

1                                   **Nonparametric Bayesian Methods**  
2                                   **in Robotics**

3                                   PROEFSCHRIFT

4                                   ter verkrijging van de graad van doctor  
5                                   aan de Universiteit Leiden,  
6                                   op gezag van de rector magnificus,  
7                                   prof. mr. C.J.J.M. Stolker,  
8                                   volgens besluit van het College voor Promoties  
9                                   te verdedigen op ...2018  
10                                  in de aula van de Universiteit  
11                                  klokke ...

12                                  door

13                                  **Anne Cornelis van Rossum,**  
14                                  **geboren op 30 december 1980**  
15                                  **te Dirksland**

## 16 Samenstelling van de promotiecommissie

17

18

Promotoren: Prof. dr. H.J. van den Herik,  
Prof. dr. ir. H. X. Lin

Copromotor: Dr. J.L.A. Dubbeldam

Promotiecommissie: Prof. dr. A. Plaat,  
Prof. dr. J.N. Kok,  
Prof. dr. ir. A.W. Heemink (TU Delft) ,  
... (Buitenlands),  
Prof. dr. ir. B.J.A. Kröse  
Prof. dr. C.M. Jonker (SIKS)



19 SIKS Dissertation Series No. ...

20 The research reported in the thesis has been carried out under the auspices of SIKS, the  
21 Dutch Research School for Information and Knowledge Systems.

22

23 ISBN ....

24 Copyright © 2018, A.C. van Rossum

25 *All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or*  
26 *transmitted, in any form or by any means, electronically, mechanically, photocopying, recording*  
27 *or otherwise, without prior permission of the author.*

28 “ *The study of mental objects with reproducible properties is called mathematics.* ”  
29

30 The Mathematical Experience (Davis and Hersch, 1981)

31 “ *The study of physical objects with reproducible properties is called science.* ”  
32

33 The dawning of the age of stochasticity, Mathematics: frontiers and perspectives  
34 (Mumford, 2000)

# CONTENTS

36	<b>1 Introduction</b>	<b>1</b>
37	1.1 Scope of the Thesis . . . . .	1
38	1.2 Bayesian Nonparametrics . . . . .	2
39	1.3 Problem Statement and Research Questions . . . . .	2
40	1.4 Research Methodology . . . . .	3
41	1.5 Main Contribution . . . . .	4
42	1.6 Organization of the Thesis . . . . .	5
43	<b>2 Related Work</b>	<b>8</b>
44	2.1 Probability Theory . . . . .	8
45	2.1.1 Measure Theory . . . . .	9
46	2.1.2 Bayesian Inference . . . . .	17
47	2.2 Five Random Processes . . . . .	22
48	2.2.1 Beta Process . . . . .	28
49	2.2.2 Gamma Process . . . . .	29
50	2.2.3 Dirichlet Process . . . . .	29
51	2.2.4 Pitman-Yor Process . . . . .	32
52	2.2.5 Hierarchical Dirichlet Process . . . . .	33
53	2.3 Inference . . . . .	34
54	2.3.1 Inverse Transform Sampling . . . . .	34
55	2.3.2 Rejection Sampling . . . . .	35
56	2.3.3 Approximate Bayesian Computation . . . . .	36
57	2.3.4 Gibbs Sampling . . . . .	37
58	2.3.5 Metropolis-Hastings Sampling . . . . .	39
59	2.3.6 Split-Merge MCMC Sampling . . . . .	39
60	2.4 Chapter Conclusions . . . . .	40
61	<b>3 Nonparametric Bayesian Line Detection</b>	<b>43</b>
62	3.1 Infinite Line Model . . . . .	44
63	3.1.1 Posterior Predictive for a Line given Other Lines . . . . .	44
64	3.1.2 Likelihood of Data given a Line . . . . .	46
65	3.1.3 Conjugate Prior for a Line . . . . .	47
66	3.1.4 Posterior Predictive for a Line given Data . . . . .	48
67	3.2 Inference for the Infinite Line Model . . . . .	48

68	3.3	Accelerating Inference for the Infinite Line Model . . . . .	50
69	3.4	Performance of the Infinite Line Model . . . . .	51
70	3.4.1	Clustering Performance . . . . .	51
71	3.4.2	Two Examples . . . . .	52
72	3.5	Chapter Conclusions . . . . .	53
73	<b>4</b>	<b>Nonparametric Bayesian Segment Estimation</b>	<b>57</b>
74	4.1	Pareto Pairs . . . . .	57
75	4.2	Generative Process to Create a Line Segment . . . . .	60
76	4.3	Inference over a Line Segment . . . . .	61
77	4.4	Results . . . . .	63
78	4.5	Chapter Conclusions . . . . .	64
79	<b>5</b>	<b>Triadic Split-Merge Sampler</b>	<b>66</b>
80	5.1	The Class of Split-Merge Samplers . . . . .	66
81	5.2	Conventional Split-Merge Sampler . . . . .	67
82	5.2.1	Acceptance for the Split Step . . . . .	68
83	5.2.2	Acceptance for the Merge Step . . . . .	69
84	5.2.3	Sequentially-Allocated Merge-Split Sampler . . . . .	70
85	5.3	Triadic split-merge sampler . . . . .	70
86	5.3.1	Acceptance for the Split Step . . . . .	71
87	5.3.2	Acceptance for the Merge Step . . . . .	73
88	5.4	Results . . . . .	74
89	5.4.1	Implementation . . . . .	75
90	5.4.2	Comparison . . . . .	76
91	5.5	Chapter Conclusions . . . . .	77
92	<b>6</b>	<b>Adversarially Trained MCMC Kernels</b>	<b>79</b>
93	6.1	Data-Driven Inference . . . . .	79
94	6.2	Learning the Transition Operator . . . . .	80
95	6.2.1	Adversarial Training . . . . .	80
96	6.2.2	Variational Autoencoders . . . . .	81
97	6.2.3	Infusion Training . . . . .	81
98	6.3	Volumetric Models . . . . .	82
99	<b>7</b>	<b>Recommender Engine</b>	<b>85</b>
100	7.1	Application . . . . .	85
101	7.2	Model of Individuals . . . . .	86
102	7.2.1	Multi-modal Normal-Uniform Distribution Model . . . . .	86
103	7.2.2	Multi-modal Von-Mises-Uniform Distribution Model . . . . .	87
104	7.2.3	Hyperparameters . . . . .	88
105	7.3	Model of Groups . . . . .	89
106	7.4	Inference . . . . .	89
107	7.5	Results . . . . .	90
108	7.5.1	Artificial Dataset . . . . .	90
109	7.5.2	Real-world Dataset . . . . .	91
110	7.6	Discussion . . . . .	92

