DATA-VARIANT KERNEL ANALYSIS
DATA-VARIANT KERNEL ANALYSIS

Yuichi Motai, Ph.D.
Sensory Intelligence Laboratory,
Department of Electrical and Computer Engineering,
Virginia Commonwealth University
Richmond, VA
The author dedicates this book in memoriam to his father, Osami Motai, who passed away on June 29, 2013.
Chapter 1  Survey

1.1 Introduction of Kernel Analysis / 1

1.2 Kernel Offline Learning / 2
   1.2.1 Choose the Appropriate Kernels / 3
   1.2.2 Adopt KA into the Traditionally Developed Machine Learning Techniques / 6
   1.2.3 Structured Database with Kernel / 9

1.3 Distributed Database with Kernel / 12
   1.3.1 Multiple Database Representation / 12
   1.3.2 Kernel Selections Among Heterogeneous Multiple Databases / 13
   1.3.3 Multiple Database Representation KA Applications to Distributed Databases / 14

1.4 Kernel Online Learning / 16
   1.4.1 Kernel-Based Online Learning Algorithms / 16
   1.4.2 Adopt “Online” KA Framework into the Traditionally Developed Machine Learning Techniques / 17
   1.4.3 Relationship Between Online Learning and Prediction Techniques / 21

1.5 Prediction with Kernels / 22
   1.5.1 Linear Prediction / 22
   1.5.2 Kalman Filter / 23
   1.5.3 Finite-State Model / 23
   1.5.4 Autoregressive Moving Average Model / 24
   1.5.5 Comparison of Four Models / 25

1.6 Future Direction and Conclusion / 26

References / 26
Chapter 2  Offline Kernel Analysis  41

2.1  Introduction / 41

2.2  Kernel Feature Analysis / 43
   2.2.1  Kernel Basics / 43
   2.2.2  Kernel Principal Component Analysis (KPCA) / 45
   2.2.3  Accelerated Kernel Feature Analysis (AKFA) / 46
   2.2.4  Comparison of the Relevant Kernel Methods / 48

2.3  Principal Composite Kernel Feature Analysis (PC-KFA) / 49
   2.3.1  Kernel Selections / 49
   2.3.2  Kernel Combinatory Optimization / 52

2.4  Experimental Analysis / 54
   2.4.1  Cancer Image Datasets / 54
   2.4.2  Kernel Selection / 56
   2.4.3  Kernel Combination and Reconstruction / 58
   2.4.4  Kernel Combination and Classification / 59
   2.4.5  Comparisons of Other Composite Kernel Learning Studies / 60
   2.4.6  Computation Time / 61

2.5  Conclusion / 61

References / 62

Chapter 3  Group Kernel Feature Analysis  69

3.1  Introduction / 69

3.2  Kernel Principal Component Analysis (KPCA) / 71

3.3  Kernel Feature Analysis (KFA) for Distributed Databases / 73
   3.3.1  Extract Data-Dependent Kernels Using KFA / 73
   3.3.2  Decomposition of Database Through Data Association via Recursively Updating Kernel Matrices / 75

3.4  Group Kernel Feature Analysis (GKFA) / 78
   3.4.1  Composite Kernel: Kernel Combinatory Optimization / 79
   3.4.2  Multiple Databases Using Composite Kernel / 81

3.5  Experimental Results / 83
   3.5.1  Cancer Databases / 83
   3.5.2  Optimal Selection of Data-Dependent Kernels / 84
   3.5.3  Kernel Combinatory Optimization / 84
   3.5.4  Composite Kernel for Multiple Databases / 86
   3.5.5  K-NN Classification Evaluation with ROC / 87
   3.5.6  Comparison of Results with Other Studies on Colonography / 89
   3.5.7  Computational Speed and Scalability Evaluation of GKFA / 90

3.6  Conclusions / 91

References / 92
Chapter 4  Online Kernel Analysis  

4.1 Introduction / 97  

4.2 Kernel Basics: A Brief Review / 99  
   4.2.1 Kernel Principal Component Analysis / 99  
   4.2.2 Kernel Selection / 100  

4.3 Kernel Adaptation Analysis of PC-KFA / 101  

4.4 Heterogeneous vs. Homogeneous Data for Online PC-KFA / 102  
   4.4.1 Updating the Gram Matrix of the Online Data / 103  
   4.4.2 Composite Kernel for Online Data / 104  

