\) is the PCA-based features of the test image after transformation by formula (4.21) and it is assumed the first decision table has \(n\) conditional attributes.

Correspondingly, each elementary set \(E_i\) of the first probabilistic decision table in the first hierarchy is represented by a binary-valued vector:

\[
E_i = [att_1^E, ..., att_i^E, ..., att_n^E]^T,
\]

where \(att_i^E\) is the \(i^{th}\) binary condition attribute and its value is 0 or 1. Then, the real-valued vector representing the test image starts to be compared to the binary-valued vector representing each elementary set, based on the distance function \(d(x_i, E)\) defined below:

**Definition 5.1** The distance between the real-valued vector of the test image and the binary-valued vector of each elementary set is defined as the absolute value of Euclidean distance between \(f(x_i)\) and \(att_i^E\) in the \(i^{th}\) dimension of the feature space, for which the absolute value of the Euclidean distance attains the maximum among the absolute value of each Euclidean distances in each dimension. That is:

\[
d(x_i, E) = MAX_{i=1,2,...,n} (|f(x_i) - att_i^E|),
\]

where \(|.|\) is the absolute value function.

In Table 5.4, a sample decision table at the 1st layer of a hierarchy is listed, and Table 5.5 is the same table after being pruned. Based on the two tables, it will be explained, in detail, how to evaluate the distance between an unknown image and each elementary set.
### Table 5.4 A sample decision table at 1st layer

| $E_i$ | $A_1$ | $A_2$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|-------|-------|----------|-----------|--------|
| $E_0$ | 1     | 1     | 0.1875   | 1.0       | POS    |
| $E_1$ | 1     | 0     | 0.3125   | 0.2       | BND    |
| $E_2$ | 0     | 1     | 0.3229   | 0.0       | NEG    |
| $E_3$ | 0     | 0     | 0.1770   | 0.0       | NEG    |

### Table 5.5 Pruned sample decision table at 1st layer

| $E_i$ | $A_1$ | $A_2$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|-------|-------|----------|-----------|--------|
| $E_0$ | 1     | 1     | 0.1875   | 1.0       | POS    |
| $E_1$ | 1     | 0     | 0.3125   | 0.2       | BND    |

Assuming there is a test image working with Decision Table 1, after it is processed by Haar-wavelet, PCA and formula (4.21), $f(x_1) = 0.9999$ and $f(x_2) = 0.9658$. According to (5.3), the distance between the test image and the first two elementary sets $E_0$ and $E_1$ is calculated, and the results are listed below:

\[
d(x, E_0) = \max_{i=1,2} \left( |f(x_1) - att^E_1|, |f(x_2) - att^E_2| \right) = \max_{i=1,2} \left( |0.9999 - 1|, |0.9658 - 1| \right) = \max_{i=1,2} (0.0001, 0.0342) = 0.0342
\]

\[
d(x, E_1) = \max_{i=1,2} \left( |f(x_1) - att^E_1|, |f(x_2) - att^E_2| \right) = \max_{i=1,2} \left( |0.9999 - 1|, |0.9658 - 0| \right) = \max_{i=1,2} (0.0001, 0.9658) = 0.9658
\]

After the distance between each elementary set has been calculated, the test image is classified into an elementary set that can satisfy the two conditions below:

**Condition 1.** The distance between the elementary set and the test image is at a minimum among all elementary sets.

**Condition 2.** Such a distance is no higher than the value of a predefined threshold $d$. 

113
the value of which can be identified heuristically. Based on the experimental results, it is identified as 0.500001.

Based on the two conditions, it is obvious $d(x, E_0) < 0.500001 < d(x, E_1)$, so the test image is classified into elementary set $E_0$.

After identifying in which elementary set the test image should be classified, the probability of classifying the test image into the target set $X$ must be identified as well, which is based on the rough region location of the lowest distance elementary set, and the following rules:

**Rule 1.** If such an elementary set $E$ is located in the positive region, the conditional probability of the target set $X$, $P(X|E)$, is selected as the probability of classifying the test image into the decision class $X$ of the first hierarchy, and the test image is no longer matched to any other decision tables in the first hierarchy. In the aforementioned example, the test image is classified into $E_0$ and is located in $POS$, and $P(X|E_0) = 1.0$, so the probability of classifying the test image into the target set $X$ is 100%.

**Rule 2.** If the lowest distance is greater than the predefined threshold value, the test image is considered an object that does not match (approximately of course) any elementary set $E$ of the hierarchy. In this case, $P(X)$ is selected as the probability of “classifying” the test image into the decision class $X$, and the test image is no longer matched to any other decision table in the first hierarchy.

**Rule 3.** If the closest-match elementary set is located in the boundary region, the process passes to the second decision table in the first hierarchy, and the procedure is
repeated. According to the description of pruning the decision table in Section 5.2, if the last decision table of the hierarchy has a boundary region, the elementary sets that can be classified into the absolute positive region $POS^*(X)$ are retained and others are removed. That is, after the pruning, the boundary region of the last decision table only has the elementary sets $E$ belonging to $POS^*(X)$. Thus, when matching the elementary sets in the last decision table in the first hierarchy, if the lowest distance elementary set is located in the boundary region, it is an elementary set belonging to $POS^*(X)$. The conditional probability of the target set $X$, $P(X|E)$, is selected as the probability of classifying the object into the decision class $X$ of the first hierarchy. After working with the first hierarchy, the test image will continue to work with the second hierarchy in the same way and so on, until all hierarchies are utilized.

**Step 3. Evaluate the probability of classifying a test image into each decision class associated with each hierarchy**

After identifying the probability of the test image belonging to each decision class $X$, based on each hierarchy (corresponding to each decision class $X$), the probabilities are ranked, and the test image will eventually be classified into the decision class $X$ of a hierarchy with the highest probability.
6. Updating Decision Table Hierarchies by Incremental Learning

Chapter 6 covers how the hierarchies of probabilistic decision tables can update their structures and retain their prior knowledge in the process of incremental learning in the meantime. The representativeness of images in the training set is often insufficient, so following the addition of new images to the training set, the hierarchies must be modified to reflect the presence of the new images.

The algorithm in [244] proposed four strategies in the context of classic rough set theory, describing how to classify a new image into an appropriate elementary set during the process of incremental learning. In this thesis, for the first time, the algorithm in [244] was extended and implemented in the framework of the VPRS model. Different from classic rough set theory, because the decision attribute of each elementary set in VPRS model usually is not assigned a singular value, the second and the third strategies in [244] can be merged. In Section 6.1, the implement of these strategies in the context of VPRS model was proposed. Section 6.2 covers the theoretical aspects of the implementation. The steps of how the hierarchies update their structures based on the strategies of [244] in VPRS model are presented in Section 6.3.

6.1 Incremental Learning and Rough Decision Tables

Decision tables learned from data, here denoted as rough decision tables, were
originally proposed by Pawlak in the framework of rough set theory [147][151]. The probabilistic decision tables and their hierarchies [248-249], defined in the context of the Variable Precision Rough Set Theory (VPRS model)[243], generalized the notion of rough decision tables. In [244], an incremental algorithm was proposed, which adopted the hierarchy of probabilistic decision tables as the classifier and was applied in the framework of classic rough set theory [147]. Four strategies were proposed and when a new case surfaces, the hierarchy of probabilistic decision tables is updated with different strategies based upon which category the new case can be classified.

According to formulas (2.8), (2.9) and (2.10), under the VPRS model, the value of the decision attributes of each object in an elementary set can be different. After adding a new case to an elementary set, the new set may migrate from one region to the other, as from $BND$ to $POS$, or from $POS$ to $BND$ or even to $NEG$ (although this is very unlikely). Therefore, when the strategies are applied under the VPRS model, the need to shift the affected elementary set to an appropriate area should be considered over and above implementing the corresponding actions of each strategy described in [244].

Since elementary sets in the VPRS model usually have no singular value of the decision attribute attached to them, there is no need to consider whether the decision attributes of elementary sets can match the decision attribute value of the newly added image. In the VPRS model, the second and the third strategies in [244] can be merged as one strategy. In this thesis, the algorithm presented in [244] is extended in the framework of the VPRS model as three strategies. The theoretical aspects of the algorithm will be
covered in the next Section and their details are explained as follows:

(1) If a new case falls into an elementary set located in the boundary region of one of the layers of the hierarchy, the count of the cases of the matching elementary set and the count of the universe of the affected layer are adjusted. Then, if necessary, the matching elementary set is shifted from the boundary region to an appropriate region (the positive or negative region), and re-compute new probabilistic decision tables for all affected subordinate layers. Finally, the $\lambda - \text{dependency}$ and $\gamma - \text{dependency}$ of the hierarchy are re-computed and used to evaluate the quality of the updated hierarchy.

(2) A new case falls into an elementary set located in the positive or negative region of one of the layers of the hierarchy. Such a new case should match or be close enough to the pattern of the conditional attributes of that elementary set. The count of the cases of the matching elementary set and the count of the universe of the affected layer are adjusted. The necessity of shifting the affected elementary set from its current region to a new approximate region is considered. As an example, if the affected elementary set is located in the positive region, it is determined if it should be shifted into the boundary region or, although highly unlikely, the negative region. After shifting the affected elementary set to a new region, if required, the new probabilistic decision tables are computed for all affected subordinate layers. Finally, the $\lambda - \text{dependency}$ and $\gamma - \text{dependency}$ are computed for the adapted hierarchy of the probabilistic decision table.

(3) If the new case is unable to match any elementary set of one of the layers of the
hierarchy, the universe and the information table of the affected layer are expanded by creating a new elementary set located in the positive or negative region. The dependency of the hierarchy is re-computed and the probabilistic decision table of the affected layer is modified by adding an extra row.

### 6.2 Theoretical Aspects of the Incremental Learning Algorithm

In this Section, the theoretical aspects of the incremental learning algorithm [244] applied under the VPRS model are introduced. It is assumed that there are $k$ new images, denoted as $I_{\text{inc}} = \{I_1, \ldots, I_t, \ldots, I_k\}$, available for incremental learning. Initially, the hierarchy built by the images in the training set has $t$ probabilistic decision tables denoted as $DT = \{DT_1, \ldots, DT_j, \ldots, DT_t\}$. Each table $DT_i$ that is built by $d_i$ images in the training set has $n_i$ conditional attributes denoted as $C_i = \{c_{i1}, \ldots, c_{ij}, \ldots, c_{in_i}\}$, and the number of elementary sets of decision table $DT_i$ is $2^{n_i} - p_i$, where $0 \leq p_i < 2^{n_i}$. When the strategies, described in the previous section, are applied to update the $i^{th}$ probabilistic decision table $DT_i$ in the hierarchy, they can be implemented as the algorithm listed in Table 6.1. The subprograms are executed according to the algorithm in Table 6.1 and can be implemented as the algorithms listed in Table 6.2 and Table 6.3.

The algorithm in Table 6.3 only consists of Primitive operations [201], so its running time is constant. The algorithm in Table 6.2 primarily consists of a “while” loop and a “for” loop which is a triple loop. The running time of the assignment statement and the
statement of the Boolean expression evaluation is constant [201], so the innermost loop of the “for” loop is executed \( n_{i+1} \) times, the loop in the middle is executed \( 2^{n_{i+1}} - p_{i+1} \) times, and the outermost loop is \( d_{i+1} \) times. The running time of the “for” loop denoted as \( RT_{for} \) is:

\[
RT_{for} = n_{i+1}(2^{n_{i+1}} - p_{i+1})d_{i+1},
\]

(6.1)

Table 6.1 Algorithm implementing three strategies[244]

<table>
<thead>
<tr>
<th></th>
<th>Input:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>- ( E_j ) containing the image ( I_j ),</td>
</tr>
<tr>
<td></td>
<td>- The region where ( E_j ) is in,</td>
</tr>
<tr>
<td></td>
<td>- Decision tables ( DT^<em>_j, DT^</em>_{i+1}, \ldots, DT^*_t ) of the hierarchy</td>
</tr>
<tr>
<td>2</td>
<td>if ((E_j \text{ in BND})) {</td>
</tr>
<tr>
<td></td>
<td>(</td>
</tr>
<tr>
<td></td>
<td>if necessary, moving ( E_j ) from BND to POS or NEG {</td>
</tr>
<tr>
<td></td>
<td>re-compute decision tables for all affected subordinate layers</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>else if ((E_j \text{ in POS or NEG})) {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(</td>
</tr>
<tr>
<td></td>
<td>if necessary, moving ( E_j ) from POS or NEG to BND {</td>
</tr>
<tr>
<td></td>
<td>re-compute decision tables for all affected subordinate layers</td>
</tr>
<tr>
<td></td>
<td>}else{continue}</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
<tr>
<td>else {</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(</td>
</tr>
<tr>
<td></td>
<td>create a new ( E ) in POS or NEG</td>
</tr>
<tr>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
**Table 6.2** Algorithm of re-computing decision tables for all affected subordinate layers

1. **Input:**
   - Images located in BND of DTᵢ: \( I_{BND}^{DTᵢ} = \{I₁, ..., Iᵢ₋₁, Iᵢ \} \)
   - Decision tables DTᵢ, DTᵢ₊₁,..., DTᵣ of the hierarchy

2. while (DTᵢ₊₁ is available and re-computation is needed) {
   for (i=0, i<dᵢ₊₁, i++) {
     // For each image \( Iᵢ \) in \( I_{BND}^{DTᵢ₋₁} \), do the operations below:
     for (j=0, j<\((2ⁿᵢ₋₁ − pᵢ₋₁) \), j++) {
       for (l=0, l<nᵢ₋₁, l++) {
         if \( Iᵢ[l] ≠ Eᵢ[k] \) flag=true
       }
     }
   }
   decide which elementary set the image \( Iᵢ \) can be classified into
  }

3. calculate card(X), P(E), P(X|E), P(X=E), and λ - dependency of DTᵢ₊₁

4. decide which region POS, NEG, or BND each \( Eᵢ \) should be classified into
}
//end while loop

**Table 6.3** Algorithm for creating a new elementary set in POS or NEG

1. **Input:**
   - \( DTᵢ \) having \( 2ⁿᵢ − pᵢ \) elementary sets and represented by an ArrayList in Java when it is implemented
   - The image \( Iᵢ \),

2. define a new elementary set \( E_{2ⁿᵢ₋₁} \) containing \( Iᵢ \)
   calculate cardinality of \( E_{2ⁿᵢ₋₁} \)
   calculate \( P(E_{2ⁿᵢ₋₁}) \)
   calculate \( P(X|E_{2ⁿᵢ₋₁}) \)
   decide which the region, POS or NEG, \( E_{2ⁿᵢ₋₁} \) should be in
   \[ DTᵢ = DTᵢ + E_{2ⁿᵢ₋₁} \]
The running time of the “while” loop denoted as $RT_{\text{while}}$ depends on the availability of the decision table $DT_{i+1}$ and the necessity of re-computing $DT_{i+1}$. As the worst case, all of the decision tables below the decision table $DT_{i}$ have been affected and have to be re-computed, so the “while” loop has to be executed $(t - i)$ times. If an elementary set needs to be moved, the algorithm in Table 6.2 will be called. This means that when the algorithm in Table 6.2 is called, at least, the decision table $DT_{i+1}$ is available and needs to be re-computed. Thus, as the best case, the “while” loop is only executed 1 time to re-compute the decision table $DT_{i+1}$. Thus, the running time of the algorithm in Table 6.2 can be formulated by the following theorem:

**Theorem 6.1** Given the hierarchy having $t$ probabilistic decision tables:

(a) Each decision table in the hierarchy (denoted as $DT_{i}$) is built by $d_{i}$ images in the training set, has $n_{i}$ conditional attributes and has $2^{n_{i}} - p_{i}$ elementary sets.

(b) As the worst case, the running time of the algorithm given in Table 6.2 is:

$$RT_{\text{while}} = \sum_{j=i+1}^{t} n_{j}(2^{n_{j}} - p_{j})d_{j}.$$  

**Proof:** According to (6.1), for each decision table $DT_{j}$ below the decision table $DT_{i}$ that has been affected, its running time is $n_{j}(2^{n_{j}} - p_{j})d_{j}$.

If all of the decision tables below the decision table $DT_{j}$ have been affected, which means that the “while” loop has to be executed $(t - i)$ times, the total running time is:

$$RT_{\text{while}} = n_{i+1}(2^{n_{i+1}} - p_{i+1})d_{i+1} + n_{i+2}(2^{n_{i+2}} - p_{i+2})d_{i+2} + \ldots + n_{t}(2^{n_{t}} - p_{t})d_{t}$$

$$= \sum_{j=i+1}^{t} n_{j}(2^{n_{j}} - p_{j})d_{j} \quad \Box$$
Based on formula (6.2), it can be summarized that the complexity of the algorithm given in Table 6.2 is mainly determined by the number of the conditional attributes of each decision table. When the number of the conditional attributes is increased, the complexity of the algorithm is always increased exponentially.

When forming the hierarchy, the decision table $DT_{i+1}$ is built by the images classified into the boundary region of the decision table $DT_i$. The size of the training set used to build each decision table can be described via the inequality below:

$$d_i \geq d_{z+1} \geq ... \geq d_{i+1} \geq ... \geq d_{r+1} \geq d_r.$$  \hspace{1cm} (6.3)

The number of images in the training set used to build the decision table $DT_i$ is always greater than or equal to the number of images in the training set used to build the decision table $DT_{i+1}$. For any decision table $DT_i$ having $n_j$ conditional attributes, the number of its elementary sets can be:

$$1 \leq 2^n - p_j \leq 2^n.$$  \hspace{1cm} (6.4)

Hence, as a worst case scenario, the following theorem can be formulated:

**Theorem 6.2** Given the hierarchy having $t$ probabilistic decision tables:

(a) Each decision table in the hierarchy (denoted as $DT_i$) is built by $d_i$ images in the training set, has $n_i$ conditional attributes, and has $2^n - p_i$ elementary sets, as a worst case scenario.

(b) The complexity of the algorithm given in Table 6.2 for re-computing the decision tables for all affected subordinate layers is:
\[ O(d_t n_{\text{max}} 2^{n_{\text{max}}}), \quad (6.5) \]

where \( n_{\text{max}} 2^{n_{\text{max}}} \) is the maximum among \( n_{i+1} 2^{n_{i+1}}, \ldots, n_i 2^n \).

**Proof:** According to (6.2), the running time can be

\[ RT_{\text{while}} = \sum_{j=i+1}^t n_j (2^n_j - p_j) d_j. \]

According to (6.3), \( d_1 \geq d_2 \geq \ldots d_i \geq d_{i+1} \geq \ldots \geq d_t \) (i.e., the size of the training set used to build the first decision table is the maximum), so the inequality below can hold:

\[ RT_{\text{while}} = \sum_{j=i+1}^t n_j (2^n_j - p_j) d_j \leq \sum_{j=i+1}^t n_j (2^n_j - p_j) d_1. \]

According to (6.4), \( 1 \leq 2^n_j - p_j \leq 2^n_j \), so the inequality below can hold:

\[ \sum_{j=i+1}^t n_j (2^n_j - p_j) d_1 = d_1 \sum_{j=i+1}^t n_j (2^n_j - p_j) \leq d_1 \sum_{j=i+1}^t n_j 2^n_j. \]

The number of the conditional attributes of each decision table can be different, so the inequality below can hold:

\[ d_1 \sum_{j=i+1}^t n_j 2^n_j \leq d_1 \sum_{j=i+1}^t n_{\text{max}} 2^{n_{\text{max}}} = d_1 (t-i) n_{\text{max}} 2^{n_{\text{max}}} \]

where \( n_{\text{max}} 2^{n_{\text{max}}} \) is the maximum among \( n_{i+1} 2^{n_{i+1}}, \ldots, n_i 2^n \), and \( n_j 2^n \leq n_{\text{max}} 2^{n_{\text{max}}} \).

