DATA MINING IN AGRICULTURE

Springer Optimization and Its Applications

VOLUME 34

Managing Editor
Panos M. Pardalos (University of Florida)

Editor—Combinatorial Optimization
Ding-Zhu Du (University of Texas at Dallas)

Advisory Board
J. Birge (University of Chicago)
C.A. Floudas (Princeton University)
F. Giannessi (University of Pisa)
H.D. Sherali (Virginia Polytechnic and State University)
T. Terlaky (McMaster University)
Y. Ye (Stanford University)

Aims and Scope
Optimization has been expanding in all directions at an astonishing rate during the last few decades. New algorithmic and theoretical techniques have been developed, the diffusion into other disciplines has proceeded at a rapid pace, and our knowledge of all aspects of the field has grown even more profound. At the same time, one of the most striking trends in optimization is the constantly increasing emphasis on the interdisciplinary nature of the field. Optimization has been a basic tool in all areas of applied mathematics, engineering, medicine, economics and other sciences.

The *Springer Optimization and Its Applications* series publishes undergraduate and graduate textbooks, monographs and state-of-the-art expository works that focus on algorithms for solving optimization problems and also study applications involving such problems. Some of the topics covered include nonlinear optimization (convex and nonconvex), network flow problems, stochastic optimization, optimal control, discrete optimization, multiobjective programming, description of software packages, approximation techniques and heuristic approaches.

For other titles published in this series, go to
www.springer.com/series/7393

DATA MINING IN AGRICULTURE

By

ANTONIO MUCHERINO
University of Florida, Gainesville, FL, USA

PETRAQ J. PAPAJORGJI
University of Florida, Gainesville, FL, USA

PANOS M. PARDALOS
University of Florida, Gainesville, FL, USA

Springer

Antonio Mucherino
Institute of Food & Agricultural
Information Technology Office
University of Florida
P.O. Box 110350
Gainesville, FL 32611
USA
amucherino@ufl.edu

Petraq J. Papajorgji
Institute of Food & Agricultural
Information Technology Office
University of Florida
P.O. Box 110350
Gainesville, FL 32611
USA
petraq@ifas.ufl.edu

Panos M. Pardalos
Department of Industrial & Systems Engineering
University of Florida
303 Weil Hall
Gainesville, FL 32611-6595
USA
pardalos@ise.ufl.edu

ISSN 1931-6828
ISBN 978-0-387-88614-5 e-ISBN 978-0-387-88615-2
DOI 10.1007/978-0-387-88615-2
Springer Dordrecht Heidelberg London New York

Library of Congress Control Number: 2009934057

© Springer Science+Business Media, LLC 2009
All rights reserved. This work may not be translated or copied in whole or in part without the written permission of the publisher (Springer Science+Business Media, LLC, 233 Spring Street, New York, NY 10013, USA), except for brief excerpts in connection with reviews or scholarly analysis. Use in connection with any form of information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed is forbidden.
The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Dedicated to Sonia
who supported me morally
during the preparation of this book.

To the memory of my parents
Eleni and Jorgji Papajorgji
who taught me not to betray my principles
even in tough times.

Dedicated to my father and mother
Miltiades and Kalypso Pardalos
for teaching me to love nature
and to grow my own garden.

Preface

Data mining is the process of finding useful patterns or correlations among data. These patterns, associations, or relationships between data can provide information about a specific problem being studied, and information can then be used for improving the knowledge on the problem. Data mining techniques are widely used in various sectors of the economy. Initially they were used by large companies to analyze consumer data from different perspectives. Data was then analyzed and useful information was extracted with the goal of increasing profitability.

The idea of using information hidden in relationships among data inspired researchers in agricultural fields to apply these techniques for predicting future trends of agricultural processes. For example, data collected during wine fermentation can be used to predict the outcome of the fermentation while still in the early days of this process. In the same way, soil water parameters for a certain soil type can be estimated knowing the behavior of similar soil types.

The principles used by some data mining techniques are not new. In ancient Rome, the famous orator Cicero used to say *pares cum paribus facillime congregantur* (*birds of a feather flock together* or literally *equals with equals easily associate*). This old principle is successfully applied to classify unknown samples based on known classification of their neighbors. Before writing this book, we thoroughly researched applications of data mining techniques in the fields of agriculture and environmental studies. We found papers describing systems developed to classify apples, separating good apples from bad ones on a conveyor belt. We found literature describing a system that classifies chicken breast quality, and others describing systems able to predict climate forecasting and soil classification, and so forth. All these systems use various data mining techniques.