4.5 Long-Term Sequential Trajectories with Self-Monitoring / 104  
   4.5.1 Reevaluation of Large Online Data / 105  
   4.5.2 Validation of Decomposing Online Data into Small Chunks / 106  

4.6 Experimental Results / 107  
   4.6.1 Cancer Datasets / 107  
   4.6.2 Selection of Optimum Kernel and Composite Kernel for Offline Data / 108  
   4.6.3 Selection of Optimum Kernel and Composite Kernel for the New Online Sequences / 110  
   4.6.4 Classification of Heterogeneous Versus Homogeneous Data / 111  
   4.6.5 Online Learning Evaluation of Long-term Sequence / 112  
   4.6.6 Evaluation of Computational Time / 116  

4.7 Conclusions / 117  

References / 117  

Chapter 5  Cloud Kernel Analysis  

5.1 Introduction / 121  

5.2 Cloud Environments / 123  
   5.2.1 Server Specifications of Cloud Platforms / 123  
   5.2.2 Cloud Framework of KPCA for AMD / 124  

5.3 AMD for Cloud Colonography / 125  
   5.3.1 AMD Concept / 125  
   5.3.2 Data Configuration of AMD / 126  
   5.3.3 Implementation of AMD for Two Cloud Cases / 129  
   5.3.4 Parallelization of AMD / 132  

5.4 Classification Evaluation of Cloud Colonography / 135  
   5.4.1 Databases with Classification Criteria / 135  
   5.4.2 Classification Results / 137  

5.5 Cloud Computing Performance / 140  
   5.5.1 Cloud Computing Setting with Cancer Databases / 140  
   5.5.2 Computation Time / 142
Chapter 6  
Predictive Kernel Analysis  

6.1 Introduction / 153
6.2 Kernel Basics / 154
   6.2.1 KPCA and AKFA / 155
6.3 Stationary Data Training / 157
   6.3.1 Kernel Selection / 157
   6.3.2 Composite Kernel: Kernel Combinatory Optimization / 159
6.4 Longitudinal Nonstationary Data with Anomaly/Normal Detection / 160
   6.4.1 Updating the Gram Matrix Based on Nonstationary Longitudinal Data / 160
   6.4.2 Composite Kernel for Nonstationary Data / 162
6.5 Longitudinal Sequential Trajectories for Anomaly Detection and Prediction / 163
   6.5.1 Anomaly Detection of Nonstationary Small Chunks Datasets / 164
   6.5.2 Anomaly Prediction of Long-Time Sequential Trajectories / 167
6.6 Classification Results / 169
   6.6.1 Cancer Datasets / 169
   6.6.2 Selection of Optimum Kernel and Composite Kernel for Stationary Data / 170
   6.6.3 Comparisons with Other Kernel Learning Methods / 172
   6.6.4 Anomaly Detection for the Nonstationary Data / 174
6.7 Longitudinal Prediction Results / 175
   6.7.1 Large Nonstationary Sequential dataset for Anomaly Detection / 175
   6.7.2 Time Horizontal Prediction for Risk Factor Analysis of Anomaly Long-Time Sequential Trajectories / 178
   6.7.3 Computational Time for Complexity Evaluation / 179
6.8 Conclusions / 180

References / 181

Chapter 7  
Conclusion  

Appendix A  

Appendix B  Representative Matlab codes  

B.1 Accelerated Kernel Feature Analysis / 196
B.2 Experimental Evaluations / 198
B.3 Group Kernel Analysis / 201
B.4 Online Composite Kernel Analysis / 206
B.5 Online Data Sequences Control / 208
B.6 Alignment Factor / 209
B.7 Cloud Kernel Analysis / 210
B.8 Plot Computation Time / 211
B.9 Parallelization / 212

Index 215
LIST OF FIGURES

Figure 1.1 General RBF-NN [97]. The input and output layers perform functions similarly to a normal neural network. The hidden layer uses the RBF kernel, denoted as its activation function as shown in Equation 1.4.

Figure 1.2 Principle of kernel PCA. (a) nonlinear in input space. (b) high-dimensional feature space corresponding to nonlinear input space. The contour lines of constant projections generate the principal eigenvectors. Kernel PCA does not actually require $\Phi$ to be mapped into $F$. All necessary computations are executed through the use of a kernel function $k$ in input space $\mathbb{R}^2$.

