

# Statistics and ML – 1

A. Naive Bayesian Classifiers

B. The Expectation-Maximization (EM) algorithm

Edoardo Milotti

Advanced Statistics for Physics

A.Y. 2023-24

# Statistics and Machine Learning (ML)

## Preface

Pattern recognition has its origins in engineering, whereas machine learning grew out of computer science. However, these activities can be viewed as two facets of the same field, and together they have undergone substantial development over the past ten years. In particular, Bayesian methods have grown from a specialist niche to become mainstream, while graphical models have emerged as a general framework for describing and applying probabilistic models. Also, the practical applicability of Bayesian methods has been greatly enhanced through the development of a range of approximate inference algorithms such as variational Bayes and expectation propagation. Similarly, new models based on kernels have had significant impact on both algorithms and applications.

## 1.2. Probability Theory

A key concept in the field of pattern recognition is that of uncertainty. It arises both through noise on measurements, as well as through the finite size of data sets. Probability theory provides a consistent framework for the quantification and manipulation of uncertainty and forms one of the central foundations for pattern recognition. When combined with decision theory, discussed in Section 1.5, it allows us to make optimal predictions given all the information available to us, even though that information may be incomplete or ambiguous.



**PATTERN RECOGNITION  
AND MACHINE LEARNING  
CHRISTOPHER M. BISHOP**

# Contents

<b>Preface</b>	<b>vii</b>
<b>Mathematical notation</b>	<b>xi</b>
<b>Contents</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Example: Polynomial Curve Fitting	4
1.2 Probability Theory	12
1.2.1 Probability densities	17
1.2.2 Expectations and covariances	19
1.2.3 Bayesian probabilities	21
1.2.4 The Gaussian distribution	24
1.2.5 Curve fitting re-visited	28
1.2.6 Bayesian curve fitting	30
1.3 Model Selection	32
1.4 The Curse of Dimensionality	33
1.5 Decision Theory	38
1.5.1 Minimizing the misclassification rate	39
1.5.2 Minimizing the expected loss	41
1.5.3 The reject option	42
1.5.4 Inference and decision	42
1.5.5 Loss functions for regression	46
1.6 Information Theory	48
1.6.1 Relative entropy and mutual information	55
Exercises	58

<b>2 Probability Distributions</b>	<b>67</b>
2.1 Binary Variables	68
2.1.1 The beta distribution	71
2.2 Multinomial Variables	74
2.2.1 The Dirichlet distribution	76
2.3 The Gaussian Distribution	78
2.3.1 Conditional Gaussian distributions	85
2.3.2 Marginal Gaussian distributions	88
2.3.3 Bayes' theorem for Gaussian variables	90
2.3.4 Maximum likelihood for the Gaussian	93
2.3.5 Sequential estimation	94
2.3.6 Bayesian inference for the Gaussian	97
2.3.7 Student's t-distribution	102
2.3.8 Periodic variables	105
2.3.9 Mixtures of Gaussians	110
2.4 The Exponential Family	113
2.4.1 Maximum likelihood and sufficient statistics	116
2.4.2 Conjugate priors	117
2.4.3 Noninformative priors	117
2.5 Nonparametric Methods	120
2.5.1 Kernel density estimators	122
2.5.2 Nearest-neighbour methods	124
Exercises	127
<b>3 Linear Models for Regression</b>	<b>137</b>
3.1 Linear Basis Function Models	138
3.1.1 Maximum likelihood and least squares	140
3.1.2 Geometry of least squares	143
3.1.3 Sequential learning	143
3.1.4 Regularized least squares	144
3.1.5 Multiple outputs	146
3.2 The Bias-Variance Decomposition	147
3.3 Bayesian Linear Regression	152
3.3.1 Parameter distribution	152
3.3.2 Predictive distribution	156
3.3.3 Equivalent kernel	159
3.4 Bayesian Model Comparison	161
3.5 The Evidence Approximation	165
3.5.1 Evaluation of the evidence function	166
3.5.2 Maximizing the evidence function	168
3.5.3 Effective number of parameters	170
3.6 Limitations of Fixed Basis Functions	172
Exercises	173

<b>4</b>	<b>Linear Models for Classification</b>	<b>179</b>
4.1	Discriminant Functions . . . . .	181
4.1.1	Two classes . . . . .	181
4.1.2	Multiple classes . . . . .	182
4.1.3	Least squares for classification . . . . .	184
4.1.4	Fisher's linear discriminant . . . . .	186
4.1.5	Relation to least squares . . . . .	189
4.1.6	Fisher's discriminant for multiple classes . . . . .	191
4.1.7	The perceptron algorithm . . . . .	192
4.2	Probabilistic Generative Models . . . . .	196
4.2.1	Continuous inputs . . . . .	198
4.2.2	Maximum likelihood solution . . . . .	200
4.2.3	Discrete features . . . . .	202
4.2.4	Exponential family . . . . .	202
4.3	Probabilistic Discriminative Models . . . . .	203
4.3.1	Fixed basis functions . . . . .	204
4.3.2	Logistic regression . . . . .	205
4.3.3	Iterative reweighted least squares . . . . .	207
4.3.4	Multiclass logistic regression . . . . .	209
4.3.5	Probit regression . . . . .	210
4.3.6	Canonical link functions . . . . .	212
4.4	The Laplace Approximation . . . . .	213
4.4.1	Model comparison and BIC . . . . .	216
4.5	Bayesian Logistic Regression . . . . .	217
4.5.1	Laplace approximation . . . . .	217
4.5.2	Predictive distribution . . . . .	218
	Exercises . . . . .	220
<b>5</b>	<b>Neural Networks</b>	<b>225</b>
5.1	Feed-forward Network Functions . . . . .	227
5.1.1	Weight-space symmetries . . . . .	231
5.2	Network Training . . . . .	232
5.2.1	Parameter optimization . . . . .	236
5.2.2	Local quadratic approximation . . . . .	237
5.2.3	Use of gradient information . . . . .	239
5.2.4	Gradient descent optimization . . . . .	240
5.3	Error Backpropagation . . . . .	241
5.3.1	Evaluation of error-function derivatives . . . . .	242
5.3.2	A simple example . . . . .	245
5.3.3	Efficiency of backpropagation . . . . .	246
5.3.4	The Jacobian matrix . . . . .	247
5.4	The Hessian Matrix . . . . .	249
5.4.1	Diagonal approximation . . . . .	250
5.4.2	Outer product approximation . . . . .	251
5.4.3	Inverse Hessian . . . . .	252

5.4.4	Finite differences . . . . .	252
5.4.5	Exact evaluation of the Hessian . . . . .	253
5.4.6	Fast multiplication by the Hessian . . . . .	254
5.5	Regularization in Neural Networks . . . . .	256
5.5.1	Consistent Gaussian priors . . . . .	257
5.5.2	Early stopping . . . . .	259
5.5.3	Invariances . . . . .	261
5.5.4	Tangent propagation . . . . .	263
5.5.5	Training with transformed data . . . . .	265
5.5.6	Convolutional networks . . . . .	267
5.5.7	Soft weight sharing . . . . .	269
5.6	Mixture Density Networks . . . . .	272
5.7	Bayesian Neural Networks . . . . .	277
5.7.1	Posterior parameter distribution . . . . .	278
5.7.2	Hyperparameter optimization . . . . .	280
5.7.3	Bayesian neural networks for classification . . . . .	281
	Exercises . . . . .	284
<b>6</b>	<b>Kernel Methods</b>	<b>291</b>
6.1	Dual Representations . . . . .	293
6.2	Constructing Kernels . . . . .	294
6.3	Radial Basis Function Networks . . . . .	299
6.3.1	Nadaraya-Watson model . . . . .	301
6.4	Gaussian Processes . . . . .	303
6.4.1	Linear regression revisited . . . . .	304
6.4.2	Gaussian processes for regression . . . . .	306
6.4.3	Learning the hyperparameters . . . . .	311
6.4.4	Automatic relevance determination . . . . .	312
6.4.5	Gaussian processes for classification . . . . .	313
6.4.6	Laplace approximation . . . . .	315
6.4.7	Connection to neural networks . . . . .	319
	Exercises . . . . .	320
<b>7</b>	<b>Sparse Kernel Machines</b>	<b>325</b>
7.1	Maximum Margin Classifiers . . . . .	326
7.1.1	Overlapping class distributions . . . . .	331
7.1.2	Relation to logistic regression . . . . .	336
7.1.3	Multiclass SVMs . . . . .	338
7.1.4	SVMs for regression . . . . .	339
7.1.5	Computational learning theory . . . . .	344
7.2	Relevance Vector Machines . . . . .	345
7.2.1	RVM for regression . . . . .	345
7.2.2	Analysis of sparsity . . . . .	349
7.2.3	RVM for classification . . . . .	353
	Exercises . . . . .	357