Because \( t-i < t \), the inequality below can hold:

\[ d_1 (t-i) n_{\text{max}} 2^{n_{\text{max}}} < d_t n_{\text{max}} 2^{n_{\text{max}}} \]

Therefore, in sum, the inequality below can hold

\[ RT_{\text{while}} = \sum_{j=i+1}^t n_j (2^n_j - p_j) d_j \leq d_1 \sum_{j=i+1}^t n_j (2^n_j - p_j) \leq d_1 \sum_{j=i+1}^t n_j 2^n_j \]

\[ \leq d_1 (t-i) n_{\text{max}} 2^{n_{\text{max}}} < d_t n_{\text{max}} 2^{n_{\text{max}}} \]
So, as the worst case, the complexity is \( O(d_i t n_{max} 2^{n_{max}}) \). □

The algorithm in Table 6.1 primarily consists of a conditional statement. The cost of an “if” statement in the worst case is the greatest of the costs for the clauses of the various branches [201]. In Table 6.1, its first branch consists of Primitive operations and an “if” statement containing a function call that calls the subprogram to re-compute the decision tables for all affected subordinate layers. The algorithm of this subprogram is listed in Table 6.2. The running time of a function call is the time for the function setup which is constant plus the time required for the execution of the function body [201]. Hence, the complexity of the first branch is determined by the complexity of the algorithm in Table 6.2. Similarly, because the second branch will call the same subprogram, when needed, to re-compute the decision tables, its complexity is also determined by the complexity of the algorithm in Table 6.2. The third branch has an assignment statement and a function call calling the subprogram, the algorithm of which is listed in Table 6.3. The complexity of the algorithm in Table 6.3 is constant. Therefore, in the worst case, the complexity of the incremental learning algorithm in Table 6.1 denoted as \( RT_{incr} \) is:

**Theorem 6.3** Given the hierarchy having \( t \) probabilistic decision tables:

(a) Each decision table in the hierarchy (denoted as \( DT_i \)) is built by \( d_i \) images in the training set, has \( n_i \) conditional attributes, and has \( 2^n - p_i \) elementary sets.

(b) In the worst case, the complexity of the algorithm given in Table 6.3 \( RT_{incr} \) is always
less than $O(d, t_{max} 2^{n_{max}})$. That is:

$$RT_{incr.} < O(d, t_{max} 2^{n_{max}}).$$

(6.6)

Based on formula (6.6), the complexity of the incremental learning algorithm implementing various strategies can be summarized as:

(1) Its performance is primarily determined by three factors: the number of the conditional attributes of each decision table in the hierarchy, the number of the decision tables in the hierarchy and the total number of images in the training set.

(2) When the value of each factor is increased, the complexity of the algorithm is always increased. However, compared to the other two factors, the number of the conditional attributes of each decision table has the highest impact on the complexity. In other words, the size of the feature space has the highest impact on the complexity of the algorithm. If the number of the conditional attributes is increased, which means that the dimension of the feature space is increased, the complexity of the algorithm is always increased exponentially.

(3) In order to reduce the complexity, the number of the decision tables and the number of the conditional attributes of each decision table should be as small as possible, when forming the hierarchy. If they can not be taken into account simultaneously, the hierarchy can have a relatively high number of the decision tables; however, the number of conditional attributes of each decision table should be as small as possible.
6.3 Incremental Learning with Hierarchies of Probabilistic Decision Tables

In this Section, steps regarding the application of the presented methodology to incremental learning will be introduced. A complete example further explaining the working process of incremental learning is included in Appendix F. According to the description in Section 6.1, when a new image is available, various strategies (three strategies in total) can be applied in different situations. Such an image has the same identity as one of the template images. The steps of implementing incremental learning are described below:

Step 1. Form initial hierarchies of probabilistic decision tables

Given a set of facial images from $K$ participants as a training set (denoted as an initial training set), $K$ hierarchies of probabilistic decision tables can be built according to the information introduced in Chapter 3 to Chapter 5. Each hierarchy uses one participant’s images as elements for its target set $X$.

Step 2. Pre-process the new image

When a new image, having the same identity as any image in the training set, is entered, it should be pre-processed first in the same way as images in the initial training set. The procedure is almost the same as that described in the Step 1 of Section 5.3.

Firstly, the image should be modified in order to have the same size as the images in the initial training set, move the face on the new image to the middle, and change its size
to be similar to that of the images in the initial training set.

Secondly, such an image is transformed by the Haar Wavelet Transformation and several of its wavelet coefficients are selected for the purpose of incremental learning. The order of selection of the wavelet coefficients must be the same as for the images in the initial training set.

Thirdly, the selected wavelet coefficients of the new image are transformed by PCA using (4.17), (4.18) and (4.19) in Section 4.1. When being transformed, the “average” image, Eigenvectors and so on are the same as those used on the initial training set.

Fourthly, the PCA-based features of the new image are transformed by (4.21) each of which is denoted as $f(x)$, and then discretized based on the rules described in Section 4.2. In (4.21), the values of parameter $a$ is 2 and the parameter $c$ is the same as the arithmetic average of the selected PCA-based feature of all images in the initial training set.

**Step 3. Classify the new image into an appropriate elementary set**

In this step, the distance between the newly-added image $I_{new}$ and each elementary set $E_i$ of the first decision table of the hierarchy, is calculated based on formula (5.3).

Once the distances are computed, they are sorted and listed according to the minimum $d(x, E)_{min}$ to the maximum $d(x, E)_{max}$. If $d(x, E)_{min}$ is less than the value of a pre-defined threshold $\tau$ (i.e. $d(x, E)_{min} < \tau$), the new image is classified into that particular elementary set. The value of the pre-defined threshold $\tau$ can be heuristically identified.
On the other hand, if $d(x, E)_{\text{min}}$ is above the pre-defined threshold $\tau^\prime$, then the new image must be discretized by a crisp cut prior to classification. Each PCA-based feature $x_i^{\text{pca}}$ is compared to the arithmetic average $c_i$ of the selected PCA-based feature for all images in the initial training set, as defined in formula (4.22). If $f(x_i^{\text{pca}}) > f(c_i)$ after $x_i^{\text{pca}}$ and the arithmetic average $c_i$ are transformed by formula (4.21), it is assigned a value of 1, otherwise it is assigned a value of 0.

Following discretization by a crisp cut, the decision table is searched, and based on its discretized conditional attributes, a new image is added into an elementary set, the conditional attributes of which are equal to that of the new image. If no such elementary set is found, a new elementary set, having the new image, is created.

**Step 4. Modify the hierarchy of probabilistic decision tables**

After classifying the new image into an appropriate elementary set or creating a new elementary set, a method to modify the hierarchy of the decision table can be developed, based upon the following factors:

1. The conditional attribute of the new image and the conditional attribute of the elementary set, into which the new image is classified;
2. The region in which the elementary set is located;
3. Strategies described in Section 6.1;
4. $\lambda$ – dependency of the hierarchy after adding the new image.

In practice, the factors must be considered together. The hierarchy must be modified by taking into consideration all the factors. For example, if the new case is unable to
match any elementary set of one of the layers of the hierarchy, Strategy 3 in Section 6.1 can be applied.

After the decision table has been changed, a second problem requiring consideration is the quality of the hierarchy of decision tables. Strategies described in Section 6.1 can explain the modification of the hierarchy, but cannot ensure the quality of the new hierarchy is higher than, or at least the same as, the quality prior to adding a new image. Obviously, if the quality of the new hierarchy deteriorates, such a hierarchy will not work well. If this is the case, the best method is to add the new image into the training set and use the new training set to completely re-generate the hierarchy according to all the steps described in Chapters 3 to 5.

In order to identify whether a hierarchy must be completely re-generated, the value of $\lambda$–dependency of the hierarchy can be used as a factor. After the hierarchy of decision tables is modified, based on strategies in Section 6.1, $\lambda$–dependency of the hierarchy is calculated. If the new $\lambda$–dependency of the hierarchy is significantly less than that of the hierarchy prior to adding the new image, the hierarchy should be completely re-generated; if not, the hierarchy can be retained.

Once all the steps are completed, it may be concluded that the image has been successfully added into the initial training set, and the hierarchy has been well modified. The aforementioned operations can be repeated on each new image until no new images are available.
7. Experiments and Discussions

In Chapter 7, the implementation of various experiments is covered. By evaluating the performance of the hierarchies in the experiments, it is determined whether or not the presented methodology can be applied to face recognition and under what circumstances it produces the best results. Section 7.1 and Section 7.2 primarily describe the process of preparing experimental datasets and the basic idea of cross-validation. Section 7.3 covers the experimental results for cross-validation. The results of the hierarchies, which are formed directly by the photos in the training set, are introduced in Section 7.4. Finally, Section 7.5 presents the results of the hierarchies after incremental learning.

7.1 Photos for Experiments and Experimental Datasets

This Section describes how to collect the photos for experiments and how to prepare the datasets. Facial photos in the experiments are digital photos taken from two sources: public online databases and the University of Regina. The public online digital photos are from the AR Face Database [116], which is publicly available and is free to academics. As described in previous chapters, when applying the presented methodology under the context of Variable Precision Rough Set Theory (VPRS model), the values of many parameters must be identified heuristically such as the value of parameter $a$ in formula (4.21), and the value of threshold $\tau$ used in Step 1 of Section 4.2 for the purpose of discretization. Before the proposed methodology was developed, a set of experiments
were performed to assist in determining the values of the parameters [27]. Photos were collected from the AR Face Database for experimental sets.

Compared to the digital photos collected from the University of Regina, the photos collected from the public databases, have several disadvantages. Especially, when they are used, the factors as described below are difficult to be controlled.

1. The number of the participants and the number of photos of each participant

The number of the participants and the number of photos of each participant in the public databases are usually fixed. Some public databases only contain the photos from a small number of participants or each participant only has a small number of photos. For example, ORL face database [13][176][190-191] only has photos from 10 participants. Yale Database [23] only consists of images from 15 participants, using 11 images from each participant. When testing the presented methodology, especially testing the incremental learning aspect, many photos will be needed. So, when photos from the public databases are all utilized, the experiments may still not be completed.

2. Types of facial expressions

One of purposes of the experiments is to assist in determining the impact of various facial expressions on the performance of the presented methodology. The number of facial expressions in some public databases is too small. For instance, in the ORL face database, the photos of some participants were taken at different times, varying the lighting and two types of facial expressions (open/closed eyes, smiling/not smiling) only.
3. **Background and illumination conditions**

The method of principal component analysis (PCA) was used in the presented methodology. PCA can demonstrate good performance only under a controlled background and illumination condition. It may fail under varying illumination conditions such as processing photos taken outside the building under natural daylight [101][197]. Photos from some public databases, for example the photos from Caltech face database [86], were taken outside the building with different backgrounds.

4. **Accessibility and availability of all potential facial features**

Facial photos from some public databases may not contain all potential facial features. According to [1], the facial contouring and background are very useful for face recognition. However, photos from some public database, for example from Yale Face Database B, only contain facial features such as the mouth, nose and eyes. The facial contouring and background of each participant are not included.

Hence, that photos collected from the University of Regina were used when conducting cross-validation experiments and systematically testing the presented methodology strictly following the procedure proposed in Chapter 3 to Chapter 6. These photos were collected from the Science Faculty after the ethical clearance was received from the University of Regina Research Ethics Board.

According to [1], the factors that can significantly affect the performance of a face recognition system include illumination variations, pose changes, facial expressions, time delay and occlusions. The performance of the method of principal component analysis
(PCA) can deteriorate if the background and illumination condition is not well controlled [101][197]. In other words, when PCA is applied, the photos can have variations, but such variations cannot be large. In addition, the size of the photos must be the same, and the size of the face on each photo must also be similar. Thus, when collecting the photos, they were taken of each participant indoor at three office locations without time delay. In all cases, the participant sat in front of the camera without pose changes or occlusions. The backgrounds were plain and white with similar but controlled fluorescent lighting. At each location, 24 photos were taken (6 photos of each facial expression) with varying facial expressions. There were many facial expressions, so four typical expressions were selected [116]: neutral, smile, scream and anger. Finally, participant’s photos were divided into three groups (24 photos/each group) based on where they were taken - each group contains all photos taken in one location.

As each person participated in this research project voluntarily and anonymously, their names and other personal information would not be released without the appropriate permission. Each participant was assigned an ID composed of a letter followed by 3 numbers (such as m-002, w-003 or m-005). Correspondingly, each photo was identified partially based on the ID of the participant plus the following information:

1. Index numbers of the photos having the same expression and taken at the same location;
2. Index number of the facial expression;
3. Index number of the location.
To explain further, if a photo is named *m-002-12-3.jpg*, then *m-002* is the ID number of the particular participant; *l* represents the index number of the photos having the same expression. A *2* represents a smiling expression and a *3* represents a photo taken at the third location. After photos are collected, they are processed manually one at a time (Figure 7.1). The portion having the face is identified and then the rest is cut out (as described by the first two photos in Figure 7.1). Next, the face photos are resized into 64 x 64 pixels and changed into black-and-white (as demonstrated by the last two photos in Figure 7.1) and used to comprise experimental sets.

Many face databases have been publicly available and free to academics; however, the unique standard protocol that can evaluate the performance of various face recognition algorithms is still not available [1]. The performance of a face recognition algorithm is typically characterized by three factors that are called false acceptance rate, false rejection rate and accuracy rate (also called correct identification rate). Thus, when addressing various experimental results in this chapter, the performance of the presented methodology is primarily evaluated by its accuracy rate, which is defined as:

**Figure 7.1 Steps of processing photo manually**

(1)  
(2)  
(3)  
(4)
**Definition 7.1** In an experiment, given a test set having $N$ photos, if the number of photos that the presented methodology can correctly classify is $M$, the accuracy rate of the presented methodology in the experiment is:

$$\frac{M}{N} \times 100\%.$$ (7.1)

For example, if the test set has 50 photos in an experiment and the presented methodology can correctly classify 45 photos, then according to (7.1), its accuracy rate in this experiment is 90%.

### 7.2 Cross-Validation

Cross-validation is a useful model evaluation method. The basic concept is that some of the data is removed from the training set before the training begins. When the training is completed, the removed data is used to test the performance of the hierarchy as ‘new’ data. Commonly used types of cross-validation include repeated random sub-sampling validation [57], $K$-fold cross-validation and leave-one-out cross-validation.

In $K$-fold cross-validation, each original training set is randomly partitioned into $K$ sub-training sets of the same size. A single sub-training set is selected from the $K$ sub-training sets as test data for the purpose of validation, and the remaining $K-1$ sub-training sets are used as training data. The cross-validation process is repeated $K$ times (the folds), usually 10-fold cross-validation is used [118], and each of the $K$ sub-training sets are used exactly once as validation data. The $K$ results from the folds are then averaged or combined to produce a single estimation.
In comparison to repeated random sub-sampling, the advantage of this method is all training data is used for both training and validation, and each datum is used for validation once. If $K$ is equal to the number of samples in a training set, then $K$-fold cross-validation becomes leave-one-out cross-validation. Validation involves using a single sample from the training each time, for the purpose of validation, and the remaining data is used as training data. This can be repeated so each sample in the training set can be used once for the purpose of validation. Leave-one-out cross-validation is often computationally expensive because the number of times the training process takes place is the same as the number of samples in the training set.

$K$-fold cross-validation was selected to validate the hierarchies in the experiments, but in order to take advantage of the $K$-fold cross-validation and leave-one-out cross-validation, $K$ was set equal to the number of photos of each participant in a training set. For example, if an experimental set has photos of 4 people and 48 photos of each person, then the value of $K = 48$.

### 7.3 $K$-fold Cross-Validation Experiments

The experiments can be divided into two categories. The first category is to evaluate the performance after the hierarchies are directly formed by the facial photos in the training set. The other is to evaluate the performance after all new photos have been added to the training set and the hierarchy has been modified, as explained in Chapter 6.
7.3.1 Purpose

The purpose of the experiments is to estimate how well the hierarchy of probabilistic decision tables, having just been formed by several training data, is going to perform on future, as-yet-unseen, data. Based on the experimental results, a preliminarily comparison can be performed between the presented methodology and other methods [190-192]. If the results are satisfactory, the system test is implemented with facial photos collected from the University of Regina.

7.3.2 Hierarchies Directly Formed by Photos in a Training Set

The experiments evaluating the system performance, after the hierarchies are directly formed by the facial photos in the training set, are covered in this Section.

1. Procedure

The experiments were completed in two steps as described below:

Step 1. Prepare experimental datasets

In order to construct all datasets, photos of 16 participants (8 men and 8 women) are selected. The first two photos of each facial expression (4 facial expressions in total), of each participant, taken at three different locations, are selected for the experiments. The photos are added to 16 original training sets (96 photos of 2 men and 2 women per set). When constructing each sub-training set, one photo of each participant in the original training set is selected randomly as the photo of that sub-training set. If a photo in the
original training set has been previously selected to construct another sub-training set, it
cannot be selected again at this particular time. This way, 24 sub-training sets are
constructed from each original training set. Then, each sub-training set is used as a test
set (having 4 photos and 1 photo of each participant). The remaining sub-training sets are
used as the corresponding training set (92 photos of 4 participants and 23 photos of each
participant). Thus, each original training set produces 24 training sets and 24 test sets.

Table 7.1 Sample results of the 6th original training set in two series of experiments

<table>
<thead>
<tr>
<th>Serial</th>
<th>Number of Photos Correctly classified of 24 Sub-Training Sets</th>
</tr>
</thead>
</table>

Step 2. Implement experiments

The experiments were conducted in two series. Since photos of each participant in an
original training set are selected randomly in Step 1, the photos selected for the same
sub-experimental set can be different for the two series. Each time, during each series of
the experiments, each training set is used to build four hierarchies of probabilistic
decision tables. Each hierarchy uses all the photos of one specific participant as the
elements in the target set $X$. After the four hierarchies are completed, they are tested by
photos in the corresponding test set and the number of correctly classified photos is
recorded. The results of the 6th original training set have been listed in Table 7.1.

The process above is repeated until all the training sets and test sets are utilized. Then,
the experimental results from each original training set are averaged and combined as a
single estimation, as listed in Table 7.2.
2. Experimental Results and Evaluation

According to the experimental results obtained from the same training set, for most of the time, the performance of the hierarchies is similar to each other in two serials. Occasionally, the performance can be very different. For instance, in Table 7.1 the hierarchies formed with the 2nd training set can classify 2 photos and 4 photos in the two series respectively (i.e., the accuracy of recognition rates is 50% and 100%).