Figure 1.3 Significant research publications in various fields of KA and kernel selection for offline learning. Works related to neural network, SVM, PCA, and kernel selection are displayed according to citation number versus publishing year.

Figure 1.4 The composition of the multiple databases. The multiple databases include distributed databases and single databases, which are stored on different storage spaces connected within a network. Three key attributes of the multiple databases are (i) simultaneous access and modification, (ii) extraction from huge amounts, and (iii) distributed clustering/classification.

Figure 1.5 Several pregenerated kernels are trained to access different features in several distributed databases. The output of this kernel network is a sum of the kernel functions that maps to the data of interest. The summation and biasing make this method behave similarly to a neural network [178].

Figure 1.6 Online kernel-based learning algorithm. As a new training sample is added, the kernel is updated on the basis of the online learning algorithms described in the following section.
Figure 1.7  Key papers for Kernel Online learning based on the number of citations for the years 1998–2013, for principal component analysis (PCA), support vector machine (SVM), neural network (NN), autoregressive moving average (ARMA), and finite-state model (FSM).

Figure 1.8  Online NN learning algorithm. An artificial neural network consists of input, hidden, and output layers interconnected with directed weights \( w \), where \( W_{ij} \) denotes the input-to-hidden layer weights at the hidden neuron \( j \) and \( w_{jk} \) denotes the hidden-to-output layer weights at the output neuron \( k \) [218].

Figure 1.9  Scheme for naïve online SVM classification. The input data are mapped to the kernel Hilbert space, and the hyperplane is generated as in offline SVM. If an optimal solution cannot be found, a new kernel is selected. The process repeats for newly incoming data.

Figure 1.10  Online KPCA diagram. In the initial (offline) training phase, the initial kernel parameters and eigen-feature space are calculated. As new training sets are introduced, incremental (online) adjustments are made to the parameters and an updated eigen-space is generated.

Figure 1.11  Linear predictor with an initial kernel mapping. The input data is mapped from its original space to a new Hilbert space where \( x(n) \) is linear, by the function \( \Phi \) to allow for the use of the kernel trick. The model is optimized by minimized the function \( e(N) \), which is the difference in the current and previous predicted value.

Figure 1.12  For processing a nonlinear time series, a kernelized Kalman filter is an optimal way to execute the Kalman filter. With a nonlinear series \( x_k \) and an associated series \( x_k^\phi \), \( x_k \) is mapped to \( x_k^\phi \) by the nonlinear map \( \Phi : R^d \rightarrow H \), where \( R^d \) is the input space, \( H \) is the feature space, and \( [\Phi(X), \Phi(Y)] = k(x,y) \).

Figure 1.13  A kernel autoregressive moving average (KARMA) model with online learning for tracking of hand movement.

Figure 2.1  The computation time comparison between KPCA, SKFA, AKFA, and PC-KFA as the data size increases.

Figure 3.1  Distributed colonography with distributed image databases for colon cancer diagnosis. The hosting server collects and analyzes databases from different institutions and groups them into assembled databases.
Figure 3.2  The concept of group kernel feature analysis. The proposed criteria are to determine the nature of database by (i) decomposition, (ii) classification by heterogeneity, and (iii) combination.  74

Figure 3.3  Relationship of the criteria to determine homogenous and heterogeneous degree.  77

Figure 3.4  The steps to choose the basic kernel matrix $P'$ and the updating process according to the elements $rf$.  77

Figure 3.5  The training flow chart of reclustered databases due to the heterogeneous nature.  79

Figure 3.6  Group kernel feature analysis (GKFA). The first two steps 1 and 2 are the same as in Fig. 3.2. Step 3 of Fig. 3.2 is illustrated here for assembled databases through kernel choice in the composite kernel by Sections 3.4.1 and 3.4.2.  82

Figure 3.7  The overall GKFA steps of newly assembled databases.  83

Figure 3.8  The computation time comparison between KPCA, KFA, and GKFA as the data size increases.  91

Figure 4.1  Developed CAD for CTC at 3D imaging group at Harvard Medical School [5].  98

