

# Brief Contents

1	Introduction	1
<b>I</b>	<b>Foundations</b>	<b>19</b>
2	Probabilistic inference	21
3	Probabilistic models	41
4	Parameter estimation	73
5	Optimization algorithms	95
6	Information theory	141
7	Bayesian statistics	159
8	Bayesian decision theory	215
<b>II</b>	<b>Linear models</b>	<b>239</b>
9	Linear discriminant analysis	241
10	Logistic regression	257
11	Linear regression	293
12	Generalized linear models	341
<b>III</b>	<b>Deep neural networks</b>	<b>357</b>
13	Neural networks for unstructured data	359
14	Neural networks for images	395
15	Neural networks for sequences	431
<b>IV</b>	<b>Nonparametric models</b>	<b>457</b>
16	Exemplar-based methods ( <b>Unfinished</b> )	459
17	Kernel methods	475
18	Trees, forests, bagging and boosting	517

<b>V</b>	<b>Beyond supervised learning</b>	<b>537</b>
19	Learning with fewer labeled examples	539
20	Dimensionality reduction	571
21	Clustering	619
22	Recommender systems ( <b>Unfinished</b> )	641
23	Graph embeddings	645
<b>VI</b>	<b>Appendix: Mathematical background</b>	<b>667</b>
A	Some useful mathematics	669
B	Linear algebra	689
C	Probability	729
D	Frequentist statistics	749
E	Exercises	785

# Contents

<b>Preface</b>	<b>xv</b>
<b>1 Introduction</b>	<b>1</b>
1.1 What is machine learning?	1
1.2 Supervised learning	1
1.2.1 Classification	2
1.2.2 Regression	7
1.2.3 Overfitting and generalization	10
1.3 Unsupervised learning	11
1.3.1 Clustering	12
1.3.2 Self-supervised learning	12
1.3.3 Evaluating unsupervised learning	13
1.4 Reinforcement learning	13
1.4.1 Challenges in RL	14
1.4.2 Comparing supervised, unsupervised and reinforcement learning	15
1.5 Discussion	15
1.5.1 The relationship between ML and other fields	16
1.5.2 Structure of the book	16
1.5.3 Caveats	16
<b>I Foundations</b>	<b>19</b>
<b>2 Probabilistic inference</b>	<b>21</b>
2.1 Introduction	21
2.2 Bayes' rule	21
2.2.1 Example: testing for COVID-19	22
2.2.2 Example: The Monty Hall problem	23
2.2.3 Inverse problems	24
2.3 Bayesian concept learning	25
2.3.1 Learning a discrete concept: the number game	26
2.3.2 Learning a continuous concept: the healthy levels game	32

2.4	Bayesian machine learning	35	
2.4.1	Example: scalar input, binary output	37	
2.4.2	Example: binary input, scalar output	38	
2.4.3	Scaling up	39	
<b>3</b>	<b>Probabilistic models</b>	<b>41</b>	
3.1	Bernoulli and binomial distributions	41	
3.1.1	Definition	41	
3.1.2	Sigmoid (logistic) function	42	
3.1.3	Binary logistic regression	44	
3.2	Categorical and multinomial distributions	45	
3.2.1	Definition	45	
3.2.2	Softmax function	45	
3.2.3	Multiclass logistic regression	46	
3.2.4	Log-sum-exp trick	47	
3.3	Univariate Gaussian (normal) distribution	48	
3.3.1	Cumulative distribution function	48	
3.3.2	Probability density function	49	
3.3.3	Regression	50	
3.3.4	Why is the Gaussian distribution so widely used?	51	
3.3.5	Half-normal	52	
3.4	Some other common univariate distributions	52	
3.4.1	Student $t$ distribution	52	
3.4.2	Cauchy distribution	54	
3.4.3	Laplace distribution	54	
3.4.4	Beta distribution	55	
3.4.5	Gamma distribution	55	
3.5	The multivariate Gaussian (normal) distribution	57	
3.5.1	Definition	57	
3.5.2	Mahalanobis distance	58	
3.5.3	Marginals and conditionals of an MVN	60	
3.5.4	Example: Imputing missing values	60	
3.6	Linear Gaussian systems	61	
3.6.1	Example: inferring a latent vector from a noisy sensor	62	
3.6.2	Example: inferring a latent vector from multiple noisy sensors	63	
3.7	Mixture models	64	
3.7.1	Gaussian mixture models	65	
3.7.2	Mixtures of Bernoullis	66	
3.7.3	Gaussian scale mixtures	67	
3.8	Probabilistic graphical models	68	
3.8.1	Inference	70	
3.8.2	Plate notation	70	
<b>4</b>	<b>Parameter estimation</b>	<b>73</b>	
4.1	Introduction	73	