<b>8</b>	<b>Graphical Models</b>	<b>359</b>
8.1	Bayesian Networks . . . . .	360
8.1.1	Example: Polynomial regression . . . . .	362
8.1.2	Generative models . . . . .	365
8.1.3	Discrete variables . . . . .	366
8.1.4	Linear-Gaussian models . . . . .	370
8.2	Conditional Independence . . . . .	372
8.2.1	Three example graphs . . . . .	373
8.2.2	D-separation . . . . .	378
8.3	Markov Random Fields . . . . .	383
8.3.1	Conditional independence properties . . . . .	383
8.3.2	Factorization properties . . . . .	384
8.3.3	Illustration: Image de-noising . . . . .	387
8.3.4	Relation to directed graphs . . . . .	390
8.4	Inference in Graphical Models . . . . .	393
8.4.1	Inference on a chain . . . . .	394
8.4.2	Trees . . . . .	398
8.4.3	Factor graphs . . . . .	399
8.4.4	The sum-product algorithm . . . . .	402
8.4.5	The max-sum algorithm . . . . .	411
8.4.6	Exact inference in general graphs . . . . .	416
8.4.7	Loopy belief propagation . . . . .	417
8.4.8	Learning the graph structure . . . . .	418
	Exercises . . . . .	418
<b>9</b>	<b>Mixture Models and EM</b>	<b>423</b>
9.1	$K$ -means Clustering . . . . .	424
9.1.1	Image segmentation and compression . . . . .	428
9.2	Mixtures of Gaussians . . . . .	430
9.2.1	Maximum likelihood . . . . .	432
9.2.2	EM for Gaussian mixtures . . . . .	435
9.3	An Alternative View of EM . . . . .	439
9.3.1	Gaussian mixtures revisited . . . . .	441
9.3.2	Relation to $K$ -means . . . . .	443
9.3.3	Mixtures of Bernoulli distributions . . . . .	444
9.3.4	EM for Bayesian linear regression . . . . .	448
9.4	The EM Algorithm in General . . . . .	450
	Exercises . . . . .	455
<b>10</b>	<b>Approximate Inference</b>	<b>461</b>
10.1	Variational Inference . . . . .	462
10.1.1	Factorized distributions . . . . .	464
10.1.2	Properties of factorized approximations . . . . .	466
10.1.3	Example: The univariate Gaussian . . . . .	470
10.1.4	Model comparison . . . . .	473
10.2	Illustration: Variational Mixture of Gaussians . . . . .	474

10.2.1	Variational distribution . . . . .	475
10.2.2	Variational lower bound . . . . .	481
10.2.3	Predictive density . . . . .	482
10.2.4	Determining the number of components . . . . .	483
10.2.5	Induced factorizations . . . . .	485
10.3	Variational Linear Regression . . . . .	486
10.3.1	Variational distribution . . . . .	486
10.3.2	Predictive distribution . . . . .	488
10.3.3	Lower bound . . . . .	489
10.4	Exponential Family Distributions . . . . .	490
10.4.1	Variational message passing . . . . .	491
10.5	Local Variational Methods . . . . .	493
10.6	Variational Logistic Regression . . . . .	498
10.6.1	Variational posterior distribution . . . . .	498
10.6.2	Optimizing the variational parameters . . . . .	500
10.6.3	Inference of hyperparameters . . . . .	502
10.7	Expectation Propagation . . . . .	505
10.7.1	Example: The clutter problem . . . . .	511
10.7.2	Expectation propagation on graphs . . . . .	513
	Exercises . . . . .	517
<b>11</b>	<b>Sampling Methods</b>	<b>523</b>
11.1	Basic Sampling Algorithms . . . . .	526
11.1.1	Standard distributions . . . . .	526
11.1.2	Rejection sampling . . . . .	528
11.1.3	Adaptive rejection sampling . . . . .	530
11.1.4	Importance sampling . . . . .	532
11.1.5	Sampling-importance-resampling . . . . .	534
11.1.6	Sampling and the EM algorithm . . . . .	536
11.2	Markov Chain Monte Carlo . . . . .	537
11.2.1	Markov chains . . . . .	539
11.2.2	The Metropolis-Hastings algorithm . . . . .	541
11.3	Gibbs Sampling . . . . .	542
11.4	Slice Sampling . . . . .	546
11.5	The Hybrid Monte Carlo Algorithm . . . . .	548
11.5.1	Dynamical systems . . . . .	548
11.5.2	Hybrid Monte Carlo . . . . .	552
11.6	Estimating the Partition Function . . . . .	554
	Exercises . . . . .	556
<b>12</b>	<b>Continuous Latent Variables</b>	<b>559</b>
12.1	Principal Component Analysis . . . . .	561
12.1.1	Maximum variance formulation . . . . .	561
12.1.2	Minimum-error formulation . . . . .	563
12.1.3	Applications of PCA . . . . .	565
12.1.4	PCA for high-dimensional data . . . . .	569

12.2	Probabilistic PCA . . . . .	570
12.2.1	Maximum likelihood PCA . . . . .	574
12.2.2	EM algorithm for PCA . . . . .	577
12.2.3	Bayesian PCA . . . . .	580
12.2.4	Factor analysis . . . . .	583
12.3	Kernel PCA . . . . .	586
12.4	Nonlinear Latent Variable Models . . . . .	591
12.4.1	Independent component analysis . . . . .	591
12.4.2	Autoassociative neural networks . . . . .	592
12.4.3	Modelling nonlinear manifolds . . . . .	595
	Exercises . . . . .	599
<b>13</b>	<b>Sequential Data</b>	<b>605</b>
13.1	Markov Models . . . . .	607
13.2	Hidden Markov Models . . . . .	610
13.2.1	Maximum likelihood for the HMM . . . . .	615
13.2.2	The forward-backward algorithm . . . . .	618
13.2.3	The sum-product algorithm for the HMM . . . . .	625
13.2.4	Scaling factors . . . . .	627
13.2.5	The Viterbi algorithm . . . . .	629
13.2.6	Extensions of the hidden Markov model . . . . .	631
13.3	Linear Dynamical Systems . . . . .	635
13.3.1	Inference in LDS . . . . .	638
13.3.2	Learning in LDS . . . . .	642
13.3.3	Extensions of LDS . . . . .	644
13.3.4	Particle filters . . . . .	645
	Exercises . . . . .	646
<b>14</b>	<b>Combining Models</b>	<b>653</b>
14.1	Bayesian Model Averaging . . . . .	654
14.2	Committees . . . . .	655
14.3	Boosting . . . . .	657
14.3.1	Minimizing exponential error . . . . .	659
14.3.2	Error functions for boosting . . . . .	661
14.4	Tree-based Models . . . . .	663
14.5	Conditional Mixture Models . . . . .	666
14.5.1	Mixtures of linear regression models . . . . .	667
14.5.2	Mixtures of logistic models . . . . .	670
14.5.3	Mixtures of experts . . . . .	672
	Exercises . . . . .	674
	<b>Appendix A Data Sets</b>	<b>677</b>
	<b>Appendix B Probability Distributions</b>	<b>685</b>
	<b>Appendix C Properties of Matrices</b>	<b>695</b>

## Who Should Read This Book?

This book is intended for anyone who is interested in using modern statistical methods for modeling and prediction from data. This group includes scientists, engineers, data analysts, data scientists, and quants, but also less technical individuals with degrees in non-quantitative fields such as the social sciences or business. We expect that the reader will have had at least one elementary course in statistics. Background in linear regression is also useful, though not required, since we review the key concepts behind linear regression in Chapter 3. The mathematical level of this book is modest, and a detailed knowledge of matrix operations is not required. This book provides an introduction to **Python**. Previous exposure to a programming language, such as **MATLAB** or **R**, is useful but not required.


<https://www.statlearning.com>

Springer Texts in Statistics

Gareth James · Daniela Witten · Trevor Hastie ·  
Robert Tibshirani · Jonathan Taylor

# An Introduction to Statistical Learning

with Applications in Python

 Springer

**Preface** **vii**

**1 Introduction** **1**

**2 Statistical Learning** **15**

- 2.1 What Is Statistical Learning? . . . . . 15
  - 2.1.1 Why Estimate  $f$ ? . . . . . 17
  - 2.1.2 How Do We Estimate  $f$ ? . . . . . 20
  - 2.1.3 The Trade-Off Between Prediction Accuracy and Model Interpretability . . . . . 23
  - 2.1.4 Supervised Versus Unsupervised Learning . . . . . 25
  - 2.1.5 Regression Versus Classification Problems . . . . . 27
- 2.2 Assessing Model Accuracy . . . . . 27
  - 2.2.1 Measuring the Quality of Fit . . . . . 28
  - 2.2.2 The Bias-Variance Trade-Off . . . . . 31
  - 2.2.3 The Classification Setting . . . . . 34
- 2.3 Lab: Introduction to Python . . . . . 40
  - 2.3.1 Getting Started . . . . . 40
  - 2.3.2 Basic Commands . . . . . 40
  - 2.3.3 Introduction to Numerical Python . . . . . 42
  - 2.3.4 Graphics . . . . . 48
  - 2.3.5 Sequences and Slice Notation . . . . . 51
  - 2.3.6 Indexing Data . . . . . 51
  - 2.3.7 Loading Data . . . . . 55
  - 2.3.8 For Loops . . . . . 59
  - 2.3.9 Additional Graphical and Numerical Summaries . . . . . 61
- 2.4 Exercises . . . . . 63

**3 Linear Regression** **69**

- 3.1 Simple Linear Regression . . . . . 70
  - 3.1.1 Estimating the Coefficients . . . . . 71
  - 3.1.2 Assessing the Accuracy of the Coefficient Estimates . . . . . 72
  - 3.1.3 Assessing the Accuracy of the Model . . . . . 77
- 3.2 Multiple Linear Regression . . . . . 80
  - 3.2.1 Estimating the Regression Coefficients . . . . . 81

- 3.2.2 Some Important Questions . . . . . 83
- 3.3 Other Considerations in the Regression Model . . . . . 91
  - 3.3.1 Qualitative Predictors . . . . . 91
  - 3.3.2 Extensions of the Linear Model . . . . . 94
  - 3.3.3 Potential Problems . . . . . 100
- 3.4 The Marketing Plan . . . . . 109
- 3.5 Comparison of Linear Regression with  $K$ -Nearest Neighbors . . . . . 111
- 3.6 Lab: Linear Regression . . . . . 116
  - 3.6.1 Importing packages . . . . . 116
  - 3.6.2 Simple Linear Regression . . . . . 117
  - 3.6.3 Multiple Linear Regression . . . . . 122
  - 3.6.4 Multivariate Goodness of Fit . . . . . 123
  - 3.6.5 Interaction Terms . . . . . 124
  - 3.6.6 Non-linear Transformations of the Predictors . . . . . 125
  - 3.6.7 Qualitative Predictors . . . . . 126
- 3.7 Exercises . . . . . 127

