brief contents

<table>
<thead>
<tr>
<th>PART 1</th>
<th>RECOMMENDATIONS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Meet Apache Mahout</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Introducing recommenders</td>
<td>13</td>
</tr>
<tr>
<td>3</td>
<td>Representing recommender data</td>
<td>26</td>
</tr>
<tr>
<td>4</td>
<td>Making recommendations</td>
<td>41</td>
</tr>
<tr>
<td>5</td>
<td>Taking recommenders to production</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>Distributing recommendation computations</td>
<td>91</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PART 2</th>
<th>CLUSTERING</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Introduction to clustering</td>
<td>117</td>
</tr>
<tr>
<td>8</td>
<td>Representing data</td>
<td>130</td>
</tr>
<tr>
<td>9</td>
<td>Clustering algorithms in Mahout</td>
<td>145</td>
</tr>
<tr>
<td>10</td>
<td>Evaluating and improving clustering quality</td>
<td>184</td>
</tr>
<tr>
<td>11</td>
<td>Taking clustering to production</td>
<td>198</td>
</tr>
<tr>
<td>12</td>
<td>Real-world applications of clustering</td>
<td>210</td>
</tr>
</tbody>
</table>
PART 3  CLASSIFICATION ............................................................ 225

13  •  Introduction to classification  227
14  •  Training a classifier    255
15  •  Evaluating and tuning a classifier  281
16  •  Deploying a classifier    307
17  •  Case study: Shop It To Me  341
1 Meet Apache Mahout 1

1.1 Mahout's story 2

1.2 Mahout's machine learning themes 3
   * Recommender engines 3  * Clustering 3  * Classification 4

1.3 Tackling large scale with Mahout and Hadoop 5

1.4 Setting up Mahout 6
   * Java and IDEs 7  * Installing Maven 8  * Installing Mahout 8  * Installing Hadoop 9

1.5 Summary 9

PART 1 RECOMMENDATIONS .................................11

Introducing recommenders 13

2.1 Defining recommendation 14
2.2 Running a first recommender engine 15
   Creating the input 15 * Creating a recommender 16
   Analyzing the output 17

2.3 Evaluating a recommender 18
   Training data and scoring 18 * Running
   RecommenderEvaluator 19 * Assessing the result 20

2.4 Evaluating precision and recall 21
   Running RecommenderlRSStatsEvaluator 21 * Problems with
   precision and recall 23

2.5 Evaluating the GroupLens data set 23
   Extracting the recommender input 23 * Experimenting with other
   recommenders 24

2.6 Summary 25

Representing recommender data 26

3.1 Representing preference data 27
   The Preference object 27 * PreferenceArray and
   implementations 28 * Speeding up collections 28
   FastBylDMap and FastlDSet 29

3.2 In-memory DataModels 30
   GenericDataModel 30 * File-based data 30 * Refreshable
   components 31 * Update files 32 * Database-based data 32
   JDBC and MySQL 32 * Configuring via JNDI 33
   Configuring programmatically 34

3.3 Coping without preference values 34
   When to ignore values 35 * In-memory representations without
   preference values 36 * Selecting compatible implementations 37

3.4 Summary 39

Making recommendations 41

4.1 Understanding user-based recommendation 42
   When recommendation goes wrong 42 * When recommendation
   goes right 42

4.2 Exploring the user-based recommender 43
   The algorithm 43 * Implementing the algorithm with
   GenericUserBasedRecommender 44 * Exploring with
   GroupLens 45 * Exploring user neighborhoods 46
   Fixed-size neighborhoods 46 * Threshold-based neighborhood 47
## CONTENTS