4.2	Maximum likelihood estimation (MLE)	73	
4.2.1	Definition	73	
4.2.2	Justification for MLE	74	
4.2.3	Example: MLE for the Bernoulli distribution	75	
4.2.4	Example: MLE for the categorical distribution	76	
4.2.5	Example: MLE for the univariate Gaussian	77	
4.2.6	Example: MLE for the multivariate Gaussian	78	
4.2.7	Example: MLE for linear regression	80	
4.3	Empirical risk minimization (ERM)	81	
4.3.1	Example: minimizing the misclassification rate	81	
4.3.2	Surrogate loss	81	
4.4	Regularization	82	
4.4.1	Example: MAP estimation for the Bernoulli distribution	83	
4.4.2	Example: MAP estimation for the multivariate Gaussian	84	
4.4.3	Example: weight decay	85	
4.4.4	Picking the regularizer using a validation set	87	
4.4.5	Cross-validation	87	
4.4.6	Early stopping	89	
4.4.7	Using more data	89	
4.5	The method of moments	91	
4.5.1	Example: MOM for the univariate Gaussian	91	
4.5.2	Example: MOM for the uniform distribution	91	
4.6	Online (recursive) estimation	92	
4.6.1	Example: recursive MLE for the mean of a Gaussian	92	
4.6.2	Exponentially-weighted moving average (EMA)	93	
4.6.3	Bayesian inference	94	
4.7	Parameter uncertainty	94	
<b>5</b>	<b>Optimization algorithms</b>	<b>95</b>	
5.1	Introduction	95	
5.1.1	Local vs global optimization	95	
5.1.2	Constrained vs unconstrained optimization	97	
5.1.3	Convex vs nonconvex optimization	97	
5.1.4	Smooth vs nonsmooth optimization	98	
5.2	First-order methods	98	
5.2.1	Descent direction	99	
5.2.2	Step size (learning rate)	99	
5.2.3	Convergence rates	101	
5.2.4	Momentum methods	102	
5.3	Second-order methods	104	
5.3.1	Newton's method	104	
5.3.2	BFGS and other quasi-Newton methods	105	
5.3.3	Trust region methods	106	
5.3.4	Natural gradient descent	107	
5.4	Stochastic gradient descent	110	

5.4.1	Application to finite sum problems	111
5.4.2	Example: SGD for fitting linear regression	111
5.4.3	Choosing the step size	112
5.4.4	Iterate averaging	113
5.4.5	Variance reduction	113
5.4.6	Preconditioned SGD	114
5.5	Constrained optimization	117
5.5.1	Lagrange multipliers	118
5.5.2	The KKT conditions	119
5.5.3	Linear programming	120
5.5.4	Quadratic programming	122
5.5.5	Mixed integer linear programming	123
5.6	Proximal gradient method	123
5.6.1	Projected gradient descent	124
5.6.2	Proximal operator for $\ell_1$ -norm regularizer	125
5.6.3	Proximal operator for quantization	126
5.7	Bound optimization	127
5.7.1	The general algorithm	127
5.7.2	The EM algorithm	128
5.7.3	Example: EM for a GMM	131
5.7.4	Example: EM for an MVN with missing data	135
5.8	Blackbox and derivative free optimization	138
5.8.1	Grid search and random search	138
5.8.2	Simulated annealing	138
5.8.3	Model-based blackbox optimization	139
<b>6</b>	<b>Information theory</b>	<b>141</b>
6.1	Entropy	141
6.1.1	Entropy for discrete random variables	141
6.1.2	Cross entropy	143
6.1.3	Joint entropy	143
6.1.4	Conditional entropy	144
6.1.5	Perplexity	145
6.1.6	Differential entropy for continuous random variables	145
6.2	Relative entropy (KL divergence)	146
6.2.1	Definition	147
6.2.2	Interpretation	147
6.2.3	Example: KL divergence between two Gaussians	147
6.2.4	Non-negativity of KL	148
6.2.5	KL divergence and MLE	148
6.2.6	Forward vs reverse KL	149
6.3	Mutual information	150
6.3.1	Definition	150
6.3.2	Interpretation	151
6.3.3	Example	151

6.3.4	Conditional mutual information	152	
6.3.5	Normalized mutual information	153	
6.3.6	MI as a “generalized correlation coefficient”	153	
6.3.7	Data processing inequality	155	
6.3.8	Sufficient Statistics	156	
6.3.9	Fano’s inequality	156	
<b>7</b>	<b>Bayesian statistics</b>	<b>159</b>	
7.1	Introduction	159	
7.1.1	Computing the posterior	159	
7.1.2	Summarizing the posterior	159	
7.2	Conjugate priors	163	
7.2.1	The beta-binomial model	164	
7.2.2	The Dirichlet-multinomial model	170	
7.2.3	The Gaussian-Gaussian model	174	
7.2.4	The multivariate Gaussian-Gaussian model	179	
7.2.5	Beyond conjugate priors	185	
7.3	Noninformative priors	185	
7.3.1	Jeffreys priors	186	
7.3.2	Invariant priors	188	
7.3.3	Reference priors	189	
7.4	Hierarchical priors	189	
7.4.1	A hierarchical binomial model	190	
7.4.2	A hierarchical Gaussian model	191	
7.5	Empirical priors	194	
7.5.1	A hierarchical binomial model	195	
7.5.2	A hierarchical Gaussian model	196	
7.6	Bayesian model selection	197	
7.6.1	Example: polynomial regression	198	
7.6.2	Bayesian Occam’s razor	198	
7.6.3	Connection between cross validation and marginal likelihood	199	
7.6.4	Bayes model averaging	201	
7.6.5	The minimum description length (MDL) principle	201	
7.6.6	Bayesian hypothesis testing	202	
7.6.7	Group comparisons	204	
7.6.8	Posterior predictive checks	206	
7.7	Approximate inference algorithms	208	
7.7.1	Grid approximation	208	
7.7.2	Laplace approximation	209	
7.7.3	Variational approximation	210	
7.7.4	Markov Chain Monte Carlo (MCMC) approximation	211	
7.7.5	Online inference using assumed density filtering	213	
<b>8</b>	<b>Bayesian decision theory</b>	<b>215</b>	
8.1	Bayesian decision theory	215	