The same situation is present when comparing the average experimental results of original training sets in two serials. For example, the averaged accuracy rates of those hierarchies formed with the 7th original training set are 100% in the first serial and

---

Table 7.2 Averaged experimental results

<table>
<thead>
<tr>
<th>Original Training Sets</th>
<th>No</th>
<th>The 1st Series of Experiments</th>
<th>The 2nd Series of Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average Correct Cases</td>
<td>Average Accuracy Rate(%)</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>3.96</td>
<td>99.00</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>3.92</td>
<td>98.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>3.96</td>
<td>99.00</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.88</td>
<td>97.00</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>3.92</td>
<td>98.00</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>3.92</td>
<td>98.00</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>4.00</td>
<td>100.00</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3.92</td>
<td>98.00</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>3.96</td>
<td>99.00</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>3.71</td>
<td>92.75</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>3.79</td>
<td>94.75</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>3.79</td>
<td>94.75</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>3.96</td>
<td>99.00</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>3.83</td>
<td>95.75</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>3.88</td>
<td>97.00</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>3.83</td>
<td>95.75</td>
</tr>
<tr>
<td>Average of Each Series</td>
<td></td>
<td>3.89</td>
<td>97.23</td>
</tr>
<tr>
<td>Average of Two Series (%)</td>
<td></td>
<td>96.84</td>
<td></td>
</tr>
</tbody>
</table>
93.75% in the second serial, respectively. This problem is partially due to the fact that the photos contained in the same training set in two series are different. Similar to other techniques in machine learning, the performance of the presented methodology depends upon the dataset forming the hierarchies. If the photos in the training sets changed, the accuracy rate of the hierarchies might also be changed. On the other hand, although the hierarchies formed by the same training set have different recognition rates in the two series, compared to existing methods, the level of performance is still satisfied as a whole. For example, the accuracy rate in [192] is about 99.24%, the methods in [191] are 86.25% - 88.75%, and the method in [176] is approximately 96.5%. As shown in Table 7.2, all accuracy rates of the hierarchies are greater than 90% and the averaged rate is approximately 96.84%.

### 7.3.3 Hierarchies after Incremental Learning

This Section describes the experiments used to evaluate the performance of the system after all the new photos have been added to the training set and the hierarchy has been modified, as presented in Chapter 6.

#### 1. Procedure

The experiments were completed in two steps, as described below:

**Step 1. Prepare experimental datasets**

Photos of the same 16 participants, using the experiments in the previous section, and a total of 16 original experimental sets were constructed. The participants of each
experimental set are the same as those of the previous section. Each original experimental set is divided into two sets: one is the original training set and the other is the add-on set. Photos of each expression taken at the 1st location are used as the photos for the original training set (96 photos, 2 men and 2 women, and 24 photos each).

When constructing each sub-training set, one photo of each participant in that original training set is selected randomly as the photo of that sub-training set. If a photo has been previously selected to construct another sub-training set, it cannot be selected again at this particular time. In this way, each initial training set is constructed into 24 sub-training sets. During the experiment, a sub-training set is used as the test set (having 4 photos and 1 photo each) each time, and the remainder of the sub-training sets are used to form the initial training set (92 photos from 4 participants and 23 photos of each participant). Each original training set can be used to construct 24 initial training sets and 24 test sets. The photos of each participant’s expression, taken at the 2nd location, are used as the photos for the add-on set (96 photos, 2 men and 2 women, 24 photos each).

**Step 2. Implement experiments**

The experiments are also deployed in two series. During each series of experiments, each time, four initial hierarchies of probabilistic decision tables are built by one initial training set. Each hierarchy uses all the photos of one specific participant in the initial training set as the elements in the target set $X$. Once an initial hierarchy is completed, it will be modified as the description in Chapter 6. Each time one photo is selected randomly from the add-on set, if not previously selected. First, the photo is pre-processed
as per the steps described in Section 6.3 and then classified into an appropriate elementary set. Next, the hierarchy of decision tables is modified based on the strategies described in Section 6.3.

**Table 7.3** Sample results of the 6th original training set in two series

<table>
<thead>
<tr>
<th>Serial</th>
<th>Number of Photos Correctly Classified of 24 Sub-Training Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3 4 4 4 4 4 2 4 4 3 3 4 4 4 4 4 4 3 4 4 3 4 4 4 4</td>
</tr>
<tr>
<td>2</td>
<td>4 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 3 4 4 2 4 4 4 4</td>
</tr>
</tbody>
</table>

**Table 7.4** Averaged experimental results

<table>
<thead>
<tr>
<th>Original Training Sets</th>
<th>No</th>
<th>The 1st Series of Experiments</th>
<th>The 2nd Series of Experiments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Average Correct Cases</td>
<td>Average Accuracy Rate(%)</td>
</tr>
<tr>
<td>1</td>
<td>3.88</td>
<td>97.00</td>
<td>3.92</td>
</tr>
<tr>
<td>2</td>
<td>3.96</td>
<td>99.00</td>
<td>3.79</td>
</tr>
<tr>
<td>3</td>
<td>3.83</td>
<td>95.75</td>
<td>3.79</td>
</tr>
<tr>
<td>4</td>
<td>3.96</td>
<td>99.00</td>
<td>3.79</td>
</tr>
<tr>
<td>5</td>
<td>3.96</td>
<td>99.00</td>
<td>3.92</td>
</tr>
<tr>
<td>6</td>
<td>3.67</td>
<td>91.75</td>
<td>3.79</td>
</tr>
<tr>
<td>7</td>
<td>3.92</td>
<td>98.00</td>
<td>3.79</td>
</tr>
<tr>
<td>8</td>
<td>3.83</td>
<td>95.75</td>
<td>3.79</td>
</tr>
<tr>
<td>9</td>
<td>3.79</td>
<td>94.75</td>
<td>3.79</td>
</tr>
<tr>
<td>10</td>
<td>3.79</td>
<td>94.75</td>
<td>3.79</td>
</tr>
<tr>
<td>11</td>
<td>3.92</td>
<td>98.00</td>
<td>3.92</td>
</tr>
<tr>
<td>12</td>
<td>3.96</td>
<td>99.00</td>
<td>3.96</td>
</tr>
<tr>
<td>13</td>
<td>3.79</td>
<td>94.75</td>
<td>3.83</td>
</tr>
<tr>
<td>14</td>
<td>3.88</td>
<td>97.00</td>
<td>3.83</td>
</tr>
<tr>
<td>15</td>
<td>3.83</td>
<td>95.75</td>
<td>3.92</td>
</tr>
<tr>
<td>16</td>
<td>3.75</td>
<td>93.75</td>
<td>3.83</td>
</tr>
<tr>
<td>Average of Each Series</td>
<td>3.86</td>
<td>96.44</td>
<td>3.84</td>
</tr>
<tr>
<td>Average of Two Series(%)</td>
<td>96.23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

After all the photos in the add-on set have been utilized for the 1st hierarchy, the same process is repeated on the remaining hierarchies. Following the modification of all the
hierarchies, they are tested by photos in the corresponding test set following the procedure described in Section 5.3 and the number of photos correctly classified is recorded. The process above is repeated until all the training sets and the test sets are utilized. A sample would be the results of the 6th original training set in two series of experiments in incremental learning being placed in Table 7.3. The experimental results such as the average number of photos correctly classified, the average accuracy rate for each original training set, and so on for two series, are listed in Table 7.4.

2. Experimental Results and Evaluation

According to the experimental results, the performance of the hierarchies is similar to that in Section 7.3.2. Occasionally, when formed by the same training set, the accuracy rate can be different. In Table 7.3, the hierarchies formed with the 7th training set can classify 2 photos and 4 photos in the two series. By analyzing the results, it is evident the hierarchies have been modified by strictly following the rules described in Section 6.1 for the process of incremental learning. After the process, its accuracy rates are always greater than 90% and the average rate is approximately 96.23%.

7.4 Experiments with Balanced and Unbalanced Datasets

In this Section, experiments with various datasets are introduced in which the hierarchies are directly formed by photos in the training set and tested by photos in the test set. The main purpose of the experiments in this section is to assist in determining
the impact of the factors introduced below on the performance of the presented methodology.

1. **Background and illumination conditions of each photo**

Photos were taken at three office locations, and the background is plain and white with similar but controlled fluorescent lighting. As addressed in Section 7.1, if the background or illumination condition is very different among photos when PCA is applied, it will fail. Thus, the goal is to determine whether the background and illumination conditions of the photos, taken at different locations, are very different or identical, by identifying if the accuracy rate of the hierarchies is significantly high or low when formed by photos taken at a specific location.

2. **The facial expression of each photo**

When preparing photos for the experiments, the facial expression of each participant can be controlled. Photos with all four facial expressions can be collected and the number of photos with each expression can be fixed. This control may not exist in practical application. This is because the facial expressions of all photos in a dataset can be of a smile, but the expression of an unknown photo can be angry.

3. **Gender ratio of the participants**

When constructing the datasets in the experiments, the gender of ratio of the participants in each training set can be controlled such as photos of two men and two women in each set. However, similar to facial expressions, the gender ratio is not always under control in practical applications. Often, the hierarchies have to be formed by a
dataset as when most of the participants are men or women. Thus, the gender ratio influence on the performance of the hierarchies must be identified as to whether the hierarchies can generate a relatively higher accuracy rate (greater than 90%) when the gender ratio is not 1:1.

4. Number of participants in the training set

The conditional attributes of the probabilistic decision table are binary-valued. When it is applied to face recognition, its conditional attributes are from selected but discretized facial features. With regard to such features, photos in the target set $X$ (from the same participant) share one of its possible values (0 or 1), and photos in the complimentary set have to share the other value (1 or 0) even if they are from different participants. When the number of participants in the training set is increased, the diversity of participants is increased with respect to age, race, complexion, and a cast of features. It is yet to be determined if the most appropriate features can be found when the number of participants in the training set is increased.

7.4.1 Experiments with Balanced Datasets

The hierarchies are formed by facial photos in the training set, so the representativeness of the photos is vital as with other machine learning methods. Unbalanced data is an important research topic in machine learning [80] [100][128][215], but it is not always easy to clearly identify if data in a dataset are unbalanced. In the presented methodology, the definition from [106] is applied. A set is considered to be a
balanced set if the total number of male and female participant(s) photos are the same; the number of photos of each participant is the same, and photos contain all the same facial expressions and the number of photos having different facial expressions is the same.

7.4.1.1 Purpose

As the first step, datasets having high representativeness (balanced datasets) are used in the experiments, and the purpose is to exam the performance of the hierarchies under very ideal conditions. Experiments were employed with 4-person data sets as an extension of the experiments described in Section 7.3.2. The averaged accuracy rate of the experiments in cross-validation is approximately 96.84%. The hierarchies were systematically formed with 4-person data sets and the results were then compared with that accuracy rate to determine whether the performance of the hierarchies could be better or worse than that in the cross-validation. The goal is to discover whether the quality of the photos collected has any significant impact on the performance of the hierarchies.

In addition, experiments with datasets with more than 4 participants were deployed. The purpose of the experiments is to estimate the relationship between the accuracy rate of the hierarchies and the size of the datasets, by comparing the accuracy rates of the hierarchies when the number of participants in the training set is increased.
7.4.1.2 Experiments with 4-person Datasets

1. Procedure

Step 1. Prepare experimental datasets

The experiments described in this section are conducted in four series. 784 experimental datasets (784 training sets and 784 testing sets) are prepared for each series. Photos are from the same participants as those for cross-validation experiments. Each experimental set has photos of 2 men and 2 women. The details of the sets are listed in Table 7.5.

Table 7.5 Photos selected for each experimental set

<table>
<thead>
<tr>
<th>Serial</th>
<th>Photos Selected for Each Set</th>
<th>Total Photos in Each Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Set</td>
<td>Test Set</td>
</tr>
<tr>
<td>1</td>
<td>First 4 photos of each (his/her) facial expression of each participant taken at three locations</td>
<td>Last 2 photos of each (his/her) facial expression of each participant taken at three locations</td>
</tr>
<tr>
<td>2</td>
<td>All photos of each participant taken at the first two locations</td>
<td>All photos of each participant taken at the 3rd location</td>
</tr>
<tr>
<td>3</td>
<td>All photos of each participant taken at the 1st location</td>
<td>All photos of each participant taken at the 3rd location</td>
</tr>
<tr>
<td>4</td>
<td>All photos of each participant taken at the 2nd location</td>
<td>All photos of each participant taken at the 3rd location</td>
</tr>
</tbody>
</table>

Step 2. Implement experiments

During the experiments, four hierarchies are built each time by one training set; then they are tested by photos in the corresponding test set, as described by the steps in Section 5.3. The accuracy rate is recorded as described in Figure 7.2 to Figure 7.5. When all test sets have been utilized in each series, the test results are averaged and the
results are used as an estimation of the performance of the hierarchy, as in Table 7.6.

<table>
<thead>
<tr>
<th></th>
<th>1st Serial</th>
<th>2nd Serial</th>
<th>3rd Serial</th>
<th>4th Serial</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Experimental Sets</strong></td>
<td>784</td>
<td>784</td>
<td>784</td>
<td>784</td>
</tr>
<tr>
<td><strong>Ave. Number of Photos correctly Classified</strong></td>
<td>87.3</td>
<td>88.54</td>
<td>88.22</td>
<td>88.3</td>
</tr>
<tr>
<td><strong>Ave. Accuracy Rate</strong></td>
<td>90.94%</td>
<td>92.20%</td>
<td>91.89%</td>
<td>91.89%</td>
</tr>
<tr>
<td><strong>Combined Ave. Accuracy Rate</strong></td>
<td><strong>91.73%</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. Experimental Results and Evaluations

Although photos selected from each training set in different series are different, according to the results listed in Table 7.6, the accuracy rates of different series are similar to each other. The accuracy rate is not significantly improved nor deteriorated by selecting photos taken at a particular location. It indicated the quality of photos of each participant, taken at different locations with the same facial expressions, can be considered identical, and all of photos can be used in the experiments.

Among the experiments, the average accuracy rate of the four series is approximately 91.73%. Such a rate is similar to but relatively lower than that of the cross-validation, which is about 96.84%. If photos of 16 participants (8 men and 8 men) are obtained and each experimental set has photos of 2 men and 2 women, then, theoretically, 1820 experimental sets can be constructed. However, only 16 photos were selected in the cross-validation experiments, and with respect to the experiments described in this section, 784 were selected. Hence, if the change is lower than the cross-validation, it is acceptable and understandable. The diversity of the experimental data is increased when
the number of experimental sets is increased.

**Figure 7.2** Accuracy rate (%) of each training set in the 1\textsuperscript{st} series

**Figure 7.3** Accuracy rate (%) of each training set in the 2\textsuperscript{nd} series
Figure 7.4 Accuracy rate (%) of each training set in the 3rd series

Figure 7.5 Accuracy rate (%) of each training set in the 4th series
7.4.1.3 Experiments with Datasets having More than 4 Participants

1. Procedure

Step 1. Prepare experimental datasets

The first four photos of each facial expression of each participant, taken at all locations (48 photos), are always selected as the photos for the training set; the remaining photos of each participant (24 photos) are selected for the test set. The remaining details of the sets have been listed in Table 7.7.

Step 2. Implement experiments

Each time, during the experiment, every hierarchy is built by one training set having photos of 6, 8, or 10 participants. The photos in the corresponding test set are used to test the hierarchies. The accuracy rates are recorded as shown in Figure 7.6 to Figure 7.8 and the average accuracy rates are presented in Table 7.7.

Table 7.7 Photos selected for each experimental set and average accuracy rate

<table>
<thead>
<tr>
<th>Total Participants in Each Dataset</th>
<th>Total Experimental Sets</th>
<th>Total Photos in Each Set</th>
<th>Average Accuracy Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training Sets</td>
<td>Test Sets</td>
<td>Training Set</td>
</tr>
<tr>
<td>6</td>
<td>196</td>
<td>196</td>
<td>288</td>
</tr>
<tr>
<td>8</td>
<td>196</td>
<td>196</td>
<td>384</td>
</tr>
<tr>
<td>10</td>
<td>196</td>
<td>196</td>
<td>480</td>
</tr>
</tbody>
</table>

2. Experimental Results and Evaluations

In reviewing the results in Table 7.6 and Table 7.7, the number of participants is inversely proportional to the accuracy rate. The hierarchy built by a 4-person training set can always have the highest accuracy rate. For example, the averaged accuracy rate of the hierarchies built by 6-person datasets is approximately 87.28%, and the averaged
accuracy rate of the hierarchies built by 8-person datasets is approximately 80.05%.

The variations among photos of the same face or different faces can be huge, but their overall configurations are similar. Thus, such a change is primarily due to the number of “features” distinguishing the faces in the target set $X$ from the number of deteriorating faces in the complimentary set of the target set $X$ when the number of participants is increased.

The conditional attributes of the probabilistic decision table are binary-valued (1 or 0). When it is applied to face recognition, each conditional attribute is from one selected but discretized facial feature. For such a feature, photos in the target set $X$ (from the same participant) are sharing one of its possible values, and the photos in the complimentary set must share the other value although they are from different participants.

![Test Results of the Experiments with 6-person Datasets](image)

**Figure 7.6** Accuracy rate (%) of experiments with 6-person datasets
When comparing photos of one participant in the target set $X$ to the photos of a very few participants in the complimentary set, (comparing the photos of a man to the photos
of a woman), there can be many features easily found. When the number of participants is large, the feature will become difficult to find.

7.4.2 Experiments with Unbalanced Datasets

7.4.2.1 Purpose

In practice, a dataset is often unbalanced, so its performance must be determined when the hierarchies are formed by a dataset with relatively low representativeness (unbalanced data). Systematically testing the presented methodology in various unbalanced data will be very time-consuming, so only the two typical cases below are considered, and others are suggested for future work.

1. The type of facial expressions in the dataset is less than four expressions (such as, the photos wherein the expressions are only neutral or smiling). However, the number of photos of male participants is the same as that of female participants.

2. The total number of male participant(s) photos is not the same as that of female participant(s) photos. For instance, a dataset has photos of one male participant and three female participants and; the photos contain four facial expressions and the number of photos with each facial expression is the same.

By comparing the results of the experiments to the results of the experiments with balanced datasets, it is yet to be determined if the results would work well in the cases above.
7.4.2.2 Experiments with Datasets Having Specific Facial Expressions

1. Procedure

Step 1. Prepare experimental datasets

As described in Table 7.8, six groups of experimental sets are constructed, and then, each experimental set, in each group, is divided into a training set having photos with one facial expression and a test set having photos with other facial expressions. In each group, based on the number of participants, the experimental sets are divided into four types as described in Table 7.9 to Table 7.12.

<table>
<thead>
<tr>
<th>Group</th>
<th>Data Set</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Expression Training Set</td>
<td>angry</td>
<td>angry</td>
<td>neutral</td>
<td>neutral</td>
<td>smile</td>
<td>smile</td>
</tr>
<tr>
<td>2</td>
<td>Expression Test Set</td>
<td>neutral</td>
<td>smile</td>
<td>angry</td>
<td>smile</td>
<td>angry</td>
<td>neutral</td>
</tr>
</tbody>
</table>

Step 2. Implement experiments

The hierarchies are built by one training set each time in the experiments and then the photos in the corresponding test set are used to test the hierarchies. Finally, the experimental results are recorded and then averaged as shown in Tables 7.9 to 7.13 and Figure 7.9.

2. Experimental Results and Evaluations

Viewing the table results, the experiments can be summarized:

Firstly, similar to the hierarchies built by balanced training sets in Section 7.4.1, the
accuracy rate of the hierarchies built by training sets, having photos from 4 participants, is always higher. When the number of participants in a training set is increased, the accuracy rate is always decreased.