111	<b>8 Discussion and Conclusions</b>	<b>94</b>
112	<b>References</b>	<b>96</b>
113	<b>Appendices</b>	
114	<b>A Probabilistic Concepts</b>	<b>104</b>
115	A.1 Common Inequalities . . . . .	105
116	A.1.1 Markov's Inequality . . . . .	105
117	A.1.2 Chebyshev's Inequality . . . . .	106
118	A.1.3 Weak Law of Large Numbers . . . . .	106
119	A.1.4 Strong Law of Large Numbers . . . . .	107
120	A.1.5 Common Distributions . . . . .	107
121	<b>B Dirichlet-multinomial</b>	<b>109</b>
122	B.1 Categorical Distribution . . . . .	109
123	B.2 Multinomial Distribution . . . . .	110
124	B.3 N Categorical Distributions . . . . .	111
125	<b>C Gibbs Sampling</b>	<b>112</b>
126	C.1 Gibbs Sampling of Parameters . . . . .	114
127	<b>D Glossary</b>	<b>117</b>
128	<b>List of Figures</b>	<b>118</b>
129	<b>List of Tables</b>	<b>121</b>
130	<b>List of Abbreviations</b>	<b>122</b>
131	<b>Summary</b>	<b>124</b>
132	<b>Samenvatting</b>	<b>126</b>
133	<b>Acknowledgments</b>	<b>128</b>
134	<b>Curriculum Vitae</b>	<b>129</b>
135	<b>Publications</b>	<b>130</b>
136	<b>SIKS Dissertation Series</b>	<b>131</b>

# INTRODUCTION

## Contents

The thesis addresses nonparametric Bayesian methods in robotic vision. Nonparametric Bayesian models can be simultaneously employed to perform inference over the number of entities observed and over the shape or nature of these entities. This chapter introduces nonparametric Bayesian models, the research methodology based on the Bayesian methodology, the main contribution towards robotic vision, and the general organization of the thesis.

## Outline

The scope of thesis is to apply nonparametric Bayesian methods to robotic vision (Sect. 1.1). Bayesian nonparametric models define entities together with noise in such a way that inference can be performed in an optimal manner (Sect. 1.2). Particular problems in robotic vision that can benefit from Bayesian nonparametric methods are formulated and detailed (Sect. 1.3). The research methodology is described (Sect. 1.4). Our main contribution is to introduce nonparametric Bayesian models in robotic vision (Sect. 1.5). At the end of this chapter the organization of the thesis is given (Sect. 1.6).

## 1.1 Scope of the Thesis

In the thesis, modern Bayesian nonparametric methods are used to answer long-standing questions within computer vision and robotics. The following three challenging questions are typical examples. Is there a Bayesian form of line detection rather than applying the traditional Hough transform? Which of the nonparametric Bayesian priors can be used to detect multiple features simultaneously? What are efficient inference methods for these priors?

The scope of the thesis is the transfer of knowledge on Bayesian nonparametrics to well-described application domains. It will not establish a new body of work around a new family

of stochastic processes. The detailed application of complex models towards robotic vision is expected to help and encourage people in entirely different application domains, such as collaborative filtering, search engine optimization, and audio processing. All these different applications do not always need dedicated algorithms, but do deserve and can exploit the same optimal general inference techniques from Bayesian nonparametrics.

## 1.2 Bayesian Nonparametrics

In robotic vision (computer vision and depth perception) traditionally custom-made algorithms have been developed for a given task. There are specific methods to detect corners (e.g., Förstner and Gülch, 1987; Harris and Stephens, 1988; Shi and Tomasi, 1994), to detect edges (e.g., Sobel, 1970; Canny, 1986), to detect features (e.g., Hough, 1962), and to describe features (e.g., Lowe, 1999; Dalal and Triggs, 2005; Bay et al., 2006).

On the one hand, it is desirable that such sophisticated methods are generalizable to other application domains. On the other hand, it is important to take particular information about an application domain into account. The methods described in the previous paragraph are limited to their specific task. An example of limited generalizability can be found in the Hough transform. The Hough transform can be used to detect lines, but the way inference is performed in the algorithm does limit its application to basic forms of object detection. An example of limited specificity can be found in linear regression. Linear regression does not take into account real-world statistics.

Both generalization and specificity are formalized by a Bayesian model. A Bayesian model is general because it can be solved with general inference methods. One of such general inference methods is a Markov-Chain Monte Carlo method. It does not know anything about real-world statistics. A Bayesian model is also specific in that it can incorporate application-specific know-how by the definition of priors.

Typical problems in robotic vision will be about the recognition of several objects, multiple shapes, or objects that have multiple parts. Models that represent such objects do not have knowledge about the number of such objects, shapes, or parts. To incorporate application-specific know-how on the number of objects it is possible to define a prior that assigns a probability to this quantity. The number of objects can even be potentially infinite. The Bayesian models that define a prior on the number of objects, shapes, or parts are called nonparametric Bayesian models. This means that in contrast with conventional methods such as  $k$ -means clustering (Forgy, 1965; Lloyd, 1982) the number of objects does not need to be predefined.

## 1.3 Problem Statement and Research Questions

Many methods in robotics - and in particular in robotic vision - have been developed in times where computational resources were limited. Then, highly optimized algorithms have been developed, leveraging peculiarities of the application domain. Recent advances in Bayesian



201 methods, both with respect to concept development, as well as computational efficient so-  
202 lution strategies, now open up new ways to solve old problems. However, extending only  
203 the old methods themselves would lead to ad-hoc solution strategies that will miss benefits  
204 from potential optimal and more widely applicable algorithms.

205 This observation leads us to the formulation of our problem statement (PS).

206       **PS:**     *How can robotic problems effectively be generalized and their structure  
                  exploited in a wider Bayesian framework?*

207 The problem statement is rather general. In our research, we focus on robotic vision, in the  
208 form of point cloud recognition and depth perception. In particular, we look at objects, lines,  
209 line segments, and more complex shapes.

210 The problem statement is divided into three research questions (RQs).

**RQ 1**   How can we estimate the number of objects simultaneously with the  
                  fitting of these objects?

211       **RQ 2**   How can we estimate the number of lines simultaneously with line fitting  
                  in computer vision?