5.6 Updating and monitoring the recommender 88  
5.7 Summary 89

**Distributing recommendation computations** 91  
6.1 Analyzing the Wikipedia data set 92  
   * Struggling with scale 93  
   * Evaluating benefits and drawbacks of distributing computations 93  
6.2 Designing a distributed item-based algorithm 95  
   * Constructing a co-occurrence matrix 95  
   * Computing user vectors 96  
   • Producing the recommendations 96  
   * Understanding the results 97  
   * Towards a distributed implementation 98  
6.3 Implementing a distributed algorithm with MapReduce 98  
   * Introducing MapReduce 98  
   * Translating to MapReduce: generating user vectors 99  
   * Translating to MapReduce: calculating co-occurrence 100  
   * Translating to MapReduce: rethinking matrix multiplication 101  
   * Translating to MapReduce: matrix multiplication by partial products 102  
   * Translating to MapReduce: making recommendations 105  
6.4 Running MapReduces with Hadoop 107  
   * Setting up Hadoop 107  
   * Running recommendations with Hadoop 108  
   * Configuring mappers and reducers 110  
6.5 Pseudo-distributing a recommender 110  
6.6 Looking beyond first steps with recommendations 112  
   * Running in the cloud 112  
   * Imagining unconventional uses of recommendations 113  
6.7 Summary 114

**PART 2 CLUSTERING** 115

**7 Introduction to clustering** 117  
7.1 Clustering basics 118  
7.2 Measuring the similarity of items 119  
7.3 Hello World: running a simple clustering example 120  
   * Creating the input 120  
   * Using Mahout clustering 122  
   * Analyzing the output 125
CONTENTS

7.4 Exploring distance measures 125
   Euclidean distance measure 126 * Squared Euclidean distance measure 126 * Manhattan distance measure 126 * Cosine distance measure 127 * Tanimoto distance measure 128 * Weighted distance measure 128

7.5 Hello World again! Trying out various distance measures 129

7.6 Summary 129

Representing data 130

8.1 Visualizing vectors 131
   Transforming data into vectors 132 * Preparing vectors for use by Mahout 134

8.2 Representing text documents as vectors 135
   Improving weighting with TF-IDF 136 * Accounting for word dependencies with n-gram collocations 137

8.3 Generating vectors from documents 138

8.4 Improving quality of vectors using normalization 143

8.5 Summary 144

Clustering algorithms in Mahout 145

9.1 K-means clustering 146
   All you need to know about k-means 147 * Running k-means clustering 148 * Finding the perfect k using canopy clustering 155 * Case study: clustering news articles using k-means 160

9.2 Beyond k-means: an overview of clustering techniques 163
   Different kinds of clustering problems 163 * Different clustering approaches 166

9.3 Fuzzy k-means clustering 168
   Running fuzzy k-means clustering 168 * How fuzzy is too fuzzy? 170 * Case study: clustering news articles using fuzzy k-means 170

9.4 Model-based clustering 171
   Deficiencies of k-means 172 * Dirichlet clustering 173
   Running a model-based clustering example 174
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.5</td>
<td>Topic modeling using latent Dirichlet allocation (LDA)</td>
<td>177</td>
</tr>
<tr>
<td></td>
<td>Understanding latent Dirichlet analysis 178 * TF-IDF vs. LDA 179 * Tuning the parameters of LDA 179 * Case study: finding topics in news documents 180 * Applications of topic modeling 182</td>
<td></td>
</tr>
<tr>
<td>9.6</td>
<td>Summary</td>
<td>182</td>
</tr>
<tr>
<td>10</td>
<td>Evaluating and improving clustering quality</td>
<td>184</td>
</tr>
<tr>
<td>10.1</td>
<td>Inspecting clustering output</td>
<td>185</td>
</tr>
<tr>
<td>10.2</td>
<td>Analyzing clustering output</td>
<td>187</td>
</tr>
<tr>
<td></td>
<td>Distance measure and feature selection 188 * Inter-cluster and intra-cluster distances 188 * Mixed and overlapping clusters 191</td>
<td></td>
</tr>
<tr>
<td>10.3</td>
<td>Improving clustering quality</td>
<td>192</td>
</tr>
<tr>
<td></td>
<td>Improving document vector generation 192 * Writing a custom distance measure 195</td>
<td></td>
</tr>
<tr>
<td>10.4</td>
<td>Summary</td>
<td>197</td>
</tr>
<tr>
<td>11</td>
<td>Taking clustering to production</td>
<td>198</td>
</tr>
<tr>
<td>11.1</td>
<td>Quick-start tutorial for running clustering on Hadoop</td>
<td>199</td>
</tr>
<tr>
<td></td>
<td>Running clustering on a local Hadoop cluster 199 * Customizing Hadoop configurations 201</td>
<td></td>
</tr>
<tr>
<td>11.2</td>
<td>Tuning clustering performance</td>
<td>202</td>
</tr>
<tr>
<td></td>
<td>Avoiding performance pitfalls in CPU-bound operations 203 * Avoiding performance pitfalls in I/O-bound operations 204</td>
<td></td>
</tr>
<tr>
<td>11.3</td>
<td>Batch and online clustering</td>
<td>205</td>
</tr>
<tr>
<td></td>
<td>Case study: online news clustering 206 * Case study: clustering Wikipedia articles 207</td>
<td></td>
</tr>
<tr>
<td>11.4</td>
<td>Summary</td>
<td>209</td>
</tr>
<tr>
<td>12</td>
<td>Real-world applications of clustering</td>
<td>210</td>
</tr>
<tr>
<td>12.1</td>
<td>Finding similar users on Twitter</td>
<td>211</td>
</tr>
<tr>
<td></td>
<td>Data preprocessing and feature weighting 211 * Avoiding common pitfalls in feature selection 212</td>
<td></td>
</tr>
<tr>
<td>12.2</td>
<td>Suggesting tags for artists on Lastim</td>
<td>216</td>
</tr>
<tr>
<td></td>
<td>Tag suggestion using co-occurrence 216 * Creating a dictionary of Last.fm artists 217 * Converting Last.fm tags into Vectors with musicians as features 219 * Running k-means over the Last.fm data 220</td>
<td></td>
</tr>
</tbody>
</table>
14.4 Classifying the 20 newsgroups data set with SGD 265
   * Getting started: previewing the data set 266
   * Parsing and tokenizing features for the 20 newsgroups data 268
   * Training code for the 20 newsgroups data 268