**Classification** **135**

- 4.1 An Overview of Classification . . . . . 135
- 4.2 Why Not Linear Regression? . . . . . 136
- 4.3 Logistic Regression . . . . . 138
  - 4.3.1 The Logistic Model . . . . . 139
  - 4.3.2 Estimating the Regression Coefficients . . . . . 140
  - 4.3.3 Making Predictions . . . . . 141
  - 4.3.4 Multiple Logistic Regression . . . . . 142
  - 4.3.5 Multinomial Logistic Regression . . . . . 144
- 4.4 Generative Models for Classification . . . . . 146
  - 4.4.1 Linear Discriminant Analysis for  $p = 1$  . . . . . 147
  - 4.4.2 Linear Discriminant Analysis for  $p > 1$  . . . . . 150
  - 4.4.3 Quadratic Discriminant Analysis . . . . . 156
  - 4.4.4 Naive Bayes . . . . . 158
- 4.5 A Comparison of Classification Methods . . . . . 161
  - 4.5.1 An Analytical Comparison . . . . . 161
  - 4.5.2 An Empirical Comparison . . . . . 164
- 4.6 Generalized Linear Models . . . . . 167
  - 4.6.1 Linear Regression on the Bikeshare Data . . . . . 167
  - 4.6.2 Poisson Regression on the Bikeshare Data . . . . . 169
  - 4.6.3 Generalized Linear Models in Greater Generality . . . . . 172
- 4.7 Lab: Logistic Regression, LDA, QDA, and KNN . . . . . 173
  - 4.7.1 The Stock Market Data . . . . . 173
  - 4.7.2 Logistic Regression . . . . . 174
  - 4.7.3 Linear Discriminant Analysis . . . . . 179
  - 4.7.4 Quadratic Discriminant Analysis . . . . . 181
  - 4.7.5 Naive Bayes . . . . . 182
  - 4.7.6  $K$ -Nearest Neighbors . . . . . 183
  - 4.7.7 Linear and Poisson Regression on the Bikeshare Data . . . . . 188
- 4.8 Exercises . . . . . 193



<b>5</b>	<b>Resampling Methods</b>	<b>202</b>
5.1	Cross-Validation	202
5.1.1	The Validation Set Approach	202
5.1.2	Leave-One-Out Cross-Validation	204
5.1.3	$k$ -Fold Cross-Validation	206
5.1.4	Bias-Variance Trade-Off for $k$ -Fold Cross-Validation	208
5.1.5	Cross-Validation on Classification Problems	209
5.2	The Bootstrap	212
5.3	Lab: Cross-Validation and the Bootstrap	215
5.3.1	The Validation Set Approach	216
5.3.2	Cross-Validation	217
5.3.3	The Bootstrap	220
5.4	Exercises	224
<b>6</b>	<b>Linear Model Selection and Regularization</b>	<b>229</b>
6.1	Subset Selection	231
6.1.1	Best Subset Selection	231
6.1.2	Stepwise Selection	233
6.1.3	Choosing the Optimal Model	235
6.2	Shrinkage Methods	240
6.2.1	Ridge Regression	240
6.2.2	The Lasso	244
6.2.3	Selecting the Tuning Parameter	252
6.3	Dimension Reduction Methods	253
6.3.1	Principal Components Regression	254
6.3.2	Partial Least Squares	260
6.4	Considerations in High Dimensions	262
6.4.1	High-Dimensional Data	262
6.4.2	What Goes Wrong in High Dimensions?	263
6.4.3	Regression in High Dimensions	265
6.4.4	Interpreting Results in High Dimensions	266
6.5	Lab: Linear Models and Regularization Methods	267
6.5.1	Subset Selection Methods	268
6.5.2	Ridge Regression and the Lasso	273
6.5.3	PCR and PLS Regression	280
6.6	Exercises	283
<b>7</b>	<b>Moving Beyond Linearity</b>	<b>289</b>
7.1	Polynomial Regression	290
7.2	Step Functions	292
7.3	Basis Functions	293
7.4	Regression Splines	294
7.4.1	Piecewise Polynomials	294
7.4.2	Constraints and Splines	296
7.4.3	The Spline Basis Representation	296
7.4.4	Choosing the Number and Locations of the Knots	299
7.4.5	Comparison to Polynomial Regression	299

7.5	Smoothing Splines	300
7.5.1	An Overview of Smoothing Splines	300
7.5.2	Choosing the Smoothing Parameter $\lambda$	301
7.6	Local Regression	303
7.7	Generalized Additive Models	305
7.7.1	GAMs for Regression Problems	306
7.7.2	GAMs for Classification Problems	308
7.8	Lab: Non-Linear Modeling	309
7.8.1	Polynomial Regression and Step Functions	310
7.8.2	Splines	315
7.8.3	Smoothing Splines and GAMs	317
7.8.4	Local Regression	324
7.9	Exercises	325
	<b>Tree-Based Methods</b>	<b>331</b>
8.1	The Basics of Decision Trees	331
8.1.1	Regression Trees	331
8.1.2	Classification Trees	337
8.1.3	Trees Versus Linear Models	341
8.1.4	Advantages and Disadvantages of Trees	341
8.2	Bagging, Random Forests, Boosting, and Bayesian Additive Regression Trees	343
8.2.1	Bagging	343
8.2.2	Random Forests	346
8.2.3	Boosting	347
8.2.4	Bayesian Additive Regression Trees	350
8.2.5	Summary of Tree Ensemble Methods	353
8.3	Lab: Tree-Based Methods	354
8.3.1	Fitting Classification Trees	355
8.3.2	Fitting Regression Trees	358
8.3.3	Bagging and Random Forests	360
8.3.4	Boosting	361
8.3.5	Bayesian Additive Regression Trees	362
8.4	Exercises	363
	<b>Support Vector Machines</b>	<b>367</b>
9.1	Maximal Margin Classifier	367
9.1.1	What Is a Hyperplane?	368
9.1.2	Classification Using a Separating Hyperplane	368
9.1.3	The Maximal Margin Classifier	370
9.1.4	Construction of the Maximal Margin Classifier	372
9.1.5	The Non-separable Case	372
9.2	Support Vector Classifiers	373
9.2.1	Overview of the Support Vector Classifier	373
9.2.2	Details of the Support Vector Classifier	374
9.3	Support Vector Machines	377
9.3.1	Classification with Non-Linear Decision Boundaries	378
9.3.2	The Support Vector Machine	379

11.5.1	The Hazard Function . . . . .	476
11.5.2	Proportional Hazards . . . . .	478
11.5.3	Example: Brain Cancer Data . . . . .	482
11.5.4	Example: Publication Data . . . . .	482
11.6	Shrinkage for the Cox Model . . . . .	484
11.7	Additional Topics . . . . .	486
11.7.1	Area Under the Curve for Survival Analysis . . . . .	486
11.7.2	Choice of Time Scale . . . . .	487
11.7.3	Time-Dependent Covariates . . . . .	488
11.7.4	Checking the Proportional Hazards Assumption . . . . .	488
11.7.5	Survival Trees . . . . .	488
11.8	Lab: Survival Analysis . . . . .	489
11.8.1	Brain Cancer Data . . . . .	489
11.8.2	Publication Data . . . . .	493
11.8.3	Call Center Data . . . . .	494
11.9	Exercises . . . . .	498

**12 Unsupervised Learning 503**

12.1	The Challenge of Unsupervised Learning . . . . .	503
12.2	Principal Components Analysis . . . . .	504
12.2.1	What Are Principal Components? . . . . .	505
12.2.2	Another Interpretation of Principal Components . . . . .	508
12.2.3	The Proportion of Variance Explained . . . . .	510
12.2.4	More on PCA . . . . .	512
12.2.5	Other Uses for Principal Components . . . . .	515
12.3	Missing Values and Matrix Completion . . . . .	515
12.4	Clustering Methods . . . . .	520
12.4.1	$K$ -Means Clustering . . . . .	521
12.4.2	Hierarchical Clustering . . . . .	525
12.4.3	Practical Issues in Clustering . . . . .	532
12.5	Lab: Unsupervised Learning . . . . .	535
12.5.1	Principal Components Analysis . . . . .	535
12.5.2	Matrix Completion . . . . .	539
12.5.3	Clustering . . . . .	542
12.5.4	NCI60 Data Example . . . . .	546
12.6	Exercises . . . . .	552

**13 Multiple Testing 557**

13.1	A Quick Review of Hypothesis Testing . . . . .	558
13.1.1	Testing a Hypothesis . . . . .	558
13.1.2	Type I and Type II Errors . . . . .	562
13.2	The Challenge of Multiple Testing . . . . .	563
13.3	The Family-Wise Error Rate . . . . .	565
13.3.1	What is the Family-Wise Error Rate? . . . . .	565
13.3.2	Approaches to Control the Family-Wise Error Rate . . . . .	567
13.3.3	Trade-Off Between the FWER and Power . . . . .	572
13.4	The False Discovery Rate . . . . .	573
13.4.1	Intuition for the False Discovery Rate . . . . .	578
13.4.2	The Benjamini–Hochberg Procedure . . . . .	575

13.5	A Re-Sampling Approach to $p$ -Values and False Discovery Rates . . . . .	577
13.5.1	A Re-Sampling Approach to the $p$ -Value . . . . .	578
13.5.2	A Re-Sampling Approach to the False Discovery Rate . . . . .	579
13.5.3	When Are Re-Sampling Approaches Useful? . . . . .	581
13.6	Lab: Multiple Testing . . . . .	583
13.6.1	Review of Hypothesis Tests . . . . .	583
13.6.2	Family-Wise Error Rate . . . . .	585
13.6.3	False Discovery Rate . . . . .	588
13.6.4	A Re-Sampling Approach . . . . .	590
13.7	Exercises . . . . .	593

**Index 597**

# A. Naive Bayesian Classifiers

# 1. Bayesian classification

data  $X$ , classes  $C$

$$P(C|X) = \frac{P(X|C)}{P(X)} P(C)$$

this likelihood is defined by training data

the prior is also defined by training data

we can use the prior learning to assign a class to new data

$$C_k = \arg \max_{C_k} \frac{P(X|C_k)}{P(X)} P(C_k) = \arg \max_{C_k} P(X|C_k) P(C_k)$$

Consider a vector of  $N$  attributes given as Boolean variables  $\mathbf{x} = \{x_i\}$  and classify the data vectors with a single Boolean variable.