8.1.1	Basics	215	
8.1.2	Classification problems	216	
8.1.3	ROC curves	218	
8.1.4	Precision-recall curves	220	
8.1.5	Regression problems	222	
8.1.6	Probabilistic prediction problems	223	
8.2	A/B testing	225	
8.2.1	A Bayesian approach	225	
8.2.2	Example	228	
8.3	Bandit problems	229	
8.3.1	Contextual bandits	230	
8.3.2	Markov decision processes	231	
8.3.3	Exploration-exploitation tradeoff	232	
8.3.4	Optimal solution	232	
8.3.5	Regret	234	
8.3.6	Upper confidence bounds (UCB)	235	
8.3.7	Thompson sampling	236	
8.3.8	Simple heuristics	237	
8.4	Discussion	238	
8.4.1	The separation principle and its limits	238	
8.4.2	Optimality of the Bayesian approach and its limits	238	

## II Linear models 239

### 9 Linear discriminant analysis 241

9.1	Introduction	241	
9.2	Gaussian discriminant analysis	241	
9.2.1	Quadratic decision boundaries	242	
9.2.2	Linear decision boundaries	243	
9.2.3	The connection between LDA and logistic regression	243	
9.2.4	Model fitting	244	
9.2.5	Nearest centroid classifier	246	
9.2.6	Fisher's linear discriminant analysis	246	
9.3	Naive Bayes classifiers	250	
9.3.1	Example models	251	
9.3.2	Model fitting	252	
9.3.3	Bayesian naive Bayes	253	
9.3.4	The connection between naive Bayes and logistic regression	253	
9.4	Generative vs discriminative classifiers	254	
9.4.1	Advantages of discriminative classifiers	254	
9.4.2	Advantages of generative classifiers	255	
9.4.3	Handling missing features	255	

### 10 Logistic regression 257



10.1	Introduction	257	
10.2	Binary logistic regression	257	
10.2.1	Linear classifiers	257	
10.2.2	Nonlinear classifiers	258	
10.2.3	Maximum likelihood estimation	260	
10.2.4	Stochastic gradient descent	263	
10.2.5	Perceptron algorithm	264	
10.2.6	Iteratively reweighted least squares	264	
10.2.7	MAP estimation	266	
10.2.8	Standardization	267	
10.3	Multinomial logistic regression	268	
10.3.1	Linear and nonlinear classifiers	268	
10.3.2	Maximum likelihood estimation	269	
10.3.3	Gradient-based optimization	271	
10.3.4	Bound optimization	271	
10.3.5	MAP estimation	273	
10.3.6	Maximum entropy classifiers	273	
10.3.7	Hierarchical classification	274	
10.3.8	Handling large numbers of classes	275	
10.4	Bayesian logistic regression	277	
10.4.1	Approximating the posterior predictive	277	
10.4.2	Laplace approximation	278	
10.4.3	MCMC approximation	280	
10.4.4	Variational inference	282	
10.4.5	Online inference using assumed density filtering	286	
10.5	Preprocessing discrete input data	288	
10.5.1	One-hot encoding	288	
10.5.2	Feature crosses	289	
10.5.3	Dealing with text	289	
<b>11</b>	<b>Linear regression</b>	<b>293</b>	
11.1	Introduction	293	
11.2	Standard linear regression	293	
11.2.1	Terminology	293	
11.2.2	Least squares estimation	294	
11.2.3	Other approaches to computing the MLE	298	
11.2.4	Measuring goodness of fit	301	
11.3	Ridge regression	303	
11.3.1	Computing the MAP estimate	304	
11.3.2	Connection between ridge regression and PCA	305	
11.3.3	Choosing the strength of the regularizer	307	
11.4	Robust linear regression	307	
11.4.1	Robust regression using the Student $t$ distribution	307	
11.4.2	Robust regression using the Laplace distribution	309	
11.4.3	Robust regression using Huber loss	310	

