Mining Relational and Nonrelational Data with IBM Intelligent Miner for Data
Using Oracle, SPSS, and SAS As Sample Data Sources

December 1998
First Edition (December 1998)

This edition applies to Version 2.1.2 of IBM Intelligent Miner for Data for use with the AIX V 4.3.1 Windows NT operating systems and to Version 2.1.2 of IBM DataJoiner for use with the AIX or Windows NT operating systems.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. QXXE Building 80-E2
650 Harry Road
San Jose, California 95120-6099

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.
# Contents

## Figures ................................................................. vii

## Preface ............................................................... xi
How This Book Is Organized ........................................... xi
The Team That Wrote This Redbook ................................ xii
Comments Welcome .................................................... xiii

## Part 1. Introduction to Data Mining and the Products Used ........................................ 1

### Chapter 1. An Analytical Approach to Data Mining .............................................. 3
1.1 The Challenge of the Information Age .................................................. 3
1.2 Data Mining ............................................................... 3
1.3 Data Mining Tools ........................................................ 4
1.4 From Data to Knowledge ..................................................... 6
  1.4.1 Setting Goals ......................................................... 7
  1.4.2 Data Collection ...................................................... 8
  1.4.3 Sampling ............................................................. 8
  1.4.4 Data Preparation ................................................... 8
  1.4.5 Dimension Reduction .............................................. 9
  1.4.6 Data Modeling ...................................................... 10
  1.4.7 Decision Making ................................................... 10
  1.4.8 Measuring the Results ............................................ 11

### Chapter 2. Introduction to Intelligent Miner for Data and DataJoiner .................... 13
2.1 Intelligent Miner ......................................................... 13
  2.1.1 Overview of the Intelligent Miner .................................. 13
  2.1.2 Working with Databases .......................................... 13
  2.1.3 The User Interface .................................................. 14
  2.1.4 Data Preparation Functions ....................................... 15
  2.1.5 Statistical and Mining Functions ................................ 16
  2.1.6 Processing IM Functions ......................................... 17
  2.1.7 Creating and Visualizing the Results ............................. 17
  2.1.8 Creating Data Mining Operations ................................ 18

2.2 DataJoiner ............................................................... 19
  2.2.1 The Multidatabase Solution ....................................... 20
  2.2.2 Heterogeneous Data Access ....................................... 20
  2.2.3 Database Integration .............................................. 22
  2.2.4 Heterogeneous Replication ....................................... 24
  2.2.5 Global Query Optimization ....................................... 25
  2.2.6 Web Support ....................................................... 26
Part 2. Accessing Relational Data

Chapter 3. System Architecture for Mining Oracle Databases

3.1 The Oracle Database Environment
3.2 Mining Oracle Databases Using IBM's Intelligent Miner for Data
   3.2.1 Using Fast Extract with Oracle Databases
   3.2.2 The DataJoiner Interface to Oracle Databases
3.3 System Architecture

Chapter 4. Install and Configure DataJoiner with Oracle

4.1 Product Installation on AIX
   4.1.1 Install Oracle Client
   4.1.2 Connecting to the Oracle Server from the Oracle Client
   4.1.3 Install DataJoiner
4.2 DataJoiner Configuration on AIX
   4.2.1 Configuring DataJoiner to Access Oracle
   4.2.2 Creating a DataJoiner Instance
   4.2.3 Configuring DataJoiner to Accept DB2 Client Connections
   4.2.4 Configuring DataJoiner Nicknames
4.3 Test Access on AIX
4.4 Product Installation on Windows NT
   4.4.1 Install Oracle Client
   4.4.2 Connecting to the Oracle Server from the Oracle Client
   4.4.3 Install DataJoiner
4.5 DataJoiner Configuration on Windows NT
   4.5.1 Configuring DataJoiner to Access Oracle
4.6 Test Access on Windows NT

Chapter 5. Fast Extract Utility

5.1 Installation
5.2 Use

Chapter 6. Sample Mining of Oracle Data

6.1 Intelligent Miner for Data Installation on AIX
6.2 The Mining Scenario
6.3 The Mining Solution
6.4 Using Intelligent Miner to Implement the Solution
   6.4.1 Creating a Data Object for the CUSTOMER Table
   6.4.2 Setting Up Demographic Clustering
   6.4.3 Applying the Model
   6.4.4 Generating the Target Customer Set
# Figures

1. Knowledge Discovery and Data Mining ........................................ 7
2. Intelligent Miner Main Window .............................................. 14
3. Sample Clustering Visualization ............................................. 18
4. Sequence - Settings Window ................................................ 19
5. IBM Business Intelligence Architecture .................................. 22
6. IBM Object-Relational Architecture ....................................... 24
7. IBM Information Warehouse Architecture .................................. 25
8. The Oracle Database Environment .......................................... 30
9. Using Flat Files to Mine Oracle Databases ................................ 33
10. Mining Oracle Databases through DB2 ..................................... 33
11. Accessing Oracle Tables with DataJoiner ................................. 35
12. Architecture for Mining Oracle DB with DataJoiner and IM .......... 36
13. Data Mining Environment for Oracle - UNIX ............................. 36
14. Data Mining Environment for Oracle - Windows NT .................. 37
15. Oracle Installation Settings ................................................ 60
16. Installation Options ......................................................... 60
17. Client Configuration ......................................................... 61
18. Installation Completed ....................................................... 61
19. SQL*Plus Logon ............................................................... 62
20. SQL*Plus Window ............................................................. 62
21. NT User Manager ............................................................ 63
22. New User ......................................................................... 63
23. Group Memberships ............................................................ 64
24. NT User Manager: DataJoiner Administrator ............................ 64
25. DataJoiner for Window NT Installation .................................... 65
26. Windows NT Services .......................................................... 66
27. Service Startup Options ...................................................... 66
28. Oracle Server Mapping ....................................................... 67
29. Testing the Oracle Connection ............................................... 68
30. Test SELECT Results .......................................................... 68
31. Intelligent Miner Client Main Screen ...................................... 79
32. Server Selection Screen in Preferences Window ....................... 79
33. Data Format and Settings Screen for Building Model ................ 80
34. Database Table or View Screen for Building Model .................. 81
35. Field Parameters Screen for Building Model ............................ 81
36. Saving Your Mining Base ..................................................... 83
37. Mining Functions and Settings Screen for Building Model .......... 83
38. Input Data Screen for Building Model ...................................... 84
39. Expression Builder Screen for Building Model ......................... 85
40. Input Fields Screen for Building Model .................................... 86
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>41. Results Screen for Building Model</td>
<td>87</td>
</tr>
<tr>
<td>42. Progress Screen for Building Model</td>
<td>87</td>
</tr>
<tr>
<td>43. Generated Demographic Clustering Model</td>
<td>88</td>
</tr>
<tr>
<td>44. Variable Distributions in Cluster 1</td>
<td>89</td>
</tr>
<tr>
<td>45. Distribution for the Gender Variable in Cluster 1</td>
<td>89</td>
</tr>
<tr>
<td>46. Mining Functions and Settings Screen for Applying Model</td>
<td>90</td>
</tr>
<tr>
<td>47. Input Data Screen for Applying Model</td>
<td>91</td>
</tr>
<tr>
<td>48. Expression Builder Screen for Applying Model</td>
<td>92</td>
</tr>
<tr>
<td>49. Mode Parameters Screen for Applying Model</td>
<td>92</td>
</tr>
<tr>
<td>50. Input Fields Screen for Applying Model</td>
<td>93</td>
</tr>
<tr>
<td>51. Output Fields Screen for Applying Model</td>
<td>94</td>
</tr>
<tr>
<td>52. Data Format and Settings Screen for Applying Model</td>
<td>94</td>
</tr>
<tr>
<td>53. Database Table or View Screen for Applying Model</td>
<td>96</td>
</tr>
<tr>
<td>54. Progress Screen for Applying Model</td>
<td>96</td>
</tr>
<tr>
<td>55. Statistics Functions and Settings Screen</td>
<td>97</td>
</tr>
<tr>
<td>56. Input Data Screen for Statistical Analysis</td>
<td>98</td>
</tr>
<tr>
<td>57. Expression Builder for Statistical Analysis</td>
<td>99</td>
</tr>
<tr>
<td>58. Statistics Screen</td>
<td>100</td>
</tr>
<tr>
<td>59. Output Fields Screen for Statistical Analysis</td>
<td>100</td>
</tr>
<tr>
<td>60. Data Format and Settings Screen for Output of Statistical Analysis</td>
<td>101</td>
</tr>
<tr>
<td>61. Database Table or View Screen for Output of Statistical Analysis</td>
<td>102</td>
</tr>
<tr>
<td>62. Output Data Screen for Statistical Analysis</td>
<td>102</td>
</tr>
<tr>
<td>63. Results Screen for Statistical Analysis</td>
<td>103</td>
</tr>
<tr>
<td>64. Progress Screen for Statistical Analysis</td>
<td>104</td>
</tr>
<tr>
<td>65. Variable Distributions for Target Customers</td>
<td>104</td>
</tr>
<tr>
<td>66. ODBC Functional Components</td>
<td>110</td>
</tr>
<tr>
<td>67. Using Flat Files to Mine ODBC Data Sources</td>
<td>111</td>
</tr>
<tr>
<td>68. Mining ODBC Data Sources through DB2</td>
<td>112</td>
</tr>
<tr>
<td>69. Accessing ODBC Data Sources with DataJoiner</td>
<td>113</td>
</tr>
<tr>
<td>70. The ODBC Environment</td>
<td>114</td>
</tr>
<tr>
<td>71. ODBC Data Source Administrator</td>
<td>115</td>
</tr>
<tr>
<td>72. Create New Data Source</td>
<td>116</td>
</tr>
<tr>
<td>73. SPSS Data Source Setup</td>
<td>116</td>
</tr>
<tr>
<td>74. SPSS System Data Source</td>
<td>117</td>
</tr>
<tr>
<td>75. Creating a DB2 Instance</td>
<td>117</td>
</tr>
<tr>
<td>76. Connecting to the Database</td>
<td>118</td>
</tr>
<tr>
<td>77. Testing the SPSS ODBC Connection</td>
<td>119</td>
</tr>
<tr>
<td>78. SAS Universal ODBC Setup</td>
<td>122</td>
</tr>
<tr>
<td>79. SAS ODBC Driver Configuration - Genera</td>
<td>122</td>
</tr>
<tr>
<td>80. SAS ODBC - Servers</td>
<td>123</td>
</tr>
<tr>
<td>81. Local DDE Options</td>
<td>123</td>
</tr>
<tr>
<td>82. SAS ODBC - Libraries</td>
<td>124</td>
</tr>
<tr>
<td>83. SAS System Data Sources</td>
<td>124</td>
</tr>
</tbody>
</table>
84. Trial Driver Expiration ............................................. 126
85. NT DDESHARE ...................................................... 127
86. DDE Share Properties ............................................. 128
87. DDE Share Name Permissions ..................................... 128
88. DDE Shares .......................................................... 129
89. Trusted Share Properties .......................................... 129
90. IM Install: Welcome ............................................... 132
91. IM Install: Components ........................................... 132
92. IM Install: Domain/User .......................................... 133
93. Start IM Server ..................................................... 134
94. IM Preferences ...................................................... 134
95. Intelligent Miner: Main Window ................................. 136
96. Data TaskGuide Data Format and Settings ..................... 137
97. Data TaskGuide Field Parameters ............................... 137
98. Data TaskGuide Summary ......................................... 138
99. BivariateStatistics ................................................. 139
100. Frequency Charts .................................................. 140
101. Transform Variable ................................................ 140
102. Visualizing the Transformation ................................... 141
103. Setting the Prediction Object .................................... 142
104. The Result Object .................................................. 143
105. Neural Prediction Settings, Mode Parameters .................. 144
106. Neural Prediction Settings, Output Fields ....................... 144
107. Sequence TaskGuide ............................................... 145
108. SAS Loan Application Data Set .................................... 147
109. SAS Loan Application Variables ................................... 147
110. Data Visualization: Frequency Charts ......................... 148
111. Encode Missing Values ............................................ 149
112. Calculate Values .................................................... 150
113. Value Set Filtering .................................................. 151
114. Logarithmic Variable Definition .................................. 152
115. Normalized Frequency Chart ...................................... 152
116. Neural Classification Results ..................................... 153
117. Tree Classification Confusion Matrix .............................. 154
118. Decision Tree ....................................................... 154
119. Test Data Classification Comparison ............................. 155
120. Sequence Settings ................................................... 156
121. Mining Relational Data Sources ................................... 159
122. Mining Nonrelational Data Sources ............................... 161
123. Linear Discriminant Analysis ..................................... 186
124. Complexity of Decision Boundaries ............................... 187
125. Multilayer Neural Network ........................................ 191
126. Radial-Basis Function Network ................................... 193
Mining Relational and Nonrelational Data with IM for Data
IBM Intelligent Miner for Data enables you to mine structured data stored in conventional databases or flat files. Its mining algorithms have been successfully used by customers and Business Partners alike, to address business problems in such areas as customer relationship marketing and fraud and abuse detection. Using Intelligent Miner you can increasingly leverage the data warehouse and more quickly derive business value from that investment.

This redbook will help you to install, tailor, and configure the IBM Intelligent Miner for Data and IBM DataJoiner products to mine any non-IBM data sources. Oracle 8 and two nonrelational Open Database Connectivity (ODBC) data sources--SPSS and SAS--are used to demonstrate DataJoiner acting as a heterogeneous data server for Intelligent Miner in the AIX and Windows NT environments.

How This Book Is Organized

Chapter 1 contains an introduction to data mining principles and concepts, and a summary of our conclusions. You will this chapter useful for developing an understanding of the basic ideas and terminology of data mining.

Chapter 2 describes using IBM DataJoiner to access data sources from Intelligent Miner for Data and presents an overview of the Intelligent Miner for Data.

Chapter 3 provides an overview of a typical Oracle database environment and describes the architecture used in this book to access an Oracle data source from Intelligent Miner for Data.

Chapter 4 describes the setup for accessing an Oracle data source from IBM DataJoiner, covering all steps from installation to configuration.

Chapter 5 describes another access method for using Oracle data from Intelligent Miner for Data.

Chapter 6 presents a sample mining process from Intelligent Miner for Data using the Oracle data source access described in Chapters 3 and 4.

Chapter 7 contains a short description of an Open Database Connectivity (ODBC) database environment.
Chapter 8 describes the setup necessary to access SPSS data from Intelligent Miner for Data through IBM DataJoiner. It covers the ODBC and DataJoiner installation and configuration.

Chapter 9 describes the setup necessary to access SAS as an ODBC data source.

Chapter 10 presents a sample mining process using the SPSS and SAS data made accessible through the setup described in Chapters 8 and 9.

Chapter 11 summarizes the different access methods and data sources described in this book. It covers the main differences among the various data sources and access methods.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Joerg Reinschmidt is a Data Mining Specialist at the International Technical Support Organization, San Jose Center. He writes extensively and teaches IBM classes worldwide on all areas of DB2 Universal Database and Internet access to legacy data. Before joining the ITSO in 1998, Joerg worked in the Solution Partnership Center in Germany as a DB2 Specialist, supporting independent software vendors (ISVs) to port their applications to use IBM data management products.

Rahul Bhattacharya is an IT Specialist with IBM Global Services in India. He has five years of experience in the field of UNIX and databases. He holds a degree in Electrical and Electronics Engineering from BITS, Pilani. His areas of expertise include UNIX clustering and business intelligence.

Paul Harris is a project manager in Sydney, Australia. He has 13 years of experience in the decision support field and has worked on many development projects for IBM clients in the banking and insurance sectors.

Athanasios Karanasos is a data management specialist in IBM UK. He studied mathematics at Aristotle University in Thessaloniki, Greece, and computational intelligence at the University of Essex, Colchester, UK. His areas of expertise include knowledge discovery in large databases and data mining.
Thanks to the following people for their invaluable contributions to this project:

Ute Baumbach
IBM Germany

Gary Chan
IBM Santa Teresa Laboratory

David Farb
SPSS Inc., Chicago, IL

Christoph Lingenfelder
IBM Germany

Susan Monje
IBM Santa Teresa Laboratory

Norbert Friedrich Poch
IBM Germany

Marc Romano
IBM Santa Teresa Laboratory

Werner Staub
IBM Germany

---

Comments Welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in "ITSO Redbook Evaluation" on page 221 to the fax number shown on the form.
- Use the electronic evaluation form found on the Redbooks Web sites:
  - For Internet users: http://www.redbooks.ibm.com
  - For IBM Intranet users: http://w3.itso.ibm.com
- Send us a note at the following address:

  redbook@us.ibm.com
Part 1. Introduction to Data Mining and the Products Used
Chapter 1. An Analytical Approach to Data Mining

Data mining has become one of the most popular techniques for building intelligent decision support systems. Borrowing tools and methods used in statistical mathematics and machine learning, data mining has proven to be a valuable tool in the hands of decision makers.

In this chapter we investigate the basic principals of data mining. We review the steps that a decision maker must make to derive knowledge from data, and the data mining tools that he or she can use for more productive and robust data mining operations. We discuss data mining in the context of knowledge discovery, providing a framework for modern decision support systems.

1.1 The Challenge of the Information Age

In a business world characterized by intense competition and globalization, information has become a valuable commodity. On the basis of information, companies can make strategic decisions and keep themselves ahead of the competition. However, the road from enterprise data to information, and finally to the anticipation of business events, is not an easy one.

Over the last 20 years computers have been used to capture detailed transaction data across a range of corporate enterprises. Retail sales, telecommunications, banking, and credit card operations are examples of transaction-intensive industries. These transactional systems were designed to capture detailed information about every aspect of business. In the process they have brought “data overload” to host computers. In addition, the use of personal computers and departmental databases isolated potentially useful data where there was need for consolidation of database records tending towards a single customer view. The need for database consolidation gave birth to the data warehousing concept and massive centralized data repositories, increasing the need for more powerful data exploitation tools.

1.2 Data Mining

To meet this modern business requirement, a new discipline in data management called data mining was created. We can loosely define data mining as the process of automatically extracting valid, useful, previously unknown, and ultimately comprehensible information from large databases and using it to make crucial business decisions.
Data mining uses a discovery-based approach in which classification, prediction, cluster formation, and association functions are employed to determine the key relationships and trends in the data. Data mining algorithms can look at numerous multidimensional data relationships concurrently, highlighting those that are dominant or exceptional, and transforming data into business knowledge.

In this way data mining differs from other data analysis methods. Multidimensional analysis, relational online analytical processing (OLAP), and ad hoc query support a verification-based approach in which the user hypothesizes about specific data interrelationships and then uses these tools to verify or refute those hypotheses. Since an external agent, usually a human analyst, is responsible for generating hypothesis tests, the effectiveness of this verification-based analysis is limited by a number of factors. We rely on the experience of the analyst to pose appropriate questions and quickly return results, to manage a usually large number of attributes required in the analysis, and deal with the complexity of the attribute space.

1.3 Data Mining Tools

Data mining draws on theories and techniques used in the fields of machine learning, databases, statistical mathematics, and arithmetic analysis. The underlying basis for those fields is the extraction of knowledge or patterns of information from data in the context of large databases.

Tools that have been used successfully in related research areas have been modified to meet the special requirements of data mining. Data mining requires a data model that is mostly data driven. The data model should demonstrate robustness even under heavy tailed distributions and/or the existence of high levels of noise in the data. It should provide means for building a representation of the observed data that has predictive validity, that not only detect patterns hidden in the available historical data (memorization) but also alerts user variation in the case of presenting previously unseen stimuli to the data model (generalization).

Variable selection, clustering, classification, association rule formation, and prediction are some of the most popular data mining functions. Each one of them can be performed using a range of tools, most of them outlined in Appendix B, “Data Mining Functions” on page 185. Data mining tools must be selected according to our requirements. Therefore, knowing the advantages, disadvantages, and limitations of each tool is essential for performing a successful data mining operation.
Data mining operates on large data volumes and “real-world” probability distributions. It also operates in a time and computationally constrained environment. This requires scalability and statistically robust performance from a proposed tool.

Statistical robustness is influenced by assumptions made before the construction of the data representation model. In linear regression, for instance, we assume that the predictors are totally independent. If one of the predictors is correlated to one or more of the others, we are facing large coefficient errors, and in a strong dependency even problems computing the actual value of the coefficients. Although stepwise regression methods have been developed, based on a T-test, assumptions concerning the distribution of the residuals and the dependent variable are still made. This will require the analyst to introduce normality to the input space by power transformations and plot the predicted value against the standardized residuals checking for uniform distribution.

Besides the assumptions that we have to make about the nature of our data another factor that influences the robustness of our model is the quality of the historical data used in the data mining operation. Observations with missing values are a common phenomenon in real-world data sets. Some data mining tools are more sensitive to missing values than others. For instance linear regression, discriminant analysis, and artificial neural networks (ANNs) do not include any mechanism for imputing missing values. However, decision trees can use surrogate rules, available at nonterminal nodes, if the splitting attribute is unobserved.

Another question that we have to address is how complex our data model must be. ANNs and nonlinear regression are powerful tools that can approximate any nonlinear function. Linear regression and analysis of variance account only for linear relationships between the variables of interest. However, linear models have indeed a different behavior under noise in the data set. If we are facing a data set with a significant degree of white noise, then a simple model like a regression line will have better generalization abilities. On the other hand a multilayer perceptron (MLP) will tend to fit any noise, making the model though better in memorization, worse in generalization. Facing the “bias/variance” problem data mining tools require the least degree of susceptibility to noise as possible, keeping their nonlinear properties at the same time. Methods in ANNs and decision trees have been developed in this direction, namely, penalty functions for ANNs that penalize unnecessary model complexity, and pruning methods for decision trees that reduce the complexity of decision rules. However, these methods do not guarantee a better model, leaving the researcher responsible for choosing the right tool.
Data mining tools can be seen from yet another point of view. Some of them are parametric models like the ANNs or regression, whereas others, like the decision trees, are nonparametric. Parametric models require a method for finding optimal parameters for the model. To perform parameter optimization we can use statistical methods or numerical analysis methods. Statistical methods are sensitive to assumptions. On the other hand parameter optimization methods based on numerical analysis are influenced by local mining in the error space. In other words there is no guarantee that an error function used in a learning algorithm to find optimal weights for a ANN will reach global minima. The error function can be trapped to a local minima even with a strong momentum. This ill behavior requires the analyst or the data mining algorithm to alter the circuit of the ANN, usually the number of hidden neurons, in order to perform error descent in a more simple space. Generally speaking no learning algorithm for ANN is proven to be NP-complete. Therefore we can not be sure that a solution to a problem will be found in a time which is polynomial in size of the problem and the accuracy required. This makes ANNs lag behind decision trees in a time constrained environment.

1.4 From Data to Knowledge

Until now we have briefly examined tools performing some data mining function. In this section we present a knowledge discovery methodology based on data mining. Knowledge discovery starts with data collection, data preparation, and dimension reductions. Then, through data mining functions, it delivers a decision support system. Figure 1 on page 7 illustrates this process.
1.4.1 Setting Goals

Clearly the goal of data mining is to provide the foundation for a robust and comprehensive decision support system (DSS). When we develop a DSS, as in any IT project, we need to clearly define the requirement and deliverable of the system at the beginning. Thus the questions that we will address using
data mining have to be managed to meet the requirement of the underlying DSS and, ultimately, the general strategy of the company.

1.4.2 Data Collection

After defining the requirements and deliverables of our data mining operation we need to collect the necessary historical data. The desired database tables may be hosted in one or more databases. For instance, marketing databases contain customer purchasing data, demographic data, and life-style data. An analyst in a marketing department needs to combine customer purchase data with demographic data in order to produce a mailing list for a direct marketing campaign.

1.4.3 Sampling

In most of data mining operations the volume of data sets that we mine is extremely large. Data mining tools designed to cope with this fundamental requirement are taking advantage of parallel computing to increase computational power and data mining databases to optimize passes over the available data.

However, there are some things that an analyst can do to reduce the time spent mining large data sets. The most common task toward this direction is sampling. In data sampling we extract a subset out of the original data for further analysis. This subset is usually considerably smaller than the original data set. The exact ratio between sample and original data depends on our domain knowledge and the type of the data mining function that we want to perform. The way that we draw our sample is usually random, stratified or N samples, without replacement. Again sampling methods are determined by the properties of the original data set. For prediction we usually use random sampling, and for classification, in order to preserve the presence of examples belonging to each class, we use stratified sampling.

1.4.4 Data Preparation

The preprocessing stage usually starts with the normalization of the input space. In real-world data sets variables may have values that differ significantly. This phenomenon may influence the way that a data mining tool will assign relative importance to these variables in determining the required output. For example, taking the Euclidean norm to compute the distance between two observations in cluster formation, the distance will be infused much more by variables with values several orders of magnitude larger than others. This will require a rescaling of the form:
where \( i = 1, 2,..., n \) labels the available examples, \( \bar{x} \) is the mean, and \( \sigma^2 \) is the variance of the examples.

In other situations a nonlinear transformation may be required to introduce normality in the data set. The above linear transformation will resolve to a transformed data set with mean zero and unit standard deviation. However, we can perform a power transformation of the form \( x' = \log(x - a) \). This nonlinear transformation will probably give a more “bell” shaped frequency chart of the underlying variable.

Missing data is a usual phenomenon when we mine real-world data sets. There are methods that we can use in order to remedy such a deficiency before the data is used for modeling. We can discard examples with missing values or impute missing fields using our domain knowledge or various heuristics. Which one to use depends on the quantity of the available examples. If the data set is sufficiently large and the number of examples with missing values considerably small, then we can discard those examples from the data set. However, this simple solution cannot be followed if all of the available examples are important to future modeling. In that case missing values can be imputed by a sample mean or median of the available values. We can also use clustering methods replacing examples with missing values with the representative values of the cluster that they belong to. More sophisticated and computational expensive methods suggest the use of regression and neural networks to estimate missing values. In this approach missing values are represented as outputs of a neural network with inputs of nonmissing variables or regression functions over the other variables.

### 1.4.5 Dimension Reduction

It is recognized that the effective dimensionality of the data is less than the apparent dimensionality since variables are correlated to each other. Data mining algorithms that work on an input space with a dimension close to intrinsic dimensionality are more successful in terms of both generalization abilities and time performance. The most simple approach to dimension reduction is feature selection. In feature selection only a subset of the original input variables is selected to take part in the data modeling. The subset is selected according to some selection criterion that judges whether one subset is better than another.
A more generic approach to dimension reduction is the principal component analysis. Using principal components instead of the actual variables may resolve to a smaller dimensionality of the input space, but again we are limited by the linear abilities of the actual technique. We may therefore not be able to capture nonlinear correlations and consequently overestimate the true dimension of the data. Feed-forward multilayer networks can be used to perform nonlinear dimensionality reduction, overcoming some of the limitations of the principal component analysis. An ANN with \( n \) input nodes, a hidden layer with \( m < n \) nodes, and \( n \) output nodes can be used. This auto-associative ANN will map the \( n \)-dimension input space to an \( m \)-dimension space. Then the output layer of the network can be removed and the remaining input and hidden layers can be used as input layers to another ANN that performs the data mining operation.

### 1.4.6 Data Modeling

After preparing our data and performing a data mining function we reach for a solution represented by our data model. On this point there is a debate regarding the validity of the final product. Though methods for evaluating data models exist (validation, cross validation), there is always a degree of skepticism over the performance of the model in an online situation operating on new data.

A solution to this problem is always to have more than one data model of the same data set. This will allow us to increase the confidence level of a certain business decision. Toward this direction a comity of models can be constructed. A comity can be a set of models coming from the same data mining tool (homogenic) or a collection of models coming from different data mining tools (heterogenic). Homogenic comity of data models can be used to take the average over the predictions generated or assign a pattern to the class with the larger score. Heterogenic comities can be used to compare models and through that different data mining approaches to the same problem. Lift charts and confusion tables can be used to compare results from different data models. The model with the higher lift or the one that gives less positive or negative error in the confusion table will be a desirable candidate.

### 1.4.7 Decision Making

Until now we have described how we can extract information from large volumes of data. We need to take this one step further and convert the information generated through data mining to knowledge and eventually business actions. Having passed through the first cycle of data mining iterations, discovering the optimal data model(s), we need to put the selected data model(s) online, delivering a decision support system. Thus, the
decision support system will encapsulate any hidden information in the company’s databases. Decision makers can then take advantage of the machine’s intelligence and together with their own experience reach more effective and revolutionary ideas.

1.4.8 Measuring the Results

After business decisions have been taken and applied in the marketplace we need to measure the results of the data mining effectiveness. This will require monitoring factors of success like sales or return on investment (ROI) through time. In addition trends must be identified as soon as possible with a high degree of accuracy for adjusting the data mining strategy.

Multidimensional analysis can be used successfully for this matter since it can time stamp data volumes, provided that one dimension of the analysis represents time. For instance, feeding new data into an OLAP engine that captures the response of the market to a marketing campaign, based on data mining we can isolate populations with negative response. Then we can reapply data mining to that specific population in order to understand why the original campaign did not positively affect them.
Chapter 2. Introduction to Intelligent Miner for Data and DataJoiner

This chapter presents a high-level, functional overview of the IBM Intelligent Miner for Data and IBM DataJoiner products.

2.1 Intelligent Miner

In Chapter 1, “An Analytical Approach to Data Mining” on page 3 we defined data mining as the process of automatically extracting valid, useful, previously unknown, and ultimately comprehensible information from large databases and using it to make crucial business decisions.

The Intelligent Miner (IM) helps us perform data mining tasks. Through an intuitive graphical user interface (GUI) you can visually design data mining operations. You can choose tools and customize them to meet your requirements. The available tools cover the whole spectrum of data mining functions. In addition IM selects data, explores it, transforms it, and visually interprets the results for productive and efficient knowledge discovery.

2.1.1 Overview of the Intelligent Miner

The IM is based on a client-server architecture. The server executes mining and processing functions and can host historical data and mining results. The client is powered with administrative and visualization tools and can be used to visually build a data mining operation, execute it on the server, and have the results returned for visualization and further analysis. In addition the IM application programing interface (API) provides C++ classes and methods as well as C structures and functions for application programers.

2.1.2 Working with Databases

Most functions of the IM can use input data from either flat files or database tables. If you want to access DB2 tables, you only require authorization to access that database and the permission to query the appropriate tables. If you want to access files in Oracle, SPSS, or SAS format, DataJoiner may be installed as a middleware product.