The learning procedure must yield:

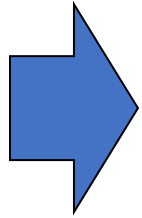
$$P(y)$$

it is easy to obtain it as an empirical distribution from a histogram of training class data:  $y$  is Boolean, the histogram has just two bins, and a hundred examples suffice to determine the empirical distribution to better than 10%.

$$P(\mathbf{x}|y)$$

there is a bigger problem here: the arguments have  $2^{N+1}$  different values, and we must estimate  $2(2^N-1)$  parameters ... for instance, with  $N = 30$  there are more than 2 billion parameters!

How can we reduce the huge complexity of learning?



we assume the conditional independence of the  $x_n$ 's:  
**naive Bayesian learning**

for instance, with just two attributes

$$P(x_1, x_2 | y) = P(x_1 | x_2, y) P(x_2 | y) = P(x_1 | y) P(x_2 | y)$$

conditional independence assumption

with more than 2 attributes

$$P(\mathbf{x} | y) \approx \prod_{k=1}^N P(x_k | y)$$

Therefore:

$$\begin{aligned} P(y_k|\mathbf{x}) &= \frac{P(\mathbf{x}|y_k)}{P(\mathbf{x})} P(y_k) = \frac{P(\mathbf{x}|y_k)}{\sum_j P(\mathbf{x}|y_j)P(y_j)} P(y_k) \\ &\approx \frac{\prod_{n=1}^N P(x_n|y_k)}{\sum_j P(y_j) \prod_{n=1}^N P(x_n|y_j)} P(y_k) \end{aligned}$$

and we assign the class according to the rule (MAP)

$$y = \arg \max_{y_k} \frac{\prod_{n=1}^N P(x_n|y_k)}{\sum_j P(y_j) \prod_{n=1}^N P(x_n|y_j)} P(y_k)$$

## ***More general discrete inputs***

If any of the  $N$   $x$  input variables has  $J$  different values, and if there are  $K$  classes, then we must estimate in all  $NK(J-1)$  free parameters with the Naive Bayes Classifier (this includes normalization) (compare this with the  $K(J^N-1)$  parameters needed by a complete classifier)



## ***Continuous inputs and discrete classes – the Gaussian case***

$$P(x_n | y_k) = \frac{1}{\sqrt{2\pi\sigma_{nk}^2}} \exp\left[-\frac{(x_n - \mu_{nk})^2}{2\sigma_{nk}^2}\right]$$

here we must estimate  $2NK$  parameters + the shape of the distribution  $P(y)$  (this adds up to another  $K-1$  parameters)

Gaussian special case with class-independent variance and Boolean classification (two classes only):

$$P(y = 0|\mathbf{x}) = \frac{P(\mathbf{x}|y = 0)P(y = 0)}{P(\mathbf{x}|y = 0)P(y = 0) + P(\mathbf{x}|y = 1)P(y = 1)}$$

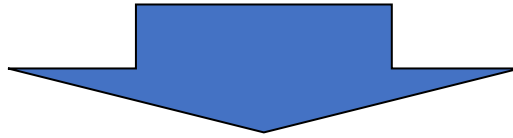
$$P(x_n|y = 0) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{(x_n - \mu_{n0})^2}{2\sigma_n^2}\right]$$

$$P(x_n|y = 1) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left[-\frac{(x_n - \mu_{n1})^2}{2\sigma_n^2}\right]$$

$$\begin{aligned}
P(y = 0 | \mathbf{x}) &= \frac{P(\mathbf{x} | y = 0)P(y = 0)}{P(\mathbf{x} | y = 0)P(y = 0) + P(\mathbf{x} | y = 1)P(y = 1)} \\
&= \frac{1}{1 + \frac{P(\mathbf{x} | y = 1)P(y = 1)}{P(\mathbf{x} | y = 0)P(y = 0)}} \\
&= \frac{1}{1 + \frac{P(y = 1)}{P(y = 0)} \prod_{n=1}^N \exp \left[ -\frac{(x_n - \mu_{n1})^2}{2\sigma_n^2} + \frac{(x_n - \mu_{n0})^2}{2\sigma_n^2} \right]} \\
&= \frac{1}{1 + \exp \left\{ \ln \left( \frac{P(y = 1)}{P(y = 0)} \right) + \sum_{n=1}^N \left[ \frac{(\mu_{n1} - \mu_{n0})x_n}{\sigma_n^2} + \frac{\mu_{n0}^2 - \mu_{n1}^2}{2\sigma_n^2} \right] \right\}}
\end{aligned}$$

$$w_0 = \ln\left(\frac{P(y=1)}{P(y=0)}\right) + \sum_{n=1}^N \left[ \frac{\mu_{n0}^2 - \mu_{n1}^2}{2\sigma_n^2} \right]$$

$$w_n = \frac{(\mu_{n1} - \mu_{n0})}{\sigma_n^2}$$



logistic shape

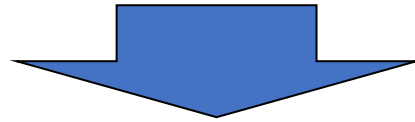
$$P(y=0|\mathbf{x}) = \frac{1}{1 + \exp\left(w_0 + \sum_{n=1}^N w_n x_n\right)}$$



$$P(y=1|\mathbf{x}) = 1 - P(y=0|\mathbf{x}) = \frac{\exp\left(w_0 + \sum_{n=1}^N w_n x_n\right)}{1 + \exp\left(w_0 + \sum_{n=1}^N w_n x_n\right)}$$

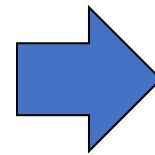
Finally, an input vector belongs to class  $y = 0$  if

$$\frac{P(y = 0|\mathbf{x})}{P(y = 1|\mathbf{x})} > 1$$

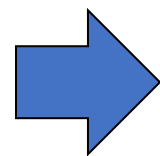


$$P(y = 0|\mathbf{x}) = \frac{1}{1 + \exp\left(w_0 + \sum_{n=1}^N w_n x_n\right)}$$

$$P(y = 1|\mathbf{x}) = \frac{\exp\left(w_0 + \sum_{n=1}^N w_n x_n\right)}{1 + \exp\left(w_0 + \sum_{n=1}^N w_n x_n\right)}$$



$$\exp\left(w_0 + \sum_{n=1}^N w_n x_n\right) < 1$$



$$w_0 + \sum_{n=1}^N w_n x_n < 0$$

# B. The Expectation-Maximization (EM) algorithm

The EM algorithm is used to maximize likelihood with incomplete information (e.g., "latent variables"), and it has two main steps that are iterated until convergence:

### E. expectation of the log-likelihood, averaged with respect to missing data:

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n-1)}) = E_{\mathbf{y}} \left[ \log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}) \middle| \mathbf{x}, \boldsymbol{\theta}^{(n-1)} \right]$$

$$= \int_Y \left[ \log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}) \right] p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}^{(n-1)}) d\mathbf{y}$$

likelihood measured data missing data parameters (with respect to which we want to maximize the expression)  
previous parameter estimate (constant values)

### M. maximization of the averaged log-likelihood with respect to parameters:

$$\boldsymbol{\theta}^{(n)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(n-1)})$$


## Example: an experiment with an exponential model (Flury and Zoppè)

Light bulbs fail following an exponential distribution with mean failure time  $\theta$

*To estimate the mean two experiments are performed*

1.  $n$  light bulbs are tested, all failure times  $u_i$  are recorded
2.  $m$  light bulbs are tested, only the total number  $r$  of bulbs failed up to time  $t$  are recorded

1. 
$$\mathcal{L} = \prod_{i=1}^n \frac{1}{\theta} \exp\left(-\frac{u_i}{\theta}\right) = \frac{1}{\theta^n} \exp\left(-\frac{\sum_i u_i}{\theta}\right) = \frac{1}{\theta^n} \exp\left(-\frac{n\langle u \rangle}{\theta}\right)$$

2. 
$$\mathcal{L} = \prod_{i=1}^m \frac{1}{\theta} \exp\left(-\frac{v_i}{\theta}\right)$$
 missing data!



combined likelihood

$$\frac{1}{\theta^n} \exp\left(-\frac{n\langle u \rangle}{\theta}\right) \cdot \prod_{i=1}^m \frac{1}{\theta} \exp\left(-\frac{v_i}{\theta}\right)$$

log-likelihood

$$-n \ln \theta - \frac{n\langle u \rangle}{\theta} - \sum_{i=1}^m \left( \ln \theta + \frac{v_i}{\theta} \right)$$

expected failure time for a bulb  
that is still burning at time  $t$

$$t + \theta$$

expected failure time for a bulb  
that is not burning at time  $t$

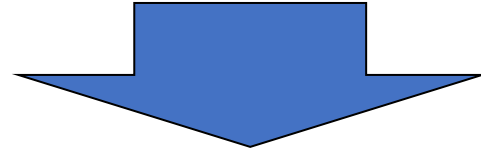
$$\theta - \frac{t \exp(-t/\theta)}{1 - \exp(-t/\theta)}$$