11.4.4	Robust regression by randomly or iteratively removing outliers	311
11.5	Lasso regression	311
11.5.1	MAP estimation with a Laplace prior ( $\ell_1$ regularization)	311
11.5.2	Why does $\ell_1$ regularization yield sparse solutions?	312
11.5.3	Hard vs soft thresholding	313
11.5.4	Regularization path	315
11.5.5	Comparison of least squares, lasso, ridge and subset selection	316
11.5.6	Variable selection consistency	317
11.5.7	Group lasso	318
11.5.8	Elastic net (ridge and lasso combined)	321
11.5.9	Optimization algorithms	321
11.6	Bayesian linear regression	323
11.6.1	Computing $p(\mathbf{w} \mathcal{D}, \sigma^2)$ with Gaussian prior	323
11.6.2	Computing $p(\mathbf{w}, \sigma^2 \mathcal{D})$ with Gaussian-Gamma prior	327
11.6.3	Uninformative priors	329
11.6.4	Sparsity-promoting priors	331
11.6.5	Hierarchical priors	334
11.6.6	Empirical Bayes (Automatic relevancy determination)	336
11.6.7	Online inference (recursive least squares)	339
<b>12</b>	<b>Generalized linear models</b>	<b>341</b>
12.1	Introduction	341
12.2	The exponential family	341
12.2.1	Definition	341
12.2.2	Examples	342
12.2.3	Log partition function is cumulant generating function	346
12.2.4	MLE for the exponential family	348
12.2.5	Exponential dispersion family	348
12.3	Generalized linear models (GLMs)	349
12.3.1	Examples	349
12.3.2	Maximum likelihood estimation	351
12.3.3	GLMs with non-canonical link functions	352
12.4	Probit regression	353
12.4.1	Latent variable interpretation	353
12.4.2	Maximum likelihood estimation	354
12.4.3	Bayesian inference	355
12.4.4	Ordinal probit regression	355
12.4.5	Multinomial probit models	356
<b>III</b>	<b>Deep neural networks</b>	<b>357</b>
<b>13</b>	<b>Neural networks for unstructured data</b>	<b>359</b>
13.1	Introduction	359
13.2	Multilayer perceptrons (MLPs)	360

13.2.1	The XOR problem	360	
13.2.2	Differentiable MLPs	361	
13.2.3	Activation functions	362	
13.2.4	Example models	364	
13.2.5	The importance of depth	368	
13.2.6	Connections with biology	370	
13.3	Backpropagation	372	
13.3.1	Forwards pass	372	
13.3.2	Backwards pass	372	
13.3.3	Automatic differentiation	374	
13.3.4	Computation graphs	376	
13.4	Training neural networks	378	
13.4.1	Tuning the learning rate	378	
13.4.2	Vanishing and exploding gradients	378	
13.4.3	Residual connections	380	
13.4.4	Batch normalization	381	
13.4.5	Parameter initialization	383	
13.5	Regularization	385	
13.5.1	Early stopping	385	
13.5.2	Weight decay	385	
13.5.3	Sparse DNNs	385	
13.5.4	Dropout	387	
13.5.5	Bayesian neural networks	387	
13.6	Other kinds of feedforward networks	388	
13.6.1	Radial basis function networks	388	
13.6.2	Mixtures of experts	389	
<b>14</b>	<b>Neural networks for images</b>	<b>395</b>	
14.1	Introduction	395	
14.2	Basics	395	
14.2.1	Convolution in 1d	395	
14.2.2	Convolution in 2d	397	
14.2.3	Convolution as matrix-vector multiplication	398	
14.2.4	Boundary conditions and strides	398	
14.2.5	Pooling layers	401	
14.2.6	Normalization layers	402	
14.2.7	Putting it altogether	403	
14.3	Image classification using CNNs	403	
14.3.1	Common datasets	403	
14.3.2	Common models	407	
14.4	Solving other discriminative vision tasks with CNNs	411	
14.4.1	Image tagging	411	
14.4.2	Object detection	412	
14.4.3	Human pose estimation	413	
14.4.4	Image segmentation	413	

14.5	Generating images by inverting CNNs	416
14.5.1	Converting a trained classifier into a generative model	416
14.5.2	Image priors	416
14.5.3	Visualizing the features learned by a CNN	418
14.5.4	Deep Dream	419
14.5.5	Neural style transfer	420
14.6	Adversarial Examples	423
14.6.1	Whitebox (gradient-based) attacks	424
14.6.2	Blackbox (gradient-free) attacks	425
14.6.3	Real world adversarial attacks	426
14.6.4	Defenses based on robust optimization	426
14.6.5	Why models have adversarial examples	427
<b>15</b>	<b>Neural networks for sequences</b>	<b>431</b>
15.1	Introduction	431
15.2	Recurrent neural networks (RNNs)	431
15.2.1	Vec2Seq (sequence generation)	431
15.2.2	Seq2Vec (sequence classification)	434
15.2.3	Seq2Seq (sequence translation)	435
15.2.4	Beam search	437
15.2.5	Backpropagation through time	437
15.2.6	Gating and long term memory	438
15.3	1d CNNs	440
15.3.1	1d CNNs for sequence classification	440
15.3.2	Causal 1d CNNs for sequence generation	441
15.4	Attention	442
15.4.1	Seq2seq with attention	443
15.4.2	Seq2vec with attention	444
15.4.3	Attention as a soft dictionary lookup	444
15.4.4	Soft vs hard attention	446
15.5	Transformers	446
15.5.1	Self-attention	447
15.5.2	Multi-headed attention	448
15.5.3	Positional encoding	449
15.5.4	Putting it altogether	449
15.5.5	Comparing transformers, CNNs and RNNs	450
15.6	Efficient transformers	451
15.6.1	Fixed non-learnable localized attention patterns	451
15.6.2	Learnable sparse attention patterns	452
15.6.3	Memory and recurrence methods	453
15.6.4	Low-rank and kernel methods	453