Therefore, given the scientific interest and the positive results obtained using the data mining techniques, we thought that it was time to provide future specialists in agriculture and environment-related fields with a textbook that will explain basic techniques and recent developments in data mining. Our goal is to provide students and researchers with a book that is easy to read and understand. The task was challenging. Some of the data mining techniques can be transformed into optimization problems, and their solutions can be obtained using appropriate optimization methods. Although this transformation helps finding a solution to the problem, it makes the presentation difficult to understand by students that do not have a strong mathematical background.

The clarity of the presentation was the major obstacle that we worked hard to overcome. Thus, whenever possible, examples in Euclidean space are provided and corresponding figures are shown to help understand the topic. We make abundant use of MATLAB® to create examples and the corresponding figures that visualize the solution. Besides, each technique presented is ranked using a well-known publication on the relevance of data mining techniques. For each technique, the reader will find published examples of its use by researchers around the world and simple examples that will help in its understanding. We made serious efforts to shed light on when to use the method and the quality of the expected results. An entire chapter is dedicated to the validation of the techniques presented in the book, and examples in MATLAB are used again to help the presentation. Another chapter discusses the potential implementation of data mining techniques in a parallel computing environment; practical applications often require high-speed computing environments. Finally, one appendix is devoted to the MATLAB environment and another one is dedicated to the implementation of one of the presented data mining techniques in C programming language.

It is our hope that readers will find this book to be of use. We are very thankful to our students that helped us shape this course. As always, their comments were useful and appropriate and helped us create a consistent course. We thank Vianney Houles, Guillermo Baigorria, Erhun Kundakcioglu, Sepehr M. Nasseri, Neng Fan, and Sonia Cafieri for reading all the material and for finding subtle inconsistencies. Last but certainly not least, we thank Vera Tomaino for reading the entire book very carefully and for working all exercises. Her input was very useful to us.

Finally, we thank Springer for trusting and giving us another opportunity to work with them.

Gainesville, Florida
January 2009

Antonio Mucherino
Petraq J. Papajorgji
Panos M. Pardalos

Contents

Preface ... vii

List of Figures ... xiii

1 Introduction to Data Mining ... 1
1.1 Why data mining? ... 1
1.2 Data mining techniques ... 3
1.2.1 A brief overview ... 3
1.2.2 Data representation ... 6
1.3 General applications of data mining ... 10
1.3.1 Data mining for studying brain dynamics ... 11
1.3.2 Data mining in telecommunications ... 12
1.3.3 Mining market data ... 13
1.4 Data mining and optimization ... 14
1.4.1 The simulated annealing algorithm ... 17
1.5 Data mining and agriculture ... 19
1.6 General structure of the book ... 20

2 Statistical Based Approaches ... 23
2.1 Principal component analysis ... 23
2.2 Interpolation and regression ... 30
2.3 Applications ... 36
2.3.1 Checking chicken breast quality ... 37
2.3.2 Effects of energy use in agriculture ... 40
2.4 Experiments in MATLAB® ... 40
2.5 Exercises ... 44

3 Clustering by *k*-means ... 47
3.1 The basic *k*-means algorithm ... 47
3.2 Variants of the *k*-means algorithm ... 56
3.3 Vector quantization ... 62

3.4 Fuzzy *c*-means clustering ... 64
3.5 Applications ... 67
3.5.1 Prediction of wine fermentation problem ... 68
3.5.2 Grading method of apples ... 71
3.6 Experiments in MATLAB ... 73
3.7 Exercises ... 80

4 *k*-Nearest Neighbor Classification ... 83
4.1 A simple classification rule ... 83
4.2 Reducing the training set ... 85
4.3 Speeding *k*-NN up ... 88
4.4 Applications ... 89
4.4.1 Climate forecasting ... 91
4.4.2 Estimating soil water parameters ... 93
4.5 Experiments in MATLAB ... 96
4.6 Exercises ... 103

5 Artificial Neural Networks ... 107
5.1 Multilayer perceptron ... 107
5.2 Training a neural network ... 111
5.3 The pruning process ... 113
5.4 Applications ... 114
5.4.1 Pig cough recognition ... 116
5.4.2 Sorting apples by watercore ... 118
5.5 Software for neural networks ... 121
5.6 Exercises ... 122