**RQ 3**   How can we recognize more general 3D objects?

## 212   1.4   Research Methodology

213 The research methodology advocated in the thesis follows the Bayesian methodology (cf.  
214 Savage, 1972; Jaynes, 2003). So, our research methodology exists out of two phases. In the  
215 first phase a Bayesian model is defined. This model exists of (1) a definition of parameters  
216 and relations between these parameters, (2) a definition of the noise, and (3) the data. In  
217 the second phase, the Bayesian method dictates all remaining unknowns, from the number  
218 of parameters to the values of the parameters. To perform Bayesian inference efficiently new  
219 methods are required if the model is complex (as is in the case of robotic vision).

220 The Bayesian methodology aims to establish the rationale for practical questions. The fol-  
221 lowing two questions are clear examples.

- 222       ◦ If we observe a single point in an image, can we expect it to be part of a line?
- 223       ◦ If we have two lines and we live in a world with squares, what are we able to infer?

224 The two questions tap into our capabilities to define models that makes our prior knowledge  
225 explicit. Moreover, if we are able to quickly assign (1) points to segments, (2) segments to  
226 lines, (3) objects to categories, we can enrich it with all corresponding group properties  
227 without the need to have them observed for this individual.

228 In robotic vision we take as an example the task of line detection. Both the Hough transform  
229 (Hough, 1962) and the RANSAC method (Bolles and Fischler, 1981) do detect lines, but they  
230 do not explicitly take noise into account. By applying Bayesian methodology to these tasks,  
231 the inference method becomes optimal in an information-theoretic sense. Also frequentist  
232 statisticians agree that nonparametric Bayesian models are consistent in the sense that they  
233 approach the underlying true distribution (Wasserman, 1998). There is no need to search  
234 for another method to infer lines in a line detection task. If someone would find a method  
235 that outperforms a Bayesian method it is either (1) because the signal or noise has not been  
236 correctly modeled after all, or (2) because the method overfits with respect to the available  
237 data. If approximations are used with respect to optimal Bayesian inference (either varia-  
238 tional approximations or Markov-Chain Monte Carlo), there are theoretical guarantees on  
239 convergence.

240 A well known problem with nonparametric Bayesian models is the curse of dimensionality.  
241 Compared to maximum likelihood methods or other non-probabilistic methods that do not  
242 take noise into account at all, the nonparametric Bayesian models require significant com-  
243 putational resources. Our research methodology first establishes the correct models, even  
244 if solving them seems computationally infeasible. Our approach is to develop subsequently  
245 approximations using more sophisticated samplers, so that the theoretical guarantees on  
246 convergence are preserved.

247 Due to the fact that the models are optimal by construction, there are no experiments re-  
248 quired to address the optimality in particular. However, experiments are still required to  
249 establish whether the models make sense. Yet, the methodology does also have limitations.  
250 For instance, we will not search over different noise models and limit priors to a particular  
251 hierarchical level.

## 252 1.5 Main Contribution

253 Our contribution to robotic vision can be subdivided into three parts that correspond with  
254 the three research questions.

255 The first part addresses the problem of inference about objects from a nonparametric Bayes-  
256 ian perspective. Contemporary methods in robotic vision do not allow for astute statements  
257 about their performance. In practice, this means that when using computer vision to detect  
258 cells under a microscope, someone cannot be confident about the number of detected cells.  
259 An autonomous cleaning robot in a supermarket cannot be confident about the aisle it is driv-  
260 ing into. To be able to properly take into account models and uncertainty simultaneously,  
261 Bayesian models have found mainstream adoption. State-of-the-art Bayesian methods that  
262 reason about the number of objects alongside object models are a recent object of study (cf.  
263 Ferguson, 1973; Hjort, 1990; Lijoi and Prünster, 2010; Joho et al., 2011). The thesis applies  
264 such nonparametric Bayesian models towards the applications of robotic vision and depth  
265 perception. Models such as the infinite line model and the infinite line segment model are  
266 introduced.

267 The second part addresses the problem of high-dimensional data. To efficiently sample more  
268 complex geometric structures, new MCMC (Markov-Chain Monte Carlo, Sect. 2.3.4) meth-  
269 ods are required. The thesis introduces such an MCMC sampler, namely a new Split-Merge  
270 sampler, and applies it to complex geometric structures.

271 The third part addresses more complex robotic vision problems, in the form of object recog-  
272 nition of point clouds in 3D. It combines nonparametric Bayesian inference with models  
273 from deep learning.

## 274 1.6 Organization of the Thesis

275 **Chapter 1** (this chapter) introduces the problem of contemporary methods in computer  
276 vision and depth perception. Due to the fact that these methods are not optimal  
277 by construction, it is hard to articulate how they perform. The need for a Bayesian  
278 methodology is sketched briefly. The problem statement and the research questions  
279 are formulated. Moreover, the research methodology is described and the organization  
280 of the thesis is outlined.

281 **Chapter 2** describes (1) probability theory using measure theory, (2) random measures  
282 known as random processes of which five are described as nonparametric Bayesian  
283 models, and (3) six inference methods that infer model parameters of such nonpara-  
284 metric Bayesian models given the data. It is followed by a discussion that indicates  
285 which parts will be most useful for chapters 3 and 4.

286 **Chapter 3** examines a first nonparametric Bayesian model, i.e., the infinite line model. The  
287 infinite line model represents a countably infinite set of lines. Gibbs sampling is used  
288 to perform simultaneous inference over (1) the number of lines and (2) line parameter  
289 values such as slope and intercept.

290 **Chapter 4** examines a second nonparametric Bayesian model, i.e., the infinite line segment  
291 model. The infinite line segment model represents a countably infinite set of line  
292 segments. A split-merge MCMC sampling method is used to perform simultaneous  
293 inference over (1) the number of line segments and (2) line segment parameter val-  
294 ues such as slope, intercept, and segment size. Chapter 2-4 answer the first research  
295 question.

296 **Chapter 5** investigates a new MCMC method, the Triadic Split-Merge sampler. It is tailored  
297 to clustering problems and accelerates inference of the models in chapter 3 and 4. This  
298 chapter answers the second research question.

299 **Chapter 6** examines a third nonparametric Bayesian model, particularly aimed at volumet-  
300 ric inference. This chapter answers the third research question.

301 **Chapter 7** applies the hierarchical sampling method to the domain of recommender en-  
302 gines. It estimates simultaneously the number of user types with a fitting procedure  
303 for the individual user. I want to change this chapter to something relevant to point  
304 clouds.

305 **Chapter 8** discusses the relevance of the developed models and inference methods. The  
306 answers to the research questions are discussed. Then the problem statement is an-  
307 swered and conclusions are formulated. Finally, recommendations are given and fu-  
308 ture research is envisaged.