**Table 7.9** Experiments with datasets having 4-participants

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Photos in Each Training Set</td>
<td>72</td>
</tr>
<tr>
<td>Total Photos in Each Test Set</td>
<td>72</td>
</tr>
<tr>
<td>Total Training Sets</td>
<td>784</td>
</tr>
<tr>
<td>Total Test Sets</td>
<td>784</td>
</tr>
<tr>
<td>Ave. # of Photos Correctly Classified</td>
<td>64.52</td>
</tr>
<tr>
<td>Ave. Accuracy Rate (%)</td>
<td>89.61</td>
</tr>
</tbody>
</table>

**Table 7.10** Experiments with datasets having 6-participants

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Photos in Each Training Set</td>
<td>108</td>
</tr>
<tr>
<td>Total Photos in Each Test Set</td>
<td>108</td>
</tr>
<tr>
<td>Total Training Sets</td>
<td>112</td>
</tr>
<tr>
<td>Total Test Sets</td>
<td>112</td>
</tr>
<tr>
<td>Ave. # of Photos Correctly Classified</td>
<td>88.62</td>
</tr>
<tr>
<td>Ave. Accuracy Rate (%)</td>
<td>82.05</td>
</tr>
</tbody>
</table>

**Table 7.11** Experiments with datasets having 8-participants

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Photos in Each Training Set</td>
<td>144</td>
</tr>
<tr>
<td>Total Photos in Each Test Set</td>
<td>144</td>
</tr>
<tr>
<td>Total Training Sets</td>
<td>140</td>
</tr>
<tr>
<td>Total Test Sets</td>
<td>140</td>
</tr>
<tr>
<td>Ave. # of Photos Correctly Classified</td>
<td>115.96</td>
</tr>
<tr>
<td>Ave. Accuracy Rate (%)</td>
<td>80.53</td>
</tr>
</tbody>
</table>
Secondly, facial expressions have an impact on the performance of the hierarchies. For example, if the facial expression of the photo in the training set is angry, the accuracy rate of the corresponding hierarchy is always lower than the other accuracy rates whether or not the hierarchy is built by a training set having photos of 4, 6, 8 or 10 participants. If a hierarchy is built by a training set having photos with a neutral expression, and tested by a test set having photos with a smiling expression, the accuracy rate of that hierarchy is always higher than the other accuracy rates whether or not the hierarchy is built by a training set having photos of 4, 6, 8 or 10 participants.

Thirdly, compared to the experimental results in Section 7.4.1, the averaged accuracy rate of the hierarchies, as shown in Table 7.13, built by training sets only having a specific facial expression, is close to hierarchies built by balanced training sets. In a few cases, the averaged accuracy rate of the hierarchies, built by the training sets with a specific facial expression can be higher. For example, as shown in Table 7.6, the averaged accuracy rate of hierarchies built by a balanced 4-person dataset is approximately 91.73%, and the averaged rate of hierarchies built by the unbalanced 4-person dataset is

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
<th>Group 4</th>
<th>Group 5</th>
<th>Group 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Photos in Each Training Set</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>Total Photos in Each Test Set</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>Total Training Sets</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Total Test Sets</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Ave. # of Photos Correctly Classified</td>
<td>134.4</td>
<td>144.86</td>
<td>140.98</td>
<td>153.46</td>
<td>143.99</td>
<td>145.96</td>
</tr>
<tr>
<td>Ave. Accuracy Rate (%)</td>
<td>74.68</td>
<td>80.45</td>
<td>78.33</td>
<td>85.75</td>
<td>80</td>
<td>81.08</td>
</tr>
</tbody>
</table>
approximately 92.01%.

### Table 7.13 Averaged accuracy rate of experiments with datasets having different sizes and different facial expressions

<table>
<thead>
<tr>
<th>Group</th>
<th>4-person</th>
<th>6-person</th>
<th>8-person</th>
<th>10-person</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>89.61</td>
<td>82.05</td>
<td>80.5</td>
<td>74.68</td>
</tr>
<tr>
<td>2</td>
<td>92.07</td>
<td>87.388</td>
<td>84.9</td>
<td>80.45</td>
</tr>
<tr>
<td>3</td>
<td>92.38</td>
<td>83.55</td>
<td>82.5</td>
<td>78.33</td>
</tr>
<tr>
<td>4</td>
<td>93.96</td>
<td>89.038</td>
<td>87.8</td>
<td>85.25</td>
</tr>
<tr>
<td>5</td>
<td>91.67</td>
<td>89</td>
<td>85.5</td>
<td>80</td>
</tr>
<tr>
<td>6</td>
<td>92.37</td>
<td>88.025</td>
<td>86.1</td>
<td>81.08</td>
</tr>
</tbody>
</table>

| Ave. Accuracy Rate(%) of 6 Groups | 92.01 | 86.51 | 84.55 | 79.97 |

![Figure 7.9 Chart of averaged accuracy rate based on Table 7.13](image)

### 7.4.2.3 Experiments with Datasets with Unbalanced Gender Ratio

#### 1. Procedure

**Step 1. Prepare experimental datasets**

Eight groups of experiments were constructed, the details of which are listed in Table 7.14. In each experimental set, the first four photos of each facial expression, of each participant, taken at all locations are used to construct the training set, and the remainder
is for the test set. Each experimental set has photos consisting of all facial expressions, but the ratio of male participants and female participants is not 1:1.

**Step 2. Implement experiments**

Similar to experiments in the previous Sections, the hierarchies are built by one training set each time and after that photos in the test set are used to test the hierarchies. Finally, the experimental results are recorded and then averaged as shown in Table 7.15.

**Table 7.14** The number of experimental sets, participants and photos

<table>
<thead>
<tr>
<th>Group</th>
<th>Experimental Sets</th>
<th>Each Experimental Set</th>
<th>Photos/Per Participant</th>
<th>Photos in Each Training Set</th>
<th>Photos in Each Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>112</td>
<td>1M 3W</td>
<td>72</td>
<td>192</td>
<td>96</td>
</tr>
<tr>
<td>2</td>
<td>112</td>
<td>3M 1W</td>
<td>72</td>
<td>192</td>
<td>96</td>
</tr>
<tr>
<td>3</td>
<td>140</td>
<td>2M 4W</td>
<td>72</td>
<td>288</td>
<td>144</td>
</tr>
<tr>
<td>4</td>
<td>140</td>
<td>4M 2W</td>
<td>72</td>
<td>288</td>
<td>144</td>
</tr>
<tr>
<td>5</td>
<td>112</td>
<td>2M 6W</td>
<td>72</td>
<td>384</td>
<td>192</td>
</tr>
<tr>
<td>6</td>
<td>112</td>
<td>6M 2M</td>
<td>72</td>
<td>384</td>
<td>192</td>
</tr>
<tr>
<td>7</td>
<td>28</td>
<td>2M 8W</td>
<td>72</td>
<td>480</td>
<td>240</td>
</tr>
<tr>
<td>8</td>
<td>28</td>
<td>8M 2W</td>
<td>72</td>
<td>480</td>
<td>240</td>
</tr>
</tbody>
</table>

**Table 7.15** Average number of photos correctly classified and average accuracy rate of various groups

<table>
<thead>
<tr>
<th>Average of Photos Correctly Classified</th>
<th>Group</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>85.91</td>
<td>Average of Photos Correctly Classified</td>
<td>86.15</td>
<td>117.29</td>
<td>119.41</td>
<td>148.12</td>
<td>144.87</td>
<td>170.32</td>
<td>169.93</td>
<td></td>
</tr>
<tr>
<td>89.49</td>
<td>Average of Accuracy Rate (%)</td>
<td>89.74</td>
<td>81.45</td>
<td>82.92</td>
<td>77.14</td>
<td>75.45</td>
<td>70.97</td>
<td>70.80</td>
<td></td>
</tr>
</tbody>
</table>
2. Experimental Results and Evaluations

Based on the experimental results, it is concluded:

1. When the total number of participants is the same, the accuracy rate of hierarchies, built by the training sets in which most of the participants are male, is very close to that by training sets in which most participants are female. For example, the accuracy rate of group 1 is 89.49%, and group 2 is 89.74% in Table 7.15.

2. When hierarchies are built by the training sets with unbalanced gender ratios, the accuracy rate still depends upon the total number of participants. For example, the accuracy rate of hierarchies built by the training sets having photos of 4 participants is still the highest. When the number of participants is increased, the accuracy rate is always decreased whether the most of photos are from female or male participants.

3. Compared to the experimental results of Section 7.4.1, when the number of participants is the same, the accuracy rate of hierarchies, built by training sets with unbalanced gender ratio, is always lower.

7.4.3 A Brief Summary

Based on the experimental results in Sections 7.4.1 and 7.4.2, the performance of the presented methodology is summarized:

1. If photos are taken at different locations against a similar background and under similar lighting, their quality is identical. In contrast, where a photo is taken is not a significant factor when evaluating the experimental results.
2. When the number of participants is fixed, the performance of the hierarchies can be improved, or at least not be deteriorated, if it is trained by the training set with photos only having a specific facial expression. Its performance can always deteriorate, if the gender ratio of the training set is unbalanced.

3. Whether or not it is trained by the balanced training sets, its performance is always inversely proportional to the number of participants. This indicates the presented methodology is best suited to cases where the training set has a relatively small number of participants.

4. As summarized in Table 2.3, compared to the methods described in Section 2.1, the performance of the presented methodology can be approximately the same as those methods described in Section 2.1 in a case of being trained by the training set having a relatively small number of participants. For example, if there are 4 participants in the training set, the accuracy rates are greater than 90%.

7.5 Incremental Learning Experiments

In this Section, incremental learning experimental results are presented. Similar to the experiments in previous sections, the hierarchies are tested by datasets with different representativeness.

7.5.1 Purpose

The primary purpose of the experiments is to assist in determining if the presented
methodology is suitable for incremental learning. It has yet to be determined:

1. **How the presented methodology learns new photos**

   As described in Chapter 6, when a new object is available, if a classifier is completely standing on stability, it will totally discard the existing model and regenerate using all accumulated data so far. If a classifier is completely plastic, it can learn quite well from a new object but cannot conserve prior knowledge. Thus, it needs to determine where the presented methodology stands in regard to the stability-plasticity spectrum.

2. **Following incremental learning, can the hierarchies still correctly classify unknown photos**

   During the incremental learning process, new photos are added, which means the representativeness of the training set has been improved. After adding new photos, the diversity of the photos in the training set increases. For instance, initially the training set only has photos with a specific facial expression, but during the incremental learning process, photos with new facial expressions are added. If the hierarchies cannot correctly classify unknown photos after incremental learning, the presented methodology will not be suitable for incremental learning although it can learn from a new object and conserve prior knowledge in the mean time.

3. **The impact of several factors mentioned in previous sections**

   Based on the results from previous sections, factors such as the number of participants, facial expression and the gender ratio of the participant in the training set have an impact on the performance of the presented methodology. For example, the
accuracy rate is inversely proportional to the number of participants.

4. Whether the order of adding new photos to a training set has a significant impact

In practice, after the hierarchies are formed, new photos can be added to the initial training set at any time, when they are available. Usually, it is impossible to predict which photo will be the first photo available for addition to the initial training set, which photo will be the second photo, and so on.

7.5.2 Experiments with Balanced Datasets

The definition of balanced datasets is the same as the definition in the previous Section: in a training set or a test set, the gender ratio is 1:1, all four facial expressions are included, and the number of photos of each facial expression is the same.

1. Procedure

Step 1. Prepare experimental datasets

Based on the number of participants, four types of experimental datasets were prepared for the experiments, as shown in Table 7.16.

Step 2. Implement experiments

Experiments with balanced 4-person datasets were employed in three serials. Each time, four hierarchies are built by the initial training set. Subsequently, randomly selected photos from the add-on set are added, one by one, to the initial training set and then the hierarchies are modified according to the steps described in Section 6.3. After all photos in the add-on set have been added to the training set, the photos in the test set are used to
test the hierarchies as the procedure described in Section 5.3 and the results are recorded and averaged as shown in Figures 7.10 to 7.12 and Table 7.17.

**Table 7.16** Experimental sets for various experiments

<table>
<thead>
<tr>
<th>Serial</th>
<th>Photos Selected for Each Set</th>
<th>Total Photos in Each Set And Total Sets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial Training Set</td>
<td>Add-On Set</td>
</tr>
<tr>
<td>4-person</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>First 2 photos of each facial expression, taken at 3 locations</td>
<td>3rd and 4th photos of each facial expression taken at 3 locations</td>
</tr>
<tr>
<td>2</td>
<td>Photos of each facial expression taken at 1st location</td>
<td>Photos of each facial expression taken at 2nd location</td>
</tr>
<tr>
<td>6-person</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>First 2 photos of each facial expression, taken at 3 locations</td>
<td>3rd and 4th photos of each facial expression taken at 3 locations</td>
</tr>
<tr>
<td>8-person</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>First 2 photos of each facial expression, taken at 3 locations</td>
<td>3rd and 4th photos of each facial expression taken at 3 locations</td>
</tr>
<tr>
<td>10-person</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The second series of experiments uses the same experimental sets as those in the first series. The difference is the order of photos selected from the corresponding add-on set. The experiments in the second series are primarily counterparts of the first series. The purpose of the arrangement is to compare the accuracy rate of the two series to determine whether the order of adding photos to the training set is a factor which greatly affects the performance of the hierarchies.
The experiments with balanced 6-person, 8-person or 10-person datasets were consistently employed in two serials, but shared the same datasets. Similar to first two series of experiments with 4-person datasets, the order of adding photos to the training set is different, and the second series is the counterpart of the first experimental series. The procedure of each series is the same as that of experiments with balanced 4-person datasets.

2. Experimental Results and Evaluations

Looking through the results of the first two serials of 4-person cases, their total average accuracy rates are similar to each other, 93.31% and 93.15%. In the first series, the number of correctly classified photos, in each experiment, has been quantitatively compared, denoted as $E_{i1}$ to its counterpart in the section series denoted as $E_{i2}$, based on the following two steps:

1. The difference between $E_{i1}$ and $E_{i2}$ denoting as $DE_i = E_{i1} - E_{i2}$ is computed
2. The absolute value of $|DE_i|$ is divided by $E_{i1}$ and multiplied by 100%, which is denoted as $RD_i$

<table>
<thead>
<tr>
<th>Table 7.17</th>
<th>Averaged experimental results in each serials</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Serials in 4-person cases</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Average of photos correctly classified</td>
<td>89.58</td>
</tr>
<tr>
<td>Average of accuracy rate(%) in each serial</td>
<td>93.31</td>
</tr>
<tr>
<td>Total average of accuracy rate (%)</td>
<td>93.80</td>
</tr>
</tbody>
</table>
Based on the result, the value of \(RD_i\) in 654 experiments is smaller than 5%. The accuracy rate of most test sets in the first series is similar with their counterparts in the second series. The impact of the order of adding photos to the training set on the accuracy rate is not apparent. Compared to the results of the first two series, the accuracy rate of the third series of experiments is similar with those in the first two series. The average accuracy rates of the three serials are 93.31%, 93.15% and 94.93% respectively.

Reviewing the results of the experiments with 6-person, 8-person or 10-person datasets, the impact of the order of photos added to the training set on the accuracy rate remains unapparent. For example, the average accuracy rates of two group experiments of 6-person cases are 88.19% and 89.06%, respectively.

If the results in this section are compared to the results wherein the hierarchies are directly formed by the photos in the training set as described in Section 7.4.1, it is noted:

1. Following incremental learning, the accuracy rate of the hierarchies remains inversely proportional to the number of participants in the training set. For example, the average accuracy rate for 4-person cases, following incremental learning, is approximately 93.80%, and the rate for 6-person cases becomes 88.63%. It seems the number of participants is always an important factor. It is important whether the hierarchies are directly formed by the photos in the training set or formed by the photos in the initial training set first and then modified into incremental learning.
2. Following incremental learning, as the classifier, the hierarchies are still able to accurately classify the unknown photos into appropriate categories. Moreover, the averaged accuracy rates with datasets having a different number of participants, following incremental learning, are always increased. It is believed such a change is because the representativeness of the training set has been improved during the process of incremental learning and correspondingly, the overall performance of the hierarchies is improved.
Figure 7.11 Photos correctly classified in each experiment of series 2 of 4-person cases

Figure 7.12 Photos correctly classified in each experiment of series 3 of 4-person cases
7.5.3 Experiments with Unbalanced Datasets

The definition of unbalanced datasets in this section is the same as the definition used in previous sections: in a dataset (a training set, an add-on set or a test set), the gender ratio is not 1:1, or only a few facial expressions are included but the number of photos of each facial expression is the same.

7.5.3.1 Experiments with Datasets having Specific Facial Expressions

1. Procedure

Step 1. Prepare experimental datasets

As described in Table 7.18, four types of experimental sets are constructed for the experiments. Based on the results from Section 7.4.2.2, it appears facial expressions always have an impact on the performance of the hierarchies. If a hierarchy is built by
photos with a neutral expression, and tested by photos with a smiling expression, the accuracy rate is always higher than the others. If the facial expression is angry, the accuracy rate of the corresponding hierarchy is always lower than the others.

In each experiment, all photos (18 photos) with neutral expressions of each participant are used for the initial training set to ensure the hierarchies formed by the initial training set can have relatively high quality. For each add-on set in the first two types of experiments, all 18 photos with angry expressions are selected as its elements. The purpose is to determine the performance of the hierarchies when adding the photos that can potentially lead to a relatively low accuracy rate. For each add-on set in the last two types of experiments, all photos (18 photos) with scream expressions are selected for the add-on set. The photos with a scream expression are not selected for the purpose of incremental learning in Section 7.4.2.2, its impact on the performance of the hierarchies is yet to be determined. For each test set, all photos (18 photos) with smiles are used to help compare the results in this section to that of Section 7.4.2.2.

<table>
<thead>
<tr>
<th>Type</th>
<th>Facial Expressions</th>
<th>Number of participants</th>
<th>Number of experimental sets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Initial training set</td>
<td>Add-on set</td>
<td>Test set</td>
</tr>
<tr>
<td>1</td>
<td>neutral</td>
<td>angry</td>
<td>smile</td>
</tr>
<tr>
<td>2</td>
<td>neutral</td>
<td>angry</td>
<td>smile</td>
</tr>
<tr>
<td>3</td>
<td>neutral</td>
<td>scream</td>
<td>smile</td>
</tr>
<tr>
<td>4</td>
<td>neutral</td>
<td>scream</td>
<td>smile</td>
</tr>
</tbody>
</table>

Table 7.18 Details of each type of experimental set

Step 2. Implement experiments

Using each type of experimental sets, two series of experiments are separately
deployed, and similar to the experiments in previous sections, the order of photos selected from the add-on set is different in the two series. The process of the experiments is almost the same as the previous section experiments. Results of the experiments have been listed in Table 7.19.