<b>IV</b>	<b>Nonparametric models</b>	<b>457</b>
<b>16</b>	<b>Exemplar-based methods (Unfinished)</b>	<b>459</b>
16.0.1	K nearest neighbor (KNN) classification	459
16.1	Kernel density estimation (KDE)	463
16.1.1	Kernel functions	463
16.1.2	Parzen window density estimator	464
16.1.3	How to choose the bandwidth parameter	466
16.1.4	From KDE to KNN classification	466
16.1.5	Kernel regression	466
16.2	Learning distance metrics	469
16.2.1	Linear and convex methods	469
16.2.2	Deep metric learning	470
16.2.3	Speedup tricks	473
<b>17</b>	<b>Kernel methods</b>	<b>475</b>
17.1	Inferring functions from data	475
17.1.1	Smoothness prior	476
17.1.2	Inference from noise-free observations	476
17.1.3	Inference from noisy observations	478
17.2	Mercer kernels	478
17.2.1	Mercer's theorem	478
17.2.2	Some popular Mercer kernels	479
17.3	Gaussian processes	484
17.3.1	Noise-free observations	484
17.3.2	Noisy observations	485
17.3.3	Weight space vs function space	486
17.3.4	Numerical issues	487
17.3.5	Estimating the kernel	487
17.3.6	GPs for classification	490
17.3.7	Connections with deep learning	492
17.4	Scaling GPs to large datasets	492
17.4.1	(Sparse) variational inference	492
17.4.2	Exploiting structure in the kernel matrix	496
17.4.3	Random feature approximation	498
17.5	Support vector machines (SVMs)	500
17.5.1	Large margin classifiers	500
17.5.2	The dual problem	502
17.5.3	Soft margin classifiers	504
17.5.4	The kernel trick	505
17.5.5	Converting SVM outputs into probabilities	505
17.5.6	Connection with logistic regression	506
17.5.7	Multi-class classification with SVMs	507
17.5.8	How to choose the regularizer $C$	508
17.5.9	Kernel ridge regression	509

17.5.10	SVMs for regression	510	
17.6	Sparse vector machines	512	
17.6.1	Relevance vector machines (RVMs)	513	
17.6.2	Comparison of sparse and dense kernel methods	513	
<b>18</b>	<b>Trees, forests, bagging and boosting</b>	<b>517</b>	
18.1	Classification and regression trees (CART)	517	
18.1.1	Model definition	517	
18.1.2	Model fitting	518	
18.1.3	Regularization	519	
18.1.4	Pros and cons	519	
18.2	Ensemble learning	521	
18.2.1	Stacking	521	
18.2.2	Ensembling is not Bayes model averaging	521	
18.3	Bagging	522	
18.4	Random forests	523	
18.5	Boosting	524	
18.5.1	Forward stagewise additive modeling	525	
18.5.2	Quadratic loss and least squares boosting	525	
18.5.3	Exponential loss and AdaBoost	525	
18.5.4	LogitBoost	528	
18.5.5	Gradient boosting	530	
18.6	Interpreting tree ensembles	533	
18.6.1	Feature importance	534	
18.6.2	Partial dependency plots	535	
<b>V</b>	<b>Beyond supervised learning</b>	<b>537</b>	
<b>19</b>	<b>Learning with fewer labeled examples</b>	<b>539</b>	
19.1	Data augmentation	539	
19.1.1	Examples	539	
19.1.2	Theoretical justification	540	
19.2	Transfer learning	541	
19.2.1	Fine-tuning	541	
19.2.2	Supervised pre-training	542	
19.2.3	Unsupervised pre-training (self-supervised learning)	543	
19.2.4	Domain adaptation	544	
19.3	Meta-learning	544	
19.3.1	Model-agnostic meta-learning (MAML)	545	
19.4	Few-shot learning	546	
19.4.1	Matching networks	547	
19.5	Word embeddings	548	
19.5.1	Methods based on SVD	548	
19.5.2	Word2vec	550	