Figure 4.2  Criteria concept for homogeneous and heterogeneous online data.  103

Figure 4.3  Overall flow of online data training using HBDA with PC-KFA.  105

Figure 4.4  Online decomposition for heterogeneous sequences.  106

Figure 4.5  Training of online datasets acquired over long-term sequences.  107

Figure 4.6  The ROC performance of offline data using PC-KFA.  109

Figure 4.7  The online dataset sequences of Table 4.4.  110

Figure 4.8  Long-term data sequence used to evaluate online learning. The horizontal axis denotes the ratio of number of online data to number of offline data. The vertical axis denotes the number of total data processed using online HBDA with PC-KFA. The three long sequences are labeled as “Large Online Data,” “Medium Online Data,” and “Small Online Data,” which were used corresponding to offline training dataset size of 750, 937, and 1250, respectively.  113
Figure 4.9  Alignment factors for long-term sequences. The horizontal axis denotes the progress of the online HBDA (PC-KFA) with time (ratio of online and offline training). The solid lines denoted the mean value of the observed AF value, while the dashed lines show the range of observed AF.

Figure 4.10  AUC of Receiver operating Characteristics for three long-term sequences. The horizontal axis denotes the progress of the online HBDA (PC-KFA) with time (ratio of online-to-offline training).

Figure 4.11  Accuracy versus ratio of online data to offline data.

Figure 4.12  Computational time for online HBDA with offline PC-KFA.

Figure 5.1  A concept of Cloud Colonography with distributed image databases for colon cancer diagnosis. The cloud server hosting will collect distributed databases from different institutions and group them using KA.

Figure 5.2  An illustration of KPCA mathematical background. KPCA calculates the eigenvectors and eigenvalues by analyzing the kernel feature space of multiple institutions so that a cloud server can handle larger datasets.

Figure 5.3  A concept of AMD. The proposed kernel framework combines the Cloud Colonography datasets by analyzing the images of polyps with nonlinear big feature space.

Figure 5.4  Four representative steps of AMD. The proposed AMD consists of the four main criteria to manage databases by (i) split, (ii) combine, (iii) sort, and (iv) merge.

Figure 5.5  Cloud Colonography architecture in private cloud environment. The proposed Cloud Colonography consists of the four representative layers from CTC CAD analysis to the service desk reporting for clinical users.

Figure 5.6  An extension of AMD framework in public cloud scenario.

Figure 5.7  A proposed parallelization organization for cloud computing.

Figure 5.8  ROC with Sigmoid and Gauss group kernel. (a) Comparison between assembling databases and databases 9, 11, 13, and 16. (b) Comparison between assembling databases and databases 1, 6, 12, and 19. (c) Comparison between assembling databases and databases 5, 8, 15, and 18.
Figure 5.9 ROC with Sigmoid and poly group kernel. 139
Figure 5.10 ROC with Sigmoid and linear group kernel. 139
Figure 5.11 Data compression ratio for data size. The horizontal axis denotes the size of the data, and vertical axis denotes the compressed data ratio. In the three databases, the data compressions are relatively distinct to each other. 141
Figure 5.12 Total training time required for Cloud Colonography with AMD. 142
Figure 5.13 Mean execution time for Cloud Colonography with AMD. 143
Figure 5.14 Computational time for Gram matrix in Cloud Colonography with AMD. 144
Figure 5.15 Memory usage for Cloud Colonography with AMD. 145
Figure 6.1 The illustration of a nonincremental update of a combination vector. The trivial rows and columns of the previous Gram matrix with those of the new data are replaced as follows $\alpha_r(\text{min} \Delta) \leftarrow \alpha'_r$. 161
Figure 6.2 Training of first nonstationary batch of data using two cases to iteratively update composite kernels. Anomaly or normal detection is conducted using the proposed criterion. 163
Figure 6.3 A conceptual illustration of the division of a large nonstationary dataset into several small subsets of equal size. Large dataset is clustered into small subsets of datasets, for example, 1-2-3-4, so make the feasible size of the time sequence. 164
Figure 6.4 Nonincremental update of data at time $t + 1$. The data sequence with time index is evaluated for replacing the previous Gram matrix by the new time sequential datasets. 166
Figure 6.5 A conceptual illustration of colorectal cancer staging. The degree of cancer progress is indexed by stages 0–IV, corresponding to the size of the tumor. 167
Figure 6.6 The training of nonstationary datasets is acquired over long-time sequences to divide the huge dataset by alignment factors, and to compute the cancer stage corresponding to synthesized measurement data. 168
Figure 6.7 The ROC performance of stationary databases 1–4 using composite KFA with k-NN. AUC of four databases shows very competitive results for the other kernel learning methods shown in Fig. 6.8. 171
Figure 6.8  The ROC performance of single KFA versus composite KFA for SVM. Composite KFA approaches work on the top of SVM, showing good results of four databases ((a) Database1, (b) Database2, (3) Database3, and (d) Database4) with high AUC values. 173