**Table 7.19** Results of experiments with datasets having specific facial expressions

<table>
<thead>
<tr>
<th>Experimental Set</th>
<th>Angry Expression for Add-on Set</th>
<th>Scream Expression for Add-on Set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>6-person Cases</td>
<td>8-person Cases</td>
</tr>
<tr>
<td></td>
<td>Serial</td>
<td>Serial</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>97</td>
<td>106</td>
</tr>
<tr>
<td>2</td>
<td>103</td>
<td>89</td>
</tr>
<tr>
<td>3</td>
<td>102</td>
<td>101</td>
</tr>
<tr>
<td>4</td>
<td>106</td>
<td>104</td>
</tr>
<tr>
<td>5</td>
<td>106</td>
<td>103</td>
</tr>
<tr>
<td>6</td>
<td>84</td>
<td>86</td>
</tr>
<tr>
<td>7</td>
<td>104</td>
<td>106</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>95</td>
</tr>
<tr>
<td>9</td>
<td>102</td>
<td>105</td>
</tr>
<tr>
<td>10</td>
<td>92</td>
<td>105</td>
</tr>
<tr>
<td>Ave. of Photos Correctly Classified</td>
<td>99.6</td>
<td>100</td>
</tr>
<tr>
<td>Ave. of Accuracy Rate (%)</td>
<td>92.22</td>
<td>92.59</td>
</tr>
<tr>
<td>Total Ave. of Accuracy Rate (%)</td>
<td>92.41</td>
<td>91.06</td>
</tr>
</tbody>
</table>

2. Experimental Results and Evaluations

By comparing the results in this section with the results of the experiments with datasets having only specific facial expressions in Section 7.4.2.2, it is evident that the accuracy rate is always improved rather than decreased following the implementation of incremental learning. Following incremental learning, the representativeness of the training photos is increased (photos contain two types of facial expressions). As shown in Table 7.13, the average accuracy rate for 6-person balanced cases is approximately 86.51% and the rate for 6-person in Table 7.19 is 92.41%.
In addition, the accuracy rate of 6-person cases is always greater than the accuracy of 8-person cases, which is similar to the conclusion in Section 7.4. For instance, there are 108 photos in each test set. According to Table 7.19, if the photos with an angry expression are selected for the add-on set, the average accuracy rate for 6-person cases is approximately 92.41%, and the rate for 8-person cases is approximately 91.64%.

Finally, the average accuracy rates for 6-person cases are similar whether selecting photos with angry expressions or scream expressions as the elements for the add-on set. If the number of participants is 8, the impact of facial expression seems relatively important. The average accuracy rate of hierarchies having scream expression photos for the add-on set is approximately 87.47%, and the rate of those having angry expression photos for the add-on set is approximately 91.04%.

7.5.3.2 Experiments with Datasets having Unbalanced Gender Ratio

1. Procedure

Step 1. Prepare experimental datasets

Four types of experimental sets are constructed. In each experimental set, two photos of each facial expression of each participant (24 photos totally), taken at three different locations, are used for the initial training set. The two other photos are for the add-on set and the remaining photos are for the test set. The details are listed in Table 7.20.

Step 2. Implement experiments

Using each type of experimental sets, two series of experiments are deployed
separately, and the results are listed in Table 7.20. Similar to the experiments in previous sections, the experiments in the second series employ the same experimental sets, but the order of selected photos from the add-on set is different.

Table 7.20 Results of experiments with datasets having unbalanced gender ratio

<table>
<thead>
<tr>
<th>Experimental Set</th>
<th>2 Men and 4 Women Cases</th>
<th>4 Men and 2 Women Cases</th>
<th>2 Men and 6 Women Cases</th>
<th>6 Men and 2 Women Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Serial 1 2</td>
<td>Serial 1 2</td>
<td>Serial 1 2</td>
<td>Serial 1 2</td>
</tr>
<tr>
<td>1</td>
<td>114 106</td>
<td>126 128</td>
<td>131 154</td>
<td>168 151</td>
</tr>
<tr>
<td>2</td>
<td>103 142</td>
<td>116 110</td>
<td>155 166</td>
<td>145 146</td>
</tr>
<tr>
<td>3</td>
<td>137 135</td>
<td>142 139</td>
<td>135 136</td>
<td>172 139</td>
</tr>
<tr>
<td>4</td>
<td>114 111</td>
<td>121 136</td>
<td>151 175</td>
<td>119 112</td>
</tr>
<tr>
<td>5</td>
<td>134 131</td>
<td>139 131</td>
<td>164 126</td>
<td>168 170</td>
</tr>
<tr>
<td>6</td>
<td>134 136</td>
<td>131 131</td>
<td>160 176</td>
<td>151 146</td>
</tr>
<tr>
<td>7</td>
<td>128 131</td>
<td>129 110</td>
<td>142 119</td>
<td>134 162</td>
</tr>
<tr>
<td>8</td>
<td>129 139</td>
<td>87 103</td>
<td>139 147</td>
<td>142 129</td>
</tr>
<tr>
<td>9</td>
<td>110 107</td>
<td>136 128</td>
<td>130 144</td>
<td>155 172</td>
</tr>
<tr>
<td>10</td>
<td>132 137</td>
<td>118 112</td>
<td>145 151</td>
<td>168 161</td>
</tr>
<tr>
<td>11</td>
<td>135 135</td>
<td>135 139</td>
<td>139 159</td>
<td>154 153</td>
</tr>
<tr>
<td>12</td>
<td>130 133</td>
<td>107 87</td>
<td>135 142</td>
<td>140 133</td>
</tr>
<tr>
<td>13</td>
<td>129 131</td>
<td>124 137</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>137 127</td>
<td>128 122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>122 134</td>
<td>120 123</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>129 103</td>
<td>108 132</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>134 134</td>
<td>130 108</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>139 128</td>
<td>111 86</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>113 128</td>
<td>138 139</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>123 122</td>
<td>128 112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Photos</td>
<td>126.3 127.5 123.7 120.65</td>
<td>143.83 149.58 151.33 147.83</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Average Rate(%)</td>
<td>88.13 84.84 76.41 77.71</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 2. Experimental Results and Evaluations

By examining Table 7.20, the accuracy rate of 6-person cases is always better than 8-person cases. There are 144 photos in each test set for 6-person cases. When each experimental set has photos of 2 men and 6 women, the averaged accuracy rate is approximately 88.13%. When each experimental set has photos of 2 men and 6 women (there are 192 photos in each test set), the averaged accuracy rate is approximately
When comparing the results in this section with the results in Section 7.4, Section 7.5.2 and Section 7.5.3.1, when the number of participants is fixed, the performance of the hierarchies can always be improved or at least not be deteriorated if it is trained by a set having only photos with specific facial expressions. Its performance can always deteriorate if the gender ratio of the training set is unbalanced. In Table 7.17, the averaged accuracy rate of experiments with 6-person and balanced datasets is about 88.63%, and in Table 7.20 the rate is 86.49. The impact of gender ratio is greater than the impact of facial expression, and unlike facial expression, unbalanced gender ratio has a negative impact on the accuracy rate.

Similar to the conclusion in Section 7.4, the impact of the number of participants is still substantial on the accuracy rate. Although an experimental set has an unbalanced gender ratio, the corresponding hierarchies are still expected to have relatively high accuracy rates if the number of participants is small.

### 7.5.4 A Brief Summary

As described in Section 2.8, various algorithms proposed for the purpose of incremental learning can fall somewhere in between the stability-plasticity spectrum. During the experiments, the hierarchies of decision tables are updated when a new incoming photo, as well as observing the impact of the number of participants, facial expressions and gender ratio in the training sets on the performance. In essence, the...
purpose of the experiments in Section 7.5 is to determine where the presented methodology is positioned in between the stability-plasticity spectrums. When the presented methodology was applied to the incremental learning process, whether using balanced or unbalanced data, a case was never discovered where the presented methodology continued to re-generate the hierarchies as each new photo appeared until all photos from the add-on set were utilized. Based on observation, it is concluded the hierarchies of decision tables are not a completely stable classifier unable to learn any new cases and keep their prior knowledge in the meantime.

Based on the experimental results, the performance of the presented methodology can always be classified into two cases. The first case is to continue to update the decision tables during the incremental learning process, such as in the hierarchy 4 in Figure 7.14, hierarchy 4 in Figure 7.15, and hierarchies 1 and 2 in Figure 7.16. The hierarchies of decision tables can be a classifier that can completely satisfy the four properties proposed in [161]. The second case is that totally generating the hierarchy of decision tables alternates with updating the hierarchy of decision tables. During the incremental learning process, the hierarchies of decision tables will be totally re-generated once a few new photos are added. After the initial hierarchies are built and a new photo is added, each hierarchy should be updated first and its \( \lambda \text{-dependency} \) is also re-calculated. Then, the value of the updated \( \lambda \text{-dependency} \) is used as a factor with which to evaluate the hierarchy in the same way the method in Chapter 6 is used. Compared to its previous value, if the updated value becomes significantly low, the
hierarchies are totally regenerated. If it is not too low, the hierarchy is retained. Usually, after a few photos are added, $\lambda$–dependency becomes relatively low, so the hierarchy is totally re-generated. The number of randomly selected photos causing total regeneration was not constant in practice: it can be a few or many photos.

For example, the regeneration points are clearly visible on hierarchies 1, 2 and 3 in Figure 7.14. Hierarchies 1, 2, 3, 5 and 6 in Figure 7.15, and hierarchies 3, 4, 5, and 6 in Figure 7.16, are the discontinuity points of the curve representing the variation of the $\lambda$–dependency of the hierarchy in the incremental learning process. Moreover, this underscores the importance of the appropriate selection of cut points and attributes in the classifier construction process. The hierarchies of the decision tables learn incoming new photos according to the following rules:

1. It is not a classifier that is totally stable and unable to learn any new cases.

2. It can satisfy the four properties proposed in [161] in some cases and in other cases, it is a classifier falling between the stable-plasticity spectrum.

3. Compared to the neural network and other techniques described in Section 2.1, it is a technique with more expertise in incremental learning. As summarized in Table 2.3, after incremental learning, its performance can always be improved either large or small.

4. The impact of the number of participants is still a substantial factor with regard to the accuracy rate of the hierarchies whether built by balanced or unbalanced experimental sets.
Figure 7.14 Example of learning new photos with a balanced dataset

Figure 7.15 Example of learning each new photo with an unbalanced dataset only having specific facial expressions
Figure 7.16 Example of learning new photos with an unbalanced gender ratio
8. Conclusions

The research summary, contributions and future work are covered in this Chapter.

8.1 Research Summary

In this thesis, a new face representation and recognition methodology called “soft-cut and probabilistic distance-based classifier” was proposed, investigated and tested experimentally. The hierarchies of probabilistic decision tables [248-249], for the first time, were applied to face representation and recognition. Compared to other techniques, the hierarchies have several unique advantages.

Firstly, before a face recognition classifier can be built using other face recognition methods, a set of features having the ability to completely distinguish one face from another must be found, which is not always easy. Rather than seeking a single set of features, as described in Section 3.2, the hierarchies can work well with a group of features that simply works well on a specific face.

Secondly, each template face is usually represented by many features in other methods, which means huge amounts of calculation when matching an unknown image to each template. When the hierarchies are applied, as introduced in Section 5.1, each template image is represented by one elementary set of each decision table, which usually has a few, possibly 2 or 3, conditional attributes.

Thirdly, the classifiers, based on many techniques, must completely change their
structures to accommodate new images when available for example, in the multi-layer perceptron neural network. Instead of total re-generation, the hierarchies can update their structure according to the various strategies described in Section 6.1 and based upon which region (positive, negative, or boundary region, or none of them) the new image is classified.

In order to determine whether the presented methodology can be successfully applied to face recognition, various experiments were employed. Experimental data (facial photos) were collected from the University of Regina after the ethical clearance was received from the University of Regina Research Ethics Board. The experiments can be classified into three categories.

As described in Section 7.3, a set of $k$-fold cross-validation experiments were employed to preliminarily estimate the performance of the hierarchies directly formed by photos in the training set and the hierarchies after incremental learning.

As introduced in Section 7.4, experiments were employed to systematically evaluate the performance of the hierarchies, when directly formed by photos in the training set. The representativeness of the photos in the experimental datasets varied. The photos with all four facial expressions would be contained in some of the training sets and the gender ratio of participants was 1:1. Other photos with only specific facial expression(s) would be contained or the gender ratio was not 1:1. Finally, the number of participants in the experimental set varied.

As explained Section 7.5, experiments to systematically evaluate the performance of
the hierarchies, after incremental learning, were conducted. The primary purpose was to identify whether the hierarchies can correctly update their structures in incremental learning and following that, whether the hierarchies can correctly classify the unknown photos into different categories.

Based on the results, the performance of the presented methodology, when applied to face recognition, can be summarized as follows:

1. **When the hierarchies were directly built by photos in the training set**

   As shown in Section 7.4.1.2, Section 7.4.1.3, Section 7.4.2.2, and Section 7.4.2.3, its performance is always inversely proportional to the number of participants whether the training set is balanced or not. If the number of participants in the training set is fixed, the performance of the hierarchies can be improved or at least not deteriorate when they are trained by a set having only photos with a specific facial expression. The performance can always deteriorate if the gender ratio of the training set is unbalanced.

   Compared to the methods described in Section 2.1, it seems the performance of the presented methodology is somewhat identical to these methods (the accuracy of all methods is greater than 90%) if it is trained by the training set and the number of participants of which is relatively small (possibly 4 participants).

2. **After the hierarchies have been updated in the process of incremental learning**

   As shown in Section 7.5.2, Section 7.5.3.1, and Section 7.5.3.2, during the process of incremental learning, the hierarchies are not a classifier that is unable to learn any new data. It can satisfy the four properties proposed in [161] in some cases. That is, the photos
in the original training set do not necessarily have to be accessed to update the hierarchies when new photos surface. It can gain additional knowledge from new photos and preserve previously acquired knowledge. If new photos introduce new classes, the hierarchies can accommodate them by creating a new elementary set located in the positive or negative region. In regard to other cases, the hierarchies is a classifier falling somewhere in between the stable-plasticity spectrum (i.e., totally generating the hierarchies of decision tables alternates with updating the hierarchies of decision tables).

After incremental learning, whether they are trained by a balanced or unbalanced training set, the performance of the hierarchies can always be improved. However, the impact of the number of participants is still substantial on the accuracy rate of the hierarchies – its performance will be relatively improved if the number of participants is relatively small (for instance, only 4 participants).

### 8.2 Research Contributions

The fundamental contribution of this research is that the hierarchies of the probabilistic decision tables are, for the first time, applied to face representation and recognition in a newly proposed methodology. More specifically, the research contributions fall into several categories, as summarized below:

1. **Research how to represent each facial image with wavelet coefficients and heuristically reduce the redundancy in the coefficients.** As shown in Section 3.2 and Section 3.3, each face is represented with several groups of coefficients. If the
coefficients of each group can be from different parts or levels, it will be hard to tell, as a general rule, which level or part the coefficients are taken from. Thus, each group of coefficients in the proposed methodology is only selected from one specific sub-band (the scale, horizontal, or vertical part of levels 2, 3, or 4) to avoid the difficulty. Also, a method heuristically removing the redundancy in the coefficients has been proposed, so when processed by principal component analysis (PCA), the amount of calculation can be reduced.

2. Research how to discretize real-valued facial features using a new technique named soft-cut discretization. A feature value, close to the cut point, is possibly affected by random noise during the discretization. Traditional crisp cut methods can often assign a wrong value to such a feature. As shown in Section 4.2, the proposed technique, soft-cut discretization, can overcome these limitations. It splits the real axis of each dimension, and only those facial features classified into two specific intervals of each dimension can be discretized.

3. Research how to heuristically select the most relevant features as the conditional attributes. As described in Section 5.1, instead of trying every facial feature combination, the proposed technique in the presented methodology heuristically selects the most relevant conditional attributes for each decision table by evaluating the value of a derivative of $\lambda$–dependency (as shown in Section 5.1), so the time required to find the most relevant features can be greatly reduced.
4. **Research how to prune the hierarchies to improve their efficiency.** As described in Section 5.2, the negative regions of the decision tables can be pruned and assigned properly selected default values of output probabilities. An unknown image only needs to be matched to elementary sets in a positive region and the boundary region of each decision table so as to improve the classification speed.

5. **Research how to classify an unknown image employing a new hybrid method called the probabilistic distance-based matching method.** As described in Section 5.3, it can classify an unknown image in two steps. When matching to each hierarchy, a method that can be considered as a departure from the standard distance-based classification method is applied to overcome the limitations of the standard distance-based classification method. The unknown image is classified into an elementary set of each hierarchy having a minimum distance. Based on which region the elementary set is located, if such a minimum distance is less than a threshold value, a more appropriate probability (for example $P(X|E)$ or $P(X)$) is selected as the probability of classifying the unknown image into decision class $X$ of each hierarchy. When considering which identity the unknown image should be assigned, the values of the probabilities are compared, and the unknown image is assigned an identity having the highest probability.

6. **Research how to modify the structure of the hierarchies for the purpose of incremental learning.** As shown in Section 6.1, strategies 2 and 3 proposed in [244] can be merged in the context of Variable Precision Rough Set theory. Hence, algorithm [244] was implemented only having three strategies to reduce the amount of calculation.
8.3 Future Work

In the future, the methodology will require further improvement and testing. In particular, the following requirements will need to be fulfilled:

1. Systematically perform experiments with particular datasets having more than four participants and datasets with photos having different background colors and various illumination conditions.
2. Develop a method to improve the performance, in particular, the accuracy rate, when the number of participants is increased.
3. Find out how to improve performance when the dataset is unbalanced, in particular, an unbalanced gender ratio.
4. Find a method to identify which group of wavelet coefficients is most relevant for each decision table in the hierarchy. That is, given several identical groups of wavelet coefficients representing images, a method that can find which group is the most relevant for a particular decision table.
5. Explore how to extend the methodology to other application scenarios such as the scenario of identifying whether an unknown photo is from one of the participants in the training set or if the owner of an unknown photo is an imposter.
6. Perform experiments with real-time recognition device (i.e., camera connected to a computer trying to recognize a person while he or she is waiting).
Appendix A

Ethical Clearance
DATE: January 27, 2009

TO: Xuguang Chen  
P. O. Box 38005  
1101J Kramer Blvd., Regina, SK S4S 5W4

FROM: Dr. Bruce Plouffe  
Chair, Research Ethics Board

Re: Rough Set Approach to Face Recognition (53S0809)

Please be advised that the University of Regina Research Ethics Board has reviewed your proposal and found it to be:

☑ 1. APPROVED AS SUBMITTED. Only applicants with this designation have ethical approval to proceed with their research as described in their applications. For research lasting more than one year (Section 1F). ETHICAL APPROVAL MUST BE RENEWED BY SUBMITTING A BRIEF STATUS REPORT EVERY TWELVE MONTHS. Approval will be revoked unless a satisfactory status report is received. Any substantive changes in methodology or instrumentation must also be approved prior to their implementation.

☐ 2. ACCEPTABLE SUBJECT TO MINOR CHANGES AND PRECAUTIONS (SEE ATTACHED). Changes must be submitted to the REB and approved prior to beginning research. Please submit a supplementary memo addressing the concerns to the Chair of the REB. **Do not submit a new application**. Once changes are deemed acceptable, ethical approval will be granted.

☐ 3. ACCEPTABLE SUBJECT TO CHANGES AND PRECAUTIONS (SEE ATTACHED). Changes must be submitted to the REB and approved prior to beginning research. Please submit a supplementary memo addressing the concerns to the Chair of the REB. **Do not submit a new application**. Once changes are deemed acceptable, ethical approval will be granted.