19.5.3	RAND-WALK model of word embeddings	552
19.5.4	Word analogies	553
19.5.5	Contextual word embeddings	554
19.6	Semi-supervised learning	558
19.6.1	Self-training and pseudo-labeling	559
19.6.2	Entropy minimization	560
19.6.3	Co-training	562
19.6.4	Label propagation on graphs	563
19.6.5	Consistency regularization	564
19.6.6	Deep generative models	565
19.7	Active learning	568
19.7.1	Decision-theoretic approach	569
19.7.2	Information-theoretic approach	569
19.7.3	Batch active learning	570
<b>20</b>	<b>Dimensionality reduction</b>	<b>571</b>
20.1	Principal components analysis (PCA)	571
20.1.1	Examples	571
20.1.2	Derivation of the algorithm	573
20.1.3	Computational issues	576
20.1.4	Choosing the number of latent dimensions	578
20.2	Factor analysis	580
20.2.1	Generative model	580
20.2.2	Probabilistic PCA	582
20.2.3	EM algorithm for FA/PPCA	583
20.2.4	Unidentifiability of the parameters	585
20.2.5	Nonlinear factor analysis	587
20.2.6	Mixtures of factor analysers	588
20.2.7	Exponential family factor analysis	589
20.2.8	Factor analysis models for paired data	590
20.3	Autoencoders	593
20.3.1	Bottleneck autoencoders	594
20.3.2	Denoising autoencoders	595
20.3.3	Contractive autoencoders	595
20.3.4	Sparse autoencoders	597
20.3.5	Variational autoencoders	598
20.4	Manifold learning	603
20.4.1	What are manifolds?	603
20.4.2	The manifold hypothesis	603
20.4.3	Approaches to manifold learning	604
20.4.4	Multi-dimensional scaling (MDS)	605
20.4.5	Isomap	608
20.4.6	Kernel PCA	608
20.4.7	Maximum variance unfolding (MVU)	610
20.4.8	Local linear embedding (LLE)	611

20.4.9	Laplacian eigenmaps	612
20.4.10	t-SNE	615
<b>21</b>	<b>Clustering</b>	<b>619</b>
21.1	Hierarchical agglomerative clustering	619
21.1.1	The algorithm	619
21.1.2	Example	622
21.2	K means clustering	623
21.2.1	The algorithm	623
21.2.2	Examples	624
21.2.3	Vector quantization	625
21.2.4	The K-means++ algorithm	626
21.2.5	The K-medoids algorithm	627
21.2.6	Speedup tricks	627
21.2.7	Choosing the number of clusters $K$	628
21.3	Clustering using mixture models	631
21.3.1	Mixtures of Gaussians	631
21.4	Spectral clustering	633
21.4.1	Normalized cuts	633
21.4.2	Eigenvectors of the graph Laplacian encode the clustering	634
21.4.3	Example	634
21.4.4	Connection with other methods	635
21.5	Biclustering	636
21.5.1	Basic biclustering	636
21.5.2	Nested partition models (Crosscat)	637
<b>22</b>	<b>Recommender systems (Unfinished)</b>	<b>641</b>
22.1	Explicit feedback	641
22.1.1	Netflix competition	641
22.1.2	Matrix factorization	642
22.1.3	Autoencoders	642
22.2	Implicit feedback	642
22.2.1	Ranking loss	642
22.2.2	Neural collaborative filtering	642
22.3	Leveraging side information	642
22.3.1	Sequence-aware recommendation	642
22.3.2	Factorization machines	642
22.4	Exploration-exploitation tradeoff	642
<b>23</b>	<b>Graph embeddings</b>	<b>645</b>
23.1	Introduction	645
23.2	Graph Embedding as an Encoder/Decoder Problem	646
23.3	Shallow graph embeddings	648
23.3.1	Unsupervised embeddings	648
23.3.2	Distance-based: Euclidean methods	649



23.3.3	Distance-based: non-Euclidean methods	650	
23.3.4	Outer product-based: Matrix factorization methods	650	
23.3.5	Outer product-based: Skip-gram methods	651	
23.3.6	Supervised embeddings	652	
23.4	Graph Neural Networks	653	
23.4.1	Message passing GNNs	653	
23.4.2	Spectral Graph Convolutions	655	
23.4.3	Spatial Graph Convolutions	655	
23.4.4	Non-Euclidean Graph Convolutions	657	
23.5	Deep graph embeddings	657	
23.5.1	Unsupervised embeddings	657	
23.5.2	Semi-supervised embeddings	660	
23.6	Applications	661	
23.6.1	Unsupervised applications	661	
23.6.2	Supervised applications	663	
<b>Appendices</b>		<b>665</b>	
<b>VI Appendix: Mathematical background</b>		<b>667</b>	
<b>A Some useful mathematics</b>		<b>669</b>	
A.1	Introduction	669	
A.2	Sets, functions and relations	669	
A.2.1	Functions	669	
A.2.2	Relations	673	
A.3	Matrix calculus	674	
A.3.1	Derivatives	674	
A.3.2	Gradients	675	
A.3.3	Jacobian	676	
A.3.4	Hessian	676	
A.3.5	Gradients of commonly used functions	676	
A.4	Convexity	678	
A.4.1	Convex sets	678	
A.4.2	Convex functions	680	
A.4.3	Jensen's inequality	682	
A.4.4	Subgradients	682	
A.4.5	Taylor series approximation	683	
A.4.6	Bregman divergence	684	
A.4.7	Conjugate duality	685	
<b>B Linear algebra</b>		<b>689</b>	
B.1	Introduction	689	
B.1.1	Notation	689	
B.1.2	Vector spaces	692	
B.1.3	Norms of a vector and matrix	694	