309

310

## RELATED WORK

### 311 Contents

312

313

314

315

316

317

318

319

320

In robotics depth sensors generate point clouds. The tasks of robotic object recognition, positioning, and navigation require models that represent such point clouds. It is unclear whether the current methods that perform inference over point clouds are appropriate for these tasks. The current models do not model uncertainty explicitly. This chapter presents models that can be used for point cloud modeling and that represent uncertainty. This (partially) answers research question RQ1. The chapter concludes with recommendations for the development of point cloud inference models. They will be implemented in a new model for line inference in Chapter 3 and line segment inference in Chapter 4.

### 321 Outline

322

323

324

325

326

327

328

329

330

331

332

333

This chapter describes probability theory, and in particular, measure theory underlying random processes (Sect. 2.1). Five random processes are described, the Beta Process, the Gamma Process, the Dirichlet Process, the Pitman-Yor Process, and the Hierarchical Dirichlet Process. The random processes are presented as a Poisson process with a Lévy measure, a stick-breaking construction, and a sequential presentation (Sect. 2.2). These representations give rise to different inference methods. Six inference methods are described. Inverse transform sampling, rejection sampling, approximate Bayesian computation, Gibbs sampling, Metropolis-Hastings, and Split-Merge Markov chain Monte Carlo (Sect. 2.3). Inference about point clouds in the chapters to follow will use adaptations of the described models and inference methods for which some recommendations are given (Sect. 2.4).

## 334 2.1 Probability Theory

335 Modern probability is based on measure theory (Sect. 2.1.1). Measure theory will provide  
336 the means to formally describe random variables, random processes, and most generally,

random measures. A model represented by random measures can be fitted to the data using Bayesian inference (Sect 2.1.2).

### 2.1.1 Measure Theory

A random variable is a *function* that assigns values to a *set* of possible outcomes. The formal definition requires concepts such as “measurable function” and “probability space” from *measure theory* (Feller, 1950). Measure theory is used to generalize the notion of a random variable to that of a “random process”.

Informally, a measure generalizes the notion of size of Euclidean objects to sets and subsets. The definition of a measure is based on the definition of a  $\sigma$ -algebra. A  $\sigma$ -algebra ascribes a value to a sum of individual disjoint sets, even if they are infinite in number.

#### ▼ Definition 2.1 — $\sigma$ -algebra

A  $\sigma$ -algebra is a subset  $\Sigma \in 2^X$ , with  $X$  a set and  $2^X$  its powerset, with three requirements:

- $\Sigma$  is non-empty: at least one  $A \in X$  is in  $\Sigma$ ;
- $\Sigma$  is closed under complementation: if  $A$  in  $\Sigma$ , so is its complement  $A^c$ ;
- $\Sigma$  is closed under countable unions: if  $A_1, A_2, \dots$  in  $\Sigma$ , so is  $A = A_1 \cup A_2 \cup \dots$

The members of a  $\sigma$ -algebra are called *measurable sets*. Let  $X = \{1, 2, 3, 4\}$  and let us define a  $\sigma$ -algebra  $\Sigma = \{\emptyset, \{1\}, \{4\}, \{2, 3\}, \{1, 4\}, \{1, 2, 3\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$ . Here  $\emptyset$  denotes the empty set. The complement of  $A$  is defined with respect to  $X$ :  $A \cup A^c = X$ . An example of closure under complementation: let  $A_1 = \{1\}$ , then  $A_1^c = \{2, 3, 4\}$  and  $A_1^c$  is indeed a member of  $\Sigma$ :  $A_1^c \in \Sigma$ . An example of closure under countable unions: let  $A_1 = \{1\}$  and  $A_2 = \{2, 3\}$ , then  $A_1 \cup A_2 = \{1, 2, 3\}$  and  $A_1 \cup A_2 \in \Sigma$ .

#### ▼ Definition 2.2 — *generated* $\sigma$ -algebra

A **generated  $\sigma$ -algebra**, with  $X$  a set and  $B \in 2^X$ , is the smallest  $\sigma$ -algebra  $\sigma(B)$  that contains all sets of  $B$ .

Let  $X = \{1, 2, 3, 4\}$  and  $B = \{\{1\}, \{2, 3\}\}$ , then the generated  $\sigma$ -algebra is the set  $\sigma(B) = \{\emptyset, \{1\}, \{2, 3\}, \{1, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$ . Here the sets in  $B$  are completed to the sets in  $\sigma(B)$  by obeying the requirements of a  $\sigma$ -algebra of closure under complementation and countable unions by addition (e.g.  $\{1, 4\}$  is added due to closure under completion with respect to  $\{2, 3\}$ ).

371

The notion of a  $\sigma$ -algebra (Fremlin, 2000) can be applied to solve the so-called Banach-Tarski paradox (Banach and Tarski, 1924). This paradox describes how a unit-ball in  $\mathbf{R}^3$  can be partitioned into a finite number of disjoint infinite sets (scattering of points) and then can be reassembled into two unit-balls again. This violates the intuitive notion of preservation of volume. If the measure  $\mu$  of the union of two disjoint sets is equal to the sum of the measures of the two sets, this is called *finite additivity*:  $\mu(\bigcup_{i=1}^N A_i) = \sum_{i=1}^N \mu(A_i)$ . In probability theory  $\sigma$ -additivity extends this to infinite disjoint sets:  $\mu(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} \mu(A_i)$ . Measure theory solves the Banach-Tarski paradox by only assigning a measure to subsets that are measurable sets (Tao, 2011).

372 A *measure* assigns values to measurable sets (as stated before, measurable sets are members  
373 or subsets of  $\Sigma$ ).

374

### ▼ Definition 2.3 — *measure*

375

376 A **measure**  $\mu$  is a function from  $\Sigma$  to  $[-\infty, +\infty]$ , with three requirements:

377

- $\mu$  is non-negative:  $\mu(A) \geq 0$  for  $\forall A \in \Sigma$ ;

378

- $\mu$  has a null empty set:  $\mu(\emptyset) = 0$ ;

379

- $\mu$  is  $\sigma$ -additive:  $\mu(\bigcup_{i \in I_{\Sigma}} A_i) = \sum_{i \in I_{\Sigma}} \mu(A_i)$  for  $A_i$  disjoint.