☐ 4. UNACCEPTABLE AS SUBMITTED. The proposal requires substantial additions or redesign. Please contact the Chair of the REB for advice on how the project proposal might be revised.

Dr. Bruce Plouffe

cc. Dr. Wojciech Ziarko – Department of Computer Science

**supplementary memo should be forwarded to the Chair of the Research Ethics Board at the Office of Research Services (Research and Innovation Centre, Room 109) or by e-mail to research.ethics@uregina.ca**
Appendix B

Fundamentals of Harr Wavelet Transformation

In this Appendix, the principal concepts of the Wavelet Transform, in particular a Haar Wavelet, are introduced. Essentially, wavelets can be considered a mathematical tool to hierarchically decompose functions. The basic idea of the wavelet transform is to describe a function in terms of a coarse overall shape. The function details can range from broad to narrow, where such a function can be an image, a curve or another form of input.

B.1 One-Dimensional Haar Wavelet Transformation

If an image possesses only one row or one column of \( h \) pixels, it can be considered a one-dimensional (1D) image. The image can be represented as sequences of coefficients, and processed using the Haar Wavelet Transformation algorithm, listed in Table B.1. After applying the algorithm to the image, the first half of the image (\( h/2 \) pixels), becomes a new lower-resolution image and each of its pixel values is an average of the neighboring pixels in the image. The value of each pixel in the second half (\( h/2 \) pixels) is called a detail coefficient and is equal to the difference between the neighboring pixels of the image.

If the number of pixels in the first half of the image is greater than one, then the
operation can continue until it becomes one. After an image is processed by the algorithm in Table B.1, it will have a single coefficient representing the overall average of the original image and will be followed by a set of detailed coefficients in the order of increasing resolution. No information is gained or lost during the aforementioned operation. When needed, the image can be completely reconstructed back into its original form or reconstructed to any resolution.

One advantage with regard to representing an image in this way is that a large number of the detail coefficients turn out to be very small in magnitude [187][192]. If they are truncated or removed, only minute errors remain in the reconstructed images.

**Table B.1** Algorithm for one-dimensional Haar Wavelet Transformation

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Input:</strong> One-dimensional Haar Wavelet Transformation (I: array[1…h])</td>
</tr>
<tr>
<td>2</td>
<td>Normalize I:   [ I = I / \sqrt{h} ]</td>
</tr>
<tr>
<td>3</td>
<td>While (h&gt;1)</td>
</tr>
<tr>
<td></td>
<td>for( i=1; i&lt;h/2; i++)</td>
</tr>
<tr>
<td></td>
<td>[ I'[i] = (I[2i - 1] + I[2i]) / \sqrt{2} ]</td>
</tr>
<tr>
<td></td>
<td>[ I'[h / 2 + i] = (I[2i - 1] - I[2i]) / \sqrt{2} ]</td>
</tr>
<tr>
<td></td>
<td>[ I = I' ]</td>
</tr>
<tr>
<td></td>
<td><strong>End for</strong></td>
</tr>
<tr>
<td></td>
<td>h=h/2</td>
</tr>
<tr>
<td></td>
<td><strong>}</strong></td>
</tr>
</tbody>
</table>

While a 1D image can be represented by a sequence of coefficients, it can also be thought of as piecewise-constant function on a half-open interval \([0,1)\). If a 1D image has \(2^j\) pixels, it can be thought of as a function having \(2^j\) different pieces in a vector space denoted as \(V^j\). It can be considered a \(2^j\) vector in a \(V^j\) space. A piecewise-constant function with two intervals can always be described as a piecewise-constant function with
four intervals and each vector in $V^{j-1}$ is also contained in $V^{j}$. Similarly, each vector in $V^{j-2}$ is also contained in $V^{j-1}$, as in

$$V^{0} \subset V^{1} \subset \ldots \subset V^{j} \subset \ldots$$

The basis functions for space $V^{j}$ are called scaling functions and defined as in (B.1):

$$\phi^{j}(x) = 2^{j/2} \phi(2^{j} x - i), \quad (B.1)$$

where $i = 0, ..., 2^{j} - 1$ and $\phi(x)$ can be defined as:

$$\phi(x) = \begin{cases} 
1, & 0 \leq x < 1, \\
0, & \text{otherwise}.
\end{cases}$$

A vector space $V^{j+1}$ can contain another space $V^{j}$. A new vector space $W^{j}$ can be defined, which is considered the orthogonal complement of $V^{j}$ in $V^{j+1}$. The basis function for space $W^{j}$ is called a wavelet function and is defined in (B.2):

$$\psi^{j}(x) = 2^{j/2} \psi(2^{j} x - i), \quad (B.2)$$

where $i = 0, ..., 2^{j}$ and $\psi(x)$ can be defined as:

$$\psi(x) = \begin{cases} 
1, & 0 \leq x < 1/2, \\
-1, & 1/2 \leq x < 1, \\
0, & \text{otherwise}.
\end{cases}$$

The basis function of the space $V^{j+1}$ is a summation of the basis function $\phi^{j}$ of the space $V^{j}$ and the basis function $\psi^{j}$ of the space $W^{j}$, denoted as $V^{j+1} = V^{j} + W^{j}$. Each basis function $\psi^{j}$ of the space $W^{j}$ is orthogonal to each basis function of $\phi^{j}$ of the space $V^{j}$ [187]. According to the algorithm in Table B.1, the detail coefficients are actually the coefficients of the wavelet basis functions.
B.2 Two-dimensional Haar Wavelet Transformation

Usually, the images are two-dimensional (2D) rather than one-dimensional (1D). Given a 2D image, two processing methods are available called standard decomposition [8] and non-standard decomposition [187], as listed in Tables B.2 and B.3.

If standard decomposition is used, a 1D wavelet transform will be applied to each row of pixel values of a given image denoted as $I$. The first half of each row is the average value of the neighboring pixels, and the second half of each row contains the detail coefficients. This transformed image is denoted as $I'$. Then, a 1D transform is applied to the first half of each row of $I'$. This process is repeated until all the pixels, other than the first pixel of each row, become detail coefficients. This image is denoted as $I''$. Following this step, a 1D wavelet transform is applied to each column containing the pixel values of $I''$. The same steps are applied to each row. Figure B.1 is a sample image from [187] which describes standard decomposition.

**Table B.2** Standard decomposition

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>Input: Two-dimensional Haar Wavelet Transformation (II: array[1...h, 1...w])</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2</td>
<td>Define a 1-D array $I$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>for(row=1; row&lt;h; row++) {</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>$I = II[row, 1...w]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>applying the algorithm in table B.1 to $I$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>}</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>for(col=1; col&lt;w; col++) {</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$I = II[1...h, col]$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>applying the algorithm in table B.1 to $I$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>}</td>
</tr>
</tbody>
</table>
Alternately, if non-standard decomposition is used, a 1D wavelet transform is applied to each row and column of pixel values. A 1D wavelet transform is applied to each row of pixel values of a given image, denoted as \( II \). The first half of each row contains the average values of the neighboring pixels, and the second half of each row contains the detail coefficients. The transformed image is denoted as \( II' \). Then, a 1D transform is applied to each column of \( II' \). The steps are repeated on the quadrant containing average values in both directions until only a single overall average coefficient is left. Figure B.2 presents a sample image to describe non-standard decomposition.
**Table B.3** Non-standard decomposition

1. **Input**: Two-dimensional Haar Wavelet Transformation \( II: \text{ array}[1...h, 1...h] \)

2. **Normalize** \( II: \ I = II / \sqrt{h} \)

3. **Define a 1-D array** \( I \)

4. **While** \( h > 1 \)
   - **for** \( row=1; row<h; row++ \)
     - \( I = II[row,1...h] \)
     - **for** \( i=1; i<h/2; i++ \)
       - \( I'[i] = (I[2i-1]+I[2i]) / \sqrt{2} \)
       - \( I'[h/2+i] = (I[2i-1]-I[2i]) / \sqrt{2} \)
       - \( I = I' \)
   - **End for**
   - **End for**

   **for** \( col=1; col<w; col++ \)
   - \( I=II[1...h, col] \)
   - **for** \( i=1; i<h/2; i++ \)
     - \( I'[i] = (I[2i-1]+I[2i]) / \sqrt{2} \)
     - \( I'[h/2+i] = (I[2i-1]-I[2i]) / \sqrt{2} \)
     - \( I = I' \)
   - **End for**
   - **End for**

   \( h=h/2 \)

**End while**

Basis functions for the two decomposition methods can be divided into a two-dimensional scaling function and three wavelet functions, denoted as (B.3):
scaling function

$$\Phi \Phi(x, y) = \Phi(x) \Phi(y), \quad (B.3)$$

wavelet functions

$$\Phi \Psi(x, y) = \Phi(x) \Psi(y),$$
$$\Psi \Phi(x, y) = \Psi(x) \Phi(y),$$
$$\Psi \Psi(x, y) = \Psi(x) \Psi(y).$$

Figure B.2 Decomposing an image using Non-standard Decomposition
Appendix C

Overview of Relevant Statistical Concepts

In this Appendix, the statistical concepts as required by the presented methodology are briefly described.

1. Standard Deviation

Given a set of data $X_{data}$ having $n$ elements denoted as $X_{data} = \{X_1, \ldots, X_n\}$, the standard deviation can be used to measure the spread of the data. The mean of the set $X_{data}$, defined as $\bar{X}$, can be represented by formula (C.1):

$$\bar{X} = \frac{\sum_{i=1}^{n} X_i}{n}.$$  \hspace{1cm} (C.1)

**Definition C.1** [184] The standard deviation of a given dataset can be defined as the average distance from the mean of the dataset to the points in the set. Its formula is given in (C.2):

$$S = \sqrt{\frac{\sum_{i=1}^{n} (X_i - \bar{X})^2}{n-1}}.$$  \hspace{1cm} (C.2)

2. Variance

If the standard deviation is squared, it is defined as the variance [184], as in (C.3):

$$S^2 = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(X_i - \bar{X})}{n-1}. \hspace{1cm} (C.3)$$

The variance and standard deviation can be used to measure the spread of the data. The
variance is not used as often, compared to standard deviation.

3. Covariance

Standard deviation and variance are used to measure 1-dimensional data. If a dataset has more than one dimension, as in a set of images, the covariance will become useful, notably in a case where any relationship among the dimensions is to be identified. The covariance is measured between 2 dimensions and will become the variance when calculating covariance between one-dimensional data and itself. The formula for covariance is very similar to that of variance, defined in (C.4):

\[ \text{cov}(X, Y) = \frac{\sum_{i=1}^{n} (X_i - \bar{X})(Y_i - \bar{Y})}{n} \quad (C.4) \]

All the possible covariance values between in the different dimensions in a dataset are calculated and placed in a matrix. If the dataset has a dimension \(n\), represented as \((d_1, d_2, \ldots, d_n)\), the size of the covariance matrix will be \(n \times n\), as defined in (C.5):

\[ C^{n \times n} = \begin{bmatrix} 
\text{cov}(d_1, d_1) & \ldots & \text{cov}(d_1, d_n) \\
\vdots & \ddots & \vdots \\
\text{cov}(d_n, d_1) & \ldots & \text{cov}(d_n, d_n) 
\end{bmatrix} \quad (C.5) \]

For example, if a dataset is 3-dimensional \((x, y, \text{and } z)\), its covariance is measured between \((x, y)\), \((x, z)\), and \((y, z)\), as listed below:

\[ C^{3 \times 3} = \begin{bmatrix} 
\text{cov}(x, x) & \text{cov}(x, y) & \text{cov}(x, z) \\
\text{cov}(y, x) & \text{cov}(y, y) & \text{cov}(y, z) \\
\text{cov}(z, x) & \text{cov}(z, y) & \text{cov}(z, z) 
\end{bmatrix} \]

Two important properties can be observed in the covariance matrix \(C^{3 \times 3}\) [184]. Firstly, they are based on one of the dimensions and their own values, so the covariance
values along the main diagonal (from top left to bottom right) are variance values. Secondly, because $\text{cov}(d_i, d_j) = \text{cov}(d_j, d_i)$, the covariance matrix is symmetrical.

4. Eigenvectors and Eigenvalues

Given a linear transformation, an eigenvector $X_{eigen}$ obtained from a linear transformation is a nonzero vector. Its length can be changed and its direction can be retained when the transformation is applied. For each eigenvector obtained from a linear transformation, there is a corresponding scalar value called an eigenvalue $\lambda$ for that vector. In linear algebra, every linear transformation between finite-dimensional vector spaces is expressed as a matrix $A$. The eigenvectors and eigenvalues of the matrix $A$ is defined as:

**Definition C.2** [45] If a vector $X_{eigen} \neq 0$ satisfies the equation

$$AX_{eigen} = \lambda X_{eigen},$$

for some real or complex number $\lambda$, then $\lambda$ is called an eigenvalue of the matrix $A$, and $X_{eigen}$ is called an eigenvector corresponding to the eigenvalue $\lambda$.

By the definition, an eigenvector must be a nonzero vector, but eigenvalue could be zero. In addition, an eigenvector only has a unique eigenvalue associated to it, but an eigenvalue can have more than one eigenvector associated to it [169]. A linear transformation between finite-dimensional vector spaces can be expressed as a matrix operation but this does not mean every matrix must have eigenvectors. Only a few square matrices can have eigenvectors [184]. In PCA, eigenvectors are used to construct a matrix, whereby the size of a data set can be reduced, but the most important features to represent each datum can be extracted.
Appendix D

Overview of Related Rough Set Models

This Appendix briefly reviews three probabilistic rough set models including the 0.5 probabilistic approximations, the decision-theoretic rough set model and Bayesian rough set model. The notions of rough membership function and a rough inclusion in the aforementioned methods can be interpreted in terms of conditional probabilities or posteriori probabilities [231]. The threshold values are then applied to a rough membership function or a rough inclusion to obtain the probabilistic approximation.

D.1 the 0.5 Probabilistic Approximations

The 0.5 probabilistic approximations were suggested by Wong and Ziarko [150][212]. The basis of such a model is the majority rule. The value 0.5 was selected as a threshold value. An element $x$ of a finite non-empty set $U$ (called the universe), defined as $x \in U$, can be classified into a relevant region (lower approximation, upper approximation, or boundary region) based on the threshold value. If the conditional probability of finding the equivalence class (defined as $E$) containing the element $x$ in a set $X \subseteq U$ defined as $P(X \mid E)$ is greater than 0.5, the element $x$ is inside the lower approximation. If it is greater than or equal to 0.5, the element $x$ is inside the upper approximation. Element $x$ is in the boundary region if the conditional probability equals to 0.5.

The 0.5 probabilistic approximations consider the probabilities of three special points,
two extreme points (0 and 1) and one middle point (0.5). In practice, this does not always work well because point 0.5 is not always a good middle point. By considering other values, two additional general probabilistic approximation models, the decision-theoretic model [221][224][231-232], and the variable precision rough set model [88][243][252-253], were introduced.

D.2 Decision-Theoretic Rough Set Model

The two models are formulated to generalize the 0.5 probabilistic, but they formulate the probability approximations and interpret the threshold values in different ways. In the context of a variable precision rough set model, the required threshold values are treated as a primitive notion. The interpretation and the process of determining the threshold values are primarily based on empirical studies [221][231]. That is, their values will be determined heuristically.

The decision-theoretic rough set model is based on the Bayesian decision theory [221][224][231-232] that mainly deals with how to make a decision with the minimum risk or cost under probabilistic uncertainty. According to the Bayesian decision theory, a set of loss functions is defined as \( \lambda(a_i \mid w_j) \), in which \( w_j \) is an element of a finite set of \( s \) states (defined as \( w_j \in \Omega = \{w_1, \ldots, w_s\} \)), and \( a_i \) is an element of a finite set of \( m \) possible actions (defined as \( a_i \in A = \{a_1, \ldots, a_m\} \)). A loss function \( \lambda(a_i \mid w_j) \) will predict the loss or cost of taking action \( a_i \) in state \( w_j \). An element \( x \) with some description \( x' \) takes the action \( a_i \), and has the conditional probability (defined as \( P(w_j \mid x') \)) that the true state is
when given the description \( x' \). The expected loss associated with taking action \( a_i \) can be calculated by [221]:

\[
R(a_i \mid x') = \sum_{j=1}^{s} \lambda(a_i \mid w_j)P(w_j \mid x'). \tag{D.1}
\]

When a description \( x' \) is given, a decision rule can be defined as a function \( \tau(x') \) specifying which action \( a_i \) should be taken. The overall risk \( R' \) is the expected loss associated with a given decision rule and can be defined as [221]:

\[
R' = \sum_{x} R(\tau(x') \mid x')P(x'), \tag{D.2}
\]

where \( R(\tau(x') \mid x') \) is the conditional risk associated with \( \tau(x') \) and the summation is over the set of all possible descriptions of objects. For every description \( x' \), if a function \( \tau(x') \) can be selected so that \( R(\tau(x') \mid x') \) can be as small as possible, the overall risk \( R' \) will be minimized. Thus, the Bayesian decision procedure can be formally described as follows. For every description \( x' \) and \( i = 1...m \), the conditional risk \( R(a_i \mid x') \) is calculated according to formula (D.1). For an action, if the conditional risk can be minimized, the action will be selected. If there is more than one action enabling to minimize the conditional risk \( R(a_i \mid x') \), a tie-breaking criterion will be used.

When classifying an element \( x \) into a positive, negative or boundary region, the set of states can be \( \Omega = \{X, \neg X\} \) which means the element \( x \) is in set \( X \) or the complementary set (defined as \( \neg X \) ) of \( X \). The set of actions can be \( A = \{a_1, a_2, a_3\} \) which means the action of deciding to classify the element \( x \) into a positive, negative or boundary region. The conditional probabilities \( P(X \mid E) \) and \( P(\neg X \mid E) \) are the probabilities that an element
In the equivalence class \( E \) belongs to the set \( X \) and the complementary set of \( X \), respectively. The expected loss associated with classifying \( x \) into a positive region can be given by:

\[
R(a_1 | E) = \lambda(a_1 | X)P(X | E) + \lambda(a_1 | \neg X)P(\neg X | E).
\]

Similarly, the expected loss associated with classifying \( x \) into the negative region can be given by:

\[
R(a_2 | E) = \lambda(a_2 | X)P(X | E) + \lambda(a_2 | \neg X)P(\neg X | E).
\]

The expected loss associated with classifying \( x \) into the boundary region can be given by:

\[
R(a_3 | E) = \lambda(a_3 | X)P(X | E) + \lambda(a_3 | \neg X)P(\neg X | E).
\]

Hence, in the decision-theoretic rough set model, the expected loss associated with classifying an element \( x \) into a positive, negative or boundary region can be expressed as:

\[
R(a_i | E) = \lambda(a_i | X)P(X | E) + \lambda(a_i | \neg X)P(\neg X | E),
\]

where \( \lambda_{i1} = \lambda(a_i | X) \), \( \lambda_{i2} = \lambda(a_i | \neg X) \), and \( i=1, 2 \) or \( 3 \). Based on the value of the expected loss \( R(a_i | E) \) in (D.3), the minimum-risk decision rules can be obtained as:

\( (P) \) \quad If \( R(a_1 | E) \leq R(a_2 | E) \) and \( R(a_1 | E) \leq R(a_3 | E) \), \quad decide \ POS(X);  
\( (N) \) \quad If \( R(a_2 | E) \leq R(a_1 | E) \) and \( R(a_2 | E) \leq R(a_3 | E) \), \quad decide \ NEG(X);  
\( (B) \) \quad If \( R(a_3 | E) \leq R(a_1 | E) \) and \( R(a_3 | E) \leq R(a_2 | E) \), \quad decide \ BND(X). 