Note on mean failure time for a bulb that is not burning at time  $t$

$$p(t') \propto \frac{1}{\theta} e^{-t'/\theta} \quad 0 \leq t' \leq t$$

$$\text{normalization} = \int_0^t p(t') dt' = \int_0^t \frac{dt'}{\theta} e^{-t'/\theta} = 1 - e^{-t/\theta}$$

$$\begin{aligned} \text{mean failure time} &= \int_0^t t' p(t') dt' = \frac{1}{1 - e^{-t/\theta}} \int_0^t t' e^{-t'/\theta} \frac{dt'}{\theta} \\ &= \frac{\theta}{1 - e^{-t/\theta}} \left[ 1 - e^{-t/\theta} - (t/\theta) e^{-t/\theta} \right] \\ &= \theta - \frac{te^{-t/\theta}}{1 - e^{-t/\theta}} \end{aligned}$$



average log-likelihood

$$\begin{aligned} Q &= E \left[ -n \ln \theta - \frac{n \langle u \rangle}{\theta} + \sum_{i=1}^m \left( -\ln \theta - \frac{v_i}{\theta} \right) \right] \\ &= -(n+m) \ln \theta - \frac{n \langle u \rangle}{\theta} - \frac{r}{\theta} \left( \theta - \frac{t \exp(-t/\theta)}{1 - \exp(-t/\theta)} \right) - \frac{(m-r)}{\theta} (\theta + t) \end{aligned}$$

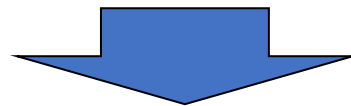
*this ends the expectation step*

the max of the mean likelihood

$$Q = -(n+m)\ln\theta - \frac{1}{\theta} \left[ n\langle u \rangle + r \left( \theta - \frac{t \exp(-t/\theta)}{1 - \exp(-t/\theta)} \right) + (m-r)(\theta + t) \right]$$

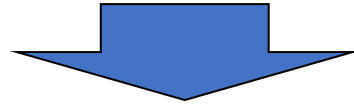
can be found by maximizing the approximate expression

$$Q \approx -(n+m)\ln\theta - \frac{1}{\theta} \left[ n\langle u \rangle + r \left( \theta^{(k)} - \frac{t \exp(-t/\theta^{(k)})}{1 - \exp(-t/\theta^{(k)})} \right) + (m-r)(\theta^{(k)} + t) \right]$$



$$\frac{dQ}{d\theta} \approx -(n+m)\frac{1}{\theta} + \frac{1}{\theta^2} \left[ n\langle u \rangle + r \left( \theta^{(k)} - \frac{t \exp(-t/\theta^{(k)})}{1 - \exp(-t/\theta^{(k)})} \right) + (m-r)(\theta^{(k)} + t) \right] = 0$$

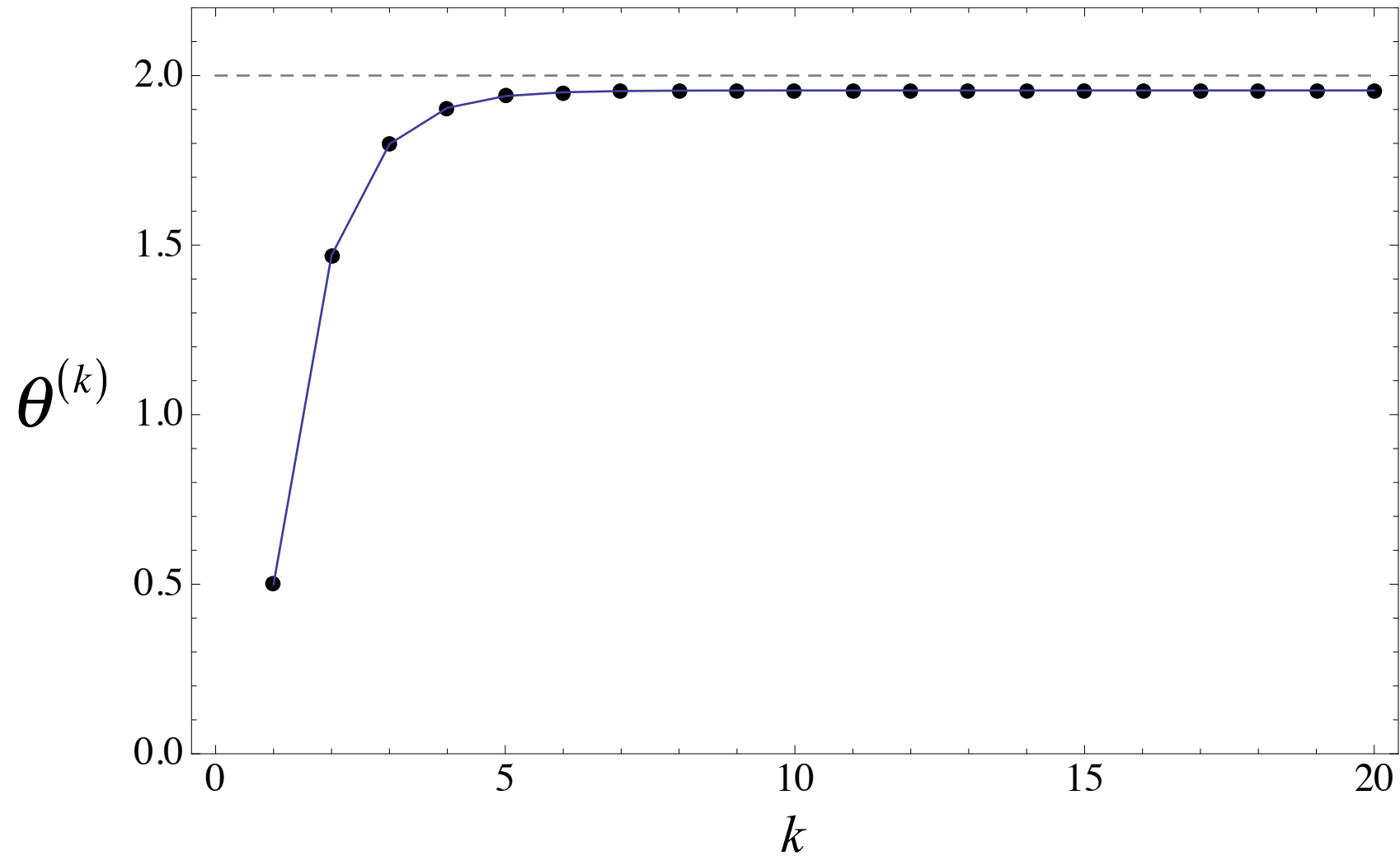
$$\frac{dQ}{d\theta} \approx -(n+m)\frac{1}{\theta} + \frac{1}{\theta^2} \left[ n\langle u \rangle + r \left( \theta^{(k)} - \frac{t \exp(-t/\theta^{(k)})}{1 - \exp(-t/\theta^{(k)})} \right) + (m-r)(\theta^{(k)} + t) \right] = 0$$



$$\theta^{(k+1)} = \frac{1}{n+m} \left[ n\langle u \rangle + r \left( \theta^{(k)} - \frac{t \exp(-t/\theta^{(k)})}{1 - \exp(-t/\theta^{(k)})} \right) + (m-r)(\theta^{(k)} + t) \right]$$

this formula summarizes expectation and maximization:  
therefore, the recipe is to iterate this until convergence ...

Example with mean failure time = 2 (a.u.), and randomly generated data ( $n = 100$ ;  $m = 100$ ).  
In this example  $r = 36$ .

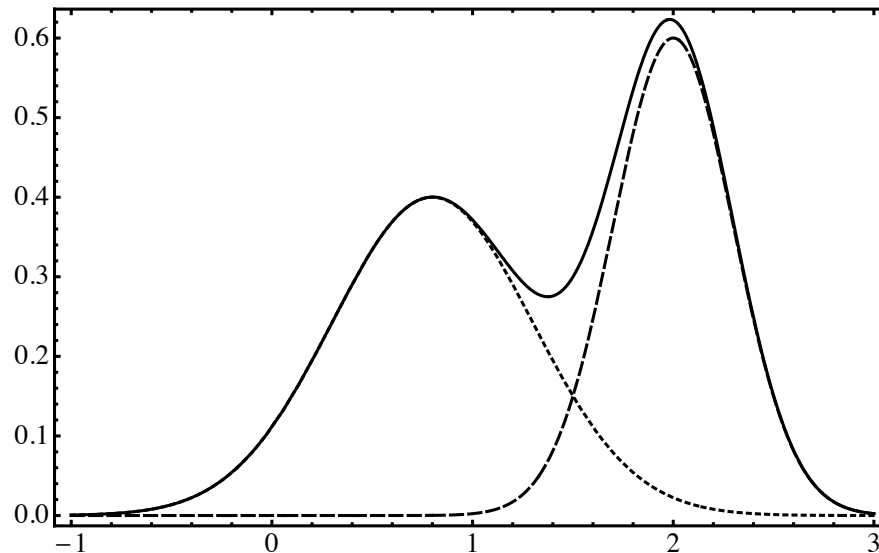


# The EM method is often used to estimate the parameters of “mixture models”.

$$p(x_n | \boldsymbol{\theta}) = \sum_{i=1}^M \alpha_i p_i(x_n | \boldsymbol{\theta}_i)$$

"hidden parameters"  
(also "latent parameters")

$$\boldsymbol{\theta} = (\underbrace{\alpha_1, \dots, \alpha_M}_{\text{"hidden parameters"}}; \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M)$$
$$\sum_{i=1}^M \alpha_i = 1$$



Example: a Gaussian mixture model (M=2)

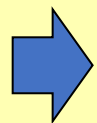


direct maximization of log likelihood

$$\begin{aligned}\log \mathcal{L}(\mathbf{x}, \boldsymbol{\theta}) &= \log \prod_n p(x_n | \boldsymbol{\theta}) = \sum_n \log p(x_n | \boldsymbol{\theta}) \\ &= \sum_n \log \left[ \sum_{i=1}^M \alpha_i p_i(x_n | \boldsymbol{\theta}_i) \right]\end{aligned}$$

difficult numerical treatment ... however we can manage with a reinterpretation of the mixture model parameters ...