There are two ways to access database management systems:

- Through the Intelligent Miner server without installing any database software in the client
- Connect directly from the client to the table letting the IM client access the database without using the IM communication interface.
2.1.3 The User Interface

The IM helps businesses reduce the cost of data mining and maximize the ROI, providing an administrative user interface based on Java. The IM user interface is simple and intuitive and provides consistency across all operations. The interface’s state-of-the-art GUI facilities include online help, task guides, and a graphical representation of the mining operations and its functions.

The main window of the IM GUI is divided into three areas (see Figure 2 on page 14):

- Mining base container. A tree view of object folders containing knowledge discovery tools.
- A contents container. An area for customized objects.
- A workarea. An area where customized objects from the contents container can be imported and assigned to a single folder (Mining base).

![Intelligent Miner Main Window](image)

*Figure 2. Intelligent Miner Main Window*
2.1.4 Data Preparation Functions

Once the desired database tables have been selected and the data to be mined has been identified, it is usually necessary to perform certain transformations on the data. IM provides a wide range of data preparation functions which help to optimize the performance of the data mining functions. Depending on the data mining technique, you can select, sample, aggregate, filter, cleanse, and/or transform data in preparation for mining. The data preparation functions are:

- Aggregate values
- Calculate values
- Clean up data sources
- Convert to lower or upper case
- Copy records to file
- Discard records with missing values
- Discretize into quantiles
- Discretize using ranges
- Encode missing values
- Encode nonvalid values
- Filter fields
- Filter records
- Filter records using a value set
- Get random sample
- Group records
- Join data sources
- Map values
- Pivot fields to records
- Run SQL statements

Data preparation functions are performed through the GUI, reducing the time and complexity of data mining operations. You can transform variables, impute missing values, and create new fields through the touch of a button. This automation of the most typical data preparation tasks is aimed at improving your productivity by eliminating the need for programming specialized routines.
2.1.5 Statistical and Mining Functions

After transforming the data we use one or more data mining functions in order to extract the desired type of information. IM provides both statistical analysis tools and state of the art machine learning algorithms for successful data mining.

Statistical functions facilitate the analysis and preparation of data, as well as providing forecasting capabilities. For example, you can apply statistical functions like regression to understand hidden relationships in the data or use factor analysis to reduce the number of input variables. Statistical functions include:

- Factor analysis
- Linear regression
- Principal component analysis
- Univariate curve fitting
- Univariate and bivariate statistics

Based on IBM research, validated through real-world applications, IM has incorporated a number of data mining algorithms as the critical suite to address a wide range of business problems. The algorithms are categorized as follows:

Association discovery: Given a collection of items and a set of records, each of which contains some number of items from the given collection, an association discovery function is an operation against this set of records which returns affinities that exist.

Sequential pattern discovery: A sequence discovery function will analyze collections of related records and will detect frequently occurring patterns of products bought over time among the collection of items.

Clustering: Clustering is used to segment a database into subsets, the clusters, with the members of each cluster sharing a number of interesting properties.

Classification: Classification is the process of automatically creating a model of classes from a set of records. The induced model consists of patterns, essentially generalizations over the records, that are useful for distinguishing the classes. Once a model is induced it can be used to automatically predict the class of other unclassified records.

Value prediction: As in classification the goal is to build a data model as a generalization over the records. However, the difference is that the target is not a class membership but an actual value.
Similar time sequences: The purpose of this process is to discover all occurrences of similar subsequences in a database of time sequences.

Clustering, classification, and value prediction can be covered by a number of different algorithms. For instance, you can perform clustering by using either the demographic or the neural network algorithm, depending on the properties of the input data set and the requirements of the data mining operation.

2.1.6 Processing IM Functions

All IM functions can be customized using two levels of expertise. Users who are not experts can accept the defaults and suppress advanced settings. However, experienced users who want to fine-tune their application have the ability to customize all settings according to their requirements.

You can additionally define the mode of your statistical and mining functions. Possible modes are:

- Training mode. In training mode a mining function builds a model based on the selected input data.
- Clustering mode. In clustering mode, the clustering functions build a model based on the selected input data.
- Test mode. In test mode, a mining function uses new data with known results to verify that the model created in training mode produces consistent results.
- Application mode. In application mode, a mining function uses a model created in training mode to predict the specified fields for every record in the new input data.

2.1.7 Creating and Visualizing the Results

Information that has been created using statistical or mining functions can be saved for further analysis in the form of result objects. Figure 3 on page 18 illustrates some of the results generated by the clustering function.

Result objects can be used in several ways:

- To visualize or access the results of a mining or statistical function
- To determine what resulting information you want to write to an output data object
- As input when running a mining function in test mode to validate the predictive model representation by the result
- As input when running a mining function in application mode to apply the model to new data
2.1.8 Creating Data Mining Operations

The IM provides a means to construct data mining operations as a sequential series of related data preparation, statistical, and mining functions. IM will execute the objects in the order in which they have been specified. A sequence can contain other sequences. You can construct standard sequences that can be reused in similar data mining operations forming part of a more complex sequence.

In IM sequence setting objects are created through the sequence task guide. You can select objects from the mining database and place them in the sequence work area setting in the order in which they are to run. In addition you can specify whether the sequence should continue if any of the setting objects fails. Figure 4 on page 19 illustrates a sequence settings window.
### 2.2 DataJoiner

Business and government agencies today maintain enormous amounts of data about their operations, from sales transactions to medical records, organized along departmental and functional lines.

Critical business data is often managed by multiple database management systems (or conventional file management systems) running on different operating systems across locations with different network protocols. This has led to fragmented data resources, or to what has come to be known as islands of information.

Turning operational data into an informational asset has become a key focus for many enterprises, but one not easily realized due to the volume and complexity of the data, its distribution, and proprietary incompatibilities.

This is compounded by the fact that much of this data is redundant and too inconsistent to support analytical processing and decision making. Providing an integrated view across the enterprise is a challenge facing many businesses today.

Data mining has become a critical form of analysis in the business intelligence field today, with many competitive applications including product
demand prediction, targeted marketing, and fraud and abuse management. In a diverse data environment this advantage is challenged.

2.2.1 The Multidatabase Solution

Many products on the market today seek to address the complexities faced by businesses with distributed and diverse data sources. Middleware is the generic term used to refer to a class of software which aims to simplify access by providing a transparent view of divergent data resources.

Database gateways are a type of middleware which partially satisfy this requirement by passing requests from one database to another, but with poor location transparency and no support for join operations across data sources.

Multidatabase servers surpass gateways by providing single-site images of data that is physically distributed across multiple systems.

DataJoiner is IBM’s solution for multivendor data access enabling users to define a single, enterprisewide view of relational and nonrelational data sources across a range of platforms and operating environments.

Through DataJoiner multilocation joins, unions, views, and other relational operations required for complex business queries can be expressed in a single SQL statement optimized to minimize network traffic. Virtual tables, or views, can be defined which allow base tables to be migrated from one location or database to another without impacting the applications that reference them.

2.2.2 Heterogeneous Data Access

DataJoiner extends the reach of data mining tools like IM by presenting a single database image of a heterogeneous environment to the business analyst as though the data were local.

DataJoiner provides complete, multiplatform DB2 support in addition to Oracle, Oracle Rdb (formerly DEC Rdb), Sybase, Microsoft SQL Server and Jet databases (including MS Access), Informix, IMS, and VSAM through Classic Connect, third-party gateways such as EDA/SQL and Cross Access, and any ODBC-X/Open compliant data source such as Tandem Non-StopSQL, SPSS, and SAS. Standard embedded SQL, ODBC, and JDBC compliant interfaces to these data sources are provided.

The complexity of establishing connections to different data sources, translating requests into native calls, and data type compatibility are
managed by DataJoiner, providing transparent read/write access to current (that is, live) data, as well as stable, or point-in-time snapshots.

Data access methods, SQL dialects, network protocols, operating systems, data types, remote connections, error codes, functional differences, and security are all handled by DataJoiner so that data appears local, accessible through standard SQL to the developer and business analyst alike.

All data manipulation language (DML) statements are supported directly by DataJoiner, with data definition language (DDL) and data control language (DCL) statements passed through to the native database system. Specific function supported by a particular vendor can also be exploited using DataJoiner’s "pass-through" mechanism.

Security is handled by the native data source with validation performed at the client remote data source, by DataJoiner, or by DataJoiner in conjunction with the data source. DataJoiner observes the locking protocols of the native data source.

DataJoiner additionally provides:

- Two-phase commit support, allowing data to be updated across multiple systems in a single transaction with data integrity
- Transparent, distributed joins and unions across data sources
- Global, heterogeneous synchronous or asynchronous data replication and warehousing

for "universal" or object-relational data management.

DataJoiner works synergistically with other components of the IBM Business Intelligence (Figure 5 on page 22) and Information Warehouse (Figure 7 on page 25) architectures.
2.2.3 Database Integration

DataJoiner is built on the DB2 server and supports all DB2 object-relational extensions including:

- **Abstract Data Types (ADTs)**
  
  These are user-defined types that describe complex column structures to the database, for example, spatial and geographic data, audio, video, multidimensional time series. This allows you to apply spatial intelligence to heterogeneous data sources to ask questions like: "Where should we open a new store?" or "Is this home location within our risk parameters?"

- **User-Defined Types (UDTs)**
  
  Examples are US_DOLLARS and EURO as extended decimals, external file links, or DATALINKS. DataJoiner supports strong typing
which can prevent misleading operations such as directly comparing two currency types.

• User-Defined Functions (UDFs)
  DataJoiner allows users to define their own SQL functions and map these to standard or user-defined functions in the native back-end data source. Encapsulation and polymorphism/function resolution are supported.

• OLAP aggregation and OLAP Server support
  Examples are CUBE, ROLLUP, and UDB Summary Table query redirection and OLAP Server support. The DataJoiner optimizer also processes distributed star schema queries efficiently.

• Collection and Reference Types

• Large Objects (LOBs)
  Examples are multimedia data types such as audio, compressed video, image, and text. DataJoiner supports queries referencing multiple large object columns, even if the underlying DBMS does not natively support this, and optimizes the communications overhead.

• Triggers
  Triggers can be defined within a DataJoiner database to perform SQL statements automatically, depending on predefined criteria.

• Stored Procedures
  Stored procedures are executable modules, comprised of SQL and application logic, that are held in the database. They greatly improve performance and minimize network traffic. DataJoiner supports the SQL92 level of function for stored procedure calls. For data sources that do not natively support stored procedures, DataJoiner will run the stored procedure locally and generate the instructions and SQL necessary for the data source to complete the request. Cascaded stored procedures across databases are also supported.

• Conditional Logic
  DataJoiner allows you to apply CASE expressions to all supported data sources.

Where possible these features are simulated by DataJoiner if native support is not available, effectively delivering an SQL3-based, object-relational API to any defined data source. See Figure 6 on page 24 for the IBM Object-Relational Architecture.
Figure 6. IBM Object-Relational Architecture

The ability of DataJoiner to simulate a function (such as recursive SQL) that is not natively supported by a remote data source is a powerful feature. DataJoiner can, for example, simulate EXCEPT and INTERSECT set operations. Where multiple open cursors are not supported natively DataJoiner will compensate by establishing separate, simultaneous connections to the database. Using this technique it can also maintain the cursor position by simulating CURSOR WITH HOLD.

2.2.4 Heterogeneous Replication

DataJoiner can capture database changes in Oracle, for example, using system-generated triggers and apply these to a supported target database (for example, Informix or DB2) on another platform. This sort of replication supports both “push” and “pull” propagation so that data can be provided on request (“pulling”), or “pushed” off-peak to avoid excessive polling in online transaction processing (OLTP) environments.

Some enterprises have a proliferation of small-scale data warehouses, or data marts, with no unified access. DataJoiner provides a viable solution to
this problem by simulating a single enterprise warehouse or cross-departmental data mart creating a "virtual warehouse" on top of pretransformed and cleansed data. Superior performance can be realized in this type of environment through DataJoiner’s query rewrite and push-down processing.

The replication features of DataJoiner are fully compatible with DataPropagator log-CAPTURE and APPLY, as well as DataPropagator Non-Relational for IMS.

2.2.5 Global Query Optimization

DataJoiner includes sophisticated global query optimization that will automatically transform queries into faster, less costly, logically equivalent SQL where I/O, sorts, and the communications overhead can be reduced. It does this by collecting and maintaining statistics on cataloged data sources. Examples are redundant DISTINCT clauses for primary key columns, and unnecessary JOINs and UNIONs. Indeed, DataJoiner’s optimizer can often provide faster single-source query execution times than the native DBMS.

Other optimization techniques employed by DataJoiner include push-down analysis. This ensures data is filtered as close to the source as possible, avoiding unnecessary I/O and network traffic. DataJoiner does this by analyzing SQL to determine which predicate(s) to apply to each data source.
When processing distributed star schema queries DataJoiner takes account of the communications overhead of pushing down a cartesian product operation and will rewrite such queries to improve performance.

The DataJoiner global catalog statistics allow it to determine the most efficient access path. In doing so it considers:

- Relative communication (I/O and CPU speed)
- Native indices and access methods
- Reference table cardinality
- Table size (pages and fractions)
- High/low values for indexed columns
- Index levels, leaf nodes, and key cardinality
- Cluster rations for clustered indices
- Distribution histograms
- Estimated function costs
- Function execution location options

DataJoiner will generate these statistics directly by querying the data source, and allows them to be entered manually when the native database does not maintain this information. By supplying cost, relative CPU, I/O, and communication cost estimates, administrators can also influence optimizer access plans.

An automatic, time-driven refresh technique maintains statistical metadata held in the DataJoiner catalog. This is specified on a table basis and indicates how many days must pass before a refresh is triggered when the table is next referenced. A refresh period of "0" will trigger a catalog refresh each time the table is referenced, while a "-1" will suppress refreshes.

### 2.2.6 Web Support

DataJoiner fully supports Java, allowing you to write stored procedures and UDFs, and Java Database Connectivity (JDBC) for transparent access to all data sources including VSAM and IMS. Web development support is extended through the Net.Data product that comes with DataJoiner. This allows knowledge workers to rapidly develop Web pages, dynamically querying any DataJoiner data source through a simple HTML-based macro language. Net.Data also supports the common gateway interface (CGI) and multiple Web server APIs, including IBM's ICAPI, Netscape's NSAPI, and Microsoft's ISAPI. It also allows the implementation of Java servlets.
Chapter 3. System Architecture for Mining Oracle Databases

In this chapter we briefly describe an Oracle database environment and define an architecture that will enable data mining on the Oracle database using the IBM IM.

The architecture described here and expanded on in later chapters does not make any assumptions about the purpose of the Oracle database. It can be an operational database, an enterprise data warehouse, or a departmental data mart. However, from the point of view of performance, stability, and ease of use, we recommend that you not implement decision support systems such as a data mining system to directly query or manipulate operational data.

3.1 The Oracle Database Environment

The version of Oracle used to set up and test the environment described in this book is Version 8.0.4 for the Server as well as the Client. We have not tested this environment with Version 7 of Oracle. However, as we go along and describe how we built this environment, we will mention anything that is significantly different for Version 7 of Oracle.

A typical Oracle database environment consists of a server platform running the Oracle RDBMS and Net8 with Protocol Adaptors for client connectivity. The clients are typically applications built on Oracle or original equipment manufacturer (OEM) software development environments or Oracle front-end tools running on personal computers using Oracle Net8 to communicate with the Oracle RDBMS over TCP/IP or IPX/SPX.

The Oracle environment we used consists of an RS/6000 server (model J50) running Oracle RDBMS Version 8.0.4 on AIX Version 4.3.1 (see Figure 8 on page 30). For the purpose of client connectivity, we also installed Net8 together with Protocol Adaptors on the RS/6000 server. The client is an IBM PC running Oracle Client Version 8.0.4 on Windows NT Workstation Version 4. Communication between the client and server is over TCP/IP. A similar environment with Oracle Version 7 would have one key difference - it would use SQL*Net instead of Net8.
The following names were used for the installation performed:

- **INSTANCE SID**: ora8
- **USERID**: analyst
- **TABLESPACE**: warehouse
- **TABLE**: customer

The *warehouse* tablespace is the default tablespace for the user and the table called *customer* resides within this tablespace. The *customer* table contains bank customer information and the columns are:

- **GENDER**: Gender of the customer
- **AGE**: Age of the customer
- **SIBLINGS**: The number of brothers/sisters the customer has
- **INCOME**: The annual income of the customer
- **CAR_COLOUR**: Colour of customer’s car
- **PRODUCT**: Retail banking product purchased by customer from bank

This table is created from a sample data file called *banking.txt* provided with the IM product on AIX. In order to access this file, you need to start IM in demo mode. You will then find the file in the `/tmp/dmtksample.<number>` directory. For instructions on starting IM in demo mode, please refer to "Appendix A. A mining Scenario" in the *Using the Intelligent Miner for Data* manual supplied with the IM product.

Here is a sample query on the *customer* table using an SQL*Plus session in our Oracle environment:
$ sqlplus analyst/analyst

SQL*Plus: Release 8.0.4.0.0 – Production on Mon Oct 12 10:53:21 1998
(c) Copyright 1997 Oracle Corporation. All rights reserved.

Connected to:
Oracle8 Release 8.0.4.0.0 – Production
PL/SQL Release 8.0.4.0.0 – Production

SQL> desc customer
Name                            Null?    Type
------------------------------- -------- ----
GENDER                                   VARCHAR2(6)
AGE                                      NUMBER(5,2)
SIBLINGS                                 NUMBER(2)
INCOME                                   NUMBER(7)
CAR_COLOUR                               VARCHAR2(7)
PRODUCT                                  VARCHAR2(1)

SQL> select * from customer where car_colour='purple' and siblings>5;

GENDER        AGE   SIBLINGS     INCOME CAR_COLOUR  P
------ ---------- ---------- ---------- ------- ----
male        27.06          6      95386 purple     2
male           32          9       9507 purple     2
female       31.2          8      21645 purple     6
female       37.6          7      46517 purple     7
male         63.3          7      54829 purple     4
female       11.08         10      2514 purple     7
male        27.04          6      44339 purple     5
male          28          6      91550 purple     7
female      24.02         10      10043 purple     4
male        31.05          6      5112 purple     5
male        45.8         14      99314 purple     3
male         32.6          6      36068 purple     5
female        49         11      8623 purple     6
female       28.05         11      15908 purple     1
female       20.04         10      226 purple     2
female       17.09          6      1703 purple     1
female       50.5         11      29786 purple     7
male        32.8          6      71465 purple     2
male          51          9      71388 purple     8

19 rows selected.

SQL>
3.2 Mining Oracle Databases Using IBM’s Intelligent Miner for Data

IM consists of:

- **A server product** that provides the engine for running statistical, data processing, and mining functions
- **A client product** that provides a Java GUI for setting up and executing these functions
- **A set of C++ APIs** for developing data mining applications using these functions

For a detailed description of the product, please refer to Chapter 2, “Introduction to Intelligent Miner for Data and DataJoiner” on page 13.

IM can use flat files or databases managed by the IBM DB2 family of products as data sources for mining. IM can also store the output data created by its statistical, data processing, and mining functions in flat files or DB2 databases.

So, in order for IM to use an Oracle database as a data source and as a target for output data, either of two approaches can be taken:

- Transfer data from Oracle tables into flat files, then execute all necessary data processing, statistical, and mining functions. Finally create an Oracle table from the final output data stored on a flat file. Figure 9 on page 33 represents this approach graphically.
- Make Oracle tables appear as DB2 tables to IM and use the same as data sources and output targets. Figure 10 on page 33 represents this approach graphically.

The Fast Extract utility available with IM enables the first approach, and the DataJoiner product from IBM enables the second. We now briefly describe these products and present an overview of the system architecture required for implementing these approaches.
3.2.1 Using Fast Extract with Oracle Databases

Fast Extract is a data collection utility supplied with IM. It allows you to transfer data out of an Oracle table on an AIX or HP-UX machine into a flat file. This flat file can then be used as input for data mining.

Once the mining is done and an output file is produced, this file will need to be transferred to the database server machine and the output data loaded back into the Oracle database. The output data from data mining algorithms usually has more fields than the input data. So, you will need to create a table
in Oracle to hold the output data and then use an Oracle utility like SQL*Load to populate the output table.

The Fast Extract utility only supports input data extraction from Oracle databases. It does not support data movement in the other direction, that is, from flat files to database tables.

The Fast Extract utility does not pad data values that it extracts. So, the resulting flat file may have a nonuniform record length. This should be corrected using file processing utilities such as sed or awk before the file is presented to IM as input data for mining.

A more detailed description of the functionality and usage of Fast Extract is provided in Chapter 5, “Fast Extract Utility” on page 71.

### 3.2.2 The DataJoiner Interface to Oracle Databases

The DataJoiner software is part of the IBM Data Management family of products. It allows users and applications to transparently communicate with multiple and diverse data sources just as if they were communicating with a DB2 database. It does this by creating aliases or nicknames in a DB2 database catalog which refer to tables or joins of tables that actually reside in other databases. Applications can then treat these nicknames just like DB2 tables and submit SQL commands against them. DataJoiner translates these SQL commands into the format recognized by the database management system (DBMS) where the tables actually reside, transfers the same to this database, and relays the result of the SQL command from this database back to the application. For further information on the DataJoiner product, please refer to Chapter 2, “Introduction to Intelligent Miner for Data and DataJoiner” on page 13.

In the current scenario, DataJoiner serves to catalog Oracle tables as nicknames which IM can access as DB2 tables. When an SQL command is issued against these tables through IM, DataJoiner first translates it to a format that can be understood by Oracle. It then sends the statement to the Oracle database and gets back the results. This transfer of SQL and data between DataJoiner and Oracle is accomplished using Net8 (or SQL*Net in the case of Oracle Version 7). So, in a sense, DataJoiner behaves as a client to Oracle. IM is unaware of the existence of the Oracle database and communicates only with DataJoiner’s DB2 database. Please note that the actual data from the Oracle database is not replicated into the DataJoiner DB2 database. Only references in the form of nicknames are stored in the DataJoiner DB2 database. However, a fully functional DB2 database is
included with DataJoiner, and applications such as IM can use this database for storing intermediate tables created during the data mining process.

Figure 11 illustrates the data transfer process using DataJoiner. The dashed representations of the tables in the DB2 database indicate that the data does not physically exist in the DB2 database.

3.3 System Architecture

Figure 12 on page 36 describes the environment that we set up to enable mining on data stored in Oracle databases. The IM client interacts with the IM server to define and execute data mining exercises. The IM server reads source data and writes output data into DB2 tables on DataJoiner. DataJoiner, in turn, maps these tables to source and target tables on Oracle and transfers data to and from Oracle, using the Oracle Net8 software. The Oracle Client on the PC is part of the initial environment that provides querying and application development facilities to end users of Oracle.
We can now change Figure 8 on page 30 to add an RS/6000 server and additional software components. The complete environment that we have used to set up and test data mining of Oracle databases with DataJoiner and Intelligent Miner for Data is shown in Figure 13 on page 36. We also installed a parallel test environment using a Windows NT based PC server. This is illustrated in Figure 14 on page 37.
A detailed description of the installation and configuration steps for DataJoiner to work with Oracle is provided in Chapter 4, "Install and Configure DataJoiner with Oracle" on page 39.
Mining Relational and Nonrelational Data with IM for Data
Chapter 4. Install and Configure DataJoiner with Oracle

In this chapter we walk you through the installation, configuration, and testing of DataJoiner to provide access to Oracle databases. All procedures have been documented for AIX as well as Windows NT.

Based on the system architecture described in Chapter 3, “System Architecture for Mining Oracle Databases” on page 29 and illustrated in Figure 13 on page 36 and in Figure 14 on page 37, the following steps need to be performed on the Data Mining Server machine (sky/candemas) in the order given:

1. Product Installation:
   1. Install Oracle Client
   2. Connect to the Oracle Server from the Oracle Client
   3. Install DataJoiner

2. DataJoiner Configuration
   1. Configure DataJoiner to access Oracle
   2. Set up a DataJoiner instance
   3. Configure DataJoiner to accept DB2 client connections
   4. Configure a DataJoiner nickname to reference an Oracle table

3. Test Access

4.1 Product Installation on AIX

In this section we describe the steps mentioned above for the AIX Data Mining Server. In our configuration this machine is an RS/6000 Model J50 running AIX Version 4.3.1. The IP address for this machine is 9.1.150.209 and its name is sky.

4.1.1 Install Oracle Client

To install the Oracle8 Client software on the RS/6000 Data Mining Server, the following tasks need to be performed:

1. Create a filesystem for the Oracle software installation

   In our setup, we created the following:
   
   • A volume group called datavg
   • A logical volume called oralv in this volume group
• A filesystem called /oracle8 on this logical volume

2. Create an Oracle administrative user

In order to do this, we created:

• An administrative group for the Oracle installation called dba

• A user called oracle in the dba group with /oracle8/product/8.0.4 as the home directory

3. Set the environment variables for the oracle user

Login as oracle and edit the .profile file to add the following environment variable definitions, shown highlighted:

```bash
PATH=/usr/bin:/etc:/sbin:/usr/ucb:$HOME/bin:/usr/X11:/sbin
JDK_HOME=/usr/jdk_base
ORACLE_BASE=/oracle8
ORACLE_HOME=$ORACLE_BASE/product/8.0.4
LIBPATH=$ORACLE_HOME/lib
LD_LIBRARY_PATH=$ORACLE_HOME/lib:$ORACLE_HOME/network/lib
ORACLE_TERM=htc
PATH=$PATH:$ORACLE_HOME/bin:$PATH:/usr/lbin:. 
LINK_CNTRL=L_PTHREADS_D7
ORAENV_ASK=NO

export PATH JDK_BASE ORACLE_BASE ORACLE_HOME ORACLE_SID LIBPATH LD_LIBRARY_PATH ORACLE_TERM LINK_CNTRL ORAENV_ASK

if [ -s "$MAIL" ]; then echo "$MAILMSG" ; fi

./usr/lbin/oraenv
```

Make sure to insert the environment variables after the existing definitions in the .profile file. Additionally, ensure that you export these variables after defining them.

Execute the .profile file after saving it. This will activate all the environment variables that you have just defined.

The environment variables required by Oracle Version 7 may be slightly different from the ones listed above. Please refer to the installation guide supplied with Oracle Version 7 for the correct variables.

4. Perform the following pre-installation tasks as root user

• Ensure that the Oracle installation CD-ROM is in the CD-ROM drive. Create a directory, /cdrom, and mount the CD-ROM filesystem on the same.

• Add the following line to the root user’s .profile file:
export ORACLE_OWNER=oracle

- Go to the `/cdrom/orainst` directory and execute the `rootpre.sh` shell script.

The `rootpre.sh` script installs the post/wait kernel extension and configures asynchronous I/O. If the kernel extension is already present on your system, `rootpre.sh` replaces the existing kernel extension and requests you to reboot the system. If asynchronous I/O is already configured, `rootpre.sh` will report this. No further action needs to be taken in this case.

5. Install the Oracle client software

- Log in as `oracle`.
- Go to the `/cdrom/orainst` directory and execute `orainst`.
- Go with the default options on all screens except for the following:
  - In the `Install Type` selection screen, select `Default Install`.
  - Read the prerelease information for any additional version- or platform-specific pre- or post-installation tasks that need to be performed.
  - In the `Installation Activity Choice` screen, select `Install, Upgrade, or De-Install Software`.
  - In the `Installation Options` screen, select `Install New Product - Do Not Create DB Objects`.
  - In the `Relink All Executables?` screen, select `Yes`.
  - In the `Software Asset Manager` screen, select the following components:
    - AIX-Based System Specific Documentation 8.0.4.0
    - Net8 8.0.4.0.0
    - Net8 External Naming Adapters 8.0.4.0.0
    - Net8 Protocol Adapters 8.0.4.0.0
    - SQL*Plus 8.0.4.0.0
  - Select `Install`.
  - Wait for the message indicating successful completion of the installation process. Check the installation log for any warning or error messages. Correct any errors by referring to the Oracle product documentation.
• Some of the above steps may be missing or slightly different for the Oracle Version 7 installation. Please refer to the installation guide supplied with Oracle for exact instructions.

• Exit the installation program.

6. Perform post-installation tasks as root

   Go to the /oracle8/product/8.0.4/orainst directory and execute the root.sh script.

Some of the above tasks may be slightly different for Oracle Version 7. Please refer to the installation guide supplied with Oracle for exact instructions. You can now proceed to test this installation.

4.1.2 Connecting to the Oracle Server from the Oracle Client

The Oracle Database Server in our installation (please refer to Figure 13 on page 36) is an RS/6000 model J50 running AIX Version 4.3.1. The IP address for this machine is 9.1.150.74 and its name is azov.

In order to allow clients to communicate with it, the Oracle Server software uses a daemon called the TNS Listener (tnslsnr). Once this daemon is started, it "listens" on port 1521, using a service called listener for clients trying to connect to the database server. The /etc/services file on your Oracle database server machine should have an entry for this service. The default port for this service is 1521. But, due to the presence of other software products on your machine which might be using the same port number, you may have the listener service configured on a different port. Please note down the exact port number that listener is using on your database server machine.

Here is the section from the /etc/services file on our database server machine (azov) where the listener is defined:
Before you start configuring your Oracle client to access the Oracle database server, you need to ensure that there is an entry for the database server machine in the /etc/hosts file of your data mining server machine.

For example, the /etc/hosts file on our data mining server machine, sky, has an entry for our Database Server machine, azov:

```
127.0.0.1 loopback localhost datajoiner # loopback (lo0) name/address
9.1.150.209 sky
9.1.150.74 azov
```

You will also need to have the following information available before configuring access to the Oracle server:

- Communication protocol being used to connect to the Oracle server
- Hostname of the Oracle database server machine
- Port number being used by the listener service on the Oracle database server machine
- Instance SID of the Oracle database instance

In our case, these are as follows:

- TCP/IP
- azov
Now, log in as oracle on the data mining server machine. Change the directory to network/admin from the oracle user's home directory. Edit the tnsnames.ora file to create an alias for the remote Oracle database on the database server machine. The alias has to be appended to the tnsnames.ora file and should have the following format:

```
<Alias> =
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL= <Protocol Name>)(Host= <Hostname>)(Port= <Port Number>))
  (CONNECT_DATA = (SID = <Instance SID>))
 )
```

In our environment, the tnsnames.ora file on sky looks like this:

```
azov =
 (DESCRIPTION =
  (ADDRESS = (PROTOCOL= TCP)(Host= azov)(Port= 1521))
  (CONNECT_DATA = (SID = ora8))
 )
```

Now, you can use SQL*Plus to submit a query against any Oracle table on the remote database on the database server machine.