In addition, tie-breaking rules should be added to ensure that each element is only classified into one region. Finally, the rules described above can be further simplified because \( P(X | [x]) + P(\neg X | [x]) = 1 \), so only the conditional probability \( P(X | E) \) and the
given loss functions are required when classifying any element $x$ in $E$.

If it is assumed the loss of classifying an element $x$ belonging to the set $X$ into the positive region is less than or equal to the loss of classifying $x$ into the boundary region, and both of these losses must be less than the loss of classifying $x$ into the negative region, the lose functions $\lambda_{11}$, $\lambda_{31}$ and $\lambda_{21}$ can have $\lambda_{11} \leq \lambda_{31} < \lambda_{21}$. Similarly, if the reverse order of losses is used for classifying an element $x$ not belonging to the set $X$, the lose functions $\lambda_{22}$, $\lambda_{32}$ and $\lambda_{12}$ can have $\lambda_{22} \leq \lambda_{32} < \lambda_{12}$. This being the case, the minimum-risk decision rules in (D.3) can be rewritten as:

$$\begin{align*}
(P) & \quad \text{If } P(X \mid E) \geq \gamma \text{ and } P(X \mid E) \geq \alpha, \quad \text{decide } POS_{\alpha,\beta}(X); \\
(N) & \quad \text{If } P(X \mid E) \leq \beta \text{ and } P(X \mid E) \leq \gamma, \quad \text{decide } NEG_{\alpha,\beta}(X); \\
(B) & \quad \text{If } \beta \leq P(X \mid E) \leq \alpha, \quad \text{decide } BND_{\alpha,\beta}(X); 
\end{align*}$$

where

$$\begin{align*}
\alpha &= \frac{\lambda_{12} - \lambda_{32}}{\lambda_{21} - \lambda_{12}}, \\
\gamma &= \frac{\lambda_{12} - \lambda_{22}}{\lambda_{21} - \lambda_{12}}, \quad \text{and} \\
\beta &= \frac{\lambda_{32} - \lambda_{22}}{\lambda_{21} - \lambda_{32}}.
\end{align*}$$

In the decision-theoretic rough set model, $POS_{\alpha,\beta}(X)$, $NEG_{\alpha,\beta}(X)$ and $BND_{\alpha,\beta}(X)$ are denoted as the positive region, negative region, and boundary region. In order to ensure that the boundary region is not empty, the parameters $\alpha$ and $\beta$ should be $\alpha > \beta$. Hence, after tie-breaking, the decision rules in the decision-theoretic rough set model can be expressed as:
That is, for example, an element $x$ belonging to set $E$ is classified in the positive region if the conditional probability $P(X \mid E)$ is greater than $\alpha$.

### D.3 Bayesian Rough Set Model

The Bayesian rough set model [58][180-182][245] focuses on the connections with Bayesian rules, in which the set approximation can be defined using the prior probability as a reference. Thus, given a non-empty finite set $U$ of objects called the universe, its positive region, negative region, and boundary region can be defined as:

**Definition D.1** [181][245]

\[
\begin{align*}
POS^* (X) &= \bigcup \{ E : P(X \mid E) > P(X) \}, \\
NEG^* (X) &= \bigcup \{ E : P(X \mid E) < P(X) \}, \\
BND^* (X) &= \bigcup \{ E : P(X \mid E) = P(X) \}.
\end{align*}
\]

In Bayesian rough set model, the positive region $POS^*(X)$ defines an area of the universe where the probability of the target set $X$ is higher than the prior probability. In continuation, the negative region $NEG^*(X)$ defines an area of the universe where the probability of the target set $X$ is lower than the prior probability. The boundary region $BND^*(X)$ defines an area characterized by the lack of certainty improvement with respect to predicting neither $X$ or $\neg X$. 

\[(P) \quad \text{If } P(X \mid E) \geq \alpha, \quad \text{decide } POS_{\alpha, \beta}^* (X); \]
\[(N) \quad \text{If } P(X \mid E) \leq \beta, \quad \text{decide } NEG_{\alpha, \beta}^* (X); \]
\[(B) \quad \text{If } \beta < P(X \mid E) < \alpha, \quad \text{decide } BND_{\alpha, \beta}^* (X). \quad (D.4)\]
Bayesian reasoning can combine prior knowledge with data-driven inverse probabilities to obtain the posterior probabilities [181]. They can be applied to evaluate the obtained information by comparing the prior probabilities with the posterior probabilities. The prior probability can be defined as [181]:

\[
P(X \mid E) = \frac{P(X)P(E \mid X)}{P(E)}.
\] (D.6)

If the prior probability of the target set \(X\) and the prior probability of the complementary set \(\neg X\) are combined, the formula [181] below will be obtained:

\[
\frac{P(X \mid E)}{P(\neg X \mid E)} = \frac{P(X)}{P(\neg X)} \cdot \frac{P(E \mid X)}{P(E \mid \neg X)}.
\] (D.7)

That is, instead of \(P(X \mid E)\) and \(P(X)\), the set approximations can be based on the inverse probabilities as formalized below:

**Theorem D.1** [181] In Bayesian rough model, the positive region \(POS^*(X)\), negative region \(NEG^*(X)\), and boundary region \(BND^*(X)\) can also be expressed as:

\[
\begin{align*}
POS^*(X) &= \bigcup \{ E : P(E \mid X) > P(E \mid \neg X) \}, \\
NEG^*(X) &= \bigcup \{ E : P(E \mid X) < P(E \mid \neg X) \}, \\
BND^*(X) &= \bigcup \{ E : P(E \mid X) = P(E \mid \neg X) \}.
\end{align*}
\] (D.8)

**Proof:** Selecting \(POS^*(X)\) as an example to prove. The others are analogous. Assuming \(P(X \mid E) > P(E)\), it implies \(P(\neg X \mid E) = 1 - P(X \mid E) < 1 - P(X) = P(\neg X)\). So, it can get

\[
(P(X \mid E) / P(X)) / (P(\neg X \mid E) / P(\neg X)) > 1,
\]

i.e., based on (D.7), \(P(E \mid X) / P(E \mid \neg X) > 1\).

By contradiction, assuming \(P(X \mid E) \leq P(X)\), it implies \(P(E \mid X) / P(E \mid \neg X) \leq 1\) in the way as above. □
Appendix E

Concept of Reducts

This Appendix briefly reviews the concept of reducts. Intuitively speaking, a reduct or the minimal set of attributes is the minimal subset of condition attributes \( C \) that can preserve the dependency with decision attributes \( D \). Research on reduct is one of the fundamental works in rough set theory. Many methods have been proposed to find all of reducts or a single reduct such as [96][180][227][229][243]. Because it has been well known that finding all of reducts or a reduct with the minimum number of attributes is NP-hard [212], many methods that can find reducts heuristically have been proposed, for instance [45][78][120][169][241].

In the original rough set model, the reducts can be defined as follows [147-148] [150][226]:

**Definition E.1** Let \( K = (U, A) \) be a knowledge representation system and let \( C, D \subseteq A \) be two subsets of attributes, called condition and decision attributes respectively. A subset \( C_{\text{sub}} \subseteq C \) is a reudct of \( C \) with respect to \( D \) if it satisfies the following two conditions:

1. \( POS_{C_{\text{sub}}}(D) = POS_C(D) \),
2. No attribute can be eliminated from \( C_{\text{sub}} \) without affecting the condition (1).

In Definition E.1, \( POS_C(D) \) is called \( C \)-positive region of \( D \) and is the set of all elements on the universe \( U \) that can be properly classified into the classes of \( U/D \) employing the knowledge that is expressed by the classification \( U/C \). \( U/C \) is the family of
all equivalence classes of the indiscernibility relation induced by the condition attributes $C$ and can also be denoted as $IND(C)$. Similarly, $U/D$ is the family of all equivalence classes of the indiscernibility relation induced by the decision attributes $D$ and can also be denoted as $IND(D)$. $C$-positive region of $D$ can be defined as:

$$POS_C(D) = \bigcup_{X \in U/D} \{ x \in U : P(X \mid E) = 1 \},$$  \hspace{1cm} (E.1)

where $E \in U / C$ and contains the element $x$ (i.e., $x \in E$).

Other than the defection in the original rough set model, over the last thirty years, many kinds of definitions of reducts in the literature of rough set theory have been proposed. For instance, in VPRS model, the idea of reducts is generalized as approximate reducts or $\beta$-reducts as shown in Definition E.2 [245]:

**Definition E.2** Let $K = (U, A)$ be a knowledge representation system and let $C, D \subseteq A$ be two subsets of attributes, called condition and decision attributes respectively. A $\beta$-reduct of the set of condition attributes $C$ with respect to a set of decision attributes $D$ is a subset $C_{\beta} \subseteq C$ which satisfies the following two conditions:

1. $\gamma(C, D, \beta) = \gamma(C_{\beta}, D, \beta),$

2. No attribute can be eliminated from $C_{\beta}$ without affecting the condition (1).

In Definition E.2, $\beta$ is a parameter determining the admissible classification error, and should be $0 \leq \beta < 0.5$. $\gamma(C, D, \beta)$ is called $\beta$-dependency, and can be defined as:

$$\gamma(C, D, \beta) = \frac{\text{card}(\cup_{X \in U/D} \{ E \in U / C : c(E, X) \leq \beta \})}{\text{card}(U)},$$  \hspace{1cm} (E.2)

where $\text{card}$ means cardinality, and $c(E, X)$ is called the relative degree of
misclassification of the set $E$ belonging to $U/C$ with respect to set $X$ belonging to $U/D$.

c\left(E, X\right)$ can be defined as:

$$
c\left(E, X\right) = \begin{cases} 
1 - \frac{\text{card}(E \cap X)}{\text{card}(E)}, & \text{if } \text{card}(E) > 0, \\
0, & \text{if } \text{card}(E) = 0.
\end{cases} \quad (E.3)
$$

In the decision-theoretic rough set model, given a set $X \in U/D$, a reduct can be defined by requiring that the positive region of $POS_{\alpha, \beta}(X)$ is not changed. Its definition is shown in Definition E.3 [226]:

**Definition E.3** Let $K = (U, A)$ be a knowledge representation system and let $C, D \subseteq A$ be two subsets of attributes, called condition and decision attributes respectively. A subset $C_{\text{sub}} \subseteq C$ is a reduct of $C$ with respect to $D$ if it satisfies the following two conditions:

1. Jointly sufficient condition:

   $$
   POS^{C_{\text{sub}}}_{\alpha, \beta}(X) = POS^{C}_{\alpha, \beta}(X),
   $$

2. Individually necessary conditions:

   for any attribute $c \in C_{\text{sub}}$, $POS^{C_{\text{sub}} - \{c\}}_{\alpha, \beta}(X) \neq POS^{C}_{\alpha, \beta}(X)$.

In Bayesian rough set model, reducts are defined according to the notion of the global relative gain function denoted as $R(X)$ and are called $R$-reducts [180-181].

$R(X)$ is defined as:

$$
R(X) = \bigcup_{X \in U/D} P(E)r(X \mid E), \quad (E.4)
$$

where $E$ belongs to $U/C$, $r(X \mid E) = \max(P(X \mid E) / P(X) - 1, P(\neg X \mid E) / P(\neg X) - 1)$, and $\neg X$ is the complementary set of $X$. Based on the notion of $R(X)$, $R$-reducts can be defined as follows [180-181]:
**Definition E.4** Let $K = (U, A)$ be a knowledge representation system and let $C, D \subseteq A$ be two subsets of attributes, called condition and decision attributes respectively. A $R$-reduct of the set of condition attributes $C$ with respect to a set of decision attributes $D$ is a subset $C_{\text{sub}} \subseteq C$:

1. If and only if it can satisfy the equality $R_{C_{\text{sub}}}(X) = R_{C}(X)$, that is, it can preserve the value of the global gain function;
2. None of subsets of $C_{\text{sub}}$ can do it.
Appendix F

Example of Incremental Learning with Hierarchies

In this Appendix, the working process of incremental learning is explained further with an example in which 4 hierarchies are built based on an initial training set having images from 4 participants as in Table F.1 to Table F.4. Each hierarchy uses all the images of one specific participant in the initial training set as the elements in the target set $X$. Once an initial hierarchy is completed, it will be modified as described in Section 6.1.

It is assumed a new image is available and its decision attribute is assigned a value of 1. When it is matched to the first decision table in the first hierarchy, based on formula (5.3), it is added to the first elementary set, $E_0$. According to the first strategy in Section 6.1, $E_0$ is an elementary set in $BND$, so the cardinality of that elementary set is adjusted from $\text{card}(E_0)=26$ into $\text{card}(E_0)=27$. Similarly, when it is matched to the second decision table in the first hierarchy, it is added to the second elementary set, $E_1$, which is in $POS$. The value of decision attributes of images classified into $E_1$ is 1, and that of the new image is 1. Thus, according to the second strategy in Section 6.1, the cardinality of that elementary set is adjusted from $\text{card}(E_1)=19$ into $\text{card}(E_1)=20$ and $\lambda-$dependency of the hierarchy denoted as $\lambda_{\text{new}}$ can be re-computed and the value of its $\lambda-$dependency is represented by the second point in Figure F.1.
Table F.1 The 1st hierarchy

### Decision table 1

| $E_i$ | $A_{23}$ | $A_{22}$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|----------|----------|----------|------------|--------|
| $E_0$ | 1        | 1        | 0.2708   | 0.8846     | BND    |
| $E_1$ | 1        | 0        | 0.2291   | 0.0454     | BND    |
| $E_2$ | 0        | 1        | 0.2812   | 0          | NEG    |
| $E_3$ | 0        | 0        | 0.2187   | 0          | NEG    |

### Decision table 2

| $E_i$ | $A_{15}$ | $A_{14}$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|----------|----------|----------|------------|--------|
| $E_0$ | 1        | 1        | 0.1458   | 0.5714     | BND    |
| $E_1$ | 1        | 0        | 0.3958   | 1          | POS    |
| $E_2$ | 0        | 1        | 0.2291   | 0          | NEG    |
| $E_3$ | 0        | 0        | 0.2291   | 0.0909     | BND    |

### Decision table 3

| $E_i$ | $A_{9}$ | $A_{7}$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|---------|---------|----------|------------|--------|
| $E_0$ | 1       | 1       | 0.2777   | 0          | NEG    |
| $E_1$ | 1       | 0       | 0.2222   | 0          | NEG    |
| $E_2$ | 0       | 1       | 0.3333   | 0.3333     | BND    |
| $E_3$ | 0       | 0       | 0.1666   | 1          | POS    |

### Decision table 4

| $E_i$ | $A_{9}$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|---------|----------|------------|--------|
| $E_0$ | 1       | 0.6666   | 0          | NEG    |
| $E_1$ | 0       | 0.3333   | 1          | POS    |
Table F.2 The 2\textsuperscript{nd} hierarchy

### Decision table 1

| $E_i$ | $A_{23}$ | $A_{22}$ | $A_{21}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|----------|----------|----------|----------|-------------|--------|
| $E_0$ | 1        | 1        | 1        | 0.1145   | 0           | NEG    |
| $E_1$ | 1        | 1        | 0        | 0.1562   | 0           | NEG    |
| $E_2$ | 1        | 0        | 1        | 0.1354   | 0           | NEG    |
| $E_3$ | 1        | 0        | 0        | 0.0937   | 0           | NEG    |
| $E_4$ | 0        | 1        | 1        | 0.125    | 1           | POS    |
| $E_5$ | 0        | 1        | 0        | 0.1562   | 0.1333      | BND    |
| $E_6$ | 0        | 0        | 1        | 0.1458   | 0.6428      | BND    |
| $E_7$ | 0        | 0        | 0        | 0.0729   | 0.1428      | BND    |

### Decision table 2

| $E_i$ | $A_{15}$ | $A_{13}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|----------|----------|----------|-------------|--------|
| $E_0$ | 1        | 1        | 0.2777   | 1           | POS    |
| $E_1$ | 1        | 0        | 0.2222   | 0.25        | BND    |
| $E_2$ | 0        | 1        | 0.3055   | 0           | NEG    |
| $E_3$ | 0        | 0        | 0.1944   | 0           | NEG    |

### Decision table 3

| $E_i$ | $A_9$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|-------|----------|-------------|--------|
| $E_0$ | 1     | 0.75     | 0           | NEG    |
| $E_1$ | 0     | 0.25     | 1           | POS    |
Table F.3 The 3rd hierarchy

### Decision table 1

| $E_i$ | $A_{23}$ | $A_{22}$ | $A_{19}$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|----------|----------|----------|-----------|------------|--------|
| $E_0$ | 1        | 1        | 1        | 0.1666    | 0          | NEG    |
| $E_1$ | 1        | 1        | 0        | 0.1041    | 0          | NEG    |
| $E_2$ | 1        | 0        | 1        | 0.125     | 0          | NEG    |
| $E_3$ | 1        | 0        | 0        | 0.1041    | 0          | NEG    |
| $E_4$ | 0        | 1        | 1        | 0.1458    | 0.0714     | BND    |
| $E_5$ | 0        | 1        | 0        | 0.1354    | 0.923      | BND    |
| $E_6$ | 0        | 0        | 1        | 0.1145    | 0.0909     | BND    |
| $E_7$ | 0        | 0        | 0        | 0.1041    | 1          | POS    |

### Decision table 2

| $E_i$ | $A_{15}$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|----------|----------|------------|--------|
| $E_0$ | 1        | 0.5789   | 0          | NEG    |
| $E_1$ | 0        | 0.421    | 0.875      | BND    |

### Decision table 3

| $E_i$ | $A_7$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|-------|----------|------------|--------|
| $E_0$ | 1     | 0.625    | 1          | POS    |
| $E_1$ | 0     | 0.375    | 0.6666     | BND    |

### Decision table 4

| $E_i$ | $A_9$ | $P(E_i)$ | $P(X|E_i)$ | Region |
|-------|-------|----------|------------|--------|
| $E_0$ | 1     | 0.3333   | 0          | NEG    |
| $E_1$ | 0     | 0.6666   | 1          | POS    |
### Table F.4 The 4th hierarchy

<table>
<thead>
<tr>
<th>Decision table 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;23&lt;/sub&gt;</td>
<td>A&lt;sub&gt;22&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>1</td>
<td>0.2708</td>
<td>0.1153</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>1</td>
<td>0</td>
<td>0.2291</td>
<td>0.9545</td>
</tr>
<tr>
<td>E&lt;sub&gt;2&lt;/sub&gt;</td>
<td>0</td>
<td>1</td>
<td>0.2812</td>
<td>0</td>
</tr>
<tr>
<td>E&lt;sub&gt;3&lt;/sub&gt;</td>
<td>0</td>
<td>0</td>
<td>0.2187</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 2</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;15&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
<td>E&lt;sub&gt;i&lt;/sub&gt;)</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>0.5416</td>
<td>0.1153</td>
<td>BND</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0</td>
<td>0.4583</td>
<td>0.9545</td>
<td>BND</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 3</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;15&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
<td>E&lt;sub&gt;i&lt;/sub&gt;)</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>0.5</td>
<td>0.0416</td>
<td>BND</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0</td>
<td>0.5</td>
<td>0.9583</td>
<td>BND</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 4</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;15&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
<td>E&lt;sub&gt;i&lt;/sub&gt;)</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>0.4791</td>
<td>0.9565</td>
<td>BND</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0</td>
<td>0.5208</td>
<td>0.0799</td>
<td>BND</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 5</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;15&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
<td>E&lt;sub&gt;i&lt;/sub&gt;)</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>0.5416</td>
<td>0.1538</td>
<td>BND</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0</td>
<td>0.4583</td>
<td>0.909</td>
<td>BND</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 6</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;15&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
<td>E&lt;sub&gt;i&lt;/sub&gt;)</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>0.4583</td>
<td>0.909</td>
<td>BND</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0</td>
<td>0.5416</td>
<td>0.1538</td>
<td>BND</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 7</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>E&lt;sub&gt;i&lt;/sub&gt;</td>
<td>A&lt;sub&gt;15&lt;/sub&gt;</td>
<td>P(E&lt;sub&gt;i&lt;/sub&gt;)</td>
<td>P(X</td>
<td>E&lt;sub&gt;i&lt;/sub&gt;)</td>
</tr>
<tr>
<td>E&lt;sub&gt;0&lt;/sub&gt;</td>
<td>1</td>
<td>0.5416</td>
<td>0.1153</td>
<td>BND</td>
</tr>
<tr>
<td>E&lt;sub&gt;1&lt;/sub&gt;</td>
<td>0</td>
<td>0.4583</td>
<td>0.9545</td>
<td>BND</td>
</tr>
</tbody>
</table>
After the modification, if the value of $\lambda_{new}$ is much lower than that of the $\lambda - dependency$ before modification that is denoted as $\lambda_{old}$ and represented by the first point in Figure F.1, then the structure of the modified hierarchy is no longer perfect. If this is the case, the hierarchy must be completely re-constructed. In practice, a threshold value of 0.975 is defined for the $\lambda - dependency$ of the hierarchy. If formula (F.1) cannot be satisfied, the hierarchy is completely re-constructed:

$$\frac{\lambda_{new}}{\lambda_{old}} \geq 0.975.$$  \hspace{1cm} (F.1)

In the example above, after adding one image and modifying the hierarchy, the $\lambda - dependency$ of the hierarchy becomes $\lambda_{new} = 0.833$. According to formula (F.1), because $\lambda_{new}$ is higher than 97.5% of $\lambda_{old}$, the whole hierarchy does not require total reconstruction.

