## Contents

<table>
<thead>
<tr>
<th>Preface</th>
<th>page xi</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Teaching a computer to distinguish cats from dogs</td>
<td>1</td>
</tr>
<tr>
<td>1.1.1 The pipeline of a typical machine learning problem</td>
<td>5</td>
</tr>
<tr>
<td>1.2 Predictive learning problems</td>
<td>6</td>
</tr>
<tr>
<td>1.2.1 Regression</td>
<td>6</td>
</tr>
<tr>
<td>1.2.2 Classification</td>
<td>9</td>
</tr>
<tr>
<td>1.3 Feature design</td>
<td>12</td>
</tr>
<tr>
<td>1.4 Numerical optimization</td>
<td>15</td>
</tr>
<tr>
<td>1.5 Summary</td>
<td>16</td>
</tr>
</tbody>
</table>

### Part I Fundamental tools and concepts

<table>
<thead>
<tr>
<th>2 Fundamentals of numerical optimization</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Calculus-defined optimality</td>
<td>21</td>
</tr>
<tr>
<td>2.1.1 Taylor series approximations</td>
<td>21</td>
</tr>
<tr>
<td>2.1.2 The first order condition for optimality</td>
<td>22</td>
</tr>
<tr>
<td>2.1.3 The convenience of convexity</td>
<td>24</td>
</tr>
<tr>
<td>2.2 Numerical methods for optimization</td>
<td>26</td>
</tr>
<tr>
<td>2.2.1 The big picture</td>
<td>27</td>
</tr>
<tr>
<td>2.2.2 Stopping condition</td>
<td>27</td>
</tr>
<tr>
<td>2.2.3 Gradient descent</td>
<td>29</td>
</tr>
<tr>
<td>2.2.4 Newton’s method</td>
<td>33</td>
</tr>
<tr>
<td>2.3 Summary</td>
<td>38</td>
</tr>
<tr>
<td>2.4 Exercises</td>
<td>38</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3 Regression</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 The basics of linear regression</td>
<td>45</td>
</tr>
<tr>
<td>3.1.1 Notation and modeling</td>
<td>45</td>
</tr>
<tr>
<td>3.1.2 The Least Squares cost function for linear regression</td>
<td>47</td>
</tr>
<tr>
<td>3.1.3 Minimization of the Least Squares cost function</td>
<td>48</td>
</tr>
</tbody>
</table>
## Contents

3.1.4 The efficacy of a learned model 50
3.1.5 Predicting the value of new input data 50

3.2 Knowledge-driven feature design for regression 51
3.2.1 General conclusions 54

3.3 Nonlinear regression and $\ell_2$ regularization 56
3.3.1 Logistic regression 56
3.3.2 Non-convex cost functions and $\ell_2$ regularization 59

3.4 Summary 61
3.5 Exercises 62

4 Classification 73
4.1 The perceptron cost functions 73
4.1.1 The basic perceptron model 73
4.1.2 The softmax cost function 75
4.1.3 The margin perceptron 78
4.1.4 Differentiable approximations to the margin perceptron 80
4.1.5 The accuracy of a learned classifier 82
4.1.6 Predicting the value of new input data 83
4.1.7 Which cost function produces the best results? 84
4.1.8 The connection between the perceptron and counting costs 85

4.2 The logistic regression perspective on the softmax cost 86
4.2.1 Step functions and classification 87
4.2.2 Convex logistic regression 89

4.3 The support vector machine perspective on the margin perceptron 91
4.3.1 A quest for the hyperplane with maximum margin 91
4.3.2 The hard-margin SVM problem 93
4.3.3 The soft-margin SVM problem 93
4.3.4 Support vector machines and logistic regression 95

4.4 Multiclass classification 95
4.4.1 One-versus-all multiclass classification 96
4.4.2 Multiclass softmax classification 99
4.4.3 The accuracy of a learned multiclass classifier 103
4.4.4 Which multiclass classification scheme works best? 104

4.5 Knowledge-driven feature design for classification 104
4.5.1 General conclusions 106

4.6 Histogram features for real data types 107
4.6.1 Histogram features for text data 109
4.6.2 Histogram features for image data 112
4.6.3 Histogram features for audio data 115