14.5 Choosing an algorithm to train the classifier 273
   * Nonparallel but powerful: using SGD and SVM 274
   * The power of the naive classifier: using naive Bayes and complementary naive Bayes 275
   * Strength in elaborate structure: using random forests 276

14.6 Classifying the 20 newsgroups data with naive Bayes 276
   * Getting started: data extraction for naive Bayes 276
   * Training the naive Bayes classifier 278
   * Testing a naive Bayes model 278

14.7 Summary 280

15 Evaluating and tuning a classifier 281
   15.1 Classifier evaluation in Mahout 282
   * Getting rapid feedback 282
   * Deciding what "good" means 282
   * Recognizing the difference in cost of errors 284

   15.2 The classifier evaluation API 284
   * Computation of AUC 285
   * Confusion matrices and entropy matrices 287
   * Computing average log likelihood 289
   * Dissecting a model 290
   * Performance of the SGD classifier with 20 newsgroups 291

   15.3 When classifiers go bad 295
   * Target leaks 295
   * Broken feature extraction 298

   15.4 Tuning for better performance 300
   * Tuning the problem 300
   * Tuning the classifier 304

   15.5 Summary 306

16 Deploying a classifier 307
   16.1 Process for deployment in huge systems 308
   * Scope out the problem 308
   * Optimize feature extraction as needed 309
   * Optimize vector encoding as needed 309
   * Deploy a scalable classifier service 310

   16.2 Determining scale and speed requirements 310
   * How big is big? 310
   * Balancing big versus fast 312
16.3 Building a training pipeline for large systems 313
   * Acquiring and retaining large-scale data 314
   * Denormalizing and downsampling 316
   * Training pitfalls 318
   * Reading and encoding data at speed 320

16.4 Integrating a Mahout classifier 324
   * Plan ahead: key issues for integration 325
   * Model serialization 330

16.5 Example: a Thrift-based classification server 332
   * Running the classification server 336
   * Accessing the classifier service 338

16.6 Summary 340

Case study: Shop It To Me 341

17.1 Why Shop It To Me chose Mahout 342
   * What Shop It To Me does 342
   * Why Shop It To Me needed a classification system 342
   * Mahout outscales the rest 343

17.2 General structure of the email marketing system 344

17.3 Training the model 346
   * Defining the goal of the classification project 346
   * Partitioning by time 348
   * Avoiding target leaks 348
   * Learning algorithm tweaks 348
   * Feature vector encoding 349

17.4 Speeding up classification 352
   * Linear combination of feature vectors 353
   * Linear expansion of model score 354

17.5 Summary 356

appendix A JVM tuning 359
appendix B Mahout math 362
appendix C Resources 367
index 369