Figure 6.9  Labeling “anomaly” with cancer stage for all TP cases of time indexing corresponding to the segmented polyp pixels. Area of pixel histogram (APH) is used for assigning each cancer stage from 0 to III. 176

Figure 6.10  Labeling “normal” with cancer stage for all FP cases of time indexing corresponding to the segmented polyp pixels. Area of pixel histogram (APH) represent each cancer stage from 0 to III. 176

Figure 6.11  Transient from normal to anomaly among longitudinal sequential datasets. The indexing of cancer stages is based on Figs. 6.9 and 6.10. The 15 cases of transition from normal to anomaly are shown. 177

Figure 6.12  Post-probabilities of anomaly cases among 15 longitudinal sequential datasets. Based on Fig. 6.11 this plot reconfigured the axis to see the sorting order how cancer stage calculated post-probabilities of anomaly cases. 178

Figure 6.13  The cancer stage of long-time sequence trajectories using (a) $k$-step prediction, (b) predictive LACK. When the horizontal time window increased 1, 5, 10, 50, and 100 ahead of the time of data sequence, the predicted values were off from the actual cancer stage value. (a) $k$-Step and (b) predictive LACK performed similar with small predicted window sizes $k$ such as $k = 1 – 10$. If $k = 50$ or 100 (under the larger prediction time frame), LACK performed much better than $k$-step. 179
<p>| Table 1.1 | Kernel selection approaches and related research | 4 |
| Table 1.2 | Hyperkernel method comparison | 5 |
| Table 1.3 | Neural network approaches related to kernel algorithm | 7 |
| Table 1.4 | SVM approaches relevant to solve optimization problem | 8 |
| Table 1.5 | PCA approaches related to kernel algorithm | 10 |
| Table 1.6 | Application of kernels for structured databases | 11 |
| Table 1.7 | Distributed database application-wise concentration | 15 |
| Table 1.8 | Comparisons of prediction and online learning algorithms | 22 |
| Table 1.9 | Quantitative comparisons of prediction error | 25 |
| Table 1.10 | Comparisons of computational speed | 26 |
| Table 2.1 | Overview of method comparison for parameters tuning | 49 |
| Table 2.2 | Colon cancer dataset 1 (low resolution) | 54 |
| Table 2.3 | Colon cancer dataset 2 (U. Chicago) | 54 |
| Table 2.4 | Colon cancer dataset 3 (BID) | 55 |
| Table 2.5 | Colon cancer dataset 4 (NorthWestern U.) | 55 |
| Table 2.6 | Breast cancer dataset | 55 |
| Table 2.7 | Lung cancer dataset | 56 |
| Table 2.8 | Lymphoma dataset | 56 |
| Table 2.9 | Prostate cancer dataset | 56 |
| Table 2.10 | Eigenvalues $\lambda$ of five kernel functions (Eqs. 2.1–2.5) and their parameters selected | 57 |
| Table 2.11 | Mean square reconstruction error of KPCA, SKFA, and AKFA with the selected kernel function | 57 |
| Table 2.12 | Mean square reconstruction error of KPCA with other 4 kernel functions | 58 |
| Table 2.13 | Linear combination $\hat{\rho}$ for selected two kernel functions | 58 |</p>
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.14</td>
<td>Mean square reconstruction error with kernel combinatory optimization</td>
<td>59</td>
</tr>
<tr>
<td>2.15</td>
<td>Classification accuracy using six nearest neighborhoods for single-kernel and two-composite-kernels with KPCA, SKFA,</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>AKFA, and PC-KFA</td>
<td></td>
</tr>
<tr>
<td>2.16</td>
<td>Overall classification comparison among other multiple kernel methods</td>
<td>60</td>
</tr>
<tr>
<td>2.17</td>
<td>PC-KFA Computation time for kernel selection and operation with KPCA, SKFA, AKFA, and PC-KFA</td>
<td>61</td>
</tr>
<tr>
<td>3.1</td>
<td>Databases</td>
<td>84</td>
</tr>
<tr>
<td>3.2</td>