On the data mining server machine, sky, you can execute the following:
You are now ready to install DataJoiner on the data mining server.

4.1.3 Install DataJoiner

In this section we walk you through the installation process for DataJoiner Version 2.1.1 on AIX. Before you begin, ensure that you have the DataJoiner distribution media in the appropriate drive.

Execute the following SMIT fastpath:

```bash
$ sqlplus analyst/analyst@azov
SQL*Plus: Release 8.0.4.0.0 - Production on Tue Oct 6 12:22:10 1998
(c) Copyright 1997 Oracle Corporation. All rights reserved.

Connected to:
Oracle8 Release 8.0.4.0.0 - Production
PL/SQL Release 8.0.4.0.0 - Production
SQL> select * from customer where car_colour='purple' and siblings>5;
GENDER        AGE   SIBLINGS     INCOME  CAR_COL P
------ ---------- ---------- ---------- ------- -
male        27.06          6 95386 purple  2
male           32          9 9507 purple  2
female       31.2          8 21645 purple  6
female       37.6          7 46517 purple  7
male        63.3          7 54829 purple  4
female      11.08         10 2514 purple  7
male        27.04          6 44339 purple  5
male           28          6 91350 purple  7
female       24.02          8 10043 purple  4
male        31.05          6 5112 purple  5
male       45.8          14 99314 purple  3
male        32.6          6 36068 purple  5
female         49         11 8623 purple  6
female       28.05         11 15908 purple  1
female       20.04         10 226 purple  2
female       17.09          6 1703 purple  1
female       50.5         11 29786 purple  7
male        32.8          6 71465 purple  2
male           51          9 71388 purple  8
19 rows selected.
SQL>
```
Select the media device for installation (the cdrom drive in our example).

From the next screen, press the F4 function key against the SOFTWARE to Install field to select the components to be installed.

The DataJoiner components we installed in order to access Oracle tables from IM for Data over TCP/IP are:

- DB2 Client Application Enabler
- DB2 Command Line Processor
- DataJoiner Code Page Conversions
- Data Joiner Communications Support - Base with TCP/IP
- DataJoiner Executables
- DataJoiner Utilities and Samples

Additionally, if you want to install the documentation, select the DataJoiner Product Library - INF - En_US and DataJoiner Product Library - Postscript - En_US file sets.

Other file sets that can be optionally selected are:
- DataJoiner Database Director
• DataJoiner Visual Explain
• DataJoiner Performance Monitor
• Replication Apply

Ensure that the **COMMIT software updates?** field is set to **yes** and hit **Enter** to continue.

Confirm successful installation in the **Command Status** screen and exit **SMIT**.

Using the **lspp** command, we can see the DataJoiner components installed on our data mining server, **sky**:

```
# lspp -l djx*
Fileset                      Level  State      Description
----------------------------------------------------------------------------
Path: /usr/lib/objrepos
  djx_02_01_01.client           2.1.2.2  COMMITTED  DB2 Client Application Enabler
  djx_02_01_01.clp              2.1.2.2  COMMITTED  DB2 Command Line Processor
  djx_02_01_01.conv             2.1.2.2  COMMITTED  DataJoiner Code Page
                             Conversions
  djx_02_01_01.cs.rte           2.1.2.2  COMMITTED  DataJoiner Communications
                             Support - Base with TCP/IP
  djx_02_01_01.db2.misc         2.1.2.2  COMMITTED  DataJoiner Utilities and
                             Samples
  djx_02_01_01.db2.rte          2.1.2.2  COMMITTED  DataJoiner Executables
  djx_02_01_01.doc.En_US.pscript 2.1.2.2  COMMITTED  DataJoiner Product Library -
                             Postscript - En_US
```

Check at the [http://www.software.ibm.com/data/datajoiner/support.html](http://www.software.ibm.com/data/datajoiner/support.html) Web site for the latest patches (Fixpak) for DataJoiner. At the time of writing, the latest Fixpak for DataJoiner Version 2.1 on AIX is Fixpak1. The name of the Fixpak file is **djx_02_01_01.U459154** and it is available at the Internet FTP site **service.boulder.ibm.com/aix/fixes/v4/other**.

### 4.2 DataJoiner Configuration on AIX

In this section we describe the steps required to configure the software.

#### 4.2.1 Configuring DataJoiner to Access Oracle

As mentioned in Chapter 3, "System Architecture for Mining Oracle Databases," DataJoiner allows users and applications to transparently communicate with multiple and diverse data sources just as if they were
communicating with a DB2 database. To do this, DataJoiner libraries must be link-edited with data source client libraries.

DataJoiner provides a script called `djxlink.sh` in the `/usr/lpp/djx_02_01_01/lib` directory to do this. The `djxlink.sh` script builds an executable file called data access module (DAM) through which DataJoiner communicates with data sources.

Each time a particular data source is accessed, DataJoiner loads the DAM for this data source, performs data transfer, and unloads the DAM. In order to reduce the system overhead of loading and unloading DAMs every time data needs to be transferred, you can set up DataJoiner to load all required DAMs at initialization. This is done by setting an environment variable called `DJXCOMM` (see 4.2.2, “Creating a DataJoiner Instance” on page 52).

4.2.1.1 Environment Variables
In our case, we need to link-edit the DataJoiner libraries with Oracle client libraries. Before you execute the `djxlink.sh` script to do this, you need to set a couple of environment variables.

Log in as `root` and add the following lines in the `.profile` file:

```
LIBPATH=/usr/lpp/djx_02_01_01/lib
ORACLE_HOME=/oracle8/product/8.0.4
export LIBPATH ORACLE_HOME
```

Execute the `.profile` file:

```
# . ./profile
```

4.2.1.2 The `djxlink.sh` Script
The `djxlink.sh` script has link-edit subroutines for creating data access modules (DAMs) for most of the data sources that DataJoiner accesses. Oracle Version 8 uses Net8 for client access. The subroutine for `net8` in the `djxlink.sh` file looks like this:
If you are using Oracle Version 7, a number of subroutines are provided in the `djxlink.sh` file for creating DAMs for several different releases of Oracle Version 7 and `sqlnet`.

These subroutines are called by a main-line function:
A complete listing of the `djxlink.sh` file is provided in Appendix A, “Scripts for Creation of Data Access Modules” on page 163.

### 4.2.1.3 The Oracle net8 Data Access Module

As root user, execute the `djxlink.sh` script from the `/usr/lpp/djx_02_01_01/lib` directory.

```
# cd /usr/lpp/djx_02_01_01/lib
# ./djxlink.sh
```

You should see the following output on your screen:
After attempting to create DAMs for all the data sources, `djxlink.sh` reports a summary of the DAMs it tried to construct and the result for each:
If `djxlink.sh` reports failure in constructing a DAM for `net8` (or in the case of Oracle version 7, `sqlnet`), check the `djxlink` file in the `/tmp` directory. This is the execution log file for `djxlink.sh`. Correct the problem listed in the log file and execute the `djxlink.sh` script again.

Some of the reasons for which DAM creation may fail are:

- DataJoiner has not been updated with the most recent patches.
- Environment variables are not set properly.
- Library levels are different from those supported by `djxlink.sh`.

The solutions to the first two problems have already been covered in this chapter. If the third problem occurs, you will need to edit a makefile called `djxlink.makefile` which is in the `/usr/lpp/djx_02_01_01/lib` directory to change the library names and paths. You will then need to run the makefile to link all needed data source types. As in the `djxlink.sh` file, there are separate sections in the `djxlink.makefile` for each data source type.

A complete listing of the `djxlink.makefile` is provided in Appendix A, “Scripts for Creation of Data Access Modules” on page 163.

### 4.2.2 Creating a DataJoiner Instance

The next step in the DataJoiner configuration is to create a DataJoiner instance. This is a normal DB2 instance with a DB2 database which users and applications can connect to. The only difference here is that the data does not reside on DB2. All SQL statements are converted and forwarded by DataJoiner to one or more non-DB2 databases which process these statements and return the output to users. However, to the end user it appears like the data is stored in and supplied by one single DB2 database.

To create the DataJoiner instance, you will need to do the following:
1. Create an administrative group for DataJoiner called *djadm*.

```
# mkgroup -a djadm
```

2. Create a user called *djinst1* in the *djadm* group. In our setup, the home directory is specified as `/data/djinst1`. You can specify any directory as long as there is enough space in the filesystem to create the database. Typically, you should not need more than 30 MB of free space in this filesystem.

```
# mkuser pgrp='djadm' home='/data/djinst1' djinst1
```

3. Assign a password to the *djinst1* user, using the `passwd djinst1` command.

4. Change the ownership of the `/data/djinst1` directory to *djinst1*.

```
# cd /data
# chown -R djinst1.djadm djinst1
```

5. Create a DataJoiner instance with *djinst1* as the instance administrator.

```
# cd /usr/lpp/djx_02_01/instance
# ./db2icrt djinst1
```

6. Log in as *djinst1* and add the following lines in the `.profile` file:

```
. $HOME/sqllib/db2profile
ORACLE_HOME=/oracle8/product/8.0.4
ORACLE_BASE=/oracle8
TNS_ADMIN=$ORACLE_HOME/network/admin
export ORACLE_HOME ORACLE_BASE TNS_ADMIN
```

The first statement executes the `db2profile` file which sets all the environment variables required by DataJoiner. Net8 requires the `ORACLE_HOME` environment variable to be set prior to starting the DataJoiner instance. The other two environment variables, `ORACLE_BASE` and `TNS_ADMIN`, are optional and are used to locate the `tnsnames.ora` file.
Edit the `db2profile` file in the `sqlib` subdirectory. Ensure that you have the `DB2COMM` environment variable set to `TCPIP` and that `net8` (`sqlnet` if you are using Oracle Version 7) is included in the `DJXCOMM` variable.

Here are the relevant sections from the `db2profile` file in our environment:

```
# Set the DB2COMM environment variable.
# DB2COMM [Not set by default, values: TCPIP APPC IPXSPX]
# Specifies which communication protocol(s) will be enabled
# at the server when the database manager is started.
# Any combination of TCPIP, APPC and IPXSPX separated by commas is valid.
# If this variable is undefined or is unset then no communication
# protocols are started.
DB2COMM=TCPIP
export DB2COMM

# DataJoiner data access module Environment Variables.
# The environment variables that are related to data access modules are:
# - DJXCOMM This variable indicates which data access modules are
#   loaded at start time (as opposed to those loaded on demand)
# - DJX_NR_CONFIG for Classic Connect data sources
# - DJX_NR_START for Classic Connect data sources
DJXCOMM=`db2ra drda drdaIP net8`
export DJXCOMM
```

7. Log out and log in again as `djinst1`. Start the DataJoiner instance, using the `db2start` command.

```
$ db2start
SQL1063N  DB2START processing was successful.
```

### 4.2.3 Configuring DataJoiner to Accept DB2 Client Connections

You now need to configure DataJoiner to allow remote users and applications to communicate with the DB2 instance you have just created.

Log in as root and edit the `/etc/services` file. Add an entry for the DataJoiner listener service. You need to provide a service name and a port number for the service. There is no restriction on the name and port number as long as they are unique. In our environment, we used the name `djlstn` and port `70000`. Here is the relevant section from the `/etc/services` file on `sky`:
4.2.4 Configuring DataJoiner Nicknames

DataJoiner uses nicknames to allow users and applications to map a two- or three-part table or view name (such as Server.Remote_AuthID.Tablename) to a data source. The nickname can be used subsequently in an SQL statement whenever the remote table, or view, is referenced.

There are several tasks that need to be performed before you can create nicknames:

1. Create a DB2 database
   - Log in as djinst1 and start DB2.

Now you need to update the DB2 database manager configuration to use the service that you have just created.

```
$ db2 update dbm cfg using svcename djlstn
DB20000I  The UPDATE DATABASE MANAGER CONFIGURATION command completed successfully.
DB21025I  Client changes will not be effective until the next time the application is started.  Server changes will not be effective until the next DB2START command.
```

Stop and restart (recycle) DataJoiner for the new configuration to be effective.

```
$ db2stop
SQL1064N  DB2STOP processing was successful.
$ db2start
SQL1063N  DB2START processing was successful.
```

```
x_st_mgrd       9000/tcp                        # IBM X Terminal
man             9535/tcp
man             9535/udp
isode-dua       17007/tcp
isode-dua       17007/udp
dtspc           6112/tcp
ipsec_sk_master  1011/udp
ipsec_sk_engine_s  4001/udp
   djlstn       70000/tcp                      # Data Joiner Listener Port
   ~
   ~
   ~
   ~
   ~/etc/services" [Read only] The cursor is at line 675 of 697 --96%-- .

Install and Configure DataJoiner with Oracle  55
• Create a database called oradb.

`
$ db2 create database oradb
DB20000I The CREATE DATABASE command completed successfully.
$

• Change the `DB2DBDFT` environment variable to `oradb` in the `db2profile` file in the `$HOME/sqllib` directory. This will make the `oradb` database that you just created the default database for this instance.

Here is the relevant section from our `db2profile` file:

```
# DB2DBDFT [Default= SAMPLE]
# is set to the database alias name of the database that will
# be implicitly connected to when applications are started.
#-----------------------------------------------
DB2DBDFT=ORADB
export DB2DBDFT
```

• Recycle the DataJoiner instance.

`
$ db2stop
SQL1064N DB2STOP processing was successful.
$ db2start
$ db2stop
SQL1064N DB2STOP processing was successful.
$
```

2. Create a mapping for the Oracle database server

In this step, you define to DataJoiner the Oracle server you want to access.

The syntax of the DDL is:

```
db2 create server mapping from <SERVER NAME> to node <NODE NAME> \
    type <DATABASE SERVER TYPE> version <VERSION NUMBER> protocol "<DAM>"
```

The `SERVER NAME` should be unique. The `NODE NAME` should be the alias that you configured in your `tnsnames.ora` file (refer to 4.1.2, “Connecting to the Oracle Server from the Oracle Client” on page 42) for the Oracle database server. In this case, `DATABASE SERVER TYPE` is `oracle`, `Version Number` is `8.0` and `PROTOCOL` is `net8`. If you are using
Oracle Version 7, your version number would be different, and
**PROTOCOL** would be *sqlnet*.

You need to connect to the database you have created before creating the server mapping. This is the command we used in our test environment:

```
$ db2 connect to oradb
   Database Connection Information
   Database product       = DB2/6000 2.1.2
   SQL authorization ID   = DJINST2
   Local database alias   = ORADB

$ db2 create server mapping from djora8 to node azov type oracle version 8.0 \
   protocol "net8"
   DB20000I  The SQL command completed successfully.
   $ 
```

Please note the back-slash (\) before the quotes (") surrounding the protocol name. If you do not use the back-slash, the shell will drop the quotes before processing the statement, and DataJoiner will return an error. If you execute this DDL from the DB2 Command Line Processor, you will not need to use the back-slashes.

```
$ db2
   (c) Copyright IBM Corporation 1993,1995
   Command Line Processor for DB2 SDK 2.1.2
   .
   .
   For more detailed help, refer to the Online Reference Manual.

   db2 => create server mapping from djora8 to node azov type oracle \
   db2 (cont.) => version 8.0 protocol "net8"
   DB20000I  The SQL command completed successfully.
   db2 =>
```

This DDL creates an entry in the **SYSCAT.SERVERS** table in DataJoiner.

<table>
<thead>
<tr>
<th>Server</th>
<th>Node</th>
<th>Dbname</th>
<th>Server_Type</th>
<th>Server_Version</th>
<th>Server_Protocol</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>DJORA8</td>
<td>AZOV</td>
<td>ORACLE</td>
<td>8.0</td>
<td></td>
<td>NET8</td>
<td></td>
</tr>
</tbody>
</table>

Once you have defined the data source server mapping, you can use additional DataJoiner DDL statements to refine access to these data sources.
3. Create a mapping for one or more Oracle database users

One of the DDLs that you can use maps a local DB2 user to an Oracle user. The Oracle user should have been created at the Oracle database server using the `create user` command with the `identified by` clause rather than the `identified externally` clause.

The syntax for this DDL is:

```
db2 create user mapping from <LOCAL USER> to server <SERVER NAME> authid <REMOTE USER NAME> password <REMOTE USER PASSWORD>
```

We created a DB2 user called `dminer` in our environment. We mapped this user to the `analyst` user on Oracle with a password of `analyst`.

```
$ db2 create user mapping from dminer to server djora8 authid analyst password analyst
DB20000I  The SQL command completed successfully.
```

You can now create a nickname for an Oracle table owned by an Oracle user, using the following syntax:

```
create nickname <NICKNAME> for <SERVER NAME.REMOTE USER.TABLE NAME>
```

In our environment, we created a nickname, `customer`, for an Oracle table, `customer`, owned by the Oracle user `analyst`. The DB2 user that maps to the `analyst` Oracle is `dminer`. So, we need to log in as `dminer` before executing the `create nickname` command. Please ensure that the `LIBPATH` environment variable is set to `/usr/lpp/djx_02_01_01/lib` and that the `/data/djinst1/db2profile` file is executed in the `dminer` user's `.profile` file.

```
$ db2 create nickname customer for djora8.analyst.customer
DB20000I  The SQL command completed successfully.
```

This nickname will now appear to the `dminer` DB2 user on DataJoiner as just another table.

4.3 Test Access on AIX

You should now be able to test your environment. Log in as the DB2 user for whom you have created the user mapping. Connect to the DB2 database you have defined in DataJoiner and submit an SQL against the nickname you have created.
In our scenario, the local DB2 user `dminer` can now connect to the DB2 database `oradb` and submit an SQL against the nickname `customer`. This is the same query that we document in 3.1, “The Oracle Database Environment” on page 29.

```
$ db2
(c) Copyright IBM Corporation 1993,1995
Command Line Processor for DB2 SDK 2.1.2
.
.
For more detailed help, refer to the Online Reference Manual.

db2 => connect to oradb

Database Connection Information

Database product       = DB2/6000 2.1.2
SQL authorization ID   = DMINER
Local database alias   = ORADB

db2 => select * from customer where car_colour='purple' and siblings>5

GENDER AGE     SIBLINGS INCOME      CAR_COLOUR PRODUCT
------ ------- -------- ----------- ---------- -------
male     27.06        6       95386 purple     2
male     32.00        9       9507 purple     2
female   31.20        8       21645 purple     6
female   37.60        7       46517 purple     7
male     63.30        7       54829 purple     4
female   11.08       10        2514 purple     7
male     27.04        6       44339 purple     5
male     28.00        6       91550 purple     7
female   24.02        6       10043 purple     4
male     31.05        6       5112 purple     5
male     45.80       14       99314 purple     3
male     32.60        6       36068 purple     5
female   49.00       11        8623 purple     6
female   28.05       11       15908 purple     1
female   20.04       10        226 purple     2
female   17.09        6       1703 purple     1
female   50.50       11       29786 purple     7
male     32.80        6       71365 purple     2
male     51.00        9       71388 purple     8

19 record(s) selected.

db2 =>
```

This completes the task of configuring DataJoiner to access Oracle. You can now use IM to run data mining tasks on your Oracle tables.
4.4 Product Installation on Windows NT

In this section we describe the steps required to configure DataJoiner and Oracle in the Windows NT environment. In the example shown the data mining server is a Pentium PC running Windows NT 4.0. The IP address for this machine is 9.1.150.155 and its name is candemas.

4.4.1 Install Oracle Client

To install the Oracle8 Client software on the Windows NT data mining server:

1. Run the Setup program on the Oracle client installation CD and OK the Oracle Installation Settings dialog (Figure 15).

![Figure 15. Oracle Installation Settings](image)

2. On the Installation Options dialog choose Oracle8 Client and click OK (Figure 16).

![Figure 16. Installation Options](image)
3. On the *Client Configuration* dialog select *Application User* and click **OK** (Figure 17).

![Client Configuration](image)

**Figure 17. Client Configuration**

When the installation is complete exit the installer (Figure 18).

![Installation Completed](image)

**Figure 18. Installation Completed**

### 4.4.2 Connecting to the Oracle Server from the Oracle Client

Before connecting to the Oracle Database Server ensure there is an entry for the server machine in the `\WinNT\System32\Drivers\Etc\hosts` file on your NT machine. This will have the IP address and name for the machine. In the ITSO environment this was: 9.1.150.74 AZOV.

To test the client connection:

1. Select **Start ->Programs -> Oracle for Windows NT -> SQL Plus 8.0**.
2. On the *Log On* dialog (Figure 19 on page 62) enter the following parameters and click **OK**:
3. To test the connection, enter the following at the SQL> prompt of the Oracle SQL*Plus window (Figure 20):
   
   desc customer

![SQL*Plus Window](image)

Figure 20. SQL*Plus Window

4. To leave SQL*Plus type `quit` at the prompt.
4.4.3 Install DataJoiner

In the section we guide you through the DataJoiner for Windows NT installation process.

1. Create a DataJoiner Administrator ID through the *NT User Manager*.
   - Select **Start -> Programs -> Administrative Tools (Common) -> User Manager** (Figure 21).

   ![NT User Manager](image1)

   **Figure 21. NT User Manager**

   - Select **User -> New User** from the *User Manager* window and enter the ID and description fields (Figure 22).

   ![New User](image2)

   **Figure 22. New User**
• Click the button and add the DataJoiner Administrator to the Administrators group (Figure 23).

![Figure 23. Group Memberships](image)

• Click the Group Memberships and New User dialogs.

• Close the User Manager window (Figure 24 on page 64) by selecting User -> Exit and log on to Windows NT using the ID you created by selecting Start -> Shutdown (with Close all programs and log on as a different user?).

![Figure 24. NT User Manager: DataJoiner Administrator](image)
2. Run the Setup program on the DataJoiner installation CD and select the options you require.
   - On the DataJoiner for Windows NT Installation window (Figure 25) click the Next > button to proceed with the installation. Depending on the installation options selected you will be prompted for the installation directory for optional components (for example, Classic Connect data mapper for IMS and VSAM, Replication Administration, DB2 Spatial Extender, and finally the DataJoiner NLS Options. For the installation type select Typical and click the Install > button.

   ![DataJoiner for Windows NT Installation](image)

   Figure 25. DataJoiner for Window NT Installation

   - When the Setup is complete click Finish to restart your computer.
   - After Windows NT has restarted you should ensure that the NT services file (\<drive>:\WinNT\System32\Drivers\Etc\Services) includes the following DataJoiner services:
     
     ```
     djlstn 80000/tcp
     djintr 80001/tcp
     ```
     These should be configured with unique port numbers for your environment.

3. If you want to start DB2 Security Server automatically, follow these steps:
   - Select Start -> Settings -> Control Panel -> Services.
   - Select the DB2 Security Server and click the Startup button (Figure 26).

![Figure 26. Windows NT Services](image)

- Select Startup Type=Automatic (Figure 27), OK the Service dialog, click the Start button in the Services window, and then click Close.

![Figure 27. Service Startup Options](image)
4.5 DataJoiner Configuration on Windows NT

In this section we describe the steps required to configure this software.

4.5.1 Configuring DataJoiner to Access Oracle

DataJoiner on Windows NT allows users and applications to access remote or local Oracle data as if it were a local DB2 table. For Oracle8 databases it does this by routing SQL through the net8 DAM. To configure this connection:

1. Create a DataJoiner DB2 instance by selecting Start -> Programs -> DataJoiner for Windows NT -> Command Window and submit the `db2icrt djinst1` command.

2. Start the DataJoiner instance with the `db2start` command.

3. Create a database: `db2 create database oradb`.

4. Connect to the database with the `db2 connect to oradb` command.

5. Create a mapping (Figure 28 on page 67) for the Oracle database server by typing:

   `db2 create server mapping from djora8 to node azov type oracle version 8.0 protocol "net8"`

6. Map the client user to the database server’s remote user:

   `db2 create user mapping from imres2 to server djora8 authid analyst password analyst`

---

Figure 28. Oracle Server Mapping
7. You can now define a local nickname for the Oracle tables you want to use from Windows NT with this command:

```
db2 create nickname customer for djora8.analyst.customer
```

### 4.6 Test Access on Windows NT

To test the configuration submit the SELECT statement shown in Figure 29 from the DB2 prompt of the DB2 command line processor.

![Figure 29. Testing the Oracle Connection](image)

The output should look like that illustrated in Figure 30.

![Figure 30. Test SELECT Results](image)
With the expected results your configuration is complete. To stop the DataJoiner instance, type:

```
connect reset
db2stop
```

at the DB2 prompt then `quit` and `exit` to close the command window.
Chapter 5. Fast Extract Utility

The IM Fast Extract utility allows you to read data out of Oracle and Sybase tables on AIX and Oracle tables on HP-UX and write the same into flat files.

In this chapter we discuss the most relevant scenario: transferring data out of Oracle on a remote AIX machine into a flat file on the IM machine. In our scenario, this translates to transferring data out of an Oracle table on azov into a flat file on sky.

There are two components in the Fast Extract utility for Oracle:

- An executable called *oracollect*, which is a client program that runs on the data mining server
- An executable called *idmpldor*, which is a server program that runs on the Oracle database server machine and actually does the job of data collection

Whenever the *oracollect* utility is called, it in turn invokes the *idmpldor* utility on the database server machine, which collects the data and passes it back to the *oracollect* utility, which then writes the data into a flat file.

A message log file called *oracollect.log* is created in the directory on the data mining server machine where the *oracollect* utility is invoked.

5.1 Installation

Installing the Fast Extract utility for Oracle involves transferring the *idmpldor* executable program from the `/usr/lpp/IMiner/bin` directory of the data mining server machine to the Oracle database server machine and changing its mode to make it an executable.

In our case, we logged in to sky as root, changed the active directory to `/usr/lpp/IMiner/bin`, FTPed to azov, and wrote the *idmpldor* file in the `/usr/bin` directory.
We then logged in as root on azov, changed the active directory to /usr/bin, and changed the idmpldor file to executable mode.

```bash
root@azov > cd /usr/bin
root@azov > chmod +x idmpldor
root@azov >
```

5.2 Use

Before you execute the oracollect utility, you must note the following information:

- Hostname of the Oracle database server machine
- AIX user ID for connecting to Oracle
- Password for this user ID
- Fully qualified pathname for the idmpldor command
- Oracle user ID
- Password for the Oracle user ID
- SID of the Oracle database
- Oracle home directory (this is the same as the ORACLE_HOME environment variable)
• Name of the file containing the SQL query to be used for data extraction
• Name of the flat file where the data has to be written

Use the following syntax to extract data out of Oracle into a flat file:

```
oracollect -H host-machinename -U host-username -P host-password \ 
-M path-of-idmpldor -u oracle-login-name -p oracle-login-password \ 
-D oracle-SID -W oracle-home-directory -q query-file -f local-filename
```

Before you execute the above command you have to create a file containing the SQL query (the query-file of the command syntax described above) that is to be executed.

A very simple query file \((getdata.sql)\) looks like this:

```
$ cat getdata.sql
select * from customer
$ 
```

Note that there is no semi-colon (;) at the end of the statement.

The \(oracollect\) command that we executed was:

```
$ /usr/lpp/IMiner/bin/oracollect -H azov -U oracle -P oracle \ 
> -M "/usr/bin/idmpldor" -u analyst -p analyst -D ora8 -W "/oracle8/product/8.0.4" \ 
> -q getdata.sql -f customer.txt
(C) Copyright IBM Corporation 1996, 1998
Local file is customer.txt
Total  2048 rows processed
$ 
```

This creates a \(customer.txt\) file in the local directory that looks like this:

```
$ head customer.txt
"female",27.06,1,6611,"red","1",
"female",49.3,2,7652,"green","2",
"male",16.08,3,1778,"red","3",
"male",14.04,0,567,"red","4",
"female",24.06,2,21334,"green","5",
"male",4.4,50,"blue","6",
"male",30.2,60346,"blue","7",
"female",53.1,0,33291,"blue","8",
"male",33.2,0,53993,"blue","1",
"female",31.6,1,982,"green","2",
$ 
```
Before the above file can be used by IM for data mining, the double-quote characters need to be removed and the fields need to be padded with blanks so that each record is of the same length. This can be done with any text processing tool like `awk` or `sed`.
Chapter 6. Sample Mining of Oracle Data

This chapter begins with the task of installing IM. We then walk you through a sample data mining exercise, using data stored in an Oracle database that is accessed through DataJoiner.

6.1 Intelligent Miner for Data Installation on AIX

We installed the following two IM components:

- Intelligent Miner for Data server - This is the processing engine which runs the mining, data processing, and statistical algorithms.
- Intelligent Miner for Data client - This is the Java interface that is used to set up mining runs.

The IM server software can be installed on an RS/6000, a System/390, an AS/400, or a PC running under Windows NT. The IM client software can be installed on an RS/6000, a PC running OS/2, or a PC running 32-bit Windows. In this section, we consider an RS/6000 server and a Windows NT client.

The IM server software can be installed on an RS/6000 machine using SMIT. The recommended memory is 512 MB or more. The minimum disk space required for installing and demonstrating the product is 100 MB. However, we recommend that in production environments, two times the amount of raw data be available as free disk space.

The version of AIX should be 4.1.5 or higher. IM also requires DB2 to be installed on the server machine. The supported versions of DB2 on AIX are:

- IBM DB2 for AIX Version 2.1.1 (therefore includes DataJoiner)
- IBM DB2 Parallel Edition Version 1.2
- IBM DB2 UDB Enterprise Edition Version 5.0 or higher
- IBM DB2 UDB Extended Enterprise Edition Version 5.0 or higher

DB2 must be installed on the AIX server before IM is installed. If you attempt to install DB2 on a machine on which IM is already installed, the DB2 installation might fail. A DB2 UDB Enterprise Edition Version 5 CD-ROM is included with the IM product package. This CD-ROM contains installation notes which you must follow to install DB2.

In order to install the IM server on AIX, select the following software packages from the installation CD-ROM, using SMIT:
If you want to develop data mining applications of your own that use the IM engine, also install the IMiner.toolkit package.

The installation directory for IM is /usr/lpp/IMiner.

After the SMIT installation task completes, you need to make the IM executables available to users. This can be done in either of two ways:

- Execute the /usr/lpp/IMiner/bin/imln script. This creates links in the /usr/bin directory to the IM executable files in the /usr/lpp/IMiner/bin directory.
- Add the /usr/lpp/IMiner/bin path to the PATH environment variable in a user’s .profile file.