380

381 The first statement defines that a measure  $\mu$  only assigns non-negative values to sets in  $\Sigma$ .  
382 The second statement equals the measure of the empty set  $\emptyset$  to 0. The third statement  
383 defines that  $\sigma$ -additivity is required. For any two sets in  $\Sigma$  the measure of the union of the  
384 sets equals the sum of the measures of the individual sets. Here  $I_{\Sigma}$  defines an index over  
385 sets in  $\Sigma$ .

386

Informally, a measure relates the concepts of *sets* and *subsets* to notions of size. A measure can be seen as a *monotonically* increasing function. Let the set  $A$  in  $X$  be the interval  $[0, 1)$ , an uncountable (infinite) set of real numbers. Define the  $\sigma$ -algebra  $\{\emptyset, A\}$ . The empty set has measure 0, the set  $A$  has measure 1. Let us define the  $\sigma$ -algebra  $\{\emptyset, A_{0,0.5}, A_{0.5,1}, A\}$ . The set  $A_{0,0.5}$  corresponds to the interval  $[0, 0.5)$  and  $A_{0.5,1}$  to  $[0.5, 1)$ . Both sets are assigned measure 0.5 and their union has measure 1. This examples shows that with  $\sigma$ -additive unions, measures can be assigned to sets that are uncountable.

387 A *measurable space*  $(X, \Sigma)$  is defined as a pair.

388

### ▼ Definition 2.4 — *measurable space*

389

390 A **measurable space**  $(X, \Sigma)$  is a pair with:

391

- $X$  a set;

392

- $\Sigma$  a  $\sigma$ -algebra over  $X$ .



393

394 A *measure space*  $(X, \Sigma, \mu)$  is defined as a triple.

395

---

▼ **Definition 2.5 — *measure space***

---

396

397 A **measure space**  $(X, \Sigma, \mu)$  is a triple with:

398

- $X$  a set;

399

- $\Sigma$  a  $\sigma$ -algebra over  $X$ ;

400

- $\mu$  a measure from  $\Sigma$  to  $[-\infty, \infty]$ .

---

401

402 A finite measure  $\mu$  assigns a finite real number to all  $A$ .

403

---

▼ **Definition 2.6 — *finite measure***

---

404

405 A **finite measure**  $\mu$  is a measure from  $\Sigma$  to  $[0, \infty)$ :

406

- $\mu$  is non-negative:  $\mu(A) \geq 0$  for  $\forall A \in \Sigma$ ;

407

- $\mu$  has a null empty set:  $\mu(\emptyset) = 0$ ;

408

- $\mu$  is  $\sigma$ -additive:  $\mu(\bigcup_{i \in I_\Sigma} A_i) = \sum_{i \in I_\Sigma} \mu(A_i)$  for  $A_i$  disjoint;

409

- $\mu$  for the whole sample space,  $X$ , is finite:  $\mu(X) = N$ .

---

410

411 A  $\sigma$ -finite measure allows  $A$  to be a countable union of sets with finite measure.

412

---

▼ **Definition 2.7 —  $\sigma$ -finite measure**

---

413

414 A  **$\sigma$ -finite measure**  $\mu$  is a finite measure with:

415

- $X$  is a countable union of sets with finite measures.

---

416

417 We will now define five measures: (A) the *probability measure* (def. 2.8), (B) the *counting*  
 418 *measure* (def. 2.10), (C) the *Borel measure* (def. 2.12), (D) the *Lebesgue measure* (def.  
 419 2.17), and (E) the *random measure* (def. 2.18). These measures are important because they  
 420 are fundamental to different branches of mathematics. In probability theory a  $\sigma$ -algebra is  
 421 interpreted as a collection of events to which probabilities are assigned. Counting measures  
 422 play a fundamental role in discrete probability distributions. In integration theory a  $\sigma$ -  
 423 algebra corresponding to the Borel and Lebesgue measures are relevant for integration in  
 424 the Euclidean space  $\mathcal{R}^n$ . In statistics a  $\sigma$ -algebra formally defines a sufficient statistics and  
 425 generalizes random variables to random functions and measures.

## 426 A Probability measure

427 A *probability measure*,  $\mathbb{P}$ , is a finite measure which satisfies the following requirements:

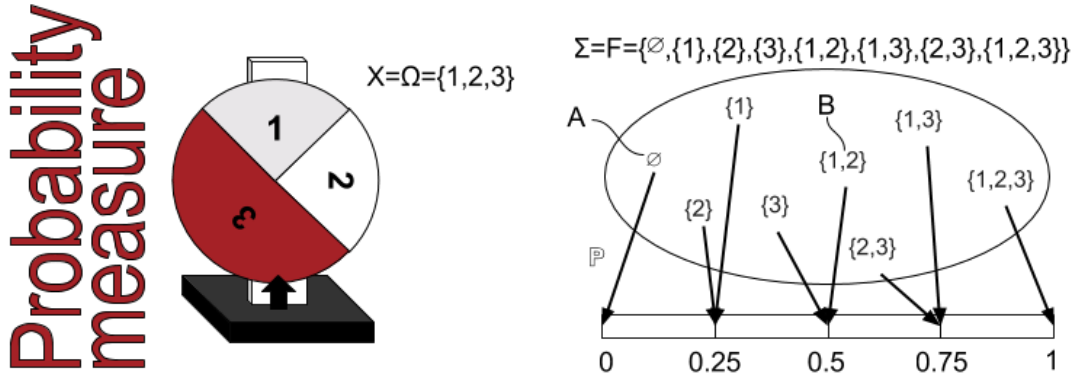
### ▼ Definition 2.8 — *probability measure*

A **probability measure**  $\mathbb{P}$  is a measure  $\mu$  with:

- $\mathbb{P}$  is non-negative:  $\mathbb{P}(A) \geq 0$  for  $\forall A \in \Sigma$ ;
- $\mathbb{P}$  has a null empty set:  $\mathbb{P}(\emptyset) = 0$ ;
- $\mathbb{P}$  is  $\sigma$ -additive:  $\mathbb{P}(\bigcup_{i \in I_{\Sigma}} A_i) = \sum_{i \in I_{\Sigma}} \mu(A_i)$  for  $A_i$  disjoint;
- $\mathbb{P}$  for the whole sample space,  $X$ , is unity:  $\mathbb{P}(X) = 1$ .

These requirements are called the Kolmogorov axioms (Kolmogorov, 1933). The probability measure is indeed an actual measure. It therefore obeys the same three requirements: (1) non-negativity for any set, (2) the existence of a null empty set, and (3)  $\sigma$ -additivity. A probability measure compared to a general measure obeys a fourth requirement, namely the restriction of the measure for the whole space  $X$  to 1. This can be seen as some kind of normalization and it influences how two probability measures have to be summed to become again a probability measure.