<td>Eigenvalues of four kernels for offline databases</td>
<td>85</td>
</tr>
<tr>
<td>3.3</td>
<td>The value of $\hat{\rho}$ for each of the composite kernels</td>
<td>86</td>
</tr>
<tr>
<td>3.4</td>
<td>GKFA for assembled database</td>
<td>87</td>
</tr>
<tr>
<td>3.5</td>
<td>Performance with Sigmoid and Gauss group kernel (CASE 1)</td>
<td>88</td>
</tr>
<tr>
<td>3.6</td>
<td>Performance with Sigmoid and Gauss group kernel (CASE2)</td>
<td>88</td>
</tr>
<tr>
<td>3.7</td>
<td>Performance with Sigmoid and Gauss group kernel (CASE3)</td>
<td>88</td>
</tr>
<tr>
<td>3.8</td>
<td>Performance with Sigmoid and poly group kernel</td>
<td>89</td>
</tr>
<tr>
<td>3.9</td>
<td>Performance with Sigmoid and linear group kernel</td>
<td>89</td>
</tr>
<tr>
<td>3.10</td>
<td>comparison of different classification methods</td>
<td>89</td>
</tr>
<tr>
<td>3.11</td>
<td>Classification performance with different methods</td>
<td>90</td>
</tr>
<tr>
<td>3.12</td>
<td>Classification accuracy to compare other methods</td>
<td>90</td>
</tr>
<tr>
<td>4.1</td>
<td>Arrangement of offline datasets</td>
<td>108</td>
</tr>
<tr>
<td>4.2</td>
<td>Eigenvalues of four kernels for offline datasets</td>
<td>108</td>
</tr>
<tr>
<td>4.3</td>
<td>Classification accuracy and mean square error of offline data</td>
<td>109</td>
</tr>
<tr>
<td>4.4</td>
<td>Online sequences generated from datasets</td>
<td>110</td>
</tr>
<tr>
<td>4.5</td>
<td>Eigenvalues of four different kernels (linear, poly, RBF, Laplace) with $\lambda$ for online data sequences</td>
<td>111</td>
</tr>
<tr>
<td>4.6</td>
<td>Homegeneous and heterogeneous categories of online sequences</td>
<td>112</td>
</tr>
<tr>
<td>4.7</td>
<td>Update of different sets of Online Sequences with Gram matrix size</td>
<td>112</td>
</tr>
<tr>
<td>4.8</td>
<td>Classification accuracy and MSE with composite kernels of online data sequences</td>
<td>113</td>
</tr>
<tr>
<td>4.9</td>
<td>Online data sequence computation time</td>
<td>117</td>
</tr>
<tr>
<td>5.1</td>
<td>Representative cloud venders</td>
<td>123</td>
</tr>
<tr>
<td>5.2</td>
<td>Desktop and private cloud server hardware specification</td>
<td>131</td>
</tr>
<tr>
<td>5.3</td>
<td>Amazon EC2 instance servers specification</td>
<td>133</td>
</tr>
<tr>
<td>Table</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>5.4</td>
<td>Databases</td>
<td>136</td>
</tr>
<tr>
<td>5.5</td>
<td>AMD with KPCA for assembled database</td>
<td>137</td>
</tr>
<tr>
<td>5.6</td>
<td>Classification accuracy for assembled database</td>
<td>140</td>
</tr>
<tr>
<td>5.7</td>
<td>Databases for cloud computing analysis</td>
<td>141</td>
</tr>
<tr>
<td>5.8</td>
<td>Averaged training computation time for three databases</td>
<td>143</td>
</tr>
<tr>
<td>5.9</td>
<td>Average total testing time per sample</td>
<td>144</td>
</tr>
<tr>
<td>5.10</td>
<td>Cost components for private cloud service</td>
<td>145</td>
</tr>
<tr>
<td>5.11</td>
<td>Cost components for public cloud Amazon EC2 (c3.8xlarge)</td>
<td>146</td>
</tr>
<tr>
<td>5.12</td>
<td>Time and memory using parallelization for private cloud</td>
<td>146</td>
</tr>
<tr>
<td>5.13</td>
<td>Time and cost using parallelization for public cloud Amazon EC2 (c3.8xlarge)</td>
<td>146</td>
</tr>
<tr>
<td>6.1</td>
<td>Arrangement of datasets</td>
<td>169</td>
</tr>
<tr>
<td>6.2</td>
<td>Eigenvalues of four kernels for stationary datasets</td>
<td>170</td>
</tr>
<tr>
<td>6.3</td>
<td>The value of $\hat{\rho}$ for each of the composite kernels</td>
<td>170</td>
</tr>
<tr>
<td>6.4</td>
<td>Classification accuracy and mean square error of stationary data</td>
<td>171</td>
</tr>
<tr>
<td>6.5</td>
<td>Overview of framework methodology</td>