$\alpha_k$  = probability of drawing the  $k$ -th component of the mixture model



new ("hidden" or "latent") variable:  $y$  = index of component (integer values only)

therefore, we **must** redefine data and parameters.

new likelihood which includes the hidden variables

$$\begin{aligned}\log \mathcal{L}'(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) &= \log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}) \\ &= \log \prod_n p(x_n, y_n | \boldsymbol{\theta}) \\ &= \sum_n \log \left[ p(x_n | y_n, \boldsymbol{\theta}) p(y_n | \boldsymbol{\theta}) \right] \\ &= \sum_n \log \left[ \alpha_{y_n} p_{y_n} \left( x_n | \boldsymbol{\theta}_{y_n} \right) \right]\end{aligned}$$

( $\boldsymbol{\theta}_i$  is the parameter vector restricted to the  $i$ -th component)

The structure is simpler now, there is no sum in the argument of the logarithm, however there are new hidden variables  $y$ .

Now we proceed by averaging the likelihood  
(Expectation step)

new parameter  
estimate

previous parameter  
estimate

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) &= E_{\mathbf{y}} \left[ \log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}) \mid \mathbf{x}, \boldsymbol{\theta}^{(i-1)} \right] \\ &= \int_Y \left[ \log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}) \right] p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}^{(i-1)}) d\mathbf{y} \\ &\rightarrow \sum_{\mathbf{y}} \left[ \log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta}) \right] p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}^{(i-1)}) \end{aligned}$$

sum instead of integral, because the  $y$   
variables are discrete

## prior probabilities in the expression of the averaged log-likelihood

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = \sum_{\mathbf{y}} [\log p(\mathbf{x}, \mathbf{y} | \boldsymbol{\theta})] p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}^{(i-1)})$$

and now we use Bayes' theorem:

$$p(y_n | x_n, \boldsymbol{\theta}) = \frac{p(x_n | y_n, \boldsymbol{\theta}) p(y_n | \boldsymbol{\theta})}{p(x_n | \boldsymbol{\theta})} = \frac{\alpha_{y_n} p_{y_n}(x_n | \boldsymbol{\theta}_{y_n})}{\sum_{k=1}^M \alpha_k p_k(x_n | \boldsymbol{\theta}_k)}$$

$$p(\mathbf{y} | \mathbf{x}, \boldsymbol{\theta}) = \prod_{n=1}^N p(y_n | x_n, \boldsymbol{\theta}) = \prod_{n=1}^N \frac{\alpha_{y_n} p_{y_n}(x_n | \boldsymbol{\theta}_{y_n})}{\sum_{k=1}^M \alpha_k p_k(x_n | \boldsymbol{\theta}_k)}$$

Therefore, using  $\log \mathcal{L}'(\mathbf{x}, \mathbf{y}, \boldsymbol{\theta}) = \sum_n \log \left[ \alpha_{y_n} p_{y_n} \left( x_n \mid \boldsymbol{\theta}_{y_n} \right) \right]$

and  $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}) = \prod_{n=1}^N p(y_n \mid x_n, \boldsymbol{\theta})$

we find

$$\begin{aligned} Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) &= \sum_{\mathbf{y}} \left[ \log p(\mathbf{x}, \mathbf{y} \mid \boldsymbol{\theta}) \right] p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta}^{(i-1)}) \\ &= \sum_{\mathbf{y}} \sum_{k=1}^N \log \left[ \alpha_{y_k} p_{y_k} \left( x_k \mid \boldsymbol{\theta}_{y_k} \right) \right] \prod_{j=1}^N p(y_j \mid x_j, \boldsymbol{\theta}^{(i-1)}) \\ &= \sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \sum_{k=1}^N \log \left[ \alpha_{y_k} p_{y_k} \left( x_k \mid \boldsymbol{\theta}_{y_k} \right) \right] \prod_{j=1}^N p(y_j \mid x_j, \boldsymbol{\theta}^{(i-1)}) \end{aligned}$$

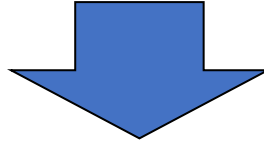
$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) &= \sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \sum_{k=1}^N \log \left[ \alpha_{y_k} p_{y_k} (x_k | \boldsymbol{\theta}_{y_k}) \right] \prod_{j=1}^N p(y_j | x_j, \boldsymbol{\theta}^{(i-1)}) \\
&= \sum_{y_1=1}^M \sum_{y_2=1}^M \dots \sum_{y_N=1}^M \sum_{k=1}^N \sum_{\ell=1}^M \delta_{\ell, y_k} \log \left[ \alpha_{\ell} p_{\ell} (x_k | \boldsymbol{\theta}_{\ell}) \right] \prod_{j=1}^N p(y_j | x_j, \boldsymbol{\theta}^{(i-1)})
\end{aligned}$$

to decouple the variables, we add one sum and one Kronecker's delta...

after the decoupling, we can use the normalization of conditional probabilities

$$\sum_{y_j=1}^M p(y_j | x_j, \boldsymbol{\theta}^{(i-1)}) = 1$$

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) = \sum_{y_1=1}^M \sum_{y_2=1}^M \cdots \sum_{y_N=1}^M \sum_{k=1}^N \sum_{\ell=1}^M \delta_{\ell, y_k} \log [\alpha_{\ell} p_{\ell}(x_k | \boldsymbol{\theta}_{\ell})] \prod_{j=1}^N p(y_j | x_j, \boldsymbol{\theta}^{(i-1)})$$



$$\begin{aligned}
 Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) &= \sum_{\ell=1}^M \sum_{k=1}^N \ln [\alpha_{\ell} p_{\ell}(x_k | \boldsymbol{\theta}_{\ell})] \sum_{y_1=1}^M \sum_{y_2=1}^M \cdots \sum_{y_N=1}^M \delta_{\ell, y_k} \prod_{j=1}^N p(y_j | x_j, \boldsymbol{\theta}^{(i-1)}) \\
 &= \sum_{\ell=1}^M \sum_{k=1}^N \ln [\alpha_{\ell} p_{\ell}(x_k | \boldsymbol{\theta}_{\ell})] \left\{ \sum_{y_1=1}^M \cdots \sum_{y_{k-1}=1}^M \sum_{y_{k+1}=1}^M \cdots \sum_{y_N=1}^M \prod_{\substack{j=1 \\ j \neq k}}^N p(y_j | x_j, \boldsymbol{\theta}^{(i-1)}) \right\} p(\ell | x_k, \boldsymbol{\theta}^{(i-1)}) \\
 &= \sum_{\ell=1}^M \sum_{k=1}^N \ln [\alpha_{\ell} p_{\ell}(x_k | \boldsymbol{\theta}_{\ell})] \left\{ \prod_{\substack{j=1 \\ j \neq k}}^N \sum_{y_j=1}^M p(y_j | x_j, \boldsymbol{\theta}^{(i-1)}) \right\} p(\ell | x_k, \boldsymbol{\theta}^{(i-1)}) \\
 &= \sum_{\ell=1}^M \sum_{k=1}^N \ln [\alpha_{\ell} p_{\ell}(x_k | \boldsymbol{\theta}_{\ell})] p(\ell | x_k, \boldsymbol{\theta}^{(i-1)})
 \end{aligned}$$

these sums all add to 1 (normalization of conditional probabilities)

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(i-1)}) &= \sum_{\ell=1}^M \sum_{k=1}^N \ln [\alpha_{\ell} p(\ell | x_k, \boldsymbol{\theta})] p_{\ell}(x_k, \boldsymbol{\theta}^{(i-1)}) \\
&= \sum_{\ell=1}^M \sum_{k=1}^N \ln \alpha_{\ell} p_{\ell}(x_k, \boldsymbol{\theta}^{(i-1)}) + \sum_{\ell=1}^M \sum_{k=1}^N \ln p(\ell | x_k, \boldsymbol{\theta}) p_{\ell}(x_k, \boldsymbol{\theta}^{(i-1)})
\end{aligned}$$

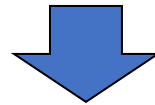
this depends only on the  $\alpha$  parameters

this term depends on the parameters of the component distributions

Thus, there are two terms that can be maximized separately.