In Fig. 2.1 the probability measure is visualized as a mapping from the probability space to the unit interval  $[0, 1]$ .



**Figure 2.1:** A probability measure  $\mathbb{P}$  mapping the probability space for 3 events to the unit interval. Left: a turning wheel representing three possible outcomes of which the third is twice as likely as the other two outcomes. Right: a probability measure  $\mathbb{P}$  assigned to each outcome. The empty set,  $A = \emptyset$ , has probability measure 0. The set of encountering either 1 or 2,  $B = \{1, 2\}$ , has probability measure 0.5. Taken from Wikipedia.

A **probability space**  $(X, \Sigma, \mathbb{P})$  is a measure space  $(X, \Sigma, \mu)$  with the probability measure  $\mathbb{P}$  as its measure  $\mu$ .

### ▼ Definition 2.9 — *probability space*

A **probability space**  $(X, \Sigma, \mathbb{P})$  is a triple with:

- $X$  a set;

- 451      ◦  $\Sigma$  a  $\sigma$ -algebra over  $X$ ;
  - 452      ◦  $\mathbb{P}$  a probability measure from  $\Sigma$  to  $[0, 1]$ .
- 

454 The triple for the probability space  $(X, \Sigma, \mathbb{P})$  is also written as  $(\Omega, \mathbb{F}, \mathbb{P})$ . The space  $X$  is the  
 455 event space  $\Omega$ , the set of *elementary outcomes*. The  $\sigma$ -algebra over subsets of  $\Omega$  is denoted  
 456 by  $\mathbb{F}$ . The probability measure  $\mathbb{P}$  assigns a value on the unit interval  $[0, 1]$  to every event in  
 457  $\mathbb{F}$ .

## 458 B Counting measure

459 The *counting measure* forms the basis for the definition of discrete probabilities (Schilling,  
 460 2005).

### 461 ▼ Definition 2.10 — *counting measure*

---

463 A **counting measure**  $\nu$  on a space  $X$  is a measure  $\mu$  with:

- 464      ◦  $\nu$  is non-negative and integer-valued for  $\forall A \in \Sigma$ ;
  - 465      ◦  $\nu < \infty$  for  $\forall A \in \Sigma$  if  $A$  bounded (of finite size);
  - 466      ◦  $\nu = \infty$  if  $\exists A \in \Sigma$  with  $A$  unbounded (infinite).
- 

468 A counting measure is a measure that is integer-valued. Every set  $A$  has a measure that is a  
 469 positive integer or zero. If and only if the set  $A$  is unbounded will its counting measure be  
 470 infinite.

## 471 C Borel measure

472 The *Borel  $\sigma$ -algebra* defines a  $\sigma$ -algebra for the real line  $\mathbb{R}$ .

### 473 ▼ Definition 2.11 — *Borel $\sigma$ -algebra*

---

475 A **Borel  $\sigma$ -algebra**  $\mathbb{B}_\sigma$  on  $\mathbb{R}$  is the smallest  $\sigma$ -algebra that contains all open subsets of  
 476  $\mathbb{R}$ :

- 477      ◦  $\mathbb{B} = \Sigma(U)$  with  $U = \bigcup U \subseteq \mathbb{R}$ :  $U$  is open.
- 

479 The Borel  $\sigma$ -algebra contains all open subsets of  $\mathbb{R}$ . The property of closure under comple-  
 480 mentation of a  $\sigma$ -algebra means that it also contains the closed subsets of  $\mathbb{R}$ . If  $A = (0, 1)$ ,  
 481 then  $A^c = \{[-\infty, 0], [1, \infty]\}$ .

482 A *Borel measure* assigns values to subsets of  $\mathbb{B}_\sigma$ .

---

**▼ Definition 2.12 — Borel measure**


---

A **Borel measure**  $\mu$  is a function from  $\Sigma = \mathbb{B}_\sigma$  to  $[-\infty, +\infty]$ , with the three measure requirements:

- $\mu$  is non-negative:  $\mu(A) \geq 0$  for  $\forall A \in \Sigma$ ;
  - $\mu$  has a null empty set:  $\mu(\emptyset) = 0$ ;
  - $\mu$  is  $\sigma$ -additive:  $\mu(\bigcup_{i \in I_\Sigma} A_i) = \sum_{i \in I_\Sigma} \mu(A_i)$  for  $A_i$  disjoint.
- 

The *Borel space* is a measurable space with a Borel  $\sigma$ -algebra rather than a general  $\sigma$ -algebra.

---

**▼ Definition 2.13 — Borel space**


---

A **Borel space**  $(X, \mathbb{B}_\sigma)$  is a pair with:

- $X$  a set;
  - $\mathbb{B}_\sigma$  a Borel  $\sigma$ -algebra over  $X$ .
- 

A *complete measure space* is a measure space in which every subset of every null set is measurable.

---

**▼ Definition 2.14 — complete measure space**


---

A **complete measure space**  $(X, \Sigma, \mu)$ :

- $S \subseteq N \in \Sigma$  and  $\mu(N) = 0 \Rightarrow S \in \Sigma$ .
- 

The Borel space is not a complete measure space. There are sets in the Borel  $\sigma$ -algebra that are of measure zero and that contain subsets that are undefined.

## D Lebesgue measure

The *Lebesgue measure* defines a size to subsets of  $\mathbb{R}^n$  that completes the Borel measure (Lebesgue, 1902). It makes use of the notion of an *outer measure*.

---

**▼ Definition 2.15 — outer measure**


---

An **outer measure**  $\phi$  on a space  $\mathbb{R}$  is a measure  $\mu$  with:

- $\phi$  is non-negative and real-valued for  $\forall A \in \Sigma$ ;
- $\phi$  has a null empty set:  $\phi(\emptyset) = 0$ ;
- $\phi$  is  $\sigma$ -subadditive:  $\phi(\bigcup_{i \in I_\Sigma} A_i) < \sum_{i \in I_\Sigma} \mu(A_i)$  for  $\forall A_i$ ;
- $\phi$  is monotone:  $A \subseteq B$  implies  $\phi(A) \leq \phi(B)$ ;

- 518     ◦  $\phi$  is translation-invariant:  $\phi(A + x) = \phi(A)$  for  $\forall A \in \Sigma$  and  $\forall x \in \mathbb{R}$ .  
 519
- 

520 An outer measure relaxes  $\sigma$ -additivity of disjoint sets of  $X$  to  $\sigma$ -subadditivity for any se-  
 521 quence of sets. Intuitively, the outer measure of a set is an upper bound on the size of a  
 522 set.