Also ensure that the .profile file for the root user has the following entries:

```
export LIBPATH=/usr/lpp/djx_02_01_01/lib
./data/djinst1/sqllib/db2profile
```

The IM client can be installed on a Windows NT system by executing the SETUP.EXE program in the \WIN32\xx directory of the IM V 2.1.2 Client CD-ROM, where xx identifies the language for your installation. For English, you would use the \WIN32\EN directory. The recommended memory is 64 MB, and the recommended diskspace is 70 MB or more.

The client installation program also installs the Java Runtime Environment (JRE) required. If you want to view the online help, you have to install a Web browser capable of displaying frames such as Microsoft Internet Explorer Version 3.0 (or higher) or Netscape Navigator for Windows 95/NT Version 3.01 (or higher).

Please refer to the Using the Intelligent Miner for Data manual supplied with the IM product for more detailed instructions on installation and setup. Also, please read the README.TXT file in the server and client product installation CD-ROMs for additional information before you install the product.

### 6.2 The Mining Scenario

We implemented a hypothetical scenario to illustrate the process of data mining on an Oracle database.
In Chapter 3, “System Architecture for Mining Oracle Databases”, we describe a table in our Oracle database called `CUSTOMER`. This table contains demographic information about a bank’s customers such as age, gender, income, number of siblings, and the color of the car they drive. It also contains information about the kind of service they have purchased from the bank.

The bank knows that its most profitable service is the Premier Checking Account. It wants to use data mining to determine the characteristics of its Premier Checking Account customers and to use this information to sell this service to its existing customers who do not have a Premier Checking Account.

In the `CUSTOMER` table, the kind of service purchased by a customer is recorded in the `PRODUCT` column. A value of 1 in this column indicates Premier Checking Account.

### 6.3 The Mining Solution

The mining process to achieve the goal stated in the previous section consists of the following steps:

1. In order to understand the characteristics of Premier Checking Account customers, we perform demographic clustering on all customer records in the `CUSTOMER` table for which the value in the `PRODUCT` field is 1. The output of this step is a model containing a number of clusters which are characterized by the distribution of all the fields in the `CUSTOMER` table except for the `PRODUCT` field.

2. We apply the model generated in step 1 to the records in the `CUSTOMER` table for which the value in the `PRODUCT` field is not 1. These are the customers who do not have a Premier Checking Account with the bank. Based on the values of its fields, a particular customer record is placed in one of the clusters created in step 1. The output of this step is a table called `SCORED` containing the original records with three additional fields:
   - `CLUSTERID` - The cluster to which a record belongs
   - `SCORE` - The strength of a record’s fit in the cluster to which it has been assigned on a scale of 0 to 1
   - `CONF` - The confidence level of an assignment

3. We select those records in the `SCORED` table for which the value of the `SCORE` field is greater than 0.7 and store it in a table called `TARGET`. We also generate univariate statistics for this group of records to better understand the distribution of values in each of the fields. The `TARGET`...
table can now be used by the bank to market the Premier Checking Account.

### 6.4 Using Intelligent Miner to Implement the Solution

To implement the above solution, the following tasks must be performed on Intelligent Miner:

1. Create a data object for the CUSTOMER table.
2. Create a demographic clustering mining setting and a result object to store the model that is generated.
3. Create a demographic clustering mining setting in application mode to apply the model generated in the previous step to the CUSTOMER table and store the output in a SCORED table.
4. Create a statistics setting to select appropriate records from the SCORED table, store them in the TARGET table, and generate univariate statistics for each of the fields.

To start the IM server, log in as root and execute the *idmstart* script.

```bash
# idmstart
LIBPATH=/usr/lpp/IMiner/lib:/usr/lpp/djx_02_01/lib
Changing ulimit -s from 32768 to unlimited
Changing ulimit -d from 131072 to unlimited
Changing ulimit -m from 32768 to unlimited
```

To start the IM client on a Windows NT machine, select Start -> Programs -> Intelligent Miner -> Intelligent Miner.

The main screen of IM is shown in Figure 31 on page 79. When you start the IM client on a machine which is not running the IM server for the first time, it will open the Preferences window at the Server tab (see Figure 32 on page 79) automatically. Here you specify the hostname of the IM server machine, the user you want to connect as, and the password. The user you specify should be a valid AIX user. This user should also have the permission to access the database tables that you want to use for mining. You can open this window later by selecting Options -> Preferences in the IM main menu.

In our example we use the *dminer* user to connect to our Data Mining Server (*sky*). The *dminer* user has access to the CUSTOMER table on Oracle through DataJoiner (see Chapter 4, “Install and Configure DataJoiner with Oracle” on page 39).
After selecting the IM server hostname, user ID, and password, click on the Add button if this is the first time you are connecting to the server. This will
add an entry for the IM server in the *Matrix* section of the window. Every subsequent time that you need to connect to the IM server, you just need to select the entry from the *Matrix* section, type in your password, and click the **Update** button. Click on **OK** to continue.

You are now ready to start with the mining configuration.

### 6.4.1 Creating a Data Object for the CUSTOMER Table

In the IM main screen, click on the *Create Data* button on the toolbar. The *Welcome* screen of the *Data - Taskguide* window appears. Click the **Next >** button to continue.

In the *Data format and settings* screen (Figure 33), select *Database Table/View* and enter *Customer* under *Settings name*. Select the *Show the advanced pages and controls* option.

![Data Format and Settings Screen for Building Model](image)

Click on the **Next >** button to continue to the *Database Table or View* screen shown in Figure 34 on page 81. Select *ORADB* from the *Database server* pulldown menu, then *DMINER* in the *Schema list*. Finally select *CUSTOMER* in the *Database Tables and Views* list and the *Read only* option under *Use mode*. Click on the **Next >** button to continue.
The next screen, Field Parameters, shows the fields in the table that you just selected (see Figure 35). It lists the field names and the corresponding DB2 data types and IM data types.
IM maps database data types to five different data types for its own use:

- **Binary** - Data containing categorical fields with a specific NULL and NONNULL value

- **Categorical** - Data containing string values. IM maps database data types of CHAR, VARCHAR, DATE, TIME, TIMESTMP, GRAPHIC, VARGRAPHIC, and LONG VARGRAPHIC to the categorical data type.

- **Continuous** - Data containing real values. The number of distinct values is unlimited. Statistics are maintained for the ranges of continuous data. IM maps database data types of DOUBLE, REAL, DECIMAL, FLOAT, NUMERIC, DEC, DOUBLE PRECISION, and PACKED to the continuous numeric data type.

- **Numeric** - Data containing integer or real values. Use this data type to let IM determine whether a numeric field should be treated as a discrete numeric or a continuous numeric field. If the number of distinct values in a field is:
  - Below a certain threshold, IM treats it as a discrete numeric field
  - Above a certain threshold, IM treats it as a continuous numeric field

  The default threshold value is 100. IM maps database data types of INTEGER and SMALLINT to the numeric data type.

- **Discrete numeric** - Data containing integer or real values. The number of distinct values is limited, for example, in age or salary data. Statistics are maintained for each value in discrete numeric data. Discrete numeric values can also be cyclic. For example, the days of a week represented by numbers from 1 to 7 are cyclic.

For our example, nothing needs to be changed in this screen. Click on the **Next >** button to continue.

You can skip the *Computed fields*, *Bucket limits*, *Valid values*, and *Cyclic fields* screens. Just click on the **Next >** button until you reach the *Summary* screen. Click the **Finish** button and your data object is ready.

The *Data folder* in your mining base will now show a *Customer* object. Before you continue, save your work (*Target Marketing* in our example) using the **Mining Base -> Save Mining Base as ...** menu item (Figure 36 on page 83).
6.4.2 Setting Up Demographic Clustering

In the IM main screen, click on the Create mining button on the toolbar. The Welcome screen of the Mining - Taskguide window appears. Click on the Next > button to proceed to the Mining functions and settings screen (Figure 37). Select Clustering -> Demographic under the Available mining functions, enter Build Model as the Settings name and select the Show the advanced pages and controls option. Click on the Next > button to continue.
In the *Input data screen* (Figure 38), select the *Customer* data object under *Available input data* and click on the button against *Filter records condition in the Advanced section*.

![Figure 38. Input Data Screen for Building Model](image)

This is where we tell IM to run clustering on only those records for which the value of the *PRODUCT* field is 1. Execute the following steps in the *Expression Builder* window (Figure 39) to achieve this:

1. Click on the **AND** button.
2. Select *Field Names* under *Category*.
3. Select *PRODUCT* under *Value*.
4. Click on the **Arg1** button.
5. Select *Constants* under *Category*.
6. Double-click on `<new constant>` under *Value*.
7. Type in "1" in the edit box that appears and hit *Enter*.
8. Select "1" under *Value*.
9. Click on the **Arg2** button.
The Filter records condition in the Advanced section of the Input data screen (Figure 38) should now show the \((\text{PRODUCT} = "1")\) entry.

Click the OK button to accept the settings and then on the Next > button to continue.

In the Mode parameters screen, the Clustering mode radio button will be selected by default. Click on the Next > button to continue.

This will take you to the Input fields screen shown in Figure 40 on page 86, where you need to specify the fields in the database which will be used to decide the similarity between members of a cluster. These fields need to be specified as active fields. In case there are some fields which should not be used for generating the clusters but you want to see the distribution of these variables in the model that is generated, you have to specify these fields as supplementary fields. Select \(\text{AGE}, \text{CAR_COLOUR}, \text{GENDER}, \text{INCOME},\) and \(\text{SIBLINGS}\) under Available fields and click on the Next > button to add these fields to the Active fields list. Click on the Next > button to continue.
You can skip the Field parameters, Additional fields parameters, Outlier treatment, Similarity matrix, and Parallel parameters screens. Just click on the Next > button in each of these screens.

In the Output fields screen, the Do not create output radio button should be selected. This implies that we will not create any database table as an output from this setting. Click on the Next > button to continue.

In the Results screen (Figure 41 on page 87), type in Model as the Results name and select the If a result with this name exists, overwrite it option. Click on the Next > button to continue.
In the Summary screen, make sure that the Run this setting immediately option is selected and click the Finish button to generate the demographic clustering model.

While IM is building the model, you will see a progress window as shown in Figure 42.
After completion, IM opens a window (Figure 43) to show the various clusters found. Each horizontal strip represents one cluster.

The clusters are ordered top to bottom in order of decreasing size. The cluster number is shown on the top right-hand corner of each strip. The size of a cluster is shown on the left of each cluster as a percentage of the total population. For example, in our case, the biggest cluster consists of 34% of the records that were present in the input table.

You can double-click on any of the clusters to get a better view of the variable distributions within this cluster. For example, if you double-click on the largest cluster, you will see a screen like the one shown in Figure 44 on page 89.
You can further double-click on any of the histograms or pie charts to get an even more detailed view. For example, if you click on the pie chart for gender, you will see a detailed representation of this variable, as shown in Figure 45.

The outer band around the pie charts and the red rectangles in the histograms indicate the distribution of that variable for the entire population.
(all the records in the table). The inner circle in the pie charts and the gray rectangles in the histograms indicate the distribution of that variable for the entire population.

In our case, we can say that the target cluster of customers purchasing the Premier Checking Account (as shown in Figure 44 on page 89) are predominantly male, less than 25 years of age, and come from a low-income group.

6.4.3 Applying the Model

The model which was generated in the previous step and stored as a result object called Model can now be applied to customers of the bank who do not have a Premier Checking Account. So, we apply this model to those records in the CUSTOMER table for which the value in the PRODUCT field is not 1.

On the IM main screen, click on the Create mining button on the toolbar. The Welcome screen of the Mining - Taskguide window appears. Click on the Next > button to proceed to the Mining functions and settings screen shown in Figure 46 and select Clustering - Demographic under Available mining functions. Enter Apply Model as the Settings name, and select the Show the advanced pages and controls option. Click on the Next > button to continue.

Figure 46. Mining Functions and Settings Screen for Applying Model
In the *Input data* screen (see Figure 47), select the *Customer* data object under *Available input data*. Click on the button against *Filter records condition* in the *Advanced* section.

![Figure 47. Input Data Screen for Applying Model](image)

This is where we tell IM to run clustering on only those records for which the value of the *PRODUCT* field is not 1. Execute the following steps in the *Expression Builder* window (Figure 48 on page 92) to achieve this:

1. Click on the **AND** button.
2. Select *Field Names* under *Category*.
3. Select *PRODUCT* under *Value*.
4. Click on the **Arg1** button.
5. Click on the **<>** button.
6. Select *Constants* under *Category*.
7. Select "1" under *Value*.
8. Click on the **Arg2** button.

The *Filter records condition* in the *Advanced* section should now show the ((PRODUCT <> "1")) entry.

Click on the **OK** button to accept the settings and then on the **Next >** button to continue.
In the *Mode parameters* screen (shown in Figure 49), click on the *Application mode* radio button and select *Model* as result. Click on the **Next >** button to continue.
In the \textit{Input fields} screen (see Figure 50), specify the fields in the database which will be used to decide the cluster in which a record belongs. These fields must be specified as active fields. Select \textit{AGE}, \textit{CAR\_COLOUR}, \textit{GENDER}, \textit{INCOME}, and \textit{SIBLINGS} under \textit{Available fields} and click on the \texttt{\textgreater \textgreater} button to add these fields to the \textit{Active fields} list. Click on the \texttt{Next} \textgreater button to continue.

![Figure 50. Input Fields Screen for Applying Model](image)

You can skip the \textit{Fields parameters}, \textit{Additional field parameters}, \textit{Outlier treatment}, \textit{Similarity matrix}, and \textit{Parallel parameters} screens. Just click on the \texttt{Next} \textgreater button in each of these screens.

In the \textit{Output fields} screen (see Figure 51 on page 94), the \textit{Create output data radio} button should be selected. This means that we are going to create a database table as an output. This table will contain all the original fields along with three new fields - \textit{CLUSTERID} (the cluster number to which a record belongs), \textit{SCORE} (the strength of a record's fit in the cluster), and \textit{CONF} (the confidence level of an assignment). The output table, however, will not contain those rows in the \textit{CUSTOMER} table which had a value of 1 in the \textit{PRODUCT ID} field.

Click on the \texttt{\textgreater \textgreater} button to add all the fields under \textit{Available fields} to \textit{Output fields}. Enter \textit{CLUSTERID} as the \textit{Cluster ID} field name, \textit{SCORE} as the \textit{Record score} field name, and \textit{CONF} as the \textit{Confidence} fields name. Click on the
Next > button to continue to the Output data screen. In this screen click on the Create data button to create a data object for the output table.

Figure 51. Output Fields Screen for Applying Model

In the Welcome screen of the Data - Taskguide window, click on the Next > button to continue and in the Data format and settings screen shown in Figure 52 select Database Table/View and enter Scored Customer under Settings name.

Figure 52. Data Format and Settings Screen for Applying Model
Then select the *Show the advanced pages and controls* option and click the *Next >* button.

In the *Database table or view* screen (Figure 53 on page 96), select *ORADB* from the *Database server* pull-down menu. You now have two options to create the *SCORED* table:

1. Type *ANALYST* in the *Schema list*, and enter *SCORED* in the edit box under the *Database tables and views* list. Enter the *server mapping* name (*DJORA8*) in the *Tablespace* field. This will create the *SCORED* table in the Oracle database. To have the table created in the Oracle database, you must define a user mapping within DataJoiner to satisfy the Oracle authorization check. In our example we created this user mapping:

   ```
   create user mapping from analyst to server djora8 authid analyst
   password analyst
   ```

   All names entered in the *Database Table or view* screen should be typed in upper case. If the table name is entered in lower case, it will be difficult to access this table from the DB2 Command Line Processor, because the DB2 CLP converts all SQL statements to upper case and thus will not find the lower-case table name. If the tablespace is entered in lower case, the table creation will fail because DataJoiner will not find the server mapping.

   See 11.2.1, “Current Functional Limitation” on page 159, for the IM fix required to create output tables directly into the Oracle database.

2. Select *DMINER* in the *Schema list* and enter *SCORED* in the edit box under the *Database tables and views* list. Use upper-case characters to specify the table name. If the table name is entered in lower case, it will be difficult to access this table from the DB2 Command Line Processor, because the DB2 CLP converts all SQL statements to upper case and thus will not find the lower-case table name. If you leave the *Tablespace* field empty, the *SCORED* table will be created as a local DataJoiner table in the default tablespace. Since this table is an intermediate table in the mining process and should not be required by production systems running off the Oracle database, this should not be a matter for concern.

With both options, select the *Read and write* option under *Use mode* and then select the *The specified table does not yet exist* option. Then click on the *Next >* button to continue.
In the Summary screen, click on the Finish button to continue. You are now back in the Output data screen of the Mining - Taskguide window. Select the Scored customer data object under Available output data and click the Next button. Make sure that the Run this settings immediately option is selected and click the Finish button to apply the model on the CUSTOMER table and generate the SCORED table.

Figure 54. Progress Screen for Applying Model
6.4.4 Generating the Target Customer Set

This is the last stage of the mining process. In this step, we select those records in the SCORED table for which the value in the SCORE field is greater than 0.7 and store them in a table called TARGET. In this manner, we eliminate those customers from the target set who do not fit strongly into any of the clusters generated in the model.

From the IM client main screen, click on the **Create statistics** button on the toolbar. The **Welcome** screen of the **Statistics - Taskguide** window appears. Click on the **Next >** button to continue.

In the **Statistics function and settings** screen shown in Figure 55, select **Bivariate Statistics** under **Available statistics functions** and enter **Analyze** as the **Settings name**. Select the **Show the advanced pages and controls** option, then click on the **Next >** button to continue.

![Figure 55. Statistics Functions and Settings Screen](image)

In the **Input data** screen (Figure 56 on page 98), select the **Scored Customer** data object under **Available input data**. Click on the **button against **Filter records condition** in the **Advanced** section.
This will open the Expression Builder window (Figure 57 on page 99) where you tell IM that statistical analysis needs to be conducted only on those records for which the value of the SCORE field is greater than 0.7.

Execute the following steps to achieve this:
1. Click on the AND button.
2. Select Field Names under Category.
3. Select SCORE under Value.
4. Click on the Arg1 button.
5. Click on the > button.
6. Select Constants under Category
7. Double-click on <new constant> under Value.
8. Type in 0.7 in the edit box that appears and hit Enter.
9. Select 0.7 under Value.
10. Click on the Arg2 button.
11. Click on the OK button to continue.
The `Filter records condition` in the `Advanced` section of the `Input data` screen should now read `((SCORE > 0.7))`.

Click on the **Next >** button to get to the `Parallel parameters` screen, where you again should click on the **Next >** button to continue.

In the `Statistics` screen (Figure 58 on page 100), select the `Compute statistics` radio button. Select all the fields in the `Available fields` section by clicking on the ![Add Fields button](image).
Figure 58. Statistics Screen

Click on the **Next >** button to continue. You can skip the **Quantiles** and **Samples** screens. Just click on the **Next >** button in each of these screens.

In the **Output fields** screen (Figure 59), select the **Create an output table** radio button and click on the **>>** button to include all the fields in the output table, then click on the **Next >** button to continue.

Figure 59. Output Fields Screen for Statistical Analysis
In the Output data screen, click on the Create data button to create a data object for the output table. In the Welcome screen of the Data Taskguide window, click on the Next button to continue.

In the Data format and settings screen (Figure 60), select Database Table/View and enter Target Customer under Settings name. Select the Show the advanced pages and controls option and click on the Next button to continue.

This time the resulting table will be created in the Oracle database to allow further analysis of the mining result with existing Oracle database query tools.

![Figure 60. Data Format and Settings Screen for Output of Statistical Analysis](image)

In the Database table or view screen (Figure 61 on page 102), select ORADB from the Database server pull-down menu. Select ANALYST in the Schema list, type TARGET in the Database tables and views list and select the The specified table does not yet exist option. Enter the server mapping name, DJORA8, in the Tablespace field and click the Next button to continue. Remember to use upper-case characters for the table name and tablespace.
You can click on the **Next** button to skip through the rest of the screens without changing anything. In the Summary screen click on the **Finish** button to continue.

You are now back to the **Output data** screen (Figure 62) of the **Statistics-Taskguide** window. Select the **Target Customer** data object under the **Available output data** and click on the **Next** button.

**Figure 61. Database Table or View Screen for Output of Statistical Analysis**

**Figure 62. Output Data Screen for Statistical Analysis**

102 Mining Relational and Nonrelational Data with IM for Data
In the Results screen (Figure 63), enter Target Customer Demographics for Results name and select the If a result with this name exists, overwrite it option, then click Next > to continue.

In the Summary screen, ensure that the Run this settings immediately option is selected and click on the Finish button to generate the target customer table and the statistical distribution for the same.

While the mining process is running, you will see a Progress window which will look like Figure 64 on page 104 once processing is complete.
IM then opens a visualization window (Figure 65) to show the distributions of all the fields in the TARGET table.

Figure 64. Progress Screen for Statistical Analysis

Figure 65. Variable Distributions for Target Customers
You can double-click on any of the pie charts or histograms to see the distribution for only that particular field.

In our scenario, the target customers are almost equally distributed among the other product types that the bank has. They are mostly women in a lower age category. Also, most of them fit into either cluster 1 or cluster 6 of the model we generated from the Premier Checking Account customers.

The \textit{TARGET} table on Oracle can now be used by the marketing department of the bank to design target marketing campaigns for the Premier Checking Account.
Part 3. Accessing Nonrelational Data
Chapter 7. System Architecture for Mining ODBC Data Sources

In today's business environment, high availability is imperative for core business applications. Faster and more predictable time to market is key, and in the world of e-commerce systems, quality is mandatory.

In the Windows environments ODBC allows applications to access databases and data formats through a standard API.

ODBC is a widely accepted application programming interface for database access. It is based on the Call Level Interface (CLI) specifications from X/Open and ISO/IEC for database APIs and uses SQL as its database access language.

7.1 Call Level Interface

CLI is an application programming interface for database access that uses function calls to invoke dynamic SQL statements. It is a programming interface for the C programming language especially for developing database applications in a client/server environment.

It was designed to support SQL access to databases from shrink-wrapped application programs. CLI was originally created by a subcommittee of the SQL Access Group (SAG). The SAG/CLI specification was published in 1992. In 1993, SAG submitted the CLI to the ANSI and ISO SQL committees.

SQL/CLI provides an international standard implementation-independent access to SQL databases. Client/server tools can easily access databases through DLLs. It supports and encourages a rich set of client/server tools.

7.2 Open Database Connectivity

ODBC 3.0 is a superset of the X/Open and ISO/IEC CLI specification for database APIs.

ODBC is an industry standard API for accessing data in both relational and nonrelational database management systems (DBMSs). Using ODBC, applications can access data stored in a variety of personal computer, minicomputer, and mainframe DBMSs, even when each DBMS uses a different data storage format and programming interface. Using ODBC, applications do not need to include specialized code for each DBMS. Instead, they use the ODBC interface and can access any ODBC-compliant database.
Thus developers can build and distribute a client/server application without targeting a specific DBMS or having to know specific details of various back-end data sources. When an application needs to get data from a data source, the application sends an ODBC function call to the ODBC Driver Manager (administered through an optionally installable 16- or 32-bit control panel), which then loads the ODBC driver required to talk to the data. The driver then translates the ODBC calls sent by the application into the SQL used by the DBMS and sends it to the back-end database. The DBMS retrieves the data and passes it back to the application through the driver and the Driver Manager (see Figure 66 on page 110).

Figure 66. ODBC Functional Components

You can code directly to the ODBC API by declaring various functions and then using them, for example, to connect, send SQL statements, retrieve results, get errors, and disconnect.

7.3 Mining ODBC Data Sources with IBM’s Intelligent Miner for Data

IM is the premier data mining product from IBM. It consists of:
• A server product that provides the engine for running statistical, data processing, and mining functions
• A client product that provides a Java GUI for setting up and executing these functions
• A set of C++ APIs for developing data mining applications using these functions

For a detailed description of the product, please refer to Chapter 2, “Introduction to Intelligent Miner for Data and DataJoiner” on page 13.

IM can use flat files or databases managed by the IBM DB2 family of products as data sources for mining. IM can also store the output data created by its statistical, data processing, and mining functions in flat files or DB2 databases.

So, in order for IM to use an ODBC datasource as mining source and as target for output data, either of two approaches can be taken:

• Transfer data from the ODBC data source into flat files, then execute all necessary data processing, statistical, and mining functions. Finally create a table from the final output data stored on a flat file in the ODBC datasource. Figure 67 represents this approach graphically.
• Make ODBC data sources appear as DB2 tables to IM and use the same as data sources and output targets. Figure 68 on page 112 represents this approach graphically.

![Figure 67. Using Flat Files to Mine ODBC Data Sources](image-url)
7.3.1 Using Flat Files with ODBC Data Sources

The first action to take when using flat files as input for IM is to create them. Use the front end shipped with the ODBC data source installed or any other standard ODBC interface. All of the standard ODBC front ends support undelimited ASCII as export or import file format, thus allowing IM to use these flat files as mining source.

Once the mining is done and an output file is produced, the file has to be transferred to the Database Server machine and the output data loaded back into the ODBC data source. The output data from data mining algorithms usually has more fields than the input data. So, you have to create a table in the ODBC source to hold the output data and then use a standard tool to load the data into this table.

7.3.2 The DataJoiner Interface to ODBC Data Sources

The DataJoiner software is part of the IBM Data Management family of products. It allows users and applications to transparently communicate with multiple and diverse data sources just as if they were communicating with a DB2 database. It does this by creating aliases or nicknames in a DB2 database catalog which refer to tables or joins of tables that actually reside in other databases. Applications can then treat these nicknames just like DB2 tables and submit SQL commands against them. DataJoiner translates these SQL commands into the format recognized by the database(s) where the tables actually reside, sends the statement to these database(s), and relays the result of the SQL command from these database(s) back to the application. For further information about the DataJoiner product, please refer...
to Chapter 2, “Introduction to Intelligent Miner for Data and DataJoiner” on page 13.

In the current scenario, DataJoiner serves to catalog ODBC data sources as nicknames which IM can access as DB2 tables. When IM issues an SQL command against these tables, DataJoiner first translates it to a format that can be understood by the ODBC data source using the ODBC driver. IM is unaware of the existence of the ODBC data source and communicates only with DataJoiner’s DB2 database. Please note that the actual data from the ODBC data source is not replicated into the DataJoiner DB2 database. Only references in the form of nicknames are stored in the DataJoiner DB2 database. However, a fully functional DB2 database is included with DataJoiner and applications such as IM can use this database for storing intermediate data tables created during the data mining process.

Figure 69 illustrates the data transfer process using DataJoiner. The dashed representation of the tables in the DB2 database indicate that the data does not physically exist in the DB2 database.

**Figure 69. Accessing ODBC Data Sources with DataJoiner**

### 7.4 System Architecture

Figure 70 on page 114 describes the environment that we set up to enable mining on data stored in ODBC data sources. The IM client interacts with the IM server to define and execute data mining exercises. The Intelligent Miner for Data server reads source data and writes output data into DB2 tables on DataJoiner. DataJoiner, in turn, maps these tables to source and target ODBC data sources and transfers data to and from the data source, using the ODBC
To describe the setup and configuration to access different data sources, we installed the software shown in Figure 70 at the indicated service levels, on a Windows NT workstation.

Figure 70. The ODBC Environment
Chapter 8. Access SPSS Data from Intelligent Miner for Data

IM can read and write SPSS data files (that is, SAV files) through the DataJoiner generic ODBC DAM in conjunction with the SPSS ODBC driver. In this chapter we describe the setup to access SPSS data sources from IM and have the SPSS data sources appear to IM like a DB2 database table.

8.1 SPSS ODBC Configuration

To configure SPSS data sources:

- Select **Start -> Settings -> Control Panel**, open the **32-bit ODBC Data Source Administrator**, select the **System DSN** tab and click the **Add** button (see Figure 71).

![Figure 71. ODBC Data Source Administrator](image)

- Select **SPSS Data Driver 32** (see Figure 72) from the **Create New Data Source** dialog and click **Finish**.
Mining Relational and Nonrelational Data with IM for Data

Figure 72. Create New Data Source

- On the Setup dialog (Figure 73) enter the Data source name and database (the directory or folder that contains the SPSS data files) and press OK.

Figure 73. SPSS Data Source Setup

- The SPSS data source should be listed in the System DSN tab (Figure 74 on page 117) and is ready to be configured in DataJoiner.
8.2 SPSS - DataJoiner Configuration

To configure the SPSS ODBC data source in DataJoiner:

- Select **Start -> Programs -> DataJoiner for Windows NT -> Command Window**. If there is not already a DataJoiner DB2 instance on the database server, create one with the `db2icrt <instance-name>` command. In Figure 75, **DJINST1** is used for the instance name.

- Start the database manager with the `db2start` command.