B.1.4	Properties of a matrix	696	
B.1.5	Special types of matrices	698	
B.2	Matrix multiplication	702	
B.2.1	Vector-Vector Products	702	
B.2.2	Matrix-Vector Products	702	
B.2.3	Matrix-Matrix Products	703	
B.2.4	Application: manipulating data matrices	705	
B.2.5	Kronecker products	707	
B.2.6	Einstein summation	708	
B.3	Matrix inversion	709	
B.3.1	The inverse of a square matrix	709	
B.3.2	Schur complements	709	
B.3.3	The matrix inversion lemma	711	
B.3.4	Matrix determinant lemma	711	
B.4	Eigenvalue decomposition (EVD)	712	
B.4.1	Basics	712	
B.4.2	Diagonalization	713	
B.4.3	Eigenvalues and eigenvectors of symmetric matrices	713	
B.4.4	Geometry of quadratic forms	714	
B.4.5	Standardizing and whitening data	714	
B.4.6	Power method	716	
B.4.7	Deflation	717	
B.4.8	Eigenvectors optimize quadratic forms	717	
B.5	Singular value decomposition (SVD)	717	
B.5.1	Basics	717	
B.5.2	Connection between SVD and EVD	718	
B.5.3	Pseudo inverse	719	
B.5.4	SVD and the range and null space of a matrix	720	
B.5.5	Truncated SVD	721	
B.5.6	Application: matrix factorization (MF)	722	
B.6	Other matrix decompositions	722	
B.6.1	LU factorization	722	
B.6.2	QR decomposition	723	
B.6.3	Cholesky decomposition	724	
B.7	Solving systems of linear equations	724	
B.7.1	Solving square systems	725	
B.7.2	Solving underconstrained systems (least norm estimation)	726	
B.7.3	Solving overconstrained systems (least squares estimation)	727	
<b>C</b>	<b>Probability</b>	<b>729</b>	
C.1	Introduction	729	
C.1.1	What is probability?	729	
C.1.2	Types of uncertainty	729	
C.1.3	Fundamental rules of probability	730	
C.2	Random variables	731	

C.2.1	Discrete random variables	731	
C.2.2	Continuous random variables	732	
C.3	Sets of related random variables	734	
C.3.1	Joint, marginal and conditional distributions	734	
C.3.2	Bayes' rule	735	
C.3.3	Independence and conditional independence	735	
C.4	Properties of a distribution	736	
C.4.1	Moments of a distribution	736	
C.4.2	Covariance	739	
C.4.3	Correlation	739	
C.4.4	Uncorrelated does not imply independent	740	
C.4.5	Correlation does not imply causation	741	
C.4.6	Simpsons' paradox	741	
C.5	Transformations of random variables	742	
C.5.1	Discrete case	743	
C.5.2	Continuous case	743	
C.5.3	Invertible transformations (bijections)	743	
C.5.4	Moments of a linear transformation	745	
C.5.5	The convolution theorem	746	
C.5.6	Central limit theorem	748	
<b>D</b>	<b>Frequentist statistics</b>	<b>749</b>	
D.1	Introduction	749	
D.2	Fisher information matrix (FIM)	749	
D.2.1	Definition	749	
D.2.2	Connection between the FIM and the Hessian of the NLL	750	
D.2.3	Examples	751	
D.2.4	Connection between FIM and KL divergence	752	
D.3	Sampling distributions	753	
D.3.1	Exact sampling distribution of the MLE	753	
D.3.2	Large sample approximation	755	
D.3.3	Bootstrap approximation	756	
D.3.4	Confidence intervals	758	
D.4	Bias and variance	759	
D.4.1	Bias of an estimator	759	
D.4.2	Variance of an estimator	759	
D.4.3	The bias-variance tradeoff	760	
D.4.4	Jackknife	762	
D.5	Frequentist decision theory	764	
D.5.1	Computing the risk of an estimator	764	
D.5.2	Consistent estimators	767	
D.5.3	Admissible estimators	767	
D.5.4	Stein's paradox	768	
D.6	Empirical risk minimization	770	
D.6.1	Empirical risk	770	

D.6.2	Structural risk	772	
D.6.3	Cross-validation	772	
D.6.4	Statistical learning theory	773	
D.7	Hypothesis testing	774	
D.7.1	Likelihood ratio test	775	
D.7.2	Null hypothesis significance testing (NHST)	776	
D.7.3	t-test	776	
D.7.4	$\chi^2$ test	777	
D.7.5	p-values	778	
D.8	Pathologies of frequentist statistics	778	
D.8.1	Confidence intervals are not credible	779	
D.8.2	p-values confuse deduction with induction	780	
D.8.3	p-values overstate evidence against the null hypothesis	780	
D.8.4	p-values depend on the stopping rule	781	
D.8.5	Why isn't everyone a Bayesian?	782	
<b>E</b>	<b>Exercises</b>	<b>785</b>	
	<b>Bibliography</b>	<b>821</b>	

# Preface

In 2012, I published a 1200-page book called “Machine learning: a probabilistic perspective”, which provided a fairly comprehensive coverage of the field of machine learning (ML) at that time, under the unifying lens of probabilistic modeling. The book was well received, and won the [De Groot prize](#) in 2013.