523     **▼ Definition 2.16 — Lebesgue outer measure**

---

524  
 525 A **Lebesgue outer measure**  $\lambda$  on a space  $\mathbb{R}^n$  is an outer measure  $\phi$  with:

- 526     ◦  $\lambda(A) = \inf \left\{ \sum_{k=1}^{\infty} l(I_k) : (I_k)_{k \in \mathbb{N}} \text{ is a sequence of open intervals with } A \subseteq \bigcup_{k=1}^{\infty} I_k \right\}$ .  
 527
- 

528 Here  $A \subseteq \mathbb{R}$  is a subset of the real line. The Lebesgue outer measure  $\lambda$  is the infimum (greatest  
 529 lower bound) of the sum of the lengths  $l(I) = b - a$  of the intervals  $I = [a, b]$ .

530 The *Lebesgue measure* is defined through the Lebesgue outer measure.

531     **▼ Definition 2.17 — Lebesgue measure**

---

532  
 533 A **Lebesgue measure**  $m$  on a space  $\mathbb{R}^n$  is a Lebesgue outer measure  $\lambda$  with:

- 534     ◦  $m(B) = \lambda(B \cup A) + \lambda(B \cup A^c)$ .  
 535
- 

536     **E Random measure and random process**

537 A measurable function is defined between two measurable spaces.

538     **▼ Definition 2.18 — measurable function**

---

539  
 540 A **measurable function**  $f : X \rightarrow Y$  fulfills:

- 541     ◦  $f^{-1}(E) \in \Sigma$      for      $\forall E \in T$ ,

542 with both  $(X, \Sigma)$  and  $(Y, T)$  measurable spaces.  
 543

---

544     A measurable function *preserves the structure* of the corresponding measurable spaces  
     (captured through the  $\sigma$ -algebras).

545 A random variable is a measurable function between two measurable spaces, with as domain  
 546 a measurable space that is a probability space.

547     **▼ Definition 2.19 — random variable**

---

548  
 549 A  $(X, \Sigma)$ -valued **random variable**  $X$  is a measurable function from probability space  
 550  $(\Omega, \mathbb{F}, \mathbb{P})$  to measurable space  $(X, \Sigma)$ .  
 551

---

552 A random variable is a  $(X, \Sigma)$ -valued random variable with the following choice for the  
 553 codomain and  $\sigma$ -algebra:

554 **▼ Definition 2.20**

555 A **random variable**  $X$  is a measurable function from probability space  $(\Omega, \mathbb{F}, \mathbb{P})$  to the  
 556 real line with the Borel  $\sigma$ -algebra  $(\mathbb{R}, \mathbb{B}_{\mathbb{R}})$ .  
 557

558

559 The codomain is the real line  $\mathbb{R}$  and the Borel  $\sigma$ -algebra.

560 Random variables can be generalized to complex random variables or random elements of any type. A *complex random variable* is a measurable function from  $\Omega$  to  $\mathbb{C}$ . A *random elephant* is a measurable function from  $\Omega$  to a suitable space of elephants (Kingman, 1993).

561 **▼ Definition 2.21**

562 A **random measure** is a function  $\xi : \Omega \times X \rightarrow [0, +\infty]$  from probability space  $(\Omega, \mathbb{F}, \mathbb{P})$   
 563 to measurable space  $(X, \Sigma)$  such that  $\xi(\cdot, X)$  is a random variable on  $(\Omega, \mathbb{F}, \mathbb{P})$  and  $\xi(\omega, \cdot)$   
 564 is a measure on  $\Sigma$ .  
 565

566

567 We have encountered a random variable, and a probability measure  $\mathbb{P}$  on the original probability space. Now, one might wonder whether probabilities are logically assigned to elements on the measurable space that is the codomain of this random variable. Why does  
 568 it map to a measurable space and not a measure space actually? This is because (through  
 569 the  $\sigma$ -algebras of both spaces, or more precisely the random variable itself) the probability  
 570 measure is *induced* on the target space. This is known as a *probability distribution*:  
 571

572 **▼ Definition 2.22**

573 Given a random variable  $X$  from  $(\Omega, \mathbb{F}, \mathbb{P})$  to  $(\mathbb{R}, \mathbb{B}_{\sigma})$ , the **probability distribution**  $\mu$  is  
 574 the induced probability measure:  $\mu(B) = \mathbb{P}(X^{-1}(B))$  for all Borel sets  $B \in \mathbb{B}_{\sigma}$ .  
 575

576

577 The measurable *function*  $X$  is inverted:  $X^{-1}(\cdot)$ . The measure  $\mu$  exists on  $(\mathbb{R}, \mathbb{B}_{\sigma})$  just as  $\mathbb{P}$   
 578 exists on  $(\Omega, \mathbb{F})$ . The notation for the measure  $\mu$  does not include the original probability  
 579 space or  $\sigma$ -algebra. The complete notation for the probability distribution  $\mu$  can be written  
 580 as a function  $f$  of  $X$ :  
 581

$$f_X(x) = f_{X, (\Omega, \mathbb{F}, \mathbb{P}), (\mathbb{R}, \mathbb{B}_{\sigma})}(x). \quad (2.1)$$

582 At the left  $X$  denotes the random variable,  $x \in \Omega$  are the (elementary) outcomes on the  
 583 sample space  $\Omega$ . At the right the complete notation adds  $\mathbb{F}, \mathbb{P}$  and  $\mathbb{R}, \mathbb{B}_{\sigma}$ . The shorthand  
 584 notation at the left will be used to indicate the real line with a Borel  $\sigma$ -algebra as codomain.

585 A random variable  $X$  is *distributed* as  $f_X(x)$ , notation:

$$X \sim f_X(x). \quad (2.2)$$

586 A *random process* is an *ordered* set of random variables. The set can be a sequence of random  
 587 variables in a time series. It can be a series of steps in the spatial domain, called a random  
 588 field.

589 **▼ Definition 2.23 — *random process***

590  
 591 A **random process**  $X$  is a collection  $\{X_t : t \in T\}$  with  $X_t$  an  $(S, \Sigma)$ -valued random vari-  
 592 able on  $\Omega$  and  $(\Omega, \mathbb{F}, \mathbb{P})$  a probability space,  $(S, \Sigma)$  a measurable space, and  $T$  a totally  
 593 *ordered* set.