Moreover, the first term must be maximized with the normalization constraint, i.e.

$$\frac{\partial}{\partial \alpha_m} \left[ \sum_{\ell=1}^M \sum_{k=1}^N \log \alpha_{\ell} p(\ell | x_k, \boldsymbol{\theta}^{(i-1)}) + \lambda \left( \sum_{\ell=1}^M \alpha_{\ell} - 1 \right) \right] = 0$$



$$\sum_{k=1}^N \frac{1}{\alpha_m} p(m | x_k, \boldsymbol{\theta}^{(i-1)}) + \lambda = 0$$



$$\sum_{k=1}^N \frac{1}{\alpha_m} p(m|x_k, \boldsymbol{\theta}^{(i-1)}) + \lambda = 0$$



$$\sum_{k=1}^N p(m|x_k, \boldsymbol{\theta}^{(i-1)}) = -\lambda \alpha_m$$



$$\sum_{m=1}^M \sum_{k=1}^N p(m|x_k, \boldsymbol{\theta}^{(i-1)}) = -\lambda \sum_{m=1}^M \alpha_m \quad \rightarrow \quad \lambda = -N \quad \rightarrow \quad \alpha_m = \frac{1}{N} \sum_{k=1}^N p(m|x_k, \boldsymbol{\theta}^{(i-1)})$$

This is as far as we can go without introducing an explicit form for the component distributions: to evaluate the other term we explicitly consider the 1D Gaussian mixture model:

$$p_\ell(x|\mu_\ell, \sigma_\ell) = \frac{1}{\sqrt{2\pi\sigma_\ell^2}} \exp\left(-\frac{(x - \mu_\ell)^2}{2\sigma_\ell^2}\right)$$



$$\sum_{\ell=1}^M \sum_{k=1}^N \ln p_\ell(x_k, \boldsymbol{\theta}) p(\ell|x_k, \boldsymbol{\theta}^{(i-1)}) = \sum_{\ell=1}^M \sum_{k=1}^N \left[ -\frac{1}{2} \ln(2\pi\sigma_\ell^2) - \frac{(x_k - \mu_\ell)^2}{2\sigma_\ell^2} \right] p(\ell|x_k, \mu_\ell^{(i-1)}, \sigma_\ell^{(i-1)})$$



$$\frac{\partial}{\partial \mu_m} \sum_{\ell=1}^M \sum_{k=1}^N \ln p_\ell(x_k, \boldsymbol{\theta}) p(\ell|x_k, \boldsymbol{\theta}^{(i-1)}) = -2 \sum_{k=1}^N \frac{(x_k - \mu_m)}{2\sigma_m^2} p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)}) = 0$$

$$\frac{\partial}{\partial \mu_m} \sum_{\ell=1}^M \sum_{k=1}^N \ln p_{\ell}(x_k, \boldsymbol{\theta}) p(\ell|x_k, \boldsymbol{\theta}^{(i-1)}) = -2 \sum_{k=1}^N \frac{(x_k - \mu_m)}{2\sigma_m^2} p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)}) = 0$$

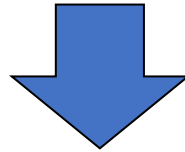


$$\mu_m = \frac{\sum_{k=1}^N x_k p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}{\sum_{k=1}^N p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}$$

moreover, if we let  $c_m = 1/\sigma_m^2$

$$\begin{aligned} \frac{\partial}{\partial c_m} \sum_{\ell=1}^M \sum_{k=1}^N \ln p_{\ell}(x_k, \boldsymbol{\theta}) p(\ell|x_k, \boldsymbol{\theta}^{(i-1)}) &= \frac{\partial}{\partial c_m} \sum_{\ell=1}^M \sum_{k=1}^N \left[ -\frac{1}{2} \ln(2\pi\sigma_{\ell}^2) - \frac{(x_k - \mu_{\ell})^2}{2\sigma_{\ell}^2} \right] p(\ell|x_k, \mu_{\ell}^{(i-1)}, \sigma_{\ell}^{(i-1)}) \\ &= \sum_{k=1}^N \left[ \frac{1}{2c_m} - \frac{1}{2}(x_k - \mu_m)^2 \right] p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)}) \\ &= \sum_{k=1}^N \left[ \frac{\sigma_m^2}{2} - \frac{1}{2}(x_k - \mu_m)^2 \right] p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)}) = 0 \end{aligned}$$

$$\frac{\partial}{\partial c_m} \sum_{\ell=1}^M \sum_{k=1}^N \ln p_{\ell}(x_k, \boldsymbol{\theta}) p(\ell|x_k, \boldsymbol{\theta}^{(i-1)}) = \sum_{k=1}^N \left[ \frac{\sigma_m^2}{2} - \frac{1}{2} (x_k - \mu_m)^2 \right] p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)}) = 0$$



$$\sigma_m^2 = \frac{\sum_{k=1}^N (x_k - \mu_m)^2 p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}{\sum_{k=1}^N p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}$$

Finally, we find the following set of recursive formulas, that combine the E and M steps:

$$p_m(x|\mu_m, \sigma_m) = \frac{1}{\sqrt{2\pi\sigma_m^2}} \exp\left(-\frac{(x - \mu_m)^2}{2\sigma_m^2}\right)$$

$$p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)}) = \frac{\alpha_m^{(i-1)} p_m(x_k|\mu_m^{(i-1)}, \sigma_m^{(i-1)})}{\sum_{k=1}^M \alpha_m^{(i-1)} p_m(x_k|\mu_m^{(i-1)}, \sigma_m^{(i-1)})}$$

$$\alpha_m^{(i)} = \frac{1}{N} \sum_{k=1}^N p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})$$

$$\mu_m^{(i)} = \frac{\sum_{k=1}^N x_k p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}{\sum_{k=1}^N p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})} \quad (\sigma_m^{(i)})^2 = \frac{\sum_{k=1}^N (x_k - \mu_m^{(i)})^2 p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}{\sum_{k=1}^N p(m|x_k, \mu_m^{(i-1)}, \sigma_m^{(i-1)})}$$

We remark that the probabilities

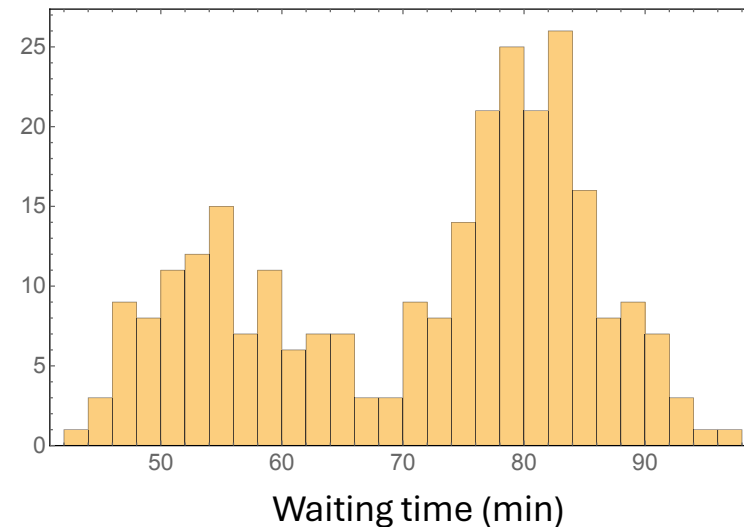
$$p(y_n | x_n, \boldsymbol{\theta}) = \frac{\alpha_{y_n} p_{y_n}(x_n | \boldsymbol{\theta}_{y_n})}{\sum_{k=1}^M \alpha_k p_k(x_n | \boldsymbol{\theta}_k)}$$

are an estimate of the frequencies of the  $y_n$  using the observed data  $x_n$ , and this amounts to a classification result (selection of one of the component distributions).

# Straightforward example: waiting times between eruptions of the Old Faithful Geiser (Yellowstone National Park – Wyoming)



Here we analyze the waiting times assuming a 2-Gaussian mixture model for the waiting time distribution (data taken from an R example)



In this case, the mixture model has two Gaussian components

$$p(w|\boldsymbol{\theta}) = \alpha N(w; \mu_1, \sigma_1) + (1 - \alpha)N(w; \mu_2, \sigma_2)$$

where the vector of parameters is

$$\boldsymbol{\theta} = (\alpha, \mu_1, \mu_2, \sigma_1, \sigma_2)$$

The resulting log likelihood with  $n$  waiting times  $w_i$  is

$$\ln \mathcal{L} = \sum_i \ln [\alpha N(w_i; \mu_1, \sigma_1) + (1 - \alpha)N(w_i; \mu_2, \sigma_2)]$$



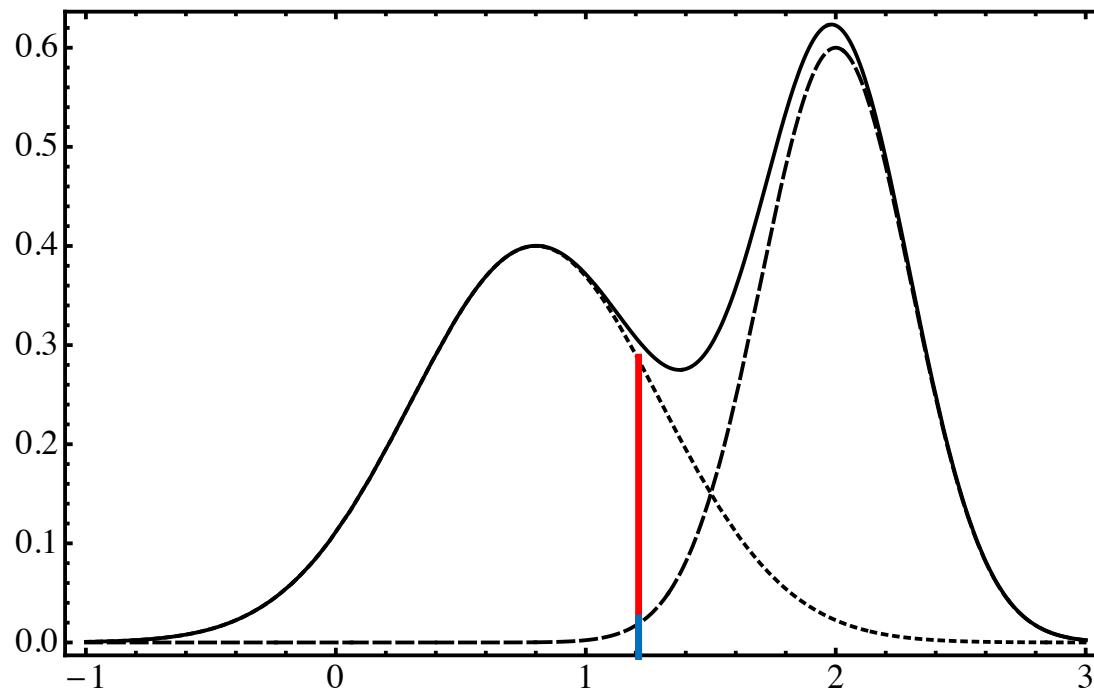
Again, we substitute the likelihood with the new one

$$\mathcal{L} = \prod_i \alpha^{y_i} (1 - \alpha)^{1-y_i} [N(w_i; \mu_1, \sigma_1)]^{y_i} [N(w_i; \mu_2, \sigma_2)]^{1-y_i}$$

where the new, unobserved data  $y_i$  are indicator variables that select extraction from the first ( $y_i = 1$ ) or the second ( $y_i = 0$ ) Gaussian.