2012 was also the year that is generally considered the start of the “deep learning revolution”. The term “deep learning” refers to a branch of ML that is based on neural networks with many layers (hence the term “deep”). Although this basic technology had been around for many years, it was not until 2012 that it started to significantly outperform other, more “classical” approaches to ML, on several challenging benchmarks. For example, [\[KSH12\]](#) used deep neural networks (DNNs) to win the ImageNet image classification challenge, [\[CMS12\]](#) used DNNs to win a different image classification challenge, and [\[DHK13\]](#) used DNNs to outperform existing methods for speech recognition by a large margin. These breakthroughs were enabled by advances in hardware technology (in particular, the repurposing of fast graphics processing units from video games to ML), data collection technology (in particular, the use of crowd sourcing to collect large labeled datasets such as ImageNet), as well as various new algorithmic ideas.

Since 2012, the field of deep learning has exploded, with new advances coming at an increasing pace. Interest in the field has also exploded, fueled by the commercial success of the technology, and the breadth of applications to which it can be applied. Therefore, in 2018, I decided to write a second edition of my book, to attempt to summarize some of this progress.

By Spring 2020, my draft of the second edition had swollen to about 1600 pages, and I was still not done. At this point, 3 major events happened. First, MIT Press told me they could not publish a 1600 page book, and that I would need to split it into two volumes. Second, the COVID-19 pandemic struck, so I decided to put the book on hold, so I could work 100% on various internal and external modeling efforts. Third, as a consequence of my “pivot” towards COVID-19 work, I realized that I would never finish the book unless I got help from others; fortunately I managed to recruit several colleagues to help me write the last  $\sim 15\%$  of “missing content”. (See acknowledgements below.)

The result is two new books, “Probabilistic Machine Learning: An Introduction”, which you are currently reading, and “Probabilistic Machine Learning: Advanced Topics”, which is the sequel to this book [\[Mur22\]](#). Together these two books attempt to present a fairly broad coverage of the field of ML c. 2020, using the same unifying lens of probabilistic modeling and Bayesian decision theory that I used in the first book.

Most of the content from the first book has been reused, but it is now split fairly evenly between

the two new books. In addition, each book has lots of new material, covering some topics from deep learning, but also advances in other parts of the field, such as generative models, variational inference and reinforcement learning. To make the book more self-contained and useful for students, I have also added some more background content, on topics such as optimization and linear algebra, that was omitted from the first book due to lack of space.

Another major change is that nearly all of the software now uses Python instead of Matlab. The new code leverages standard Python libraries, such as numpy, scipy, scikit-learn, etc. Some examples also rely on various deep learning libraries, such as [TensorFlow](#), [PyTorch](#) and [JAX](#). In addition to scripts to create some of the figures, there are Jupyter notebooks to accompany each chapter, which discuss practical aspects that we don't have space to cover in the main text. Details can be found at <http://mlbayes.ai>.

## Acknowledgements

I would like to thank the following people for helping me to write various parts of this book:

- Frederik Kunstner, Si Yi Meng, Aaron Mishkin, Sharan Vaswani, and Mark Schmidt who helped write parts of [Chapter 5 \(Optimization algorithms\)](#).
- Lihong Lig, who helped write parts of [Sec. 8.3 \(Bandit problems\)](#).
- Justin Gilmer, who helped write [Sec. 14.6 \(Adversarial Examples\)](#).
- Krzysztof Choromanski, who helped write [Sec. 15.6 \(Efficient transformers\)](#).
- Andrew Wilson, who helped write [Sec. 17.4.2 \(Exploiting structure in the kernel matrix\)](#).
- Colin Raffel, who helped write [Sec. 19.2 \(Transfer learning\)](#) and [Sec. 19.6 \(Semi-supervised learning\)](#).
- Bryan Perozzi, who helped write [Chapter 23 \(Graph embeddings\)](#).
- Zico Kolter, who helped write parts of [Chapter B \(Linear algebra\)](#).

I would like to thank John Fearnas for very carefully proofreading the entire book, Peter Cerno who also spotted many errors.

I would like to thank the following people for feedback on various sections: Sebastien Bratieres, Kai Brodersen, Daniel Galvez, Abhishek Kumar, Max Lepikhin, Aaron Michelsony, Hal Varian.

I would like to thank the following people for help with Python coding: Andrew Carr, Duane Rich, Theodore Vasiloudis. I would also like to thank those who shared their open source code (see credits in each file online).

I would like to thank all those who shared figures from their own papers (see credits in each figure caption), as well as Sandeep Choudhary for help making some of the figures, and Aurélien Géron for letting me use the Python code from his excellent book [[Gér19](#)] to make some of the figures.

Finally I would like to thank my manager at Google, Doug Eck, for letting me invest the (significant) time needed to make this book a reality. I hope my efforts to synthesize all this material together in one place will help to save you time in your journey of discovery into the “land of ML”.

Kevin Patrick Murphy  
Palo Alto, California  
December 2020.