595 A random process is a probability distribution with a domain that is a set of probability distri-  
 596 butions. A random process is a distribution over distributions, a hierarchy over distribution.

597 **2.1.2 Bayesian Inference**

598 Let  $x$  be a  $(S, \Sigma_S, \mu_S)$ -valued random variable<sup>1</sup>,  $y$  a  $(T, \Sigma_T, \mu_T)$ -valued random variable, then  
 599 we can construct  $z$ , a  $(C, \Sigma_C, \mu_C)$ -valued random variable with the latter being a subset of  
 600 the product set of  $x$  and  $y$ :  $C \in S \otimes T$ .

601 **▼ Definition 2.24 — *product space***

602  
 603 A **product space**  $(S \otimes T, \Sigma_{S \otimes T})$  has  $\sigma$ -algebra  $\Sigma_{S \otimes T} = \sigma(F \otimes G : F \in \Sigma_S, G \in \Sigma_T)$  with  
 604  $(S, \Sigma_S, \mu_S)$  and  $(T, \Sigma_T, \mu_T)$  two  $\sigma$ -finite measure spaces.

606 **▼ Definition 2.25 — *product measure***

607  
 608 A **product measure**  $\mu_{S \otimes T}$  is a measure  $\mu_{S \otimes T}(F \otimes G) = \mu_S(F) \otimes \mu_T(G)$  with  $(S, \Sigma_S, \mu_S)$   
 609 and  $(T, \Sigma_T, \mu_T)$  two  $\sigma$ -finite measure spaces.

611 The **joint probability distribution**  $P_C$  is a probability measure on the product  $\sigma$ -algebra  $\Sigma_C$   
 612 with  $C \in S \otimes T$ . As function of the random variables  $x$  and  $y$  the joint probability distribution  
 613 is written as  $_{x,y}(x, y)$ ,  $f(x, y)$ , or  $p(x, y)$ .

614 A  $\sigma$ -algebra is *independent* in the following sense.

615 **▼ Definition 2.26 — *independent  $\sigma$ -algebra***

616  
 617 Let  $(\Omega, \mathbb{F}, P)$  be a probability space and  $\mathbb{A}$  and  $\mathbb{B}$  be a sub- $\sigma$ -algebras of  $\mathbb{F}$ .  $\mathbb{A}$  and  $\mathbb{B}$  are  
 618 **independent  $\sigma$ -algebras** if:

<sup>1</sup>The lowercase  $x$  is used instead of  $X$  in the context of probability distributions as in Eq. 2.1.

619     ◦  $P(A \cap B) = P(A)P(B) \forall A \in \mathbb{A} \text{ and } B \in \mathbb{B}$   
 620

---

621 Two random variables  $x$  and  $y$  are independent if and only if the  $\sigma$ -algebras that they gen-  
 622 erate are independent.

623     ▼ **Definition 2.27 — conditional probability distribution**  
 624

---

625 Let  $(\Omega, \mathbb{F}, P)$  be a probability space,  $\mathbb{G} \subseteq \mathbb{F}$  a sub- $\sigma$ -algebra of  $\mathbb{F}$ , and  $X : \Omega \rightarrow \mathbb{R}$  a real-  
 626 valued random variable ( $\mathbb{F}$ -measurable with respect to the Borel  $\sigma$ -algebra  $\mathbb{B}_\sigma$  on  $\mathbb{R}$ ).  
 627 There exists a function  $\mu : \mathbb{B}_\sigma \times \Omega \rightarrow \mathbb{R}$  such that  $\mu(\cdot, \omega)$  is a probability measure on  $\mathbb{B}_\sigma$   
 628 for each  $\omega \in \Omega$  and  $\mu(H, \cdot) = P(X \in H | \mathbb{G})$  (almost surely) for every  $H \in \mathbb{B}_\sigma$ . For any  
 629  $\omega \in \Omega$ , the function  $\mu(\cdot, \omega) : \mathbb{B}_\sigma \rightarrow \mathbb{R}$  is called a **conditional probability distribution**  
 630 of  $X$  given  $\mathbb{G}$ .  
 631

---

632 Informally, a conditional probability is described with a sub- $\sigma$ -algebra which only presents  
 633 part of the structure of the full  $\sigma$ -algebra. As function of the random variables  $x$  and  $y$  the  
 634 conditional probability distribution of  $y$  given  $x$  is written as  $f_{Y|X}(y|x)$ ,  $f(y|x)$ , or  $p(y|x)$ .

635 The random variables  $x$  and  $\theta$  define a Bayesian model with observations  $x$  and parameters  
 636  $\theta$ .

637     ▼ **Definition 2.28 — Bayesian model**  
 638

---

639 A **Bayesian model**  $f(x, \theta)$  defines a function between observations  $x$  and parameters  
 640  $\theta$  with both  $x$  and  $\theta$  random variables.  
 641

---

642 In a **supervised learning** task both  $x$  and  $\theta$  are known. In an **unsupervised learning** task  
 643  $x$  is known, but  $\theta$  is unknown. The random variable  $\theta$  is called a hidden or latent variable.  
 644 The random variable  $\theta$  can be any random element: a random vector, a random matrix, a  
 645 random process.

646 Let the observations  $x$  be a sequence  $x_0, x_1, \dots$ , then the observations  $x_i$  can be distributed  
 647 *independent and identically*.

648     ▼ **Definition 2.29 — independent and identically distributed**  
 649

---

650 A collection of random variables  $x = \{x_0, x_1, \dots\}$  is **independent and identically dis-**  
 651 **tributed (i.i.d.)** if:

- 652     ◦ the probability distribution  $p(x_i)$  is the same for  $\forall x_i \in x$ ;
  - 653     ◦ each  $x_i$  is independent with respect to  $x_j$  with  $i \neq j$ .
- 

655 The observations  $x_i$  can be distributed in an *exchangeable* sequence in which any order is  
 656 equally likely.



---

**▼ Definition 2.30 — *exchangeable***


---

A sequence of random variables  $x = \{x_0, x_1, \dots\}$  is **exchangeable** if for any finite permutation  $\rho$  of the indices  $0, 1, \dots$ :

- the joint probability distribution of the permuted sequence  $p(x_{\rho(0)}, x_{\rho(1)}, \dots)$  equals that of the original sequence  $p(x_0, x_1, \dots)$ .
- 

The joint probability distribution of i.i.d. observations given parameters can be written as a product:

$$p(x_0, \dots, x_{k-1} | \theta) = \prod_{i=0}^{k-1} p(x_i | \theta). \quad (2.3)$