6 Support Vector Machines ... 123
6.1 Linear classifiers ... 123
6.2 Nonlinear classifiers ... 126
6.3 Noise and outliers ... 129
6.4 Training SVMs ... 130
6.5 Applications ... 131
6.5.1 Recognition of bird species ... 133
6.5.2 Detection of meat and bone meal ... 135
6.6 MATLAB and LIBSVM ... 136
6.7 Exercises ... 139

7 Biclustering ... 143
7.1 Clustering in two dimensions ... 143
7.2 Consistent biclustering ... 148
7.3 Unsupervised and supervised biclustering ... 151
7.4 Applications ... 153
7.4.1 Biclustering microarray data ... 153
7.4.2 Biclustering in agriculture ... 155
7.5 Exercises ... 159

8 Validation 161
8.1 Validating data mining techniques 161
8.2 Test set method 163
8.2.1 An example in MATLAB 163
8.3 Leave-one-out method 166
8.3.1 An example in MATLAB 166
8.4 *k*-fold method 168
8.4.1 An example in MATLAB 170

9 Data Mining in a Parallel Environment 173
9.1 Parallel computing 173
9.2 A simple parallel algorithm 176
9.3 Some data mining techniques in parallel 177
9.3.1 *k*-means 178
9.3.2 *k*-NN 179
9.3.3 ANNs 181
9.3.4 SVMs 182
9.4 Parallel computing and agriculture 184

10 Solutions to Exercises 185
10.1 Problems of Chapter 2 185
10.2 Problems of Chapter 3 191
10.3 Problems of Chapter 4 200
10.4 Problems of Chapter 5 204
10.5 Problems of Chapter 6 211
10.6 Problems of Chapter 7 216

Appendix A: The MATLAB Environment 219
A.1 Basic concepts 219
A.2 Graphic functions 224
A.3 Writing a MATLAB function 228

Appendix B: An Application in C 231
B.1 *h*-means in C 231
B.2 Reading data from a file 238
B.3 An example of main function 241
B.4 Generating random data 244
B.5 Running the applications 247

References 253

Glossary 265

Index 269

List of Figures

1.1 A schematic representation of the classification of the data mining techniques discussed in this book. 5
1.2 The codes that can be used for representing a DNA sequence. 8
1.3 Three representations for protein molecules. From left to right: the full-atom representation of the whole protein, the representation of the atoms of the backbone only, and the representation through the torsion angles Φ and Ψ. 10
1.4 The simulated annealing algorithm. 19

2.1 A possible transformation on aligned points: (a) the points are in their original locations; (b) the points are rotated so that the variability of their y component is zero. 25
2.2 A possible transformation on quasi-aligned points: (a) the points are in their original locations; (b) the points after the transformation. 26
2.3 A transformation on a set of points obtained by applying PCA. The circles indicate the original set of points. 29
2.4 Interpolation of 10 points by a join-the-dots function. 31
2.5 Interpolation of 10 points by the Newton polynomial. 33
2.6 Interpolation of 10 points by a cubic spline. 34
2.7 Linear regression of 10 points on a plane. 35
2.8 Quadratic regression of 10 points on a plane. 36
2.9 Average and standard deviations for all the parameters used for evaluating the chicken breast quality. Data from [156]. 39
2.10 The PCA method applied in MATLAB® to a random set of points lying on the line $y = x$. 41
2.11 The figure generated if the MATLAB instructions in Figure 2.10 are executed. 42
2.12 A sequence of instructions for drawing interpolating functions in MATLAB. 42

2.13 Two figures generated by MATLAB: (a) the instructions in Figure 2.12 are executed; (b) the instructions in Figure 2.14 are executed. . 43
2.14 A sequence of instructions for drawing interpolating and regression functions in MATLAB. 44