![Figure 74. SPSS System Data Source](image)

![Figure 75. Creating a DB2 Instance](image)
• Create a database and connect to it with these commands (Figure 76):

```
db2 create database spssdata
db2 connect to spssdata
```

![Figure 76. Connecting to the Database](image)

• The following command maps the SPSS ODBC data source to the DataJoiner catalog:

```
db2 create server mapping from spsserv to node "spssodbc" type generic version 2.1 protocol "generic"
```

The full syntax for this command is reprinted from the *DB2 DataJoiner Version 2 API/SQL Reference Supplement*:

```bash
>>-CREATE SERVER MAPPING FROM-server-name-TO NODE-"node-name"->

>---+-----------------------------------+--TYPE--server-type--->
| +-----DATABASE--"remote-database-name"---+

>---VERSION--server-version--PROTOCOL--"protocol-name"-------------->

>---------------------------------------------------------------------------+

  +---CPU RATIO--value---+---IO RATIO--value---+---COMM RATE--value---+
  | AuthID--name--password--| | AuthID--name------------------|

• To map the user to the server, issue this command:
db2 create user mapping from imres2 to server spsserv authid " " password " "

The full syntax for this command is:

```sql
>>CREATE USER MAPPING FROM--+-local-authid--+-user--+-
                     +--user--+-

>-TO SERVER--server-name--AUTHID--remote-authid--

>+--password--remote-password--+-CONNECTOPT--string--+
```

• Create nicknames for the SPSS data files to be used by IM:

```sql
db2 create nickname credit for spsserv.credit
```

The full syntax for this command is:

```sql
>>CREATE NICKNAME--nickname--FOR--remote-object-name--
```

At this point DataJoiner connects through ODBC to the SPSS data source and collects attributes for the global catalog.

To test the ODBC connection from the Command Window (Figure 77) issue any select statement against the table or data file; for example:

```sql
select count(*) from credit
```

![Figure 77. Testing the SPSS ODBC Connection](image-url)
The Microsoft ODBC specification defines three function levels: CORE, LEVEL1, and LEVEL2. The SPSS ODBC driver supports CORE and LEVEL1 functions with some limitations. These are documented in the SPSS ODBC Driver Version 1.1 document on the installation disk.

The Microsoft ODBC specification also defines three levels of support for SQL grammar: MINIMUM, CORE, and EXTENDED. The SPSS driver provides support for the MINIMUM SQL statements with restrictions documented.

The SPSS ODBC driver supports all MINIMUM and most EXTENDED SQL data types.
Chapter 9. Access SAS Data from Intelligent Miner for Data

IM can read SAS data sets,¹ and any data accessible through SAS/ACCESS,² through the DataJoiner generic ODBC DAM. The DAM can be configured to use the two licensed ODBC drivers provided by the SAS Institute:

- The standard SAS ODBC driver that comes with the product provides access to SAS data sets and data libraries³ locally through DDE, or remotely via TCP/IP or Network DDE. Remote access additionally requires Base SAS, SAS/SHARE, SAS/SHARE*NET, and SAS/ACCESS for each DBMS on the remote host.
- The Universal SAS ODBC driver provides access to individual data sets and does not require the SAS System software. A trial version of the Universal driver is distributed through the SAS Web site at http://www.sas.com.

**NOTE:** The SAS SQL procedure (PROC SQL), used by the server for the standard ODBC driver, does not support the NULL/NOT NULL column attribute common to most relational databases. This limitation causes errors when querying SAS data sets with missing values (NULLs), and when saving data to SAS. The Universal driver uses a third-party SQL parser that supports nullable columns and does not rely on SAS to process SQL requests.

We used both the standard and universal driver in conjunction with DataJoiner and Intelligent Miner in a Windows NT environment.

### 9.1 SAS ODBC Configuration

To configure the SAS ODBC data source in Windows NT, follow these steps:

1. Select **Start -> Settings -> Control Panel**, open the 32-bit ODBC Data Source Administrator, and select the **System DSN** tab.

2. Click the **Add** button and select the driver from the Create New Data Source dialog.

   Depending on the SAS driver you select, a dialog will display when you click the **Finish** button.

---

¹ SAS data sets are a proprietary, nonrelational file format. Physical and logical data sets are accessible to ODBC client applications such as DataJoiner.

² DataJoiner provides native support for many of the databases for which SAS/ACCESS provides individually licensed interfaces.

³ A logical reference (libref) to the physical location (directory, minidisk, or host system data set) of a collection of SAS data files and views.
1. SAS Universal ODBC Setup
   • On the SAS Universal ODBC Setup dialog enter SASUODBC as the Data Source Name and select the SAS data set you want to use by clicking the Browse button (see Figure 78).

   ![SAS Universal ODBC Setup](image)

   Figure 78. SAS Universal ODBC Setup

   • Click OK when you are finished. The SAS data is now ready to be configured in DataJoiner.

2. SAS ODBC Driver Configuration
   • The SAS ODBC Driver Configuration dialog will open to the General tab. Enter the Data Source Name and Description as illustrated in Figure 79 and select the Servers tab.

   ![SAS ODBC Driver Configuration - General](image)

   Figure 79. SAS ODBC Driver Configuration - General

   • On the Servers tab (see Figure 80 on page 123) enter the Server Name (SASSERV) and click Configure. For local data access leave the Access Method to dde.
Check whether the parameters are correct, the \textit{Local DDE Options} dialog (Figure 81), and click on the \textit{Servers} tab. The server you defined should appear in the \textit{Servers} list box, then select the \textit{Libraries} tab.

![SAS ODBC - Servers](image)

\textbf{Figure 80. SAS ODBC - Servers}

On the \textit{Libraries} tab (Figure 82 on page 124) fill in the \textit{Library Name}, \textit{Host File Name}, and \textit{Description} fields and click.

![Local DDE Options](image)

\textbf{Figure 81. Local DDE Options}
1. Select the General tab and OK the SAS ODBC Driver Configuration dialog.

3. The new system data source should appear in the ODBC Data Source Administrator control panel (Figure 83) and is ready to be configured in DataJoiner.
9.2 SAS - DataJoiner Configuration

To configure the SAS ODBC data source in DataJoiner:

1. Select Start -> Programs -> DataJoiner for Windows NT -> Command Window and create a DataJoiner database instance called DJINST1, if one does not already exist, by submitting this command:

   \texttt{db2icrt djinst1}

2. Start the database manager with the \texttt{db2start} command.

3. Create a database and connect to it with these commands:

   \texttt{db2 create database sasdata}
   \texttt{db2 connect to sasdata}

4. Map the SAS ODBC data source to the DataJoiner catalog:

   \texttt{db2 create server mapping from sasserv to node "sasodbc" database "datalib" type generic version 2.1 protocol "generic"}

5. Map the user to the server:

   \texttt{db2 create user mapping from imres2 to server sasserv authid " " password " "}

6. Create nicknames for the SAS data sets you want to use from the data library:

   \texttt{db2 create nickname asia for sasserv.datalib.asia}

For local data sources the SAS ODBC driver will invoke SAS and run PROC ODBCSERVER, through which all requests are routed, in the background.

For SAS data sources accessed through the Universal ODBC driver the final three steps above are replaced by the following:

- Map the SAS Universal ODBC data source to the DataJoiner catalog:

   \texttt{db2 create server mapping from sasuserv to node "sasuodbc" database "sasuodbc" type generic version 2.1 protocol "generic"}

- Define the user mapping:

   \texttt{db2 create user mapping from imres2 to server sasuserv authid " " password " "}

- Create the data source nickname:

   \texttt{db2 create nickname asia2 for sasuodbc.asia}
Note: If you are using a trial version of the Universal driver, you may need to OK the trial dialog (Figure 84) when an ODBC connection is made.

![Figure 84. Trial Driver Expiration](image)

To test the ODBC connection, submit any select statement against the table or data set from the Command Window, for example:

```
select count(*) from asia
```

### 9.3 SAS Server Communications

In this section we discuss the prerequisites for SAS server communications through the standard ODBC driver. The Universal driver operates independent of the SAS System and not through a SAS server process. Hence this section does not apply to client access through the Universal driver.

#### 9.3.1 Local DDE Access Method

Base SAS is required to run PROC ODBCSERVER for local data sources accessed through the standard ODBC driver’s DDE access method. If the server process is not executing when an ODBC request is made, the driver will start a SAS session with the Local DDE Options dialog (see 9.1, “SAS ODBC Configuration” on page 121) parameter settings.

If a SAS session is already open, the following statements can be submitted from the PROGRAM window to initiate the server:

```
options comamid=dde;
proc odbcserver id=server-name;
run;
```

The server-name must be the same specified on the SAS ODBC Driver Configuration: Servers tab. While the server is executing, the SAS session will not accept keyboard input.
To terminate the server, disconnect from the database and stop DB2 before ending SAS from the Windows Task List.

9.3.2 Remote Access Methods

For remote data sources the ODBC driver uses a SAS/SHARE server. This requires both SAS/SHARE*NET and SAS/SHARE software on the remote host. The SAS/SHARE server should be invoked on the remote host at system startup. The following describes the configuration using either TCP/IP or Network DDE:

• TCP/IP

If the TCP/IP communications method is to be used, PROC SERVER should be started on the host machine as follows:

```plaintext
options comamid=tcp;
proc server id=server-name;
run;
```

The server(s) should be defined in the TCP/IP services file for the client machine(s). Entries in this file associate a service name with the port number and communications protocol used by the service, which in this case is the name of the SAS/SHARE server, for example:

```plaintext
<server-name> <port number>/tcp # <comments>
```

The server name should be the same specified in the SAS ODBC Driver Configuration: Servers tab. This is case sensitive and must be 1-8 characters long.

• Network DDE

In the case of the Network DDE access method, a share must be defined under Windows NT by executing DDESHARE (see Figure 85).

```
Figure 85. NT DDESHARE
```

Select Shares -> DDE Shares and click Add a Share... .

On the DDE Share Properties dialog (Figure 86 on page 128) enter the Share Name (by convention DDE share names end with a dollar sign), Static (this should be the same name specified for the ID on the PROC
SERVER statement), and the Topic Name=SAS/SHARE on the same line as the PROC SERVER ID. Click the Permissions button.

Figure 86. DDE Share Properties

On the DDE Share Name Permissions dialog (Figure 87) select Everyone from the Name list, set Type of Access to Full Control and OK the DDE Share Name Permissions and DDE Share Properties dialogs.

Figure 87. DDE Share Name Permissions

On the DDE Shares dialog (Figure 88 on page 129) select the share you defined and click .
On the Trusted Share Properties dialog (Figure 89) check the Start Application Enable and Initiate to Application Enable checkboxes. Click on Set and OK to save the properties, then OK the DDE Shares Dialog.

Select Shares -> Exit from the DDE Share window.

On the ODBC client workstation define the data sources on the remote machine. The server name in the illustrated example would be CANDEMAS.SHR$1, where CANDEMAS identifies the remote machine, and SHR1$ is the DDE share name defined on that machine.

9.3.3 SAS ODBC Specification Compliance Levels

The SAS ODBC driver provides a callable entry point for all CORE and LEVEL1 ODBC functions. Some LEVEL1 functionality is not fully implemented.

Through the standard driver SAS supports all MINIMUM and some CORE SQL statements.
Function and grammar exceptions are documented in the "Programmer Reference" section of the SAS ODBC Driver Technical Report. This report documents data type mapping and SQLSTATE return code support.
Chapter 10. Sample Mining of ODBC Data Sources

In this chapter we cover the installation of IM in a Windows NT environment and provide two sample data mining exercises using ODBC data sources accessed through DataJoiner. The first example uses the Home Sales data from SPSS, and the second, the loan application data from SAS.

We define a goal for our data mining operations and, using the IM functionality, we show how we can achieve it. The process of building a decision support system passes through some of the most useful of the data preparation functions, demonstrating the abilities of the IM in the area of data management. Then we build a number of different data models based on the cleansed data and choose the one that satisfies our requirements. Finally we specify a sequence of data preparation and mining functions to automate the data mining operation.

10.1 Intelligent Miner Server Installation on Windows NT

Intelligent Miner for Data consists of two components:

• IM server - This is the processing engine which runs the mining, data processing, and statistical algorithms.
• IM client - This is the Java interface that is used to set up mining runs.

In a Windows environment the IM server software can be installed on a Windows NT server or workstation, with Windows NT or Windows 95 clients.

DB2 must be installed on the NT server before IM is installed. If you attempt to install DB2 on a machine on which IM is already installed, the DB2 installation might fail.

To install the IM server on Windows NT:

1. Run the Setup program in the IMServer212 directory and click Next > on the Welcome window (Figure 90 on page 132) to step through the installation process.
2. Select the components you want to install (see Figure 91), then click the **Next >** button. In this example we are installing the server and client on the same machine.

3. Enter the domain and user ID from which you will run the server process (see Figure 92 on page 133) and then click **Next >**. Select how you want
to configure your mining bases (common or individual) and their directory location before the installation proceeds.

![Image: Intelligent Miner Install: Domain/User](image)

**Figure 92. IM Install: Domain/User**

4. When the installation is complete view the README file and restart your machine.

The client installation program also installs the Java Runtime Environment (JRE) required. If you want to view the online help, you will need to install a Web browser which supports frames such as Microsoft Internet Explorer Version 3.0 (or higher) or Netscape Navigator for Windows 95/NT Version 3.01 (or higher).

Please refer to the *Using the Intelligent Miner for Data* manual which is supplied with the IM product for more detailed instructions on installation and setup. Also, please read the README.TXT file which is present in the server and client product installation CD-ROMs for additional information before you install the product.

### 10.2 Starting the Intelligent Miner Server and Clients

To start the IM server on the NT machine, log in with the server user ID specified during the installation process, select **Start -> Run**, type `Idmstart` into the text entry field and select **OK** (see Figure 93 on page 134).
To start the IM client on a Windows NT machine, select **Start -> Programs -> Intelligent Miner -> Intelligent Miner**. When started for the very first time, this will start the **Preferences** screen as shown in Figure 94.

**Figure 94. IM Preferences**

### 10.3 Home Sales Mining Scenario

In our environment, we implemented a hypothetical scenario to illustrate the process of data mining an SPSS ODBC data source.
The SPSS data used in this example contains land and home values for a group of neighborhoods. Our goal is to create a data mining operation to help us forecast or predict home sale prices based on this historical data.

The model for the available data describes hidden patterns or trends in the historical data. Based on these trends the model generalizes over unseen input data, providing a predictive tool for the analyst.

10.3.1 Implementing the Solution with Intelligent Miner

To implement the above solution, the following tasks need to be performed in Intelligent Miner:

- Create the data object
- Visualize and prepare the data
- Model the data
- Apply the data model
- Automate the process

10.3.1.1 Create the Data Object

After you have started the server, opened the client window and connected to the server through the Preferences dialog (see 10.2, “Starting the Intelligent Miner Server and Clients” on page 133), click the Create Data button from the toolbar on the Intelligent Miner main window (Figure 95 on page 136).
This will open the **Welcome** screen of the *Data - Taskguide*.

Click **Next >** to open the *Data format and settings* screen shown in Figure 96 on page 137. Select *Database Table/View* from the *Available Data Formats* group box, enter the settings name (*Home Sales Data* in our example) and click the **Next >** button.
In the *Database table or view* screen select *SPSSDATA* from the *Database Server* list, click on the schema and name of the SPSS data file, and click **Next >**.

On the *Field Parameters* dialog (shown in Figure 97) you can choose to change the data types of the columns.

![Data TaskGuide Field Parameters](image)
Click **Next >** on all screens until you reach the *Summary* window (Figure 98).

![Figure 98. Data TaskGuide Summary](image)

Click **Finish**.

### 10.3.1.2 Visualize and Prepare the Data

To prepare the data we must understand it. It is crucial to identify any missing values, outliers, and abnormal frequency charts that may negatively influence the data modeling process.

For this purpose we can create a bivariate object to generate frequency charts and descriptive statistics of our variables.

First create a bivariate object, using the *Home Sales Data*:

1. From the *IM Main window*, select *Create statistics*.
2. Click **Next >** on the *Welcome screen* of the *Statistics - TaskGuide*.
3. Select *Bivariate statistics* and enter a *Settings name*, then click **Next >**.
4. Select the *Home Sales Data* from the *Available input data* screen and click the **Next >** button.
5. Skip the *Parallel parameters* screen by clicking the **Next >** button.
6. In the *Statistics* screen select the *Compute statistics* button and select all the variables contained in the data source for calculating the descriptive statistics as illustrated in Figure 99 on page 139.
7. Skip all subsequent screens by clicking the **Next >** button until you reach the **Result** screen.

8. Enter a **Result name** and click the **Next >** button.

9. On the **Summary** screen select the **Run this settings immediately** option and click the **Finish** button.

---

Figure 99. *BivariateStatistics*

After executing the bivariate statistics object we can check the results generated. Figure 100 on page 140 illustrates the frequency charts of the underlying variables.
As you can see some of the variable frequencies are skewed to the left. To introduce normality to these variables we need to perform a nonlinear transformation. To do this, open the data object that you have created, go to the Computed Fields tab (see Figure 101), and add a new variable, using a logarithmic function for the variable you want to transform.
Then open the bivariate object, go to the Statistics tab, and add the variable that you created to the list. Finally, reexecute the bivariate object to see the new results as shown in Figure 102.

![Visualizing Prepared Data](image)

*Figure 102. Visualizing the Transformation*

### 10.3.1.3 Model the Data
To build a data model we use the Prediction Mining Object (Neural) in train mode (see Figure 103 on page 142).

The input variables are given in Figure 97 on page 137. The optimization for the neural network is set to be selected automatically by IM.
10.3.1.4 Check the Results

Finally we can execute the prediction object to generate our results. We can use the result object to visualize the performance of the neural network. The result generated is partially illustrated in Figure 104 on page 143.
10.3.1.5 Apply the Data Model

After visualizing the results and determining the performance of the neural network we compare its performance with our success expectations. If we are satisfied, the model can proceed. At this point we use the result object that the train mode generates as input to a prediction object running in application mode (Figure 105 on page 144).

In addition we have to specify an output file that will hold the predictions and confidence limits for new input examples (Figure 106 on page 144).
Figure 105. Neural Prediction Settings, Mode Parameters

Figure 106. Neural Prediction Settings, Output Fields
### 10.3.1.6 Automating the Process

Having completed each step of the data mining operation we may want to create a sequence of the individual steps. The data preparation and mining function sequence will provide us with an automated analysis tool ready for business use. Figure 107 shows how we can define a data mining operation using the sequence object.

![Sequence TaskGuide](image)

**Figure 107. Sequence TaskGuide**

### 10.4 Loan Approval Mining Scenario

In this sample mining operation we address the challenge of building a decision support system dedicated to loan assessment. The decision support system must provide the human decision maker with an evaluation of the loan application based on historical data. The outcome of the evaluation in the most simplest form will be a negative or a positive answer to the question: *“Will this person be able to repay the loan?”* The historical data will consist of old application records in which the outcome of this question is known.

In applying data mining to satisfy this requirement our goal is to build a profile of an applicant that is likely to default. Having done that we can then evaluate
any new customer application by comparing the applicant’s profile with the one
that we have built.

Applying data mining to building a decision support system dedicated to loan
assessment requires us to define a sequence of data preparation and data
mining functions that will bring us from the historical data to a valuable data
model. The data model will represent any hidden patterns in the available
data, capturing the behavior of the customers and becoming the core of our
decision support system.

10.4.1 Implementing the Solution with Intelligent Miner

We start our data mining operation by selecting the historical data needed. To
illustrate the process of accessing data alien to the IM we use data that is
provided in the SAS data format. For this task we use DataJoiner, so that we
can use the historical data set in IM like a DB2 table.

Following the data collection process we concentrate on data preparation
issues. First we investigate the observations of the data set for missing
values, outliers, and heavy tailed frequency variable charts, using the
Intelligent Miner's statistical functions. Then we apply a series of data
preparation functions depending on the properties of the variables, making
sure that the available data is ready for the data modeling process.

After delivering a cleansed data set we split it into two subsets, using random
sampling. The purpose of this task is to provide our data mining functions with
two distinct data sets - the training/validation data set and the test data set.
The training/validation set will be used to actually build the data model(s),
and the test set to finally evaluate it over previously unseen patterns.

Our concern is to use the appropriate data mining function in delivering a
robust data model that will explain or predict the behavior of past loan
applicants. IM provides two methods for performing this classification task
- decision trees and artificial neural networks. As we know, a decision tree
supplies us with a set of rules on which we can make our decisions. An
artificial neural network only gives us a classification decision. If we are not
required to explain the reasons of our classification action, we can use both
tools to fit the available data and finally choose the one that gives a model
with a confusion matrix that best meets our requirements.

Finally, after completing each step, we have to construct a sequence of the
individual functions to automate our data mining operation. This sequence
can then be deployed online and become part of the business decision
support system.
10.4.1.1 Select the Data

To build the profile of a typical applicant that is going to default we require a set of records that capture features relevant to loan applications. The old records that we are going to use must contain a combination of variables that present the most interesting characteristics such as the amount of loan applied for, the credit history of the person, and his or her current financial status.

For our convenience a data set that meets our requirements is hosted in a SAS system. Figure 108 illustrates a sample of this data and Figure 109 shows a brief description of the underlying variables.

10.4.1.2 Visualize the Data

After collecting the data necessary, we prepare it for the data mining functions. However, before conducting any data preparation it is useful to perform data visualization to understand the properties of the underlying variables. Visualization enables us to see the condition of our data set; if there are any missing values, outliers, and if the variables follow normal probability distribution.
To perform data visualization with IM we create a data settings object to host the data set. Then, using a bivariate statistics object, we generate frequency charts and other descriptive statistics for the variables contained in the original data object.

Figure 110 presents the frequency charts of all the variables. As you can see there are variables with heavy tailed charts and variables that contain missing values. In addition, a close examination of a variable’s descriptive statistics provides us with a more detailed picture of the properties of the underlying variable.

Figure 110. Data Visualization: Frequency Charts

10.4.1.3 Encode Missing Values
A number of different methods are available for encoding missing values. A simple method is to impute missing values, using our domain knowledge or the insight to the data that we have gained through visualization. For instance, missing values for a variable can be filed using the sample mean of the variable or the level with the highest frequency. Information about the variable sample mean and frequencies can be found in the bivariate object that we created. To use this information to encode missing values we create an Encode Missing Values object in IM and impute missing values, as Figure 111 illustrates.
The output of this process is a new data table that holds all old variables as well as the variables we have created by imputing the missing values. We can then use this table for further analysis by creating a new data object.

10.4.1.4 Define New Variables
In data mining operations it is often necessary to create new variables as linear or nonlinear combinations of existing variables in order to derive additional information. In our operation it will be useful to create a new variable representing the ratio between loan and value. For this task we use the calculate values object shown in Figure 112 on page 150.
10.4.1.5 Create, Train/Validate, and Test Data

Finally, having a data set that has been optimized for data mining, we can split it into two subsets - one for training/validation and one for final testing.

First, we randomly sample the table which contains the original variables, the nonmissing values variables, and the loan to value ratio variable. For this purpose a random sample object is created. This object creates a 20% sample subset of the original data forming the test data set.

Then we extract the test set from the original data, using a Filter Records object. Figure 113 on page 151 illustrates part of the process which extracts the sample data from the original data set in order to form the test/validate data set.
10.4.1.6 Normalize Variables

During the process of creating the train/validate and test data objects, we try to introduce normality in our variables by performing a power transformation. As we have noted, the loan variable is skewed to the left. We can normalize the frequency chart of this numeric variable by creating a new variable, \textit{power\_loan}, which is the logarithmic function of the original variable. Figure 114 on page 152 illustrates this process.
As you can see in Figure 115 the power transformation has introduced normality into the loan variable frequency chart.
10.4.1.7 Model the Data
Using IM's Classification Task Guide we can build two data models, one based on neural networks and one based on decision trees. First we create a neural network mining object based on the training/validation data. The model is constructed in training mode, using an architecture of one hidden layer with seven nodes (see Figure 116).

![Figure 116. Neural Classification Results](image)

Finally we build a tree classifier, using the same input variables, target and train/validation data set. The results of this classification model are given in Figure 117 and Figure 118.
As you can see from Figure 117 and Figure 118, the neural network performs better than the decision tree in classifying good applicants (marked as 0). The decision tree has better overall performance, giving better classification rates over bad applicants (marked as 1).
At this point we can decide to use the decision tree model as the basis of our decision support system. However, this requires us to take our analysis one step further and measure the performance of the decision tree and neural network over the test data. This will be the final evaluation of the data model and will prove whether or not it can accurately generalize over new patterns.

Again, as Figure 119 illustrates, the decision tree model performs better overall, making it a more desirable candidate for a final implementation.

Figure 119. Test Data Classification Comparison

10.4.1.9 Create a Sequence

Finally we can create a sequence for the data preparation and mining functions that we require in order to automate the operation. Figure 120 on page 156 illustrates the object sequence.
Figure 120. Sequence Settings
Chapter 11. Summary

In Chapter 1 we give a brief introduction to data mining. We argue that data mining has proven itself to be a valuable data exploitation tool because it fully automates the data analysis and knowledge discovery process. Unlike traditional statistical analysis tools and online analysis processes, data mining generates and examines hypotheses automatically. In addition, the data mining algorithms can deal with a very large input space without requiring external guidance.

Data mining tools can perform a number of different functions, such as classification, prediction, clustering, association discovery, and similar time sequences. The deliverables of data mining functions can be a data model, a segmented data set, or a set of affinities. Ultimately, these deliverables, which encapsulate hidden information in the available data, will be used to deduce knowledge for business advantage.

Undeniably data mining is a data-driven process. The raw material is data from a number of databases that capture information about every aspect of a business transaction. In addition, data concerning demographics and life styles can be useful to a data mining operation. Therefore, there is a need to collect and integrate data in a single database dedicated to data mining.

It is true that diversity characterizes the IT infrastructure of most businesses. Databases provided by a number of different vendors hold the data needed in data mining operations. Usually, analysts spend a considerable amount of time searching for the data they need because there is no common interface among the different data sources. Thus they have little time to concentrate on the real issues of the data mining operation; their productivity suffers, and the final product of the data mining operation is of poor quality.

In this redbook we try to address this problem. We provide data mining workers with the means to consolidate different data types into a single database.

11.1 Environments Used

Throughout this book we discuss the mining of non-IBM data sources with IBM Intelligent Miner for Data. The software we use with all of the examples is IBM DataJoiner. The data sources are Oracle Version 8.0.4, representing mining relational data sources, and SPSS and SAS, representing the nonrelational data sources.
After we configured DataJoiner to access the data sources we used on the Windows NT platform, we could not access the data with IM. Whenever we tried to execute a mining run, IM returned an error indicating an incorrect release number of a bind file. There had been no problem during the definition of the mining source.

The reason for this behavior is that IM, in the version we used, requires the DB2 Universal Database as the underlying database system. As we used DataJoiner instead, the CAE bind files required by IM were missing.

The problem is easy to solve, however. All you have to do is bind the IM bind files from a DB2 UDB CAE against the DataJoiner database. The IM bind files are located in the bin directory of the IM installation and are called:

idmpmluc.bnd
idmpmrs.bnd
idmpsdiq.bnd
idmpsdur.bnd
idmpssql.bnd

To bind these files, the DataJoiner database and node have to be cataloged at the UDB CAE workstation. The commands to bind these files are:

DB2 CONNECT TO XXX
DB2 BIND x:\im\bin\IDMPMLUC.BND BLOCKING ALL GRANT PUBLIC
DB2 BIND x:\im\bin\IDMPMRS.BND BLOCKING ALL GRANT PUBLIC
DB2 BIND x:\im\bin\IDMPSDIQ.BND BLOCKING ALL GRANT PUBLIC
DB2 BIND x:\im\bin\IDMPSDUR.BND BLOCKING ALL GRANT PUBLIC
DB2 BIND x:\im\bin\IDMPSSQL.BND BLOCKING ALL GRANT PUBLIC

Note: This problem does not occur when DataJoiner is installed on an AIX operating system.

11.2 Sample Environment for Relational Data Sources

Figure 121 on page 159 shows the environment we set up to document IM access to non-IBM relational databases. Remember that we used Oracle. Any other RDBMS accessible through DataJoiner would work.
Here is a description of the three systems:

- Mining Client: This is the end-user workstation where the mining process is defined and initiated. Mining requests from this system are sent to the Data Mining Server.

- Data Mining Server: All mining actions are performed on this system. The database access required is directed to the local database server, DataJoiner in our example. As the data itself resides on a different system, DataJoiner acts as a client to the Database Server, translating the SQL statements into a syntax understandable by the target DBMS.

- Database Server: This system provides the input data for the actual mining process and keeps the result tables created during the data mining process.

Although it might appear otherwise when you read Chapter 4, “Install and Configure DataJoiner with Oracle” on page 39, the Oracle client does not have to be installed on the data mining client. We used this client only to verify the connectivity and to access the Oracle database natively so that we could check the results we produced inside the Oracle database.

### 11.2.1 Current Functional Limitation

During our tests we determined that everything related to database access works well as described in Chapter 3, “System Architecture for Mining Oracle..."
We discovered one problem, however:

As mentioned in 6.4.3, “Applying the Model” on page 90 and visible in Figure 53 on page 96, IM for Data allows a *tablespace* to be specified together with the output table definition. This should automatically force DataJoiner to create the output table in the Oracle database, using a command syntax of:

```
create table ... in tablespace
```

When this statement is issued using a CAE, this syntax works fine, as mentioned in 6.4.3, “Applying the Model” on page 90. When the entry field in the IM GUI is used, IM checks the DataJoiner system catalogs to find an entry for the specified tablespace. As there is no entry, IM returns an error stating that the tablespace specified could not be found and therefore the table will not be created in the Oracle database. We addressed this problem to the IM development lab and received a fix that allows IM to create Oracle output tables through DataJoiner.

This fix will be available with PTF U462271, which has a prerequisite of IM Version 2.1.3.

### 11.3 Sample Environment for Nonrelational Data Sources

The environment shown in Figure 122 on page 161 looks just like the one in Figure 121 on page 159. One obvious difference is that here the environment does not use database client software for the data source. The data access is handled through DataJoiner’s generic ODBC driver. This environment will work in exactly the same way for all data sources that provide an ODBC driver. The only difference will be the configuration of the ODBC driver itself, as explained in Chapter 8, “Access SPSS Data from Intelligent Miner for Data” on page 115 and Chapter 9, “Access SAS Data from Intelligent Miner for Data” on page 121.
Here is a description of the three systems:

- **Mining Client**: This is the end-user workstation where the mining process is defined and initiated. Mining requests from this system are sent to the Data Mining Server.

- **Data Mining Server**: All mining actions are performed on this system. The database access required is directed to the local database server, DataJoiner in our example. As the data itself resides on a different system, DataJoiner acts as a client to the Database Server, translating the SQL statements into a syntax understandable by the target DBMS. In this example, DataJoiner invokes the ODBC driver to connect to the ODBC data source.

- **Database Server**: This system provides the input data for the actual mining process and keeps the result tables created during the data mining process.

### 11.3.1 Current Functional Limitation

During our tests we determined that everything related to database access works well as described in Chapter 7, “System Architecture for Mining ODBC Data Sources” on page 109 through Chapter 10, “Sample Mining of ODBC Data Sources” on page 131.

We discovered the following problems, however:
11.3.1.1 SPSS As a Data Source
In our environment we were not able to write the output data or table to the SPSS data source because of an ODBC driver problem. This problem has been addressed to the ODBC driver manufacturer.

11.3.1.2 SAS As a Data Source
As described in Chapter 8, “Access SPSS Data from Intelligent Miner for Data” on page 115, SAS provides two flavors of ODBC drivers. The "full-function" driver exhibited problems when accessing data with NULL values. This seems to be due to a missing NULL indicator within the ODBC driver.

The Universal ODBC driver supports nullable columns but is limited to read only. Thus it does not allow creating the output table within the SAS data source or writing the result back to the SAS data source.
Appendix A. Scripts for Creation of Data Access Modules

This appendix provides a listing of the two scripts used to configure data access modules (DAMS) for DataJoiner.

These scripts, both of which are found in the /usr/lpp/djx_02_01/lib directory, are:

• djxlink.sh
• djxlink.makefile

A.1 The djxlink.sh Script

#!/bin/ksh

# (C) COPYRIGHT International Business Machines Corp. 1995
# All Rights Reserved
# Licensed Materials - Property of IBM
# US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Please read the DataJoiner Planning, Installation, and Configuration Guide before using this script.

FUNCTION: This shell script attempts to link all DataJoiner data access modules.

usage: djxlink.sh

Note: This makefile must be run under root from the /usr/lpp/djx_vv_rr/lib directory.

Subroute to handle mssql DAM link-edit

mssql()
{
    echo "\n**********************************************************************\n" >> /tmp/djxlink;
    if [[ "${DJX_ODBC_LIBRARY_PATH}" = "" ]]; then
        echo "Warning: Cannot prepare djxmssql Data Access Module for use';;
        echo "Problem: DJX_ODBC_LIBRARY_PATH environment variable not set';;
        echo "Corrective action: Please set DJX_ODBC_LIBRARY_PATH environment variable to';;
        echo "the path where your ODBC driver is installed';;
        echo "';;
        results=${results}'djxmssql\tFailure'\n        return 12;
    fi
    echo "Attempting to build djxmssql Data Access Module";
    echo "Attempting to build djxmssql Data Access Module" >> /tmp/djxlink;
```bash
/bin/rm djxmssql >>/dev/null
/bin/ld -o djxmssql \
  -e _no_start \n  -e: djxmssql.exp \n  -bE: djxmssql.exp \n  -bI: libdb2e.exp \n  -bM: SRE \n  -K \n  -lc \n  -L${DJX_ODBC_LIBRARY_PATH} \n  -l:ocdbc \n  libmssql1.a \n  libdbxixts.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
if [[ $rc -ne 0 ]]; then
  echo " Was not able to find link-edit parameters for djxmssql that would work";
  echo " Was not able to find link-edit parameters for djxmssql that would work" >>
  /tmp/djxlink;
  results=${results}'djxmssql		Failure
' return 12;
fi
  echo " Successfully built djxmssql Data Access Module";
  echo " Successfully built djxmssql Data Access Module" >> /tmp/djxlink;
/bin/chmod uog+r djxmssql >> /tmp/djxlink;
results=${results}'djxmssql		Success
' return 0;
}
# Subroute to handle dblib DAM link-edit
# ↳ Subroute to handle dblib DAM link-edit

{
  echo "\n
  Warning: Cannot prepare dblib Data Access Module for use';
  echo " Problem: SYBASE environment variable not set';
  echo " Corrective action: Please set SYBASE environment variable to';
  echo " the path where your Open-Client is installed';
  echo "';
  results=${results}'dblib	Failure
' return 12;
fl
if [[ ! -f ${SYBASE}/lib/libsybdb.a ]]; then
  echo "Warning: Cannot prepare dblib Data Access Module for use';
  echo " Problem: SYBASE environment variable not correctly set';
  echo " Corrective action: Please set SYBASE environment variable to';
  echo " the path where your Open-Client is installed';
  echo "';
  results=${results}'dblib	Failure
' return 12;
fl
/bin/rm/dblib >>/dev/null
/bin/ld -o dblib \\
  -e _no_start \n  -e: sybase.exp \n  -bE: sybase.exp \n  -bI: libdb2e.exp \n  -bM: SRE \n  -K \n
164 Mining Relational and Nonrelational Data with IM for Data
-L$SYBASE/lib \
-1c \
-1sybdb \
libsybase.a \ 
libdjexits.a >>/tmp/djxlink 2>/tmp/djxlink; rc=$?; 
if [[ $rc -ne 0 ]]; then 
    echo "Was not able to find link-edit parameters for dblib that would work"; 
    echo "Was not able to find link-edit parameters for dblib that would work" >> /tmp/djxlink; 
    results=${results}'dblib		Failure
' 
    return 12; 
fi 
echo "Successfully built dblib Data Access Module"; 
echo "Successfully built dblib Data Access Module" >> /tmp/djxlink; 
/bin/chmod uog+r dblib >> /tmp/djxlink; 
results=${results}'dblib		Success
' 
return 0; 
} 

# Subroutine to handle ctlib DAM link-edit # 
# Subroutine to handle ctlib DAM link-edit 

ctlib() 
{ 
echo "\n*****************************************************************************\n" >> /tmp/djxlink; 
echo "Attempting to build ctlib Data Access Module"; 
echo "Attempting to build ctlib Data Access Module" >> /tmp/djxlink; 
if [[ "$SYBASE" = "" ]]; then 
    echo 'Warning: Cannot prepare ctlib Data Access Module for use.'; 
    echo 'Warning: Cannot prepare ctlib Data Access Module for use.' >> /tmp/djxlink; 
    echo 'Problem: SYBASE environment variable not set'; 
    echo 'Problem: SYBASE environment variable not set'; 
    echo 'Corrective action: Please set SYBASE environment variable to the path where'; 
    echo 'Corrective action: Please set SYBASE environment variable to the path where'; 
    echo 'Open-Client is installed and re-run djxlink.sh'; 
    echo 'Open-Client is installed and re-run djxlink.sh'; 
    results=${results}'ctlib		Failure
' 
    return 12; 
fi 
if [[ ! -f $SYBASE/lib/libcs.a ]]; then 
    echo 'Warning: Cannot prepare ctlib Data Access Module for use.'; 
    echo 'Warning: Cannot prepare ctlib Data Access Module for use.' >> /tmp/djxlink; 
    echo 'Problem: SYBASE environment variable not correctly set for ctlib'; 
    echo 'Problem: SYBASE environment variable not correctly set for ctlib'; 
    echo 'Corrective action: Please set SYBASE environment variable to the path where'; 
    echo 'Corrective action: Please set SYBASE environment variable to the path where'; 
    echo 'Open-Client is installed and re-run djxlink.sh'; 
    echo 'Open-Client is installed and re-run djxlink.sh'; 
    results=${results}'ctlib		Failure
' 
    return 12; 
fi 

/bin/rm ctlib >>/dev/null 

/bin/ld -o ctlib \
-e _no_start \
-EE:sybbase.exp \
-El:libdb2e.exp \
-El:M:SRE \
-X \ 
-LS$SYBASE/lib \ 
-1cmm.so \ 
-lcs.so \ 
-lct \ 
-linsck \ 
-lintl \ 
-ltcl \ 
-lxa \ 

Scripts for Creation of Data Access Modules  165
if [[ $rc -ne 0 ]]; then
        echo "  Trying variation 2"
        echo "  Trying variation 2" >> /tmp/djxlink
        /bin/ld -o ctlib
                -e _no_start
                -bE:sybase.exp
                -bI:libdb2e.exp
                -bM:SRE
                -K
                -LS($SYBASE)/lib
                -lcomn
                -lcs
                -lct
                -linpack
                -lint1
                -ltcl
                -lc
                libsybasect.a
                libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
        fi

if [[ $rc -ne 0 ]]; then
        echo "  Was not able to find link-edit parameters for ctlib that would work"
        echo "  Was not able to find link-edit parameters for ctlib that would work" >> /tmp/djxlink;
        results=${results}'ctlib		Failure
' return 12;
fi

        echo "  Successfully built ctlib Data Access Module"
        echo "  Successfully built ctlib Data Access Module" >> /tmp/djxlink;
        /bin/chmod uog+r ctlib >> /tmp/djxlink;
        results=${results}'ctlib		Success
' return 0;
}

# Subroute to handle eda DAM link-edit
# ..............................................................................
eda()
{
        echo "\n*******************************************************\n" >> /tmp/djxlink;
        echo "Attempting to build eda Data Access Module”;
        echo "Attempting to build eda Data Access Module" >> /tmp/djxlink;
        if [[ "$EDA_HOME" = "" ]]; then
                echo ’Warning: Cannot prepare eda Data Access Module for use.’;
                echo ’EDA_HOME environment variable not set’;
                echo ’Corrective action: Please set EDA_HOME environment variable to the path where’;
                echo ’edalink is installed and re-run djxlink.sh’;
                results=${results}'eda		Failure\n' return 12;
        fi
        if [[ ! -f "$EDA_HOME/edalink/leda.a" ]]; then
                echo ’Warning: Cannot prepare eda Data Access Module for use.’;
                echo ’Problem: EDA_HOME environment variable not correctly set’;

166  Mining Relational and Nonrelational Data with IM for Data
echo 'Corrective action: Please set EDA_HOME environment variable to the path where edalink';

if

/bin/rm eda >/dev/null
/bin/ld -o eda
                -e _no_start
                -bE:djmsqsql.exp
                -bI:libdb2e.exp
                -bM:SRE
                -K
                -lc
                -L$EDA_HOME\edaapi
                -L$EDA_HOME\edalink
                -leda
                -lhermes
                -ltcp
                -ledasys
                libeda.a
                libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
if [[ $rc -ne 0 ]]; then
    echo "Was not able to find link-edit parameters for eda that would work";
    echo "$() Was not able to find link-edit parameters for eda that would work" >> /tmp/djxlink;
    results=${results}'eda		Failure
    return 12;
fi

echo "Successfully built eda Data Access Module";

echo "Successfully built eda Data Access Module" >> /tmp/djxlink;
/bin/chmod uog+r eda >> /tmp/djxlink;
results=${results}'eda		Success
return 0;
}

# -----------------------------------------------
# Subroutine to handle informix DAM link-edit
# -----------------------------------------------
informix()
{
    echo "\n********************************************\n" >> /tmp/djxlink;
    echo "Attempting to build informix Data Access Module";
    echo "Attempting to build informix Data Access Module" >> /tmp/djxlink;
    if [[ "${INFORMIX_HOME}" = "" ]]; then
        echo 'Warning: Cannot prepare informix Data Access Module for use.';
        echo 'Problem: INFORMIX_HOME environment variable not set';
        echo 'Corrective action: Please set INFORMIX_HOME environment variable to the path where informix is installed and re-run djxlink.sh';
        results=${results}'informix	Failure
        return 12;
        fi
    if [[ ! -f ${INFORMIX_HOME}/lib/esql/libgen.a ]]; then
        echo 'Warning: Cannot prepare informix Data Access Module for use.';
        echo 'Problem: INFORMIX_HOME environment variable not correctly set';
        echo 'Corrective action: Please set INFORMIX_HOME environment variable to the path where informix is installed and re-run djxlink.sh';
        results=${results}'informix	Failure
        return 12;
    echo "\n********************************************\n" >> /tmp/djxlink;
    echo "Attempting to build informix Data Access Module";
    echo "Attempting to build informix Data Access Module" >> /tmp/djxlink;
    if [[ "${INFORMIX_HOME}" = "" ]]; then
        echo 'Warning: Cannot prepare informix Data Access Module for use.';
        echo 'Problem: INFORMIX_HOME environment variable not set';
        echo 'Corrective action: Please set INFORMIX_HOME environment variable to the path where informix is installed and re-run djxlink.sh';
        results=${results}'informix	Failure
        return 12;
    fi
}

Scripts for Creation of Data Access Modules 167
fi

/bin/rm informix >>/dev/null
/bin/ld -o informix
    -e _no_start \
    -bE:informix.exp \ 
    -bI:libdb2e.exp \ 
    -bM:SRE \ 
    -K \ 
    -lc \ 
    -L$(INFORMIX_HOME)/lib/esql \ 
    -lgen \ 
    -los \ 
    -lsql \ 
    libinformix.a \ 
    libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?
if [[ $rc -ne 0 ]]; then
    echo " Was not able to find link-edit parameters for informix that would work";
    echo " Was not able to find link-edit parameters for informix that would work" >>
    /tmp/djxlink;
    results=${results}'informix	Failure
' 
    return 12;
fi
    echo " Successfully built informix Data Access Module";
    echo " Successfully built informix Data Access Module" >> /tmp/djxlink;
    /bin/chmod uog+r informix >> /tmp/djxlink;
    results=${results}'informix	Success
'
    return 0;
}
# Subroute to handle informix7 DAM link-edit
#******************************************************************************
informix7()
{
    echo "\n******************************************************************************" >> /tmp/djxlink;
    echo "Attempting to build informix7 Data Access Module";
    echo "Attempting to build informix7 Data Access Module" >> /tmp/djxlink;
    if [[ "${INFORMIX_HOME}" = "" ]]; then
        echo ' Warning: Cannot prepare informix Data Access Module for use.';
        echo ' Problem: INFORMIX_HOME environment variable not set';
        echo ' Corrective action: Please set INFORMIX_HOME environment variable to the path where';
        echo ' Informix*Net is installed and re-run djxlink.sh';
        results=${results}'informix7	Failure
' 
        return 12;
    fi
    if [[ ! -d ${INFORMIX_HOME}/lib/esql ]]; then
        echo ' Warning: Cannot prepare informix Data Access Module for use.';
        echo ' Problem: INFORMIX_HOME environment variable not correctly set';
        echo ' Corrective action: Please set INFORMIX_HOME environment variable to the path where';
        echo ' Informix*Net is installed and re-run djxlink.sh';
        results=${results}'informix7	Failure
' 
        return 12;
    fi
    /bin/rm informix7 >>/dev/null
******************************************************************************
# We put the XA variations first so we pick up the oracle xa switch if it is there
******************************************************************************

168 Mining Relational and Nonrelational Data with IM for Data
echo " Trying variation 1";
echo " Trying variation 1" >> /tmp/djxlink;
/bin/ld -o informix7 \\n   -e _no_start \\
   -bE:informix.exp \\
   -bI:libdb2e.exp \\
   -bM:SRE \\
   -K \\
   -lc \\
   -L${INFORMIX_HOME}/lib/esql \\
   -L${INFORMIX_HOME}/lib \\
   ${INFORMIX_HOME}/lib/esql/libsqlshr.o \\
   ${INFORMIX_HOME}/lib/esql/libgenshr.o \\
   libinformix7.a \\
   libinformix.a \\
   libdjexits.a >> /tmp/djxlink 2>> /tmp/djxlink; rc=$?;
if 
[[ $rc -ne 0 ]]); then
  echo " Trying variation 2";
  echo " Trying variation 2" >> /tmp/djxlink;
  /bin/ld -o informix7 \\n   -e _no_start \\
   -bE:informix.exp \\
   -bI:libdb2e.exp \\
   -bM:SRE \\
   -K \\
   -lc \\
   -L${INFORMIX_HOME}/lib/esql \\
   -L${INFORMIX_HOME}/lib \\
   -lixsqlshr \\
   -lixgenshr \\
   libinformix7.a \\
   libinformix.a \\
   libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
fi;
if 
[[ $rc -ne 0 ]]); then
  echo " Trying variation 3";
  echo " Trying variation 3" >> /tmp/djxlink;
  /bin/ld -o informix7 \\n   -e _no_start \\
   -bE:informix.exp \\
   -bI:libdb2e.exp \\
   -bM:SRE \\
   -K \\
   -lc \\
   ${INFORMIX_HOME}/lib/esql/libsqlshr.o \\
   ${INFORMIX_HOME}/lib/esql/libgenshr.o \\
   libinformix7.a \\
   libinformix.a \\
   libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
fi;
if 
[[ $rc -ne 0 ]]); then
  echo " Trying permutation 4";
  echo " Trying permutation 4" >> /tmp/djxlink;
  /bin/ld -o informix7 \\n   -e _no_start \n   -bE:informix.exp \\
   -bI:libdb2e.exp \\
   -bM:SRE \\
   -K \\
   -lc \\
   -L${INFORMIX_HOME}/lib/esql \\
   -L${INFORMIX_HOME}/lib \\
   $INFORMIX_HOME}/lib/esql/libdb2e.exp \\
   SRE \\
   -K \\
   -lc \\
   -L${INFORMIX_HOME}/lib/esql \\
   -L${INFORMIX_HOME}/lib \\
   $INFORMIX_HOME}/lib/esql/libdb2e.exp \\
   $INFORMIX_HOME}/lib/esql/libinfxxa.o \\
   libinformix7.a \\
   libinformix.a \\
   libdjexits.a >> /tmp/djxlink 2>> /tmp/djxlink; rc=$?;
fi;
if 
[[ $rc -ne 0 ]]); then
  echo " Trying permutation 4";
  echo " Trying permutation 4" >> /tmp/djxlink;
  /bin/ld -o informix7 \\n   -e _no_start \\
   -bE:informix.exp \\
   -bI:libdb2e.exp \\
   -bM:SRE \\
   -K \\
   -lc \\
   -L${INFORMIX_HOME}/lib/esql \\
   -L${INFORMIX_HOME}/lib \\
   $INFORMIX_HOME}/lib/esql/libdb2e.exp \\
   SRE \\
   -K \\
   -lc \\
   -L${INFORMIX_HOME}/lib/esql \\
   -L${INFORMIX_HOME}/lib \\
   $INFORMIX_HOME}/lib/esql/libdb2e.exp \\
   $INFORMIX_HOME}/lib/esql/libinfxxa.o \\
   libinformix7.a \\
   libinformix.a \\
   libdjexits.a >> /tmp/djxlink 2>> /tmp/djxlink; rc=$?;
fi;
-bS:informix.exp \
-mlibdb2e.exp \
-Im:SRE \
-K \
-lc \
$INFORMIX_HOME)/lib/esql/libsql.o \n$INFORMIX_HOME)/lib/esql/libgen.o \nlibinformix7.a \nlibinformix.a \nlibdjeexit.a >> /tmp/djxlink 2>> /tmp/djxlink; rc=$?;

if [[ $rc -ne 0 ]]; then
  echo "Was not able to find link-edit parameters for informix7 that would work";
  echo "Was not able to find link-edit parameters for informix7 that would work" >>
  /tmp/djxlink;
  results=${results}'informix7	Failure
  return 12;
fi

/bin/chmod uog+r informix7 >> /tmp/djxlink;
results=${results}'informix7	Success
return 0;
}

# Subroute to handle sqlnet DAM link-edit
sqlnet()
{
  echo "\n********************************************************************\n" >> /tmp/djxlink;
  echo "Attempting to build sqlnet Data Access Module";
  echo "Attempting to build sqlnet Data Access Module" >> /tmp/djxlink;
  if [[ "${ORACLE_HOME}" = "" ]]; then
    echo 'Warning: Cannot prepare sqlnet Data Access Module for use.';
    echo 'Problem: ORACLE_HOME environment variable not set';
    echo 'Corrective action: Please set ORACLE_HOME environment variable to the path';
    echo 'where sql*net is installed and re-run djxlink.sh';
    results=${results}'sqlnet	Failure
    return 12;
fi
  if [[ ! -d ${ORACLE_HOME}/lib ]]; then
    echo 'Warning: Cannot prepare sqlnet Data Access Module for use.';
    echo 'Problem: ORACLE_HOME environment variable not correctly set';
    echo 'Corrective action: Please set ORACLE_HOME environment variable to the path';
    echo 'where sql*net is installed and re-run djxlink.sh';
    results=${results}'sqlnet	Failure
    return 12;
fi

# We put the XA variations first so we pick up the oracle xa switch if it is there
/bin/rm sqlnet >>/dev/null

# Often works for Oracle 7.1.4 link edit params

170 Mining Relational and Nonrelational Data with IM for Data
echo " Trying variation 1"
echo " Trying variation 1" >> /tmp/djxlink
/bin/ld -o sqnet \
  -e __no_start \n  -bE:oracle.exp \n  -bI:libdb2e.exp \n  -bM:SRE \n  -K \n  -lc \n  -lld \n  -lm \n  -bI:${ORACLE_HOME}/lib/mili.exp \n  ${ORACLE_HOME}/lib/osntab.o \n  -lS:${ORACLE_HOME}/lib \n  -lnlsrtl \n  -locic \n  -lcore \n  -lora \n  -lcv6 \n  -lsqlnet \n  -lnetwork \n  -lxa \n  liboracle.a \n  libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;

if [[ $rc -ne 0 ]]; then
  echo " Trying variation 2"
echo " Trying variation 2" >> /tmp/djxlink
/bin/ld -o sqnet \
  -e __no_start \n  -bE:oracle.exp \n  -bM:SRE \n  -K \n  -lc \n  -lld \n  -lm \n  -bI:${ORACLE_HOME}/lib/mili.exp \n  -bf:libdb2e.exp \n  ${ORACLE_HOME}/lib/osntab.o \n  ${ORACLE_HOME}/lib/libnlsrtl.a \n  ${ORACLE_HOME}/lib/libsql.a \n  ${ORACLE_HOME}/lib/libocic.a \n  ${ORACLE_HOME}/lib/libcore.a \n  ${ORACLE_HOME}/lib/libora.a \n  ${ORACLE_HOME}/lib/libcv6.a \n  ${ORACLE_HOME}/lib/libsqlnet.a \n  ${ORACLE_HOME}/lib/libnetwork.a \n  liboracle.a \n  libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
fi

if [[ $rc -ne 0 ]]; then
  echo " Trying variation 3"
echo " Trying variation 3" >> /tmp/djxlink
/bin/ld -o sqnet \n
_scripts for creation of data access modules_
-e _no_start \
-bE:oracle.exp \
-bM:SRE \
-K \
-lc \
-lld \
-lm \
-bf:libdb2e.exp \
$(ORACLE_HOME)/lib/osntab.o \ 
$(ORACLE_HOME)/lib/libnlsrtl.a \ 
$(ORACLE_HOME)/lib/libsql.a \ 
$(ORACLE_HOME)/lib/libocic.a \ 
$(ORACLE_HOME)/lib/libora.a \ 
$(ORACLE_HOME)/lib/libcv6.a \ 
$(ORACLE_HOME)/lib/libsqlnet.a \ 
$(ORACLE_HOME)/lib/libnetwork.a \ 
liboracle.a \ 
libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
fi

###############################################################
# Often works for Oracle 7.1.4 link edit parms
###############################################################
if [[ $rc -ne 0 ]]; then
  echo " Trying variation 4" >>/tmp/djxlink
  /bin/ld -o sqlnet \
    -e _no_start \
    -bE:oracle.exp \
    -bf:libdb2e.exp \
    -bM:SRE \
    -K \
    -lc \
    -lld \
    -lm \
    -bI:${ORACLE_HOME}/lib/mili.exp \ 
    $(ORACLE_HOME)/lib/osntab.o \ 
    $(ORACLE_HOME)/lib/libnlsrtl.a \ 
    $(ORACLE_HOME)/lib/libsql.a \ 
    $(ORACLE_HOME)/lib/libocic.a \ 
    $(ORACLE_HOME)/lib/libora.a \ 
    $(ORACLE_HOME)/lib/libcv6.a \ 
    $(ORACLE_HOME)/lib/libsqlnet.a \ 
    $(ORACLE_HOME)/lib/libnetwork.a \ 
    liboracle.a \ 
    libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
fi

###############################################################
# Often works for Oracle 7.2.2.3 link edit parms
###############################################################
if [[ $rc -ne 0 ]]; then
  echo " Trying variation 5" >>/tmp/djxlink
  /bin/ld -o sqlnet \
    -e _no_start \
    -bE:oracle.exp \

-bI:libdb2e.exp \
-IM:SRE \ 
-K \ 
-lc \ 
-lld \ 
-lm \ 
-bI:\$(ORACLE_HOME)/lib/mili.exp \ 
$(ORACLE_HOME)/lib/osntab.o \ 
-LS$(ORACLE_HOME)/lib \ 
-lnlslrl1 \ 
-locic \ 
-lcore \ 
-lora \ 
-lcv6 \ 
-lsqlnet \ 
-lnttcp \ 
-lnttl1 \ 
-lcore3 \ 
-lnlslrl3 \ 
-liboracle.a \ 
-libdjexits.a >>=/tmp/djxlink 2>>/tmp/djxlink; rc=$?;

fi

########################################################################
# Often works for Oracle 7.2.2.3 link edit parms
########################################################################
if [[ $rc -ne 0 ]]; then
  echo " Trying variation 6" 
  echo " Trying variation 6" >> /tmp/djxlink
  /bin/ld -o sqlnet \
    -e _no_start \ 
    -e:oracle.exp \ 
    -bI:libdb2e.exp \ 
    -IM:SRE \ 
    -K \ 
    -lc \ 
    -lld \ 
    -lm \ 
    -bI:\$(ORACLE_HOME)/lib/mili.exp \ 
    $(ORACLE_HOME)/lib/osntab.o \ 
    -LS$(ORACLE_HOME)/lib \ 
    -lnlslrl1 \ 
    -locic \ 
    -lcore \ 
    -lora \ 
    -lcv6 \ 
    -lsqlnet \ 
    -lsql \ 
    -lnttcp \ 
    -lnttl1 \ 
    -lcore3 \ 
    -lnlslrl3 \ 
    liboracle.a \ 
    libdjexits.a >>=/tpxlink 2>>/tmp/djxlink; rc=$?;
fi

########################################################################
# Often works for Oracle 7.3.2 link edit parms
########################################################################
if [[ $rc -ne 0 ]]; then

Scripts for Creation of Data Access Modules 173
echo "  Trying variation 7"
echo "  Trying variation 7" >> /tmp/djxlink
/bin/ld -o sqlnet \
  -e _no_start \
  -bE:oracle.exp \
  -bI:libdb2e.exp \
  -bM:SRE \
  -K \
  -lc \
  -lld \
  -lm \
  -bI:${ORACLE_HOME}/lib/mili.exp \
  $ORACLE_HOME/lib/osntab.o \
  -L$ORACLE_HOME/lib \
  -lepc \
  -lclient \
  -lcommon \
  -lgeneric \
  -lnr \
  -lnlartl \
  -lnlartl3 \
  -lcore \
  -lc6 \
  -lsqlnet \
  -lcore3 \
  -lsql \
  liboracle.a \
  libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
fi
if [[ $rc -ne 0 ]]; then
  echo "  Was not able to find link-edit parameters for sqlnet that would work";
  echo "  Was not able to find link-edit parameters for sqlnet that would work" >> /tmp/djxlink;
  results=${results}'sqlnet	Failure
'  
  return 12;
fi

/net8()
{
  echo "Attempting to build sqlnet Data Access Module";
  echo "Successfully built sqlnet Data Access Module" >> /tmp/djxlink;
  /bin/chmod uog+r sqlnet >> /tmp/djxlink;
  results=${results}'sqlnet	Success
'  
  return 0;
}

# Subroutine to handle net8 DAM link-edit
# --------------------------------------------------------------
net8() {
  echo "\n***********************************************************************\n" >> /tmp/djxlink;
  echo "Attempting to build net8 Data Access Module";
  echo "Successfully built net8 Data Access Module" >> /tmp/djxlink;
  /bin/ld -o net8 \
    -L$(ORACLE_HOME)/lib/ -L$(ORACLE_HOME)/rdhms/lib \
    $(ORACLE_HOME)/rdhms/lib/dopt.o $(ORACLE_HOME)/rdhms/lib/ssdBASE.o \
  -e _no_start \
  -bE:oracle.exp \
  -bI:libdb2e.exp \
  -bM:SRE \

174 Mining Relational and Nonrelational Data with IM for Data
$(_ORACLE_HOME)/lib/nautab.o $(_ORACLE_HOME)/lib/naenst.o \
$(_ORACLE_HOME)/lib/naect.o $(_ORACLE_HOME)/lib/naedhs.o \
$(_ORACLE_HOME)/rdbms/lib/xaondy.o \
-lnetv2 -lnntcp -lnetwork -lncr -lnetv2 -lnntcp -lclient -lcommon \
-lgeneric -lmm -lnlsrtl3 -lcore4 -lnlsrtl3 -lcore4 -lnlsrtl3 -lclient -lcommon \
-lgeneric -lmm -lnlsrtl3 -lcore4 -lnlsrtl3 -lcore4 -lnlsrtl3 -lclient -lcommon \
-libcrlst -lcore4 -lnlsrtl3 -lclient -lcommon -lgeneric \
-lgeneric -lnlsrtl3 -lcore4 -lnlsrtl3 -lclient -lcommon -lgeneric \
-lepc -lnlsrtl3 -lcore4 -lnlsrtl3 -lclient -lcommon -lgeneric \
/lib/crt0.r.o -lc_r -lpthreads -lodm -lm \
liboracle8.a \
libdjexits.a \
-lclntsh >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;

if [[ $rc -ne 0 ]]; then
  echo "Was not able to find link-edit parameters for net8 that would work";
  echo "Was not able to find link-edit parameters for net8 that would work" >> /tmp/djxlink;
  results=${results}'net8	Failure
' return 12;
fi

echo "Successfully built net8 Data Access Module";

/bin/chmod uog+r net8 >> /tmp/djxlink;
results=${results}'net8	Success
' return 0;
}