Then

$$\begin{aligned} \ln \mathcal{L} = \sum_i & \left[ y_i \ln \alpha + (1 - y_i) \ln(1 - \alpha) + y_i \left( -\frac{1}{2} \ln(2\pi\sigma_1) - \frac{(w_i - \mu_1)^2}{2\sigma_1^2} \right) \right. \\ & \left. + (1 - y_i) \left( -\frac{1}{2} \ln(2\pi\sigma_2) - \frac{(w_i - \mu_2)^2}{2\sigma_2^2} \right) \right] \end{aligned}$$



The probability that a given time interval belongs to the first Gaussian is

this probability is also equal to the mean value of the indicator variable

$$\begin{aligned}
 p_i &= \frac{\alpha \times N(w_i; \mu_1, \sigma_1)}{\alpha \times N(w_i; \mu_1, \sigma_1) + (1 - \alpha) \times N(w_i; \mu_2, \sigma_2)} \\
 &= \frac{\alpha^{(k)} \exp[-(w_i - \mu_1^{(k)})^2 / 2(\sigma_1^{(k)})^2] / \sqrt{2\pi(\sigma_1^{(k)})^2}}{\alpha^{(k)} \exp[-(w_i - \mu_1^{(k)})^2 / 2(\sigma_1^{(k)})^2] / \sqrt{2\pi(\sigma_1^{(k)})^2} + (1 - \alpha^{(k)}) \exp[-(w_i - \mu_2^{(k)})^2 / 2(\sigma_2^{(k)})^2] / \sqrt{2\pi(\sigma_2^{(k)})^2}}
 \end{aligned}$$

Now, averaging the log likelihood with respect to the missing data we find

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}^{(k)}) = \sum_i \left[ p_i^{(k)} \ln \alpha + (1 - p_i^{(k)}) \ln(1 - \alpha) + p_i^{(k)} \left( -\frac{1}{2} \ln(2\pi\sigma_1^2) - \frac{(w_i - \mu_1)^2}{2\sigma_1^2} \right) \right. \\ \left. + (1 - p_i^{(k)}) \left( -\frac{1}{2} \ln(2\pi\sigma_2^2) - \frac{(w_i - \mu_2)^2}{2\sigma_2^2} \right) \right]$$

(the mean value of the indicator variable is equal to the current estimate probability  $\alpha$ )

Next, we maximize with respect to all the remaining parameters, and we find:

$$\alpha^{(k+1)} = \frac{1}{N} \sum_i p_i^{(k)}$$

$$\left( \sigma_1^{(k+1)} \right)^2 = \frac{\sum_i p_i^{(k)} (w_i - \mu_1^{(k)})^2}{\sum_i p_i^{(k)}}; \quad \mu_1^{(k+1)} = \frac{\sum_i p_i^{(k)} w_i}{\sum_i p_i^{(k)}}$$

$$\left( \sigma_2^{(k+1)} \right)^2 = \frac{\sum_i (1 - p_i^{(k)}) (w_i - \mu_2^{(k)})^2}{\sum_i (1 - p_i^{(k)})}; \quad \mu_2^{(k+1)} = \frac{\sum_i (1 - p_i^{(k)}) w_i}{\sum_i (1 - p_i^{(k)})}$$

Finally, we find the following set of equations:

$$p_i^{(k)} = \frac{\alpha^{(k)} \exp[-(w_i - \mu_1^{(k)})^2 / 2(\sigma_1^{(k)})^2] / \sqrt{2\pi(\sigma_1^{(k)})^2}}{\alpha^{(k)} \exp[-(w_i - \mu_1^{(k)})^2 / 2(\sigma_1^{(k)})^2] / \sqrt{2\pi(\sigma_1^{(k)})^2} + (1 - \alpha^{(k)}) \exp[-(w_i - \mu_2^{(k)})^2 / 2(\sigma_2^{(k)})^2] / \sqrt{2\pi(\sigma_2^{(k)})^2}}$$

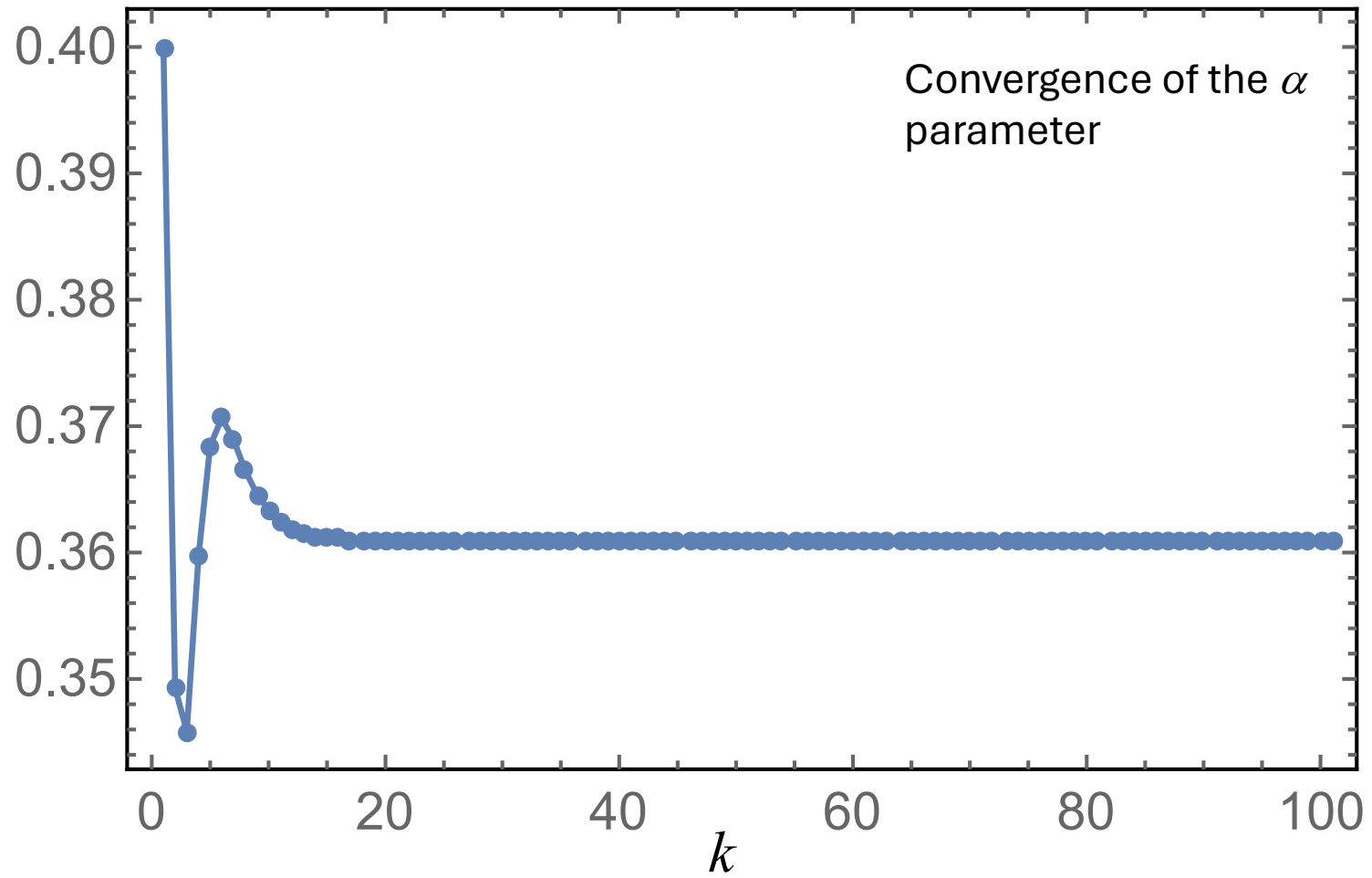
$$\alpha^{(k+1)} = \frac{1}{N} \sum_i p_i^{(k)}$$

$$\left(\sigma_1^{(k+1)}\right)^2 = \frac{\sum_i p_i^{(k)} (w_i - \mu_1^{(k)})^2}{\sum_i p_i^{(k)}};$$

$$\mu_1^{(k+1)} = \frac{\sum_i p_i^{(k)} w_i}{\sum_i p_i^{(k)}}$$

$$\left(\sigma_2^{(k+1)}\right)^2 = \frac{\sum_i (1 - p_i^{(k)}) (w_i - \mu_2^{(k)})^2}{\sum_i (1 - p_i^{(k)})};$$

$$\mu_2^{(k+1)} = \frac{\sum_i (1 - p_i^{(k)}) w_i}{\sum_i (1 - p_i^{(k)})}$$



# Comparison of the original data with the mixture model obtained with the EM algorithm