4.7 Summary 117
4.8 Exercises 118
## Part II Tools for fully data-driven machine learning

### 5 Automatic feature design for regression

5.1 Automatic feature design for the ideal regression scenario

5.1.1 Vector approximation

5.1.2 From vectors to continuous functions

5.1.3 Continuous function approximation

5.1.4 Common bases for continuous function approximation

5.1.5 Recovering weights

5.1.6 Graphical representation of a neural network

5.2 Automatic feature design for the real regression scenario

5.2.1 Approximation of discretized continuous functions

5.2.2 The real regression scenario

5.3 Cross-validation for regression

5.3.1 Diagnosing the problem of overfitting/underfitting

5.3.2 Hold out cross-validation

5.3.3 Hold out calculations

5.3.4 $k$-fold cross-validation

5.4 Which basis works best?

5.4.1 Understanding of the phenomenon underlying the data

5.4.2 Practical considerations

5.4.3 When the choice of basis is arbitrary

5.5 Summary

5.6 Exercises

5.7 Notes on continuous function approximation

### 6 Automatic feature design for classification

6.1 Automatic feature design for the ideal classification scenario

6.1.1 Approximation of piecewise continuous functions

6.1.2 The formal definition of an indicator function

6.1.3 Indicator function approximation

6.1.4 Recovering weights

6.2 Automatic feature design for the real classification scenario

6.2.1 Approximation of discretized indicator functions

6.2.2 The real classification scenario

6.2.3 Classifier accuracy and boundary definition

6.3 Multiclass classification

6.3.1 One-versus-all multiclass classification

6.3.2 Multiclass softmax classification

6.4 Cross-validation for classification

6.4.1 Hold out cross-validation

6.4.2 Hold out calculations
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.4.3</td>
<td>$k$-fold cross-validation</td>
<td>184</td>
</tr>
<tr>
<td>6.4.4</td>
<td>$k$-fold cross-validation for one-versus-all multiclass classification</td>
<td>187</td>
</tr>
<tr>
<td>6.5</td>
<td>Which basis works best?</td>
<td>187</td>
</tr>
<tr>
<td>6.6</td>
<td>Summary</td>
<td>188</td>
</tr>
<tr>
<td>6.7</td>
<td>Exercises</td>
<td>189</td>
</tr>
</tbody>
</table>

#### 7 Kernels, backpropagation, and regularized cross-validation

7.1 Fixed feature kernels  
7.1.1 The fundamental theorem of linear algebra  
7.1.2 Kernelizing cost functions  
7.1.3 The value of kernelization  
7.1.4 Examples of kernels  
7.1.5 Kernels as similarity matrices  
7.2 The backpropagation algorithm  
7.2.1 Computing the gradient of a two layer network cost function  
7.2.2 Three layer neural network gradient calculations  
7.2.3 Gradient descent with momentum  
7.3 Cross-validation via $\ell^2$ regularization  
7.3.1 $\ell^2$ regularization and cross-validation  
7.3.2 Regularized $k$-fold cross-validation for regression  
7.3.3 Regularized cross-validation for classification  
7.4 Summary  
7.5 Further kernel calculations  
7.5.1 Kernelizing various cost functions  
7.5.2 Fourier kernel calculations – scalar input  
7.5.3 Fourier kernel calculations – vector input  

#### Part III Methods for large scale machine learning

8 Advanced gradient schemes  
8.1 Fixed step length rules for gradient descent  
8.1.1 Gradient descent and simple quadratic surrogates  
8.1.2 Functions with bounded curvature and optimally conservative step length rules  
8.1.3 How to use the conservative fixed step length rule  
8.2 Adaptive step length rules for gradient descent  
8.2.1 Adaptive step length rule via backtracking line search  
8.2.2 How to use the adaptive step length rule  
8.3 Stochastic gradient descent  
8.3.1 Decomposing the gradient  
8.3.2 The stochastic gradient descent iteration  
8.3.3 The value of stochastic gradient descent
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3.4</td>
<td>233</td>
</tr>
<tr>
<td>8.3.5</td>
<td>234</td>
</tr>
<tr>
<td>8.4</td>
<td>235</td>
</tr>
<tr>
<td>8.4.1</td>
<td>236</td>
</tr>
<tr>
<td>8.4.2</td>
<td>236</td>
</tr>
<tr>
<td>8.4.3</td>
<td>238</td>
</tr>
<tr>
<td>8.4.4</td>
<td>239</td>
</tr>
<tr>
<td>8.5</td>
<td>241</td>
</tr>
<tr>
<td>8.6</td>
<td>243</td>
</tr>
<tr>
<td>8.7</td>
<td>243</td>
</tr>
</tbody>
</table>

#### Dimension reduction techniques

9.1 Techniques for data dimension reduction

9.1.1 Random subsampling

9.1.2 K-means clustering

9.1.3 Optimization of the K-means problem

9.2 Principal component analysis

9.2.1 Optimization of the PCA problem

9.3 Recommender systems

9.3.1 Matrix completion setup

9.3.2 Optimization of the matrix completion model

9.4 Summary

9.5 Exercises

#### Part IV Appendices

A Basic vector and matrix operations

A.1 Vector operations

A.2 Matrix operations

B Basics of vector calculus

B.1 Basic definitions

B.2 Commonly used rules for computing derivatives

B.3 Examples of gradient and Hessian calculations

C Fundamental matrix factorizations and the pseudo-inverse

C.1 Fundamental matrix factorizations

C.1.1 The singular value decomposition

C.1.2 Eigenvalue decomposition

C.1.3 The pseudo-inverse
<table>
<thead>
<tr>
<th>D</th>
<th>Convex geometry</th>
<th>278</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D.1 Definitions of convexity</td>
<td>278</td>
</tr>
<tr>
<td></td>
<td>D.1.1 Zeroth order definition of a convex function</td>
<td>278</td>
</tr>
<tr>
<td></td>
<td>D.1.2 First order definition of a convex function</td>
<td>279</td>
</tr>
</tbody>
</table>

References 280

Index 285