Following this modification, it is assumed another image is added to the hierarchy, and its decision attribute value is 1. The image is pre-processed and then matched to each elementary set in a similar manner to the previous image. When it is matched to the first decision table in the first hierarchy, based on formula (5.3), it is added to the elementary set $E_1$ in BND. When matched to the second decision table, it is added to the elementary set $E_3$ in BND. Hence, according to the first strategy in Section 6.1, the cardinality of $E_1$ is adjusted from $\text{card}(E_1) = 22$ into $\text{card}(E_1) = 23$, and that of $E_3$ is adjusted from $\text{card}(E_3) = 11$ into $\text{card}(E_3) = 12$. When the image is matched to the third decision table, it is added to the second elementary set, $E_1$. The value of decision attributes of images
classified into $E_1$ is 0, and that of the new image is 1. According to the second strategy in Section 6.1, the cardinality of $E_1$ is adjusted, the counters of the universe of the affected layers are adjusted and the probabilistic decision table of the affected subordinate layer, the fourth decision table of the first hierarchy, is re-computed as shown in Table F.5. Finally, the $\lambda$–dependency of the first hierarchy is re-computed and represented by the third point on Figure F.1. After this image is added, the $\lambda$–dependency of the hierarchy becomes $\lambda_{new} = 0.807$ that cannot satisfy formula (F.1). Thus, the whole hierarchy requires total reconstruction as shown in Table F.6, and the value of the corresponding $\lambda$–dependency is represented by the fourth point of Figure F.1.

**Table F.5** The 1st hierarchy after adding two images

<table>
<thead>
<tr>
<th>Decision table 1</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
<td>$A_{23}$</td>
<td>$A_{22}$</td>
</tr>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$E_2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$E_3$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
<td>$A_{15}$</td>
<td>$A_{14}$</td>
</tr>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$E_2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$E_3$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 3</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
<td>$A_9$</td>
<td>$A_7$</td>
</tr>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$E_2$</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$E_3$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Decision table 4

<table>
<thead>
<tr>
<th>$E_i$</th>
<th>$A_0$</th>
<th>$P(E_i)$</th>
<th>$P(X \mid E_i)$</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>0.8181</td>
<td>0.1111</td>
<td>BND</td>
</tr>
<tr>
<td>$E_1$</td>
<td>0</td>
<td>0.1818</td>
<td>1</td>
<td>POS</td>
</tr>
</tbody>
</table>

### Table F.6 The 1st hierarchy after complete re-construction

### Decision table 1

<table>
<thead>
<tr>
<th>$E_i$</th>
<th>$A_{23}$</th>
<th>$A_9$</th>
<th>$A_{22}$</th>
<th>$P(E_i)$</th>
<th>$P(X \mid E_i)$</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.1326</td>
<td>1</td>
<td>POS</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.1632</td>
<td>0.0624</td>
<td>BND</td>
</tr>
<tr>
<td>$E_2$</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.1326</td>
<td>0.8461</td>
<td>BND</td>
</tr>
<tr>
<td>$E_3$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.0816</td>
<td>0.125</td>
<td>BND</td>
</tr>
<tr>
<td>$E_4$</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.0714</td>
<td>0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_5$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.102</td>
<td>0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_6$</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.1938</td>
<td>0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_7$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1224</td>
<td>0</td>
<td>NEG</td>
</tr>
</tbody>
</table>

### Decision table 2

<table>
<thead>
<tr>
<th>$E_i$</th>
<th>$A_{15}$</th>
<th>$A_{14}$</th>
<th>$P(E_i)$</th>
<th>$P(X \mid E_i)$</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
<td>0.1891</td>
<td>0.1428</td>
<td>BND</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>0</td>
<td>0.3243</td>
<td>1</td>
<td>POS</td>
</tr>
<tr>
<td>$E_2$</td>
<td>0</td>
<td>1</td>
<td>0.2162</td>
<td>0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_3$</td>
<td>0</td>
<td>0</td>
<td>0.2702</td>
<td>0</td>
<td>NEG</td>
</tr>
</tbody>
</table>

### Decision table 3

<table>
<thead>
<tr>
<th>$E_i$</th>
<th>$A_9$</th>
<th>$A_8$</th>
<th>$P(E_i)$</th>
<th>$P(X \mid E_i)$</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_0$</td>
<td>1</td>
<td>1</td>
<td>0.1428</td>
<td>0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_1$</td>
<td>1</td>
<td>0</td>
<td>0.1428</td>
<td>1</td>
<td>POS</td>
</tr>
<tr>
<td>$E_2$</td>
<td>0</td>
<td>1</td>
<td>0.2857</td>
<td>0</td>
<td>NEG</td>
</tr>
<tr>
<td>$E_3$</td>
<td>0</td>
<td>0</td>
<td>0.4285</td>
<td>0</td>
<td>NEG</td>
</tr>
</tbody>
</table>

Following complete reconstruction, it is assumed the third image is added to the hierarchy, and its decision attribute value is 0. It is pre-processed and then matched to
each elementary set in a similar manner to the previous images. When it is matched to the
elementary sets of the first decision table in the first hierarchy, it is classified in \( E_5 \)
located in \( NEG \) in accordance with formula (5.3). Therefore, according to strategy 2 in
Section 6.1, the cardinality of \( E_1 \) is adjusted, the counters of the universe of the affected
layers are adjusted and the \( \lambda - dependency \) is re-computed. Because the value of \( \lambda_{new} \) is
0.969 which satisfies formula (F.1), the hierarchy does not require complete
re-construction. After the third image is added, a few new images are added in turn and
correspondingly the first hierarchy is modified according to the strategies in Section 6.1.
After these new images are added, the \( \lambda - dependency \) of the first hierarchy deteriorates.
Hence, the whole hierarchy requires total reconstruction as shown in Table F.7, and the
value of the corresponding \( \lambda - dependency \) is represented by the 33\(^{rd} \) point on Figure F.1.
After re-computing the decision tables, the remaining new images in the add-on set are
added to the initial training set in a similar manner to the previous images and
correspondingly the first hierarchy is modified.

After all the new images have been utilized for the incremental learning process in
the first hierarchy, as shown in Table F.8, the same process is repeated for the remaining
hierarchies. Table F.9, Table F.10 and Table F.11 are the remaining hierarchies following
the incremental learning process. Figure F.2 displays the values of the \( \lambda - dependency \) in
these hierarchies.
Table F.7 The 1st hierarchy after re-computing decision tables

<table>
<thead>
<tr>
<th>Decision table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
<tr>
<td>$E_2$</td>
</tr>
<tr>
<td>$E_3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
<tr>
<td>$E_2$</td>
</tr>
<tr>
<td>$E_3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
<tr>
<td>$E_2$</td>
</tr>
<tr>
<td>$E_3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
<tr>
<td>$E_2$</td>
</tr>
<tr>
<td>$E_3$</td>
</tr>
</tbody>
</table>
**Table F.8** The 1st hierarchy after all new images have been added

### Decision table 1

| $E_i$ | $A_{49}$ | $A_{28}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|---------|---------|---------|-------------|--------|
| $E_0$ | 1       | 1       | 0.2552  | 0.9387      | BND    |
| $E_1$ | 1       | 0       | 0.2447  | 0.0425      | BND    |
| $E_2$ | 0       | 1       | 0.2395  | 0           | NEG    |
| $E_3$ | 0       | 0       | 0.2604  | 0           | NEG    |

### Decision table 2

| $E_i$ | $A_{23}$ | $A_{22}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|---------|---------|---------|-------------|--------|
| $E_0$ | 1       | 1       | 0.1875  | 0.8333      | BND    |
| $E_1$ | 1       | 0       | 0.3229  | 1           | POS    |
| $E_2$ | 0       | 1       | 0.2812  | 0           | NEG    |
| $E_3$ | 0       | 0       | 0.2083  | 0.0999      | BND    |

### Decision table 3

| $E_i$ | $A_{15}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|----------|----------|-------------|--------|
| $E_0$ | 1        | 0.5263   | 0.05        | BND    |
| $E_1$ | 0        | 0.4736   | 0.8888      | BND    |

### Decision table 4

| $E_i$ | $A_{15}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|----------|----------|-------------|--------|
| $E_0$ | 1        | 0.5526   | 0.0952      | BND    |
| $E_1$ | 0        | 0.4473   | 0.8823      | BND    |

### Decision table 5

| $E_i$ | $A_{15}$ | $A_{13}$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|----------|----------|----------|-------------|--------|
| $E_0$ | 1        | 1        | 0.3947   | 0.9333      | BND    |
| $E_1$ | 1        | 0        | 0.1315   | 0.4         | BND    |
| $E_2$ | 0        | 1        | 0.2368   | 0.1111      | BND    |
| $E_3$ | 0        | 0        | 0.2368   | 0           | NEG    |

### Decision table 6

| $E_i$ | $A_5$ | $A_8$ | $P(E_i)$ | $P(X | E_i)$ | Region |
|-------|------|------|----------|-------------|--------|
| $E_0$ | 1    | 1    | 0.2413   | 0.2857      | BND    |
| $E_1$ | 1    | 0    | 0.0689   | 0           | NEG    |
| $E_2$ | 0    | 1    | 0.4137   | 0.9166      | BND    |
| $E_3$ | 0    | 0    | 0.2758   | 0.5         | BND    |
Table F.9 The 2nd hierarchy after all images have been added

### Decision table 1

| $E_i$ | $A_{23}$ | $A_{22}$ | $A_{21}$ | $P(E_i)$ | $P(X|E_i)$ | Region  |
|-------|----------|----------|----------|----------|------------|---------|
| $E_0$ | 1        | 1        | 1        | 0.1302   | 0          | NEG     |
| $E_1$ | 1        | 1        | 0        | 0.1354   | 0          | NEG     |
| $E_2$ | 1        | 0        | 1        | 0.1406   | 0          | NEG     |
| $E_3$ | 1        | 0        | 0        | 0.0937   | 0          | NEG     |
| $E_4$ | 0        | 1        | 1        | 0.1562   | 1          | POS     |
| $E_5$ | 0        | 1        | 0        | 0.0885   | 0.1176     | BND     |
| $E_6$ | 0        | 0        | 1        | 0.125    | 0.625      | BND     |
| $E_7$ | 0        | 0        | 0        | 0.1302   | 0.0399     | BND     |

### Decision table 2

| $E_i$ | $A_{15}$ | $A_{13}$ | $P(E_i)$ | $P(X|E_i)$ | Region  |
|-------|----------|----------|----------|------------|---------|
| $E_0$ | 1        | 1        | 0.2272   | 1          | POS     |
| $E_1$ | 1        | 0        | 0.2878   | 0.1578     | BND     |
| $E_2$ | 0        | 1        | 0.2272   | 0          | NEG     |
| $E_3$ | 0        | 0        | 0.2575   | 0          | NEG     |

### Decision table 3

| $E_i$ | $A_0$ | $P(E_i)$ | $P(X|E_i)$ | Region  |
|-------|-------|----------|------------|---------|
| $E_0$ | 1     | 0.6842   | 0          | NEG     |
| $E_1$ | 0     | 0.3157   | 0.5        | BND     |
Table F.10 The 3rd hierarchy after all images have been added

| E_i | A_{23} | A_{22} | A_{19} | P(E_i) | P(X| E_i) | Region |
|-----|--------|--------|--------|--------|----------|--------|
| E_0 | 1      | 1      | 1      | 0.1354 | 0        | NEG    |
| E_1 | 1      | 1      | 0      | 0.1302 | 0        | NEG    |
| E_2 | 1      | 0      | 1      | 0.0937 | 0        | NEG    |
| E_3 | 1      | 0      | 0      | 0.1406 | 0        | NEG    |
| E_4 | 0      | 1      | 1      | 0.1145 | 0.0454   | BND    |
| E_5 | 0      | 1      | 0      | 0.1302 | 0.5599   | BND    |
| E_6 | 0      | 0      | 1      | 0.0989 | 0.1578   | BND    |
| E_7 | 0      | 0      | 0      | 0.1562 | 1        | POS    |

Decision table 2

| E_i | A_{15} | P(E_i) | P(X| E_i) | Region |
|-----|--------|--------|----------|--------|
| E_0 | 1      | 0.6969 | 0        | NEG    |
| E_1 | 0      | 0.303  | 0.9      | BND    |

Decision table 3

| E_i | A_{7}  | P(E_i) | P(X| E_i) | Region |
|-----|--------|--------|----------|--------|
| E_0 | 1      | 0.6    | 1        | POS    |
| E_1 | 0      | 0.4    | 0.7499   | BND    |

Decision table 4

| E_i | A_{9}  | P(E_i) | P(X| E_i) | Region |
|-----|--------|--------|----------|--------|
| E_0 | 1      | 0.3333 | 0        | NEG    |
| E_1 | 0      | 0.6666 | 1        | POS    |

221
Table F.11 The 4th hierarchy after all images have been added

<table>
<thead>
<tr>
<th>Decision table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
<tr>
<td>$E_2$</td>
</tr>
<tr>
<td>$E_3$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Decision table 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_i$</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>$E_0$</td>
</tr>
<tr>
<td>$E_1$</td>
</tr>
</tbody>
</table>
Figure F.1 $\lambda$–dependency of the 1st hierarchy after adding new images

Figure F.2 $\lambda$–dependency of the 2nd, 3rd, and 4th hierarchies after adding new images
Bibliography


[38] Dettling, M., Buhlmann, P., Supervised Clustering of Genes, Genome Biol 2002, 3: research 0069.1-0069.15.


[42] Draper, B.A., Baek, K., Bartlett, M.S., Beveridge, J.R., Recognizing Faces with PCA and ICA, Computer Vision Image Understanding (Special Issue on Face Recognition), 2002.


[45] Eigenvalues and eigenvectors, [www documents]
URL: http://www.math.ku.edu/open/LinearAlgebraNotes-Lerner/Chapter16.pdf


[57] Govindarajan, M., Chandrasekaran, R.M., Optimal Design of Radial Basis Function for Intrusion Detection Data, Asian Journal of Information Technology, 7 (11), 2008,


[67] He, H., Garcia, E.A., Learning from Imbalanced Data Export, Knowledge and Data Engineering, IEEE Transactions on Knowledge and Data Engineering, Volume 21, No. 9, 2009, pp. 1263-1284.


[75] Hulse, J.V., Khoshgoftaar, T.M., Napolitano, A., Experimental Perspectives on
Learning from Imbalanced Data, Proceedings of the 24th International Conference on


[77] Hyvärinen, A., Survey on Independent Component Analysis, Neural Computing

[78] Inuiguchi, M., Yoshioka, Y., Kusunoki, Y., Variable-Precision Dominance-Based
Rough Set Approach and Attribute Reduction, International Journal of Approximate

[79] James, F. Peters, Ramanna, S., Feature Selection: Near Set Approach, MCD 2007,
pp.57-71.

[80] Japkowicz, N., Learning from Imbalanced Data Sets: A Comparison of Various

[81] Jung, A., an Introduction to a New Data Analysis Tool: Independent Component

[82] Kailath, T., The Divergence and Bhattacharyya Distance Measures in Signal

[83] Kanade, T., Computer Recognition of Human Faces, Birkhauser, Basel, Switzerland,

[84] Kanade, T., Picture Processing by Computer Complex and Recognition of Human


[100] Lee, K.K., Harris, C.J., Gunn, S.R., Reed, P.A.S., Control Sensitivity SVM for Imbalanced Data, Control Sensitivity SVM for Imbalanced Data, International Conference of Artificial Neural Networks and Genetic Algorithms, Prague, Czech
Republic. 2001.


[109] Liu, J., Hu, Q., Yu, D., A Comparative Study on Rough Set Based Class Imbalance


[126] Muhlbaier, M., Topalis, A., Polikar, R., Incremental Learning from Unbalanced

[127] Nguyen, G. H., Bouzerdoum, A., Phung, S.L., A Supervised Learning Approach for Imbalanced Data Sets, [WWW documents]


[134] Okada, K., Steffans, J., Maurer, T., Hong, H., Elagin, E., Neven, H., Malsburg, C. V., D., The Bochum/USC Face Recognition System and How It Fared in the FERET


[169] Section 6.1, eigenvectors and eigenvalues, [www documents]
URL: http://www.mtholyoke.edu/~jjlee/Teaching/math125summer09note5.pdf.


[196] Tian, F., Zhang, H., Lu, Y., Shi, C., Incremental Learning of Bayesian Networks with Hidden Variables, First IEEE International Conference on Data Mining (ICDM'01),
San Jose, California, 2002.


[205] Wang, E. H., Kuh, A., A Smart Algorithm for Incremental Learning, Proceeding of


[221] Yao, Y.Y., Decision-Theoretic Rough Set Models, Rough Sets and Knowledge,


[229] Yao, Y.Y., Zhao, Y., Discernibility Matrix Simplification for Constructing Attribute


[254] Ziarko, W., Probabilistic Dependencies in Multivalued Classification Tables, IPMU 2006.