3.1 A partition in clusters of a set of points. Points are marked by the same symbol if they belong to the same cluster. The two big circles represent the centers of the two clusters. 49
3.2 The Lloyd's or k-means algorithm. 50
3.3 Two possible partitions in clusters considered by the k-means algorithm. (a) The first partition is randomly generated; (b) the second partition is obtained after one iteration of the algorithm. 51
3.4 Two Voronoi diagrams in two easy cases: (a) the set contains only 2 points; (b) the set contains aligned points. 53
3.5 A simple procedure for drawing a Voronoi diagram. 53
3.6 The Voronoi diagram of a random set of points on a plane. 54
3.7 The k-means algorithm presented in terms of Voronoi diagram. 54
3.8 Two partitions of a set of points in 5 clusters and Voronoi diagrams of the centers of the clusters: (a) clusters and cells differ; (b) clusters and cells provide the same partition. 55
3.9 The h-means algorithm. 56
3.10 The h-means algorithm presented in terms of Voronoi diagram. 57
3.11 (a) A partition in 4 clusters in which one cluster is empty (and therefore there is no cell for representing it); (b) a new cluster is generated as the algorithm in Figure 3.12 describes. 59
3.12 The k-means+ algorithm. 60
3.13 The h-means+ algorithm. 60
3.14 A graphic representation of the compounds considered in datasets A, B, E and F. A and E are related to data measured within the three days that the fermentation started; B and F are related to data measured during the whole fermentation process. 69
3.15 Classification of wine fermentations by using the k-means algorithm with $k = 5$ and by grouping the clusters in 13 groups. In this analysis the dataset A is used. 71
3.16 The MATLAB function `generate`. 74
3.17 Points generated by the MATLAB function `generate`. 74
3.18 The MATLAB function `centers`. 75
3.19 The center (marked by a circle) of the set of points generated by `generate` and computed by `centers`. 76
3.20 The MATLAB function `kmeans`. 77
3.21 The MATLAB function `plotp`. 79
3.22 The partition in clusters obtained by the function `kmeans` and displayed by the function `plotp`. 79

3.23 Different partitions in clusters obtained by the function kmeans. The set of points is generated with different eps values. (a) eps = 0.10, (b) eps = 0.05. 80
3.24 Different partitions in clusters obtained by the function kmeans. The set of points is generated with different eps values. (a) eps = 0.02, (b) eps = 0. 81

4.1 (a) The 1-NN decision rule: the point **?** is assigned to the class on the left; (b) the k-NN decision rule, with $k = 4$: the point **?** is assigned to the class on the left as well. 84
4.2 The k-NN algorithm. 84
4.3 An algorithm for finding a consistent subset T_{CNN} of T_{NN}. 86
4.4 Examples of correct and incorrect classification. 86
4.5 An algorithm for finding a reduced subset T_{RNN} of T_{NN}. 87
4.6 The study area of the application of k-NN presented in [97]. The image is taken from the quoted paper. 90
4.7 The 10 validation sites in Florida and Georgia used to develop the raw climate model forecasts using statistical correction methods. . . . 92
4.8 The 10 target combinations of the outputs of FSU-GSM and FSU-RSM climate models. 92
4.9 Graphical representation of k-NN for finding the "best" match for a target soil. Image from [118]. 95
4.10 The MATLAB function knn. 97
4.11 The training set used with the function knn. 98
4.12 The classification of unknown samples performed by the function knn. 99
4.13 The MATLAB function condense: first part. 100
4.14 The MATLAB function condense: second part. 101
4.15 (a) The original training set; (b) the corresponding condensed subset T_{CNN}o btained by the function condense. 102
4.16 The classification of a random set of points performed by knn. The training set which is actually used is the one in Figure 4.15(b). 103
4.17 The MATLAB function reduce. 104
4.18 (a) The reduced subset T_{RNN} obtained by the function reduce; (b) the classification of points performed by knn using the reduced subset T_{RNN} obtained by the function reduce. 105

5.1 Multilayer perceptron general scheme. 109
5.2 The face and the smile of Mona Lisa recognized by a neural network system. Image from [200]. 115
5.3 A schematic representation of the test procedure for recording the sounds issued by pigs. Image from [45]. 117
5.4 The time signal of a pig cough. Image from [45]. 118
5.5 The confusion matrix for a 4-class multilayer perceptron trained for recognizing pig sounds. 119

5.6 X-ray and classic view of an apple. X-ray can be useful for detecting internal defects without slicing the fruit. 120

6.1 Apples with a short or long stem on a Cartesian system. 124
6.2 (a) Examples of linear classifiers for the apples; (b) the classifier obtained by applying a SVM. 124
6.3 An example in which samples cannot be classified by a linear classifier. 127
6.4 Example of a set of data which is not linearly classifiable in its original space. It becomes such in a two-dimensional space. 128
6.5 Chinese characters recognized by SVMs. Symbols from [63]. 132
6.6 The hooked crow (lat. ab.: cornix) can be recognized by an SVM based on the sounds of birds. 133
6.7 The structure of the SVM decision tree used for recognizing bird species. Image from [71]. 135
6.8 The MATLAB function `generate4libsvm`. 138
6.9 The first rows of file `trainset.txt` generated by `generate4libsvm`. 139
6.10 The DOS commands for training and testing an SVM by SVMLIB. 139

7.1 A microarray. 154
7.2 The partition found in biclusters separating the ALL samples and the AML samples. 156
7.3 Tissues from the HuGE Index set of data. 157
7.4 The partition found in biclusters of the tissues in the HuGE Index set of data. 158

8.1 The test set method for validating a linear regression model. 165
8.2 The test set method for validating a linear regression model. In this case, a validation set different from the one in Figure 8.1 is used.166
8.3 The leave-one-out method for validation. (a) The point (`x(1)`,`y(1)`) is left out; (b) the point (`x(4)`,`y(4)`) is left out. 168
8.4 The leave-one-out method for validation. (a) The point (`x(7)`,`y(7)`) is left out; (b) the point (`x(10)`,`y(10)`) is left out. 169
8.5 A set of points partitioned in two classes. 171
8.6 The results obtained applying the *k*-fold method. (a) Half set is considered as a training set and the other half as a validation set; (b) training and validation sets are inverted. 172

9.1 A graphic scheme of the MIMD computers with distributed and shared memory. 174
9.2 A parallel algorithm for computing the minimum distance between one sample and a set of samples in parallel. 178
9.3 A parallel algorithm for computing the centers of clusters in parallel.179
9.4 A parallel version of the *h*-means algorithm. 180
9.5 A parallel version of the *k*-NN algorithm. 180

9.6 A parallel version of the training phase of a neural network. 182
9.7 The tree scheme used in the parallel training of a SVM. 183
9.8 A parallel version of the training phase of a SVM. 183

10.1 A set of points before and after the application of the principal component analysis. 186
10.2 The line which is the solution of Exercise 4. 187
10.3 The solution of Exercise 7. 189
10.4 The solution of Exercise 8. 190
10.5 The solution of Exercise 9. 190
10.6 The set of points of Exercise 1 plotted with the MATLAB function `plotp`. Note that 3 of these points lie on the x or y axis of the Cartesian system. 198
10.7 The training set and the unknown point that represents a possible solution to Exercise 4. 202
10.8 A random set of 200 points partitioned in two clusters. 204
10.9 The condensed and reduced set obtained in Exercise 7: (a) the condensed set corresponding to the set in Figure 10.8; (b) the reduced set corresponding to the set in Figure 10.8. 205
10.10 The classification of a random set of points by using a training set of 200 points. 206
10.11 The classification of a random set of points by using (a) the condensed set of the set in Figure 10.8; (b) the reduced set of the set in Figure 10.8. 207
10.12 The structure of the network considered in Exercise 1. 208
10.13 The structure of the network considered in Exercise 3. 209
10.14 The structure of the network considered in Exercise 7. 211
10.15 The structure of the network required in Exercise 8. 212
10.16 The classes C^+ and C^- in Exercise 3. 213

A.1 Points drawn by the MATLAB function `plot`. 225
A.2 The sine and cosine functions drawn with MATLAB. 227
A.3 The function `fun`. 228
A.4 The graphic of the MATLAB function `fun`. 229

B.1 The function `hmeans`. 232
B.2 The prototypes of the functions called by `hmeans`. 234
B.3 The function `rand_clust`. 235
B.4 The function `compute_centers`. 236
B.5 The function `find_closest`. 237
B.6 The function `isStable`. 237
B.7 The function `copy_centers`. 238
B.8 An example of input text file. 239
B.9 The function `dimfile`. 239
B.10 The function `readfile`. 241

B.11 The function `main`. 242
B.12 The function `main` of the application for generating random sets of data. Part 1. 246
B.13 The function `main` of the application for generating random sets of data. Part 2. 247
B.14 An example of input text file for the application `hmeans`. 248
B.15 The output file provided by the application `hmeans` when the input is the file in Figure B.14 and $k = 2$. 248
B.16 An output file containing a set of data generated by the application `generate`. 249
B.17 The partition provided by the application `generate` (column A), the partition found by `hmeans` (column B) and the components of the samples (following columns) in an Excel spreadsheet. 250