# Subroutine to handle xaccess DAM link-edit
#
# For CrossAccess v2.1, the ln's are only needed if the library path in
# the file named apif.V2R1M00 is /usr/lpp/CrossAccess/V2R1M00/lib and
# apif.V2R1M00 is installed in /usr/CrossAccess/UAV2R1M00/lib.
# Subroutine to handle xaccess DAM link-edit
#
xaccess()
#
    echo "\n
/bin/ld -o xaccess
    echo "Attempting to build xaccess Data Access Module";
    ln -s -f /usr/CrossAccess/UAV2R1M00/lib/asn1.V2R1M00 `pwd`/asn1.V2R1M00
    ln -s -f /usr/CrossAccess/UAV2R1M00/lib/xvhs.V2R1M00 `pwd`/xvhs.V2R1M00
    /bin/rm xaccess >>/dev/null;
    /bin/ld -o xaccess
        -e _no_start
    -He:Xaccess.exp \
    -I/usr/CrossAccess/UAV2R1M00/lib \
    -hL:libdb2e.exp \
    -hI:/usr/CrossAccess/UAV2R1M00/samples/apif.imp \
    -rM:SRE \
    -lR \
    -lc \
    libxaccess.a \
    libdjexits.a >>/tmp/djxlink 2>>/tmp/djxlink; rc=$?;
    if [[ $rc -ne 0 ]]; then
        echo "Was not able to find link-edit parameters for xaccess that would work";
        echo "Was not able to find link-edit parameters for xaccess that would work" >> /tmp/djxlink;
        results=${results}'xaccess	Failure
' return 12;
    fi
```bash
echo "Successfully built xaccess Data Access Module";
/bin/chmod ugo+r xaccess >> /tmp/djxlink;
results=${results}'xaccess	Success\n'
return 0;
}

# Start of main-line function
# ----------------------------------------------------------------------
#
# Display usage information if any parameters supplied
#
if [[ $# -ne 0 ]]; then
    echo 'djxlink.sh:DB2 DataJoiner Data Access Module Linker';
    echo '   Link edit DB2 DataJoiner Data Access Modules with various';
    echo '   vendors client access packages to prepare the Data Access';
    echo '   module for use.';
    echo '   Usage: djxlink.sh';
    return 12;
fi

# Clean out the file we will use to log errors
/bin/rm /tmp/djxlink >> /dev/null 2>> /dev/null;

# Clear the screen and start displaying progress
# We send output of some 'odd' stuff (such as echo) to /dev/null because if we are not root
# (which we have not yet verified) we may not have write access to /tmp/djxlink
/bin/clear;
/bin/whoami

# We should be root already, exit if we are not
# We send output of some 'odd' stuff (such as echo) to /dev/null because if we are not root
# (which we have not yet verified) we may not have write access to /tmp/djxlink
# if [[ `whoami` != 'root' ]]; then
    echo 'Error: Not root user';
    echo 'Problem: This script can only be executed as root';
    echo 'Corrective action: Log out and log back in as root and reexecute djxlink.sh';
    echo ''
    echo 'Finished djxlink.sh on 'hostname' at 'date';
    echo 'Finished djxlink.sh on 'hostname' at 'date' >> /tmp/djxlink 2>> /dev/null;
    echo ''
    return 12;
fi
```

Mining Relational and Nonrelational Data with IM for Data
# Scripts for Creation of Data Access Modules

We should already be in this directory but we assure that we are

cd /usr/lpp/djx_02_01_01/lib; rc=$?;
if [[ $rc -ne 0 ]]; then
    echo 'Error: Cannot find the /usr/lpp/djx_02_01_01/lib subdirectory';
    echo 'Problem: DataJoiner V2 not correctly installed';
    echo 'Corrective action: If DB2 DataJoiner V2 is not installed please install it. If DataJoiner V2 is installed please call IBM for assistance.';
    echo '  
    echo Finished djxlink.sh on `hostname` at `date`
    echo Finished djxlink.sh on `hostname` at `date` >> /tmp/djxlink
    fi

# Track success/failure of various DAMs using 'results' variable. Prime it with header information.
results="\n\nResults of djxlink\n------------------\n";

# Link each Data Access Module (DAM)
# Link edit mssql DAM
mssql;
# Link edit dblib DAM
dblib;
# Link edit ctllib DAM
ctllib;
# Link edit eda DAM
data;
# Link edit informix DAM
informix;
# Link edit informix7 DAM
informix7;
# Link edit sqlnet DAM
sqlnet;
# Link edit net8 DAM
net8;
# Link edit xaccess DAM
xaccess;

# Finish up
echo "$\n\n************\n" >> /tmp/djxlink;
echo "$results" >> /tmp/djxlink
echo 'Finished djxlink.sh on `hostname` at `date`
echo 'Finished djxlink.sh on `hostname` at `date` >> /tmp/djxlink
echo '  
    echo Finished djxlink.sh on `hostname` at `date`
    echo Finished djxlink.sh on `hostname` at `date` >> /tmp/djxlink
    echo ' 
    echo ''
    return 0
A.2 The djxlink.makefile Script

# ----------------------------------------------------------------------
# Accessing Data Sources Through Visigenic ODBC Drivers
# ----------------------------------------------------------------------

main: dblib ctlib sqlnet Eda Xaccess Generic informix informix7 informix7c informix72 mssqlodbc

DJX_ODBC_LIBRARY_PATH = ./visigenics
DJX_ODBC_LIBRARY = -lodbc
mssqlodbc: djmsql.exp libdb2e.exp libmssql.a
    ld -o mssqlodbc \
    -e _no_start \
    -bE:djmsql.exp \ 
    -bI:libdb2e.exp \ 
    -bM:SRE \ 
    -K -lc \ 
    -L. \ 
    -L $(DJX_ODBC_LIBRARY_PATH) \ 
    $(DJX_ODBC_LIBRARY) \
    -lmssql
Scripts for Creation of Data Access Modules

# Accessing Data Sources Through Sybase Open Client

dblib: libsybase.a sybase.exp
   ld -o dlib \
      -e _no_start \n      -bE:sybase.exp \n      -bI:libdb2e.exp \n      -bM:SRE \n      -K \n      -L$(SYBASE)/lib \n      -lc\n      -lsybdb \n      libsybase.a \n      libdjexits.a

# Accessing Data Sources Through Sybase Open Client Using Multiple 
# Levels of Open Client CT-Lib.

cplib: libsybasect.a sybase.exp
   ld -o cplib \
      -e _no_start \n      -bE:sybase.exp \n      -bI:libdb2e.exp \n      -bM:SRE \n      -K \n      -L$(SYBASE)/lib \n      -lcomn \n      -lcs \n      -lct \n      -linck \n      -lintl \n      -ltcl \n      -lc\n      libsybasect.a \n      libdjexits.a

# Accessing Data Sources Through Oracle SQL*Net

# Note: Oracle changes the link-edit requirements depending on a great 
# many factors, one of which is the version and level of sql*net 
# you have installed. Below is a sample of different commands that 
# are likely to work for different version/releases of sql*net.

Please uncomment the section that most closely matches your 
environment.

Please consult the DataJoiner FAQ for more up-to-date information 

# Oracle 7.0.16

sqlnet: liboracle.a oracle.exp
   ld -o sqlnet \
      -e _no_start \n      -bE:oracle.exp \n      -bI:libdb2e.exp \n      -bM:SRE \n
-K -lc -ldd -lm \
-bI:$ORACLE_HOME/lib/mili.exp \
$ORACLE_HOME/lib/osntab.o \
$ORACLE_HOME/lib/libnlsrtl.a \
$ORACLE_HOME/lib/libsql.a \
$ORACLE_HOME/lib/libocic.a \
$ORACLE_HOME/lib/libcore.a \
$ORACLE_HOME/lib/libcv6.a \
$ORACLE_HOME/lib/libsqlnet.a \
$ORACLE_HOME/lib/libnetwork.a \
liboracle.a \
libdjexits.a

# Oracle 7.1.4
#sqlnet: liboracle.a oracle.exp
ld -o sqlnet \
  -e _no_start \
  -bE:oracle.exp \
  -bI:libdb2e.exp \
  -bM:SRE \
  -K -lc -ldd -lm \
  -bI:$ORACLE_HOME/lib/mili.exp \
  $ORACLE_HOME/lib/osntab.o \
  -L$(ORACLE_HOME)/lib \
  -lnlsrtl -locic -lcore -lora -lcv6 -lsqnet -lnetwork \
  liboracle.a \
  libdjexits.a

# Oracle 7.2.2.3
#sqlnet: liboracle.a oracle.exp
ld -o sqlnet \
  -e _no_start \
  -bE:oracle.exp \
  -bI:libdb2e.exp \
  -bM:SRE \
  -K -lc -ldd -lm \
  -bI:$ORACLE_HOME/lib/mili.exp \
  $ORACLE_HOME/lib/osntab.o \
  -L$(ORACLE_HOME)/lib \
  -lnlsrtl -locic -lcore -lora -lcv6 -lsqnet -lnnttcp -lnntli -lcore3 -lnlsrtl3 \
  liboracle.a \
  libdjexits.a

# 7.3.2 Oracle
#sqlnet: liboracle.a oracle.exp
ld -o sqlnet \
  -e _no_start \
  -bE:oracle.exp \
  -bI:libdb2e.exp \
  -bM:SRE \
  -K -lc -ldd -lm \
  -bI:$ORACLE_HOME/lib/mili.exp \
  $ORACLE_HOME/lib/osntab.o \
  -L$(ORACLE_HOME)/lib \
  -lepc -lcclient -lcommon -lgeneric -lincr -lnlsrtl1 -lnlsrtl3 -lcore -lcv6 -lsqnet -lcore3 \
  -lsql \
  liboracle.a \
  libdjexits.a

180  Mining Relational and Nonrelational Data with IM for Data
# Oracle 8.0 (net8)

net8: liboracle8.a oracle.exp

```
ld -o net8
-LS $(ORACLE_HOME)/lib/ -LS $(ORACLE_HOME)/rdhms/lib/ \
$ (ORACLE_HOME)/rdhms/lib/defopt.o $ (ORACLE_HOME)/rdhms/lib/ssdbaed.o \ 
-e _no_start \ 
-bE:oracle.exp \ 
-bI:libdb2e.exp \ 
-bM:SRE \ 
$ (ORACLE_HOME)/lib/nautab.o $ (ORACLE_HOME)/lib/naeet.o \ 
$ (ORACLE_HOME)/lib/naect.o $ (ORACLE_HOME)/lib/naedhs.o \ 
$ (ORACLE_HOME)/rdhms/lib/xaondy.o \ 
-netv2 -lnetcp -lnetwork -incr -lnetv2 -lnetcp -lnetwork -lcclient -llcommon \ 
-lgeneric -lm -lnlsrtl3 -lcore4 -lnlsrtl3 -lcore4 -lnlsrtl3 -lnetv2 -lnetcp \ 
-1network -incr -lnetv2 -lnetcp -lnetwork -lcclient -llcommon -lgeneric \ 
-lepc -lnlsrtl3 -lcore4 -lnlsrtl3 -lcore4 -lnlsrtl3 -lclient -llcommon \ 
-lgeneric -lnlsrtl3 -lcore4 -lnlsrtl3 -lcore4 -lnlsrtl3 -lclient -llcommon \ 
/lib/crt0_r.o -lc_r -lpthreads -lodm -lm \ 
liboracle8.a \ 
libdjexits.a \ 
-1cint.sh
```

# Accessing Data Sources Through EDA/SQL

```
Eda: libeda.a Eda.exp
```

```
ld -o Eda
-e _no_start \
-bE:Eda.exp \
-bI:libdb2e.exp \
-K -lc \
-L/home/eda/EDA2.1/PRODUCT/edaapi \
-L/home/eda/EDA2.1/PRODUCT/edalink \
-leda -lhermes -1tcp -ledasys\ 
libeda.a \ 
libdjexits.a
```

# Accessing Data Sources Through the Generic Data Access Module

```
Generic: generic_nofunc.o libgeneric.a generic.exp
```

```
ld -o Generic \
-e _no_start \
-bE:generic.exp \
-bI:libdb2e.exp \
-bM:SRE \
-K -lc \
-LS $(DJX_GENERIC_LIBRARY_PATH) \ 
-l$(DJX_GENERIC_LIBRARY) \ 
libgeneric.a
```

---

Scripts for Creation of Data Access Modules   181
# Accessing Data Sources Through CrossAccess 2.1
# For CrossAccess v2.1, the ln's are only needed if the library path in
# the file named apif.V2R1M00 is /usr/lpp/CrossAccess/V2R1M00/lib and
# apif.V2R1M00 is installed in /usr/CrossAccess/UAV2R1M00/lib.
# ----------------------------------------------------------------------

Xaccess: libxaccess.a Xaccess.exp libdjexits.a
\n\nln -s -f /usr/CrossAccess/UAV2R1M00/lib/asn1.V2R1M00 `pwd`/asn1.V2R1M00
ln -s -f /usr/CrossAccess/UAV2R1M00/lib/xvhs.V2R1M00 `pwd`/xvhs.V2R1M00
\n\nld -o Xaccess \n    -e _no_start \n    -bE:Xaccess.exp \n    -L/usr/CrossAccess/UAV2R1M00/lib \n    -bI:libdxe.exp \n    -bI:/usr/CrossAccess/UAV2R1M00/samples/apif.imp \n    -bM:SRE \n    -K -l -c \n    libxaccess.a \n    libdjexits.a

# Accessing Data Sources Through the Informix Data Access Module (Informix Version 5)
# You need to change one line in the following link-edit statement
# before trying to make the Informix access module:
# 1. Change "$(DJX_INFORMIX_LIBRARY_PATH)" to the directory where your
# Informix Version 5 driver’s libraries are.
# ----------------------------------------------------------------------

DJX_INFORMIX_LIBRARY_PATH = /home/informix/lib/esql
DJX_INFORMIX_LIBRARY = -l gen -l os -l sql
informix: libinformix.a informix.exp
\n\nld -o informix \n    -e _no_start \n    -bE:informix.exp \n    -bI:libdxe.exp \n    -bM:SRE \n    -K -l -c \n    $(DJX_INFORMIX_LIBRARY_PATH) \n    $(DJX_INFORMIX_LIBRARY) \n    libinformix.a

# Accessing Data Sources Through the Informix Data Access Module (Informix Version 7)
# using the ESQL/RT library
# You need to change one line in the following link-edit statement
# before trying to make the Informix access module:
# 1. Change "$(DJX_INFORMIX7_LIBRARY_PATH)" to the directory where your
# Informix Version 7 ESQL/RT driver’s libraries are.
# ----------------------------------------------------------------------

DJX_INFORMIX7_LIBRARY_PATH = /home/informix7/lib/esql
informix7: libinformix7.a libinformix.a informix.exp
ld -o informix7 \
-e _no_start \
-bE:informix.exp \
-bI:libdb2e.exp \
-bM:SRE \
-K -lc \n
$DJX_INFORMIX7_LIBRARY_PATH)/libsqlshr.o \
$DJX_INFORMIX7_LIBRARY_PATH)/libgenshr.o \
libinformix7.a \
libinformix.a

# Accessing Data Sources Through the Informix Data Access Module (Informix Version 7)
# using the ESQL/C library
#
# You need to change one line in the following link-edit statement
# before trying to make the Informix access module:
#
# 1. Change "$DJX_INFORMIX_LIBRARY_PATH" to the directory where your
#    Informix Version 7 ESQL/C driver’s libraries are.
#
DJX_INFORMIX7_LIBRARY_PATH = /home/informix7/lib/esql
informix7c: libinformix7.a libinformix.a informix.exp
ld -o informix7c \
-e _no_start \
-bE:informix.exp \
-bI:libdb2e.exp \
-bM:SRE \
-K -lc \n
$DJX_INFORMIX7_LIBRARY_PATH)/libsql.o \
$DJX_INFORMIX7_LIBRARY_PATH)/libgen.o \
libinformix7.a \
libinformix.a

# Accessing Data Sources Through the Informix Data Access Module (Informix Version 7.2)
#
# You need to change one line in the following link-edit statement
# before trying to make the Informix access module:
#
# 1. Change "$DJX_INFORMIX72_LIBRARY_PATH" to the directory where your
#    Informix Version 7.2 driver’s libraries are.
#
DJX_INFORMIX72_LIBRARY_PATH = /home/informix7/lib/esql
DJX_INFORMIX72LIBRARY = -lixsqlshr -lixgenshr
informix72: libinformix.a libinformix7.a informix.exp
ld -o informix72 \
-e _no_start \
-bE:informix.exp \
-bI:libdb2e.exp \
-bM:SRE \
-K -lc \n
-L$DJX_INFORMIX72_LIBRARY_PATH) \
$DJX_INFORMIX72_LIBRARY) \
libinformix7.a \
libinformix.a

Scripts for Creation of Data Access Modules  183
Appendix B. Data Mining Functions

In this appendix we describe each data mining function in detail to provide you with an understanding of their mathematical concepts and uses.

B.1 Linear Discriminant Analysis

In the simplest form a discriminant function is a linear combination of \( N \) independent variables \( X_i, i = 0, ..., n \) that mostly discriminate a dependent variable \( Y \):

\[
Y = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n
\]

Our goal is to optimize the coefficient \( \alpha_i, i = 0, ..., n \) values based on the available historical data.

Given a data set of examples with attributes \( x_1, x_2 \) and two membership classes \( C_1, C_2 \), the problem can be represented as an equation:

\[
y(x_1, x_2) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2
\]

such that an example \( (x', x') \) will be assigned to class \( C_1 \) if \( (x', x') < 0 \) and to class \( C_2 \) if \( (x', x') \geq 0 \).

If we geometrically interpret the discriminant equation, the decision boundary will correspond to a \((d-1)\)-dimensional hyperplane in a \( d \)-dimensional space. We can apply a number of different methods to determine the position and orientation of the decision boundary line. Fisher's linear discriminant, least-squares, and the use of single layer ANNs (Perceptron) are some of them. Figure 123 on page 186 illustrates how Perceptron can be used to define linear decision boundaries.
(a) A linear decision boundary is represented by \( y(\hat{x}) = 0 \) in a two dimension Euclidean space. The orientation of the line is determined by the parameter vector \( \alpha = (\alpha_1, \alpha_2) \), and the distance from \((0,0)\) is determined by the \( \alpha_0 \).

(b) The optimum values of the parameters can be found by using Perceptron’s interactive learning algorithm.

Another form of linear discriminant analysis is the logistic discrimination where we apply a monotonically increasing function (activation function) to the linear combination of the input variables. This resolves to a discriminant output that can be interpreted as posterior probabilities which is more convenient in classification problems.

In this simplistic approach to discriminant analysis we have just described we assumed that the decision boundaries can be linearly separated using a straight or an S-shape line (see Figure 124 on page 187). A generalized linear discriminant permits a range of possible decision boundaries. This can be achieved by defining a set of \( M \) fixed nonlinear function \( f_j(\hat{x}) \) called basis functions. The discriminant function will then take the form:

\[
y(\hat{x}) = \sum_{j=1}^{M} \alpha_j f_j(\hat{x}) + \alpha_0
\]

The above function indeed can approximate, with an arbitrary degree of accuracy, any uniformly continuous function over compacta.
(a) and (b) show a line defined by the Perceptron with a scale and logistic sigmoid output function, respectively. (c) corresponds to a generalized linear function. Clearly, it can cope with a much more complex space of examples.

B.2 Linear Regression

With linear discriminant analysis our goal was to approximate the probabilities of membership of the different classes represented as functions of the input space. However, we can approximate a function in terms of an average over a quantity (regression) and use this approximation for predicting or forecasting the value of a depended variable.

Linear regression is concerned with estimating or forecasting a dependent variable \( Y \) called response, given an infinite number of conditions. The conditions under which we try to estimate or forecast the value of \( Y \) are represented through a set of \( M \) independent variables \( X_i, i = 1, \ldots, m \).

Provided a data set that holds a series of \( N \) observations, we could represent historical knowledge and the way that the predictors influence the response through a design matrix:

\[
X = \begin{bmatrix}
X_{11} & \cdots & X_{1M} & Y_1 \\
X_{21} & \cdots & X_{2M} & Y_2 \\
\vdots & \cdots & \vdots & \vdots \\
X_{1n} & \cdots & X_{mn} & Y_n \\
\end{bmatrix}
\]

Furthermore the model of estimation can be represented in a general form as:

\[
Y = \alpha_m X_m + \ldots + \alpha_1 X_1 + \alpha_0 + \varepsilon \quad (1)
\]
where $\varepsilon$ is the error term.

With the above linear model, the problem can be represented as a system of equations to be solved:

$$y_1 = \alpha_0 + \alpha_1 x_{11} + \ldots + \alpha_m x_{m1} + \varepsilon_1$$

$$y_2 = \alpha_0 + \alpha_1 x_{12} + \ldots + \alpha_m x_{m2} + \varepsilon_2$$

$$\vdots$$

$$y_n = \alpha_0 + \alpha_1 x_{1n} + \ldots + \alpha_m x_{mn} + \varepsilon_n$$

In order to use statistical analysis to calculate coefficient $\alpha_i$ and perform hypothesis testing we make the assumption that $\varepsilon \sim N(0, \sigma^2)$. This assumption is critical to the robustness of the regression model. Heavy tailed probability distributions of the independent variables will resolve to correlated error terms, a common phenomenon in real-world data mining operations.

Having made the necessary assumptions, our problem now is to find the optimal values for the coefficients provided that the linear model described in (1) is valid. We can apply the least squares method minimizing the error term $\varepsilon$:

$$\sum_{i=1}^{N} \varepsilon_i^2$$

After computing the coefficient, we have to test the hypothesis that our linear model probably represents a strong relation between the independent and dependent variables.

The null and alternative hypotheses are:

$$H_0 : \alpha_1 = \alpha_2 = \ldots = \alpha_m = 0$$

$$H_1 : \text{At least one of } \alpha_1, \ldots, \alpha_m \neq 0$$

An $F$-test can be constructed as follows:
where $F$ is the ratio donating the significant of the variant explained by the regression line compared to the sum of squared error. $F_{m,n-m-1,a}$ is the critical value and $a$ donates the significant level.

In the case that we cannot reject the null hypothesis, there is not enough evidence to justify the existence of a linear relation between $Y$ and $X_1, X_2, ..., X_m$. However, we cannot reject the possibility of a nonlinear relationship (polynomial or exponential) between the dependent and independent variables.

### B.3 Tree-Structured Classifiers

In decision tree induction a classification tree partitions the space of possible observations (root) into subregions (nodes) until we reach a terminal node (leaf). Thus a decision tree can be seen as a hierarchical way to describe a partition of the space of examples and eventually classify the example by the label of the leaf it reaches.

In each terminal node we apply a question on which the next split is based. The question may have the form of a statistical test or a function measuring the impotence of a feature. The feature that has been chosen by the splitting criterion can be used for partitioning the node under question.

Consider a set of attributes $A = \{A_1, A_2, ..., A_n\}$ on which a node might be split and a set of classes $C = \{C_1, C_2, ..., C_k\}$. Given a set of classified examples $E$ there is a known probability distribution $p_{ij}$ over attributes and classes $(E \times C)$ and a marginal distribution $p_i$ over $C$. If we consider the feature $A_i$, which has levels $\alpha_1, \alpha_2, ..., \alpha_m$, to grow the tree the probability distribution in the chilled leaf corresponding to $A_i = \alpha_t$ would be $p(C_j | \alpha_t) = \frac{p_{ij}}{p_i}$ over the $k$ classes in $C$. 
We can evaluate our decision to use attribute \( A \) in partitioning the example space by measuring the impurity of the child nodes. Two commonly used measures of impurity are:

The Gini Index \( f(p) = \sum_{i \neq j} p_i p_j \)

The Entropy Function \( f(p) = -\sum_t p_t \log p_t \) with \( 0 \log 0 = 0 \)

As we expect, a measure of impurity will be close to zero if it concentrates on one class. Nodes with a measure of impurity close to zero will become terminal nodes (leaf) since growing the tree further will not resolve to average purer children.

Having grown the tree and reached to a leaf we can represent each branch as an If Then rule.

Starting from the root node and following a branch we construct conjunction of feature levels used to discriminate each node. Finally, the whole tree can be represented as a set of rules using AND and OR operators.

### B.4 Feed-Forward Neural Networks

Today artificial neural networks (ANNs) incorporating methods from statistical and numerical analysis have developed into successful data mining tools. From a data mining perspective a successful neural network is not one that reserves biological neural network behavior, but one that is able to adapt to large data sets, in terms of both volume and dimensionally.

In this process two things are important, the topology of the ANN and the learning rule. Both are connected to each other and influence the “intelligent” behavior of ANN. First we design an ANN. We define the number of layers, the number of neurons in each layer, and the type of activation function for each neuron. Then a learning rule can be applied to alter the weights matrix or/and the topology of the ANN in order to meet some merit of fitness. This process is known as the training of the ANN. Having completed the training process an ANN can be used to predict the outcome or classify new patterns. As we can see the “language” that an ANN is using to represent its state and potentially useful business information cannot be interpreted in a convenient form. Therefore an ANN is considered a black box and as far as data mining
is concerned, prediction and classification are more important than explanation of the relationships between the underlying variables.

Multi-Layer Perceptron (MLP) is one of the most popular ANNs used in data mining. It is a feed-forward multilayer neural network. Feed-forward means that all neurons are ordered from the input neurons to the output neurons so all connections go from a neuron to one with a higher order. The neurons are organized in layers. We have an input layer, an output layer, and a number of hidden layers. An MLP with one hidden layer is represented graphically in Figure 125.

An ANN such as in Figure 125 on page 191 represents this function:

\[ y_k = f_k\left(\alpha_k + \sum_{j \rightarrow k} w_{jk} f_j\left(\alpha_j + \sum_{i \rightarrow j} w_{ij} x_i\right)\right) \]

Thus, MLP can be seen as a nonlinear functional mapping between a set of input variables and a set of output variables.

Since we have established a circuit for the ANN we need to “train” it, that is, optimize the model’s parameters. The parameters in this case are the weights \( w_{ij}, w_{jk} \). In order to perform the optimization a number of different methods can be used. The most popular is the back-propagation method, which
consists of three steps. First the pattern is presented to the network through an input vector or signal. The signal is then passed successively from the input layer to the hidden layer (forward propagation), calculating the activation functions and finally outputting a signal through the activation functions of the output layer neurons. In the second step of the process we compute the total error between the output signal and the true target. Finally, using the generalized delta rule, we can “back-propagate” the underlying error from the output layer back to the input layer, adjusting the existing weights. Usually we iterate for a number of times until the sum-of-squares error function that the generalized delta rule uses as an error function is satisfying small.

Another popular form of feed-forward ANN is the radial-basis function network (RBFN). The circuit of the network consists of three or four layers. Figure 126 on page 193 illustrates the topology of a three-layer RBFN.

As with MLP, an RBFN can be seen as a mapping between the input variables and the output variables through a function:

$$y_k(\mathbf{x}) = \sum_{j=1}^{m} w_k \phi_j \| \mathbf{x} - \mu_j \| + w_{k0}$$

where $\phi_j$, $j = 1, 2, \ldots, m$ are the basis functions and $\| \mathbf{x} \|$ is the norm of the space.

A number of basis functions with different properties can be used. Most popular is the Gaussian function:

$$G(x) = \exp \left( -\frac{x^2}{2} \right)$$

Parameter optimization in an RBFN is been performed in two steps.

1. First we organize the available data into clusters. The centers of the clusters (representatives) $\mu_j$ become the centers of the basis function $\phi_j$ and the components of a representative $\mathbf{\mu}_j$ the $\mu_{ji}$, $i = 1, 2, \ldots, n$ weights.

2. The second stage consists of the final tuning of the network. For this task we can use an iterative training algorithm like the delta rule to optimize the $w_{ji}$ weights.

RBFN is considered to be an exact interpolation model and therefore useful for solving prediction problems where data neighborhoods are more important than decision boundaries. The degree of sensitivity to the example
space and to interpolation can be controlled through the number of clusters formed and the shape of the basis functions used.

Figure 126. Radial-Basis Function Network

B.5 Projection Methods

So far, in the data mining tools that we have seen, an external agent provided the data mining engine with a set of classified examples. The data mining engine, after "examining" the historical data, could generalize over newly presented patterns. However, some data mining operations require tools that operate over unclassified examples, where the interest is not to predict the outcome of an event or classify a feature vector but to indicate groups or to show affinities of examples revealing structures in the data.

Projection methods belong to this "unsupervised" data mining category of tools. Suppose a data set consists of \( n \) vectors with \( m \) components (\( n \geq m \)). Our goal is to project the \( m \)-dimension input space into a smaller \( q \)-dimension space. Each dimension of the rank-\( q \) reconstructed data space corresponds to a linear combination of the original features. The features that compose each component are selected in order to maximize some measure of "interestingness."

Principal component analysis is an example of input space projection. In principal component analysis the interest is on variance. The first component
is a combination of variables accounting for the largest of the variance in the original data set, the second is the combination of the second largest variance but uncorrected (orthogonal) with the first, and so on. We require also that the projection of the original m-dimension input space into the q-dimension space using the first q principal components be as accurate as possible. A measure of accuracy or fitness could be the sum of square distances from the original data point to their projections into the principal component space.

A classical way to perform principal component analysis is to construct the correlation matrix of the observations. The eigenvalues of the correlation matrix will give the proportion of the variance in the original data explained by the corresponding principal component. Thus, we can use the cumulative total of variance explained in order to choose how many components we can keep. Usually we use the first two or three components because then we are able to visualize them in a two- or three-dimensional space. The benefit of that is that any clusters formed by the original variables can be used for further analysis.

### B.6 Cluster Formation

As with principal component analysis cluster formation is an “unsupervised” method. Our goal is to group observations into subsets, called clusters, maximizing the degree of similarity within clusters and minimizing similarity between clusters.

One of the most popular clustering methods is the k-means clustering. A typical algorithm of this method will start with randomly selecting the centers for k clusters. Each example is assigned to the cluster with the closest center. The cluster centers are computed again and the examples are reassigned to clusters based on the distance to the new cluster centers. This process is repeated until there is no change in the grouping of the examples. This algorithm eventually will minimize the:

\[
\sum_{i} \| x_i - m_k(0) \|^2
\]

which donates the sum of squared distances from each example \( x_i \) to its cluster center \( m_k \). As we can see, the minimization over the cluster centers is easy. However the difficulty is to minimize the squared distances over classes:

\[
\min_{k} \sum_{i} \sum_{k} \| x_i - m_k(0) \|^2
\]
Another popular clustering method is the self-organized feature map. It can be represented as a simple neural network with an input layer that corresponds to the features of an input vector and an output layer that corresponds to the clusters that we want to form. The weight vector of an output neuron corresponds to the representative of the cluster that the output neuron represents. The examples are organized into a one- or two-dimensional feature-map space. The available examples are then presented to the network and, depending on the output of the network, a particular neighborhood of the feature-map space is updated. The algorithm will iterate over the input space until it reaches equilibrium.

B.7 Association Rules

Today’s databases hold large volumes of data that capture the shopping behavior of people. For instance customers of a supermarket purchase a list of items. Those observations saved into a database can be used to identify associations between items (association rules). A typical association rule will be:

“30% of the transactions that contain beer and potato chips also contain diapers; 2% of all transactions contain all of these items”.

The 30% is called the confidence of the rule, and the 2%, the support of the rule. The challenge for data mining is to discover a statistical significant set of association rules given specific minimum support and minimum confidence.

Let \( A = \{ \alpha_1, \alpha_2, \ldots, \alpha_n \} \) be a set of items and \( D = \{ T_1, T_2, \ldots, T_m \} \) a set of transactions. Each transaction \( T_i \) is a set of items such that \( T_i \subseteq A \). An association rule is an implication of the form \( G \Rightarrow B \), where \( G \subseteq A \), \( B \subseteq A \), \( A \cap B = 0 \). The association rule \( G \Rightarrow B \) has a confidence level in the transaction set \( A \) donated by the conditional probability \( p(B|G) \) and a support level equal to the number of transactions \( T_i \) that contain \( A \cup B \).