<td>172</td>
</tr>
<tr>
<td>6.6</td>
<td>Classification Accuracy to compare other multiple kernel methods</td>
<td>172</td>
</tr>
<tr>
<td>6.7</td>
<td>Classification accuracy to compare other multiple kernel methods using UCI machine learning repository</td>
<td>174</td>
</tr>
<tr>
<td>6.8</td>
<td>Eigenvalues $\lambda$ of four different kernels for nonstationary data sequences #1–#10 for the base nonstationary Sets #1–#3</td>
<td>175</td>
</tr>
<tr>
<td>6.9</td>
<td>Anomaly detection with classification accuracy and AUC for nonstationary data sequences #1–#10, and for the base nonstationary sets #1–#3</td>
<td>175</td>
</tr>
<tr>
<td>6.10</td>
<td>Average error of prediction using several representative prediction window sizes</td>
<td>180</td>
</tr>
<tr>
<td>6.11</td>
<td>Computational time (millisecond) of each module for each dataset</td>
<td>180</td>
</tr>
<tr>
<td>7.1</td>
<td>Summary of framework data-variant KA methodology</td>
<td>186</td>
</tr>
<tr>
<td>A.1</td>
<td>Acronyms definitions</td>
<td>189</td>
</tr>
<tr>
<td>A.2</td>
<td>Symbol definitions</td>
<td>191</td>
</tr>
</tbody>
</table>
Kernel methods have been extensively studied in pattern classification and its applications for the past 20 years. Kernel may refer to diverse meanings in different areas such as Physical Science, Mathematics, Computer Science, and even Music/Business. For the area of Computer Science, the term “kernel” is used in different contexts (i) central component of most operating systems, (ii) scheme-like programming languages, and (iii) a function that executes on OpenCL devices. In machine learning and statistics, the term kernel is used for a pattern recognition algorithm. The kernel functions for pattern analysis, called kernel analysis (KA), is the central theme of this book. KA uses “kernel trick” to replace feature representation of data with similarities to other data. We will cover KA topics ranging from the fundamental theory of kernel functions to applications. The overall structure starts from Survey in Chapter 1. On the basis of the KA configurations, the remaining chapters consist of Offline KA in Chapter 2, Group KA in Chapter 3, Online KA in Chapter 4, Cloud KA in Chapter 5, and Predictive KA in Chapter 6. Finally, Chapter 7 concludes by summarizing these distinct algorithms.

Chapter 1 surveys the current status, popular trends, and developments on KA studies, so that we can oversee functionalities and potentials in an organized manner:

- Utilize KA with different types of data configurations, such as offline, online, and distributed, for pattern analysis framework.
- Adapt KA into the traditionally developed machine learning techniques, such as neural networks (NN), support vector machines (SVM), and principal component analysis (PCA).
- Evaluate KA performance among those algorithms.

Chapter 2 covers offline learning algorithms, in which KA does not change its approximation of the target function, once the initial training phase has been absorbed. KA mainly deals with two major issues: (i) how to choose the appropriate kernels for offline learning during the learning phase, and (ii) how to adopt KA into the traditionally developed machine learning techniques such as NN, SVM, and PCA, where the (nonlinear) learning data-space is placed under the linear space via kernel tricks.

Chapter 3 covers group KA as a data-distributed extension of offline learning algorithms. The data used for Chapter 3 is now extended into several databases. Group KA for distributed data is explored to demonstrate the big-data analysis with the comparable performance of speed and memory usages.