A general approach to the problem is to find all combinations of items that have transaction support above minimum support (frequent itemset) and then use these itemsets to construct the association rules. If a subset \( G \) of items of a frequent itemset \( B \) is itself a frequent itemset, a rule of the form \( G \Rightarrow B \) holds, if:

\[
\frac{\text{support}(B)}{\text{support}(G)} > \text{min confidence}
\]
However, this approach leads us to a very large number of output patterns that just “overfit” the available data. To avoid false discoveries we require means for testing for statistical significance.

For instance we can cast the minimum support requirement in a hypotheses testing framework as follows:

\[ H_0 : p(G) = t \]
\[ H_1 : p(G) > t \]

where the density function \( p(G) \) donates the probability that a transaction contains a given itemset \( G \) and \( t \) is the support threshold. Thus, the probability that transactions in our data set will contain \( G \) will be given by a binomial distribution \( p'(G) = p(G) \) over \( n \) trials. Finally, we can compare \( p'(G) \) (with a given critical value \( t_0 \), depended on the significant level of the hypothesis test) and reject it only if \( p'(G) > t_0 \).
Appendix C. Special Notices

This publication is intended to help data mining and information system specialists understand how IBM Intelligent Miner for Data accesses non-IBM data sources. The information in this publication is not intended as the specification of any programming interfaces that are provided by the IBM Intelligent Miner for Data and IBM DataJoiner products. See the PUBLICATIONS section of the IBM Programming Announcement for IBM Intelligent Miner for Data and Data Joiner products for more information about what publications are considered to be product documentation.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM’s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM’s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Corporation, Dept. 600A, Mail Drop 1329, Somers, NY 10589 USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM ("vendor") products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer’s ability to evaluate and integrate
them into the customer’s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

The following document contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples contain the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

<table>
<thead>
<tr>
<th>Term</th>
<th>Trademark of</th>
<th>Country/Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>AS/400</td>
<td></td>
</tr>
<tr>
<td>CICS</td>
<td>DATABASE 2</td>
<td></td>
</tr>
<tr>
<td>DB2</td>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td>IBM</td>
<td>IMS</td>
<td></td>
</tr>
<tr>
<td>Information Warehouse</td>
<td>Intelligent Miner</td>
<td></td>
</tr>
<tr>
<td>MVS</td>
<td>MVS/ESA</td>
<td></td>
</tr>
<tr>
<td>Net.Data</td>
<td>OS/2</td>
<td></td>
</tr>
<tr>
<td>OS/390</td>
<td>OS/400</td>
<td></td>
</tr>
<tr>
<td>RS/6000</td>
<td>S/390</td>
<td></td>
</tr>
<tr>
<td>Visual Warehouse</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following terms are trademarks of other companies:

Java and HotJava are trademarks of Sun Microsystems, Incorporated.

Microsoft, SQL Server, Windows, Windows NT, and the Windows 95 logo are
trademarks or registered trademarks of Microsoft Corporation.

Oracle, Net8, SQL*Plus, SQL*Net are trademarks or registered trademarks of Oracle Corporation.

Pentium, MMX, ProShare, LANDesk, and ActionMedia are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

SAS, SAS/SHARE, SAS/SHARE*NET, SAS/ACCESS, PROC SQL are trademarks or registered trademarks of SAS Institute, Incorporated.

SPSS is trademarks or registered trademarks of SPSS, Incorporated.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names may be trademarks or service marks of others.
Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see “How to Get ITSO Redbooks” on page 203.

- *Data Modeling Techniques for Data Warehousing*, SG24-2238
- *From Multiple Operational Data to Data Warehousing and Business Intelligence*, SG24-5174
- *DataJoiner Implementation and Usage Guide*, SG24-2566
- *Discovering Data Mining*, SG24-4839

D.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

<table>
<thead>
<tr>
<th>CD-ROM Title</th>
<th>Subscription Number</th>
<th>Collection Kit Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>System/390 Redbooks Collection</td>
<td>SBOF-7201</td>
<td>SK2T-2177</td>
</tr>
<tr>
<td>Networking and Systems Management Redbooks Collection</td>
<td>SBOF-7370</td>
<td>SK2T-6022</td>
</tr>
<tr>
<td>Transaction Processing and Data Management Redbook</td>
<td>SBOF-7240</td>
<td>SK2T-8038</td>
</tr>
<tr>
<td>Lotus Redbooks Collection</td>
<td>SBOF-6899</td>
<td>SK2T-8039</td>
</tr>
<tr>
<td>Tivoli Redbooks Collection</td>
<td>SBOF-6898</td>
<td>SK2T-8044</td>
</tr>
<tr>
<td>AS/400 Redbooks Collection</td>
<td>SBOF-7270</td>
<td>SK2T-2849</td>
</tr>
<tr>
<td>RS/6000 Redbooks Collection (HTML, BkMgr)</td>
<td>SBOF-7230</td>
<td>SK2T-8040</td>
</tr>
<tr>
<td>RS/6000 Redbooks Collection (PostScript)</td>
<td>SBOF-7205</td>
<td>SK2T-8041</td>
</tr>
<tr>
<td>RS/6000 Redbooks Collection (PDF Format)</td>
<td>SBOF-8700</td>
<td>SK2T-8043</td>
</tr>
<tr>
<td>Application Development Redbooks Collection</td>
<td>SBOF-7290</td>
<td>SK2T-8037</td>
</tr>
</tbody>
</table>

D.3 Other Publications

These publications are also relevant as further information sources:

• C. M. Bishop, *Neural Networks for Pattern Recognition*, Cambridge University Press, 1995
How to Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at http://www.redbooks.ibm.com/.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Redbooks Web Site on the World Wide Web**
  
  http://w3.itso.ibm.com/

- **PUBORDER** – to order hardcopies in the United States

- **Tools Disks**
  
  To get LIST3820s of redbooks, type one of the following commands:

  TOOLCAT REDPRINT
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)

  To get BookManager BOOKs of redbooks, type the following command:

  TOOLCAT REDBOOKS

  To get lists of redbooks, type the following command:

  TOOLS SENDTO USDIST MKTOOLS MKTOOLS GET ITSOCAT TXT

  To register for information on workshops, residencies, and redbooks, type the following command:

  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1998

- **REDBOOKS Category on INEWS**

- **Online** – send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL

---

Redpieces

For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.html). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few
How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

• **Online Orders** – send orders to:
  - IBMMAIL
    - In United States: usib6fp1 at ibmmail
    - In Canada: caibmbkz at ibmmail
    - Outside North America: dkibmbsh at ibmmail
  - Internet
    - usib6fp1@ibmmail.com
    - lmannix@vnet.ibm.com
    - bookshop@dk.ibm.com

• **Telephone Orders**
  - United States (toll free): 1-800-879-2755
  - Canada (toll free): 1-800-IBM-4YOU
  - Outside North America (long distance charges apply):
    - (+45) 4810-1320 - Danish
    - (+45) 4810-1420 - Dutch
    - (+45) 4810-1540 - English
    - (+45) 4810-1670 - Finnish
    - (+45) 4810-1220 - French
  - (+45) 4810-1620 - Italian
  - (+45) 4810-1270 - Norwegian
  - (+45) 4810-1120 - Spanish
  - (+45) 4810-1170 - Swedish

• **Mail Orders** – send orders to:
  - IBM Publications
    - P.O. Box 29570
    - Raleigh, NC 27626-0570
    - USA
  - IBM Direct Services
    - 144-4th Avenue, S.W.
    - Calgary, Alberta T2P 3N5
  - IBM Direct Services (Outside North America)
    - Sortemosevej 21
    - DK-3450 Allerød

• **Fax** – send orders to:
  - United States (toll free): 1-800-445-9269
  - Canada: 1-800-267-4455
  - Outside North America (long distance charge):
    - (+45) 48 14 2207

• **1-800-IBM-4FAX (United States) or (+1) 408 256 5422 (Outside USA)** – ask for:
  - Index # 4421 Abstracts of new redbooks
  - Index # 4422 IBM redbooks
  - Index # 4420 Redbooks for last six months

• **On the World Wide Web**
  - Redbooks Web Site: http://www.redbooks.ibm.com

204  Mining Relational and Nonrelational Data with IM for Data
For information so current it is still in the process of being written, look at "Redpieces" on the Redbooks Web Site (http://www.redbooks.ibm.com/redpieces.html). Redpieces are redbooks in progress; not all redbooks become redpieces, and sometimes just a few
IBM Redbook Order Form

Please send me the following:

<table>
<thead>
<tr>
<th>T</th>
<th>O</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

First

Last

Compa

Address

City

Postal

Country

Teleph

Telefax

VAT

☐ Invoice to

☐ Credit

Credit

Card

Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.
Glossary

A

adaptive connection. A numeric weight used to describe the strength of the connection between two processing units in a neural network. The connection is called adaptive because it is adjusted during training. Values typically range from zero to one, or -0.5 to +0.5.

aggregate. To summarize data in a field.

application programming interface (API). A functional interface supplied by the operating system or a separate orderable licensed program that allows an application program written in a high-level language to use specific data or functions of the operating system or the licensed program.

architecture. The number of processing units in the input, output, and hidden layer of a neural network. The number of units in the input and output layers is calculated from the mining data and input parameters. An intelligent data mining agent calculates the number of hidden layers and the number of processing units in those hidden layers.

associations. The relationship of items in a transaction in such a way that items imply the presence of other items in the same transaction.

attribute. Characteristics or properties that can be controlled, usually to obtain a required appearance. For example, color is an attribute of a line. In object-oriented programming, a data element defined within a class.

B

back propagation. A general-purpose neural network named for the method used to adjust weights while learning data patterns. The Classification - Neural mining function uses such a network.

boundary field. The upper limit of an interval as used for discretization using ranges of a processing function.

bucket. One of the bars in a bar chart showing the frequency of a specific value.

C

categorical values. Discrete, nonnumerical data represented by character strings; for example, colors or special brands.

chi-square test. A test to check whether two variables are statistically dependent or not. Chi-square is calculated by subtracting the expected frequencies (imaginary values) from the observed frequencies (actual values). The expected frequencies represent the values that were to be expected if the variable question were statistically independent.

classification. The assignment of objects into groups or categories based on their characteristics.

cluster. A group of records with similar characteristics.

cluster prototype. The attribute values that are typical of all records in a given cluster. Used to compare the input records to determine whether a record should be assigned to the cluster represented by these values.

clustering. A mining function that creates groups of data records within the input data on the basis of similar characteristics. Each group is called a cluster.

confidence factor. Indicates the strength or the reliability of the associations detected.

continuous field. A field that can have any floating point number as its value.

D

DATABASE 2 (DB2). An IBM relational database management system.

database table. A table residing in a database.
**database view.** An alternative representation of data from one or more database tables. A view can include all or some of the columns contained in the database table or tables on which it is defined.

**data field.** In a database table, the intersection from table description and table column where the corresponding data is entered.

**data format.** There are different kinds of data formats, for example, database tables, database views, pipes, or flat files.

**data table.** A data table, regardless of the data format it contains.

**data type.** There are different kinds of Intelligent Miner data types, for example, discrete numeric, discrete nonnumeric, binary, or continuous.

**discrete.** Pertaining to data that consists of distinct elements such as character or to physical quantities having a finite number of distinctly recognizable values.

**discretization.** The act of making mathematically discrete.

**Euclidean distance.** The square root of the sum of the squared differences between two numeric vectors. The Euclidean distance is used to calculate the error between the calculated network output and the target output in neural classification, to calculate the difference between a record and a prototype cluster value in neural clustering. A zero value indicates an exact match; larger numbers indicate greater differences.

**F**

**field.** A set of one or more related data items grouped for processing. In this document, with regard to database tables and views, field is synonymous with column.

**file.** A collection of related data that is stored and retrieved by an assigned name.

**file name.** (1) A name assigned or declared for a file. (2) The name used by a program to identify a file.

**flat file.** (1) A one-dimensional or two-dimensional array; a list or table of items. (2) A file that has no hierarchical structure.

**formatted information.** An arrangement of information into discrete units and structures in a manner that facilitates its access and processing. Contrast with narrative information.

**F-test.** A statistical test that checks whether two estimates of the variances of two independent samples are the same. In addition, the F-test checks whether the null hypothesis is true or false.

**function.** Any instruction or set of related instructions that perform a specific operation.

**fuzzy logic.** In artificial intelligence, a technique using approximate rules of inference in which truth values and quantifiers are defined as possibility distributions that carry linguistic labels.
input data. The metadata of the database table, database view, or flat file containing the data you specified to be mined.

input layer. A set of processing units in a neural network which present the numeric values derived from user data to the network. The number of fields and type of data in those fields are used to calculate the number of processing units in the input layer.

instance. In object-oriented programming, a single, actual occurrence of a particular object. Any level of the object class hierarchy can have instances. An instance can be considered in terms of a copy of the object type frame that is filled in with particular information.

interval. A set of real numbers between two numbers either including or excluding both of them.

interval boundaries. Values that represent the upper and lower limits of an interval.

item category. A categorization of an item. For example, a room in a hotel can have the following categories: Standard, Comfort, Superior, Luxury. The lower category is called the child item category. Each child item category can have several parent item categories. Each parent item category can have several grandparent item categories.

item description. The descriptive name of a character string in a data table.

item ID. The identifier for an item.

item set. A collection of items. For example, all items bought by one customer during one visit to a department store.

Kohonen Feature Map. A neural network model comprised of processing units arranged in an input layer and output layer. All processors in the input layer are connected to each processor in the output layer by an adaptive connection. The learning algorithm used involves competition between units for each input pattern and the declaration of a winning unit. Used in neural clustering to partition data into similar record groups.

large item sets. The total volume of items above the specified support factor returned by the Associations mining function.

learning algorithm. The set of well-defined rules used during the training process to adjust the connection weights of a neural network. The criteria and methods used to adjust the weights define the different learning algorithms.

learning parameters. The variables used by each neural network model to control the training of a neural network which is accomplished by modifying network weights.

lift. Confidence factor divided by expected confidence.

metadata. In databases, data that describes data objects.

mining. Synonym for analyzing or searching.

mining base. A repository where all information about the input data, the mining run settings, and the corresponding results is stored.

model. A specific type of neural network and its associated learning algorithm. Examples include the Kohonen Feature Map and back propagation.

narrative information. Information that is presented according to the syntax of a natural language. Contrast with formatted information.

neural network. A collection of processing units and adaptive connections that is designed to perform a specific processing function.
Neural Network Utility (NNU.) A family of IBM application development products for creating neural network and fuzzy rule system applications.

nonsupervised learning. A learning algorithm that requires only input data to be present in the data source during the training process. No target output is provided; instead, the desired output is discovered during the mining run. A Kohonen Feature Map, for example, uses nonsupervised learning.

NP-complete. In the context on neurocomputing: A learning algorithm is NP-complete if it converges to a solution in time polynomial in size of the problem and the accuracy required.

O

offset. (1) The number of measuring units from an arbitrary starting point in a record, area, or control block, to some other point. (2) The distance from the beginning of an object to the beginning of a particular field.

operator. (1) A symbol that represents an operation to be done. (2) In a language statement, the lexical entity that indicates the action to be performed on operands.

output data. The metadata of the database table, database view, or flat file containing the data being produced or to be produced by a function.

output layer. A set of processing units in a neural network which contain the output calculated by the network. The number of outputs depends on the number of classification categories or maximum cluster value in neural classification and neural clustering, respectively.

P

pass. One cycle of processing a body of data.

prediction. The dependency and the variation of one field’s value within a record on the other fields within the same record. A profile is then generated that can predict a value for the particular field in a new record of the same form, based on its other field values.

processing unit. A processing unit in a neural network is used to calculate an output by summing all incoming values multiplied by their respective adaptive connection weights.

Q

quantile. One of a finite number of nonoverlapping subranges or intervals, each of which is represented by an assigned value.

$Q$ is an $N\%$ -quantile of a value set $S$ when:

- Approximately $N$ percent of the values in $S$ are lower than or equal to $Q$.
- Approximately $(100-N)$ percent of the values are greater than or equal to $Q$.

The approximation is less exact when there are many values equal to $Q$. $N$ is called the quantile label. The 50%.-quantile represents the median.

R

radial basis function (RBF). In data mining functions, radial basis functions are used to predict values. They represent functions of the distance or the radius from a particular point. They are used to build up approximations to more complicated functions.

record. A set of one or more related data items grouped for processing. In reference to a database table, record is synonymous with row.

region. (Sub)set of records with similar characteristics in their active fields. Regions are used to visualize a prediction result.

round-robin method. A method by which items are sequentially assigned to units. When an item has been assigned to the last unit in the series, the next item is assigned to the first again. This process is repeated until the last item has been assigned. The Intelligent Miner uses this method, for example, to store records in output files during a partitioning job.
**rule.** A clause in the form head<= body. It specifies that the head is true if the body is true.

**rule body.** Represents the specified input data for a mining function.

**rule group.** Covers all rules containing the same items in different variations.

**rule head.** Represents the derived items detected by the Associations mining function.

---

**S**

**scale.** A system of mathematical notation; fixed-point or floating-point scale of an arithmetic value.

**scaling.** To adjust the representation of a quantity by a factor in order to bring its range within prescribed limits.

**scale factor.** A number used as a multiplier in scaling. For example, a scale factor of 1/1000 would be suitable to scale the values 856, 432, -95, and /182 to lie in the range from -1 to +1, inclusive.

**self-organizing feature map.** See Kohonen Feature Map

**sensitivity analysis report.** An output from the Classification - Neural mining function that shows which input fields are relevant to the classification decision.

**sequential patterns.** Intertransaction patterns such that the presence of one set of items is followed by another set of items in a database of transactions over a period of time.

**similar time sequences.** Occurrences of similar sequences in a database of time sequences.

**Structured Query Language (SQL).** An established set of statements used to manage information stored in a database. By using these statements, users can add, delete, or update information in a table, request information through a query, and display results in a report.

**supervised learning.** A learning algorithm that requires input and resulting output pairs to be presented to the network during the training process. Back propagation, for example, uses supervised learning and makes adjustments during training so that the value computed by the neural network will approach the actual value as the network learns from the data presented. Supervised learning is used in the techniques provided for predicting classifications as well as for predicting values.

**support factor.** Indicates the occurrence of the detected association rules and sequential patterns based on the input data.

**symbolic name.** In a programming language, a unique name used to represent an entity such as a field, file, data structure, or label. In the Intelligent Miner you specify symbolic names, for example, for input data, name mappings, or taxonomies.

---

**T**

**taxonomy.** Represents a hierarchy or a lattice of associations between the item categories of an item. These associations are called taxonomy relations.

**taxonomy relation.** The hierarchical associations between the item categories you defined for an item. A taxonomy relation consists of a child item category and a parent item category.

**trained network.** A neural network containing connection weights that have been adjusted by a learning algorithm. A trained network can be considered a virtual processor; it transforms inputs to outputs.

**training.** The process of developing a model which understands the input data. In neural networks, the model is created by reading the records of the input and modifying the network weights until the network calculates the desired output data.

**translation process.** Converting the data provided in the database to scaled numeric values in the appropriate range for a mining kernel using neural networks. Different techniques are used depending on whether the
data is numeric or symbolic. Also, converting neural network output back to the units used in the database.

**transaction.** A set of items or events that are linked by a common key value, for example, the articles (items) bought by a customer (customer number) on a particular date (transaction identifier). In this example, the customer number represents the key value.

**transaction ID.** The identifier for a transaction, for example, the date of a transaction.

**transaction group.** The identifier for a set of transactions. For example, a customer number can represent a transaction group that includes all purchases of a particular customer during the month of May.

**V**

**vector.** A quantity usually characterized by an ordered set of numbers.

**W**

**weight.** The numeric value of an adaptive connection representing the strength of the connection between two processing units in a neural network.

**winner.** The index of the cluster which has the minimum Euclidean distance from the input record. Used in the Kohonen Feature Map to determine which output units will have their weights adjusted.
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADT</td>
<td>abstract data type</td>
</tr>
<tr>
<td>ANN</td>
<td>artificial neural network</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>API</td>
<td>application programming interface</td>
</tr>
<tr>
<td>ASCII</td>
<td>American National Standard Code for Information Interchange</td>
</tr>
<tr>
<td>DAM</td>
<td>data access module</td>
</tr>
<tr>
<td>DBMS</td>
<td>database management system</td>
</tr>
<tr>
<td>DCL</td>
<td>data control language</td>
</tr>
<tr>
<td>DDL</td>
<td>data definition language</td>
</tr>
<tr>
<td>DML</td>
<td>data manipulation language</td>
</tr>
<tr>
<td>DW</td>
<td>data warehouse</td>
</tr>
<tr>
<td>DSS</td>
<td>decision support system</td>
</tr>
<tr>
<td>GUI</td>
<td>graphical user interface</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines Corporation</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>I/O</td>
<td>input/output</td>
</tr>
<tr>
<td>IM</td>
<td>Intelligent Miner</td>
</tr>
<tr>
<td>IMS</td>
<td>Information Management System</td>
</tr>
<tr>
<td>ISAPI</td>
<td>Internet Server Application Programming Interface</td>
</tr>
<tr>
<td>IT</td>
<td>information technology</td>
</tr>
<tr>
<td>JRE</td>
<td>Java runtime environment</td>
</tr>
<tr>
<td>JDBC</td>
<td>java database connectivity</td>
</tr>
<tr>
<td>LIS</td>
<td>large item set</td>
</tr>
<tr>
<td>LOB</td>
<td>large object</td>
</tr>
<tr>
<td>MLP</td>
<td>multilayer perceptron</td>
</tr>
<tr>
<td>NSAPI</td>
<td>Netscape Application Programming Interface</td>
</tr>
<tr>
<td>ODBC</td>
<td>Open Database Connectivity</td>
</tr>
<tr>
<td>OEM</td>
<td>original equipment manufacturer</td>
</tr>
<tr>
<td>OLAP</td>
<td>online analytical processing</td>
</tr>
<tr>
<td>OLTP</td>
<td>online transaction processing</td>
</tr>
<tr>
<td>RBF</td>
<td>radial basis function</td>
</tr>
<tr>
<td>RBFN</td>
<td>radial basis function network</td>
</tr>
<tr>
<td>RDBMS</td>
<td>relational database management system</td>
</tr>
<tr>
<td>ROI</td>
<td>return on investment</td>
</tr>
<tr>
<td>SAG</td>
<td>SQL Access Group</td>
</tr>
<tr>
<td>SQL</td>
<td>structured query language</td>
</tr>
<tr>
<td>TCP/IP</td>
<td>Transmission Control Protocol/Internet Protocol</td>
</tr>
<tr>
<td>UDF</td>
<td>user-defined function</td>
</tr>
<tr>
<td>UDT</td>
<td>user-defined type</td>
</tr>
<tr>
<td>VSAM</td>
<td>Virtual Storage Access Method</td>
</tr>
</tbody>
</table>

© Copyright IBM Corp. 1998
Index

A
abstract data type 22
aggregate values 15
algorithm categorization 16
analysis
   factor 16
   principal component 10, 16
application mode 17
architecture
   Business Intelligence 21
   Information Warehouse 21
   Object-Relational 23
association discovery 16

B
binary data 82
bivariate statistic 16
Business Intelligence 21

C
calculate values 15
Call Level Interface
   see CLI 109
CASE expression 23
categorical data 82
classification 16
CLI 109
cluster result screen 88
clustering 16
clustering mode 17
conditional logic 23
configure
   DataJoiner on AIX 47
   DataJoiner on Windows NT 67
   DataJoiner SAS access 125
   Oracle client on AIX 42
   Oracle client on Windows NT 61
   SAS ODBC 121
   SPSS ODBC 115
confusion table 10
continuous data 82
create
   AIX group 53
   AIX user ID 53
   data mining operations 18
database 67
DataJoiner instance on Windows NT 67
DB2 database on AIX 55
DDE share 127
filesystem 39
IM data object 80
mining 83
mining base 80
mining sequence 145
nickname 58, 119
nickname on Windows NT 68, 125
ODBC datasource 116
Oracle administrative group 40
Oracle administrative user on AIX 40
results 17
server mapping 67, 118, 125
server mapping on AIX 56
user mapping 58, 67, 118, 125
Windows NT user 63

D
data
   abstract type 22
   access module 48
   binary 82
categorical 82
collection 8
collection utility 33, 71
continuous 82
discrete numeric 82
folder 82
heterogenous access 20
integrity 21
mart 24
model evaluation 10
modeling 10
numeric 82
preparation 8, 138
preparation function 15
sample file 30
sampling 8
selection 147
user-defined type 22
visualization 147
visualize 138
data control language 21
data definition language 21
data manipulation language 21
data mining 3
definition 3
operations creation 18
sample scenario on AIX 76
setting goals 7
tools 4
data replication
asynchronous 21
synchronous 21
data warehouse 3
database gateway 20
DataJoiner
client access configuration 54
configure on Windows NT 67
configure SPSS 117
create instance on Windows NT 67
create nickname 125
create nickname on AIX 58
create nickname on Windows NT 68
create server mapping on AIX 56
create server mapping on Windows NT 67
create user mapping on AIX 58
create user mapping on Windows NT 67
DB2DBDFT 56
install on Windows NT 63
introduction 19
listener service 54
NLS Option 65
ODBC DAM 115
ODBC interface 112
optional components on Windows NT 65
Oracle access on Windows NT 67
Oracle interface 34
recycle instance 55
SAS configuration 125
server mapping 125
start instance 54
support 26
user mapping 125
db2icrt 67, 117
db2profile 54
DDESHARE 127
decision making 10
demographic clustering 83
dimension reduction 9
discovery
association 16
sequential pattern 16
discrete numeric data 82
discretize
quantiles 15
ranges 15
distributed joins 21

E
encode missing values 148
environment
ODBC 109
Oracle 29
variables on AIX 40
Windows 109
Euclidean distance 8
evaluate data model 10

F
factor analysis 16
Fast Extract 71
idmpldor 71
install 71
oracollect 71
using 33
utility 32
feature selection 9
field
filter 15
pivot 15
filter
fields 15
records 15
filter records 84
frequency chart 139
function
data mining 16
data preparation 15
penalty 5
processing Intelligent Miner 17
statistical mining 16
user-defined 23

G
generalization 4
global query optimization 25
goals setting 7
heterogeneous replication 24

Information Warehouse 21
input
data 84
fields 85
Install
Fast Extract 71
install
DataJoiner on Windows NT 63
Intelligent Miner on Windows NT 131
Oracle client on AIX 39
Oracle client on Windows NT 60
instance
DataJoiner on AIX 53
DataJoiner on Windows NT 67
Intelligent Miner
AIX installation 75
automate process 145
client main screen 79
components on Windows NT 132
create data object 80
data preparation functions 15
data selection 147
data visualization 147
environment variables on AIX 76
Fast Extract 32
idmstart 78
mining scenario on AIX 76
mining solution on AIX 77
modes 17
overview 13
packages on AIX 75
prerequisites 75
sequence 145
starting the client 78
task guide 14
user interface 14
using Fast Extract 33
using on AIX 78
Windows installation 131
working with databases 13

language
data control 21
data definition 21
data manipulation 21
Large Object
see LOB 23
lift chart 10
linear discriminant analysis 185
linear regression 16
LOB 23
logical volume 39

memorization 4
middleware 20
mining
algorithm 16
function 16
functions 83
ODBC datasources 110
settings 83
mining base
create 80
save 82
mining process
automate 145
mode 17
application 17
clustering 17
demo 30
test 17
training 17
model
parametric 6
modeling
data 10
multidatabase
server 20
solution 20

Net.Data 26
normalize variables 151
numeric data 82
Object-Relational architecture 23
ODBC
   API 110
   configure SPSS 115
   DataJoiner interface 112
   driver 110
   driver manager 110
   environment 109
   function call 110
   SAS compliance 129
   SAS configuration 121
   SAS driver 121
   SPSS compliance level 120
OLAP
   aggregation server 23
   online analytical processing
      see OLAP 23
   open database connectivity
      see ODBC 109
Oracle
   client installation on AIX 39
   client installation on Windows NT 60
   connect client on Windows NT 61
   create administrative group 40
   create administrative user on AIX 40
   database environment 29
   environment 29
   environment variables on AIX 40
   functional limitation 159
   installation options on Windows NT 60
   post-installation tasks on AIX 42
   pre-installation tasks on AIX 40
   SQL*Plus 61
oracollect 71
outlier treatment 86
output
   fields 86

parametric model 6
penalty function 5
pivot field 15
preprocessing 8
principal component analysis 10, 16
PROC SQL 121
processing
   Intelligent Miner functions 17
propagation
   pull 24
   push 24
query optimization 25
recursive SQL 24
recycle
   DataJoiner instance 55
result
   create mining 17
   measuring 11
   object 17
   visualize 17
results screen 86
sampling 8
SAS
   data libraries 121
   data set 121
   DataJoiner configuration 125
   functional limitation 161
   local DDE access 126
   Network DDE 127
   ODBC compliance 129
   ODBC configuration 121
   ODBC driver 121
   remote access 127
   server communication 126
   TCP/IP 127
   universal driver 121
   universal ODBC configuration 122
   universal ODBC setup 122
   sequential pattern discovery 16
   similar time sequence 17
   star schema 26
   statistical function 16
   statistics
      bivariate 16
ITSO Redbook Evaluation

Mining Relational and Nonrelational Data with IBM Intelligent Miner for Data Using Oracle, SPSS, and SAS As Sample Data Sources
SG24-5278-00

Your feedback is very important to help us maintain the quality of ITSO redbooks. Please complete this questionnaire and return it using one of the following methods:

• Use the online evaluation form found at http://www.redbooks.ibm.com
• Fax this form to: USA International Access Code + 1 914 432 8264
• Send your comments in an Internet note to redbook@us.ibm.com

Which of the following best describes you?
_ Customer _ Business Partner _ Solution Developer _ IBM employee
_ None of the above

Please rate your overall satisfaction with this book using the scale:
(1 = very good, 2 = good, 3 = average, 4 = poor, 5 = very poor)

Overall Satisfaction __________

Please answer the following questions:

Was this redbook published in time for your needs? Yes___ No___

If no, please explain:

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

What other redbooks would you like to see published?

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________

Comments/Suggestions: (THANK YOU FOR YOUR FEEDBACK!)

________________________________________________________________________
________________________________________________________________________
________________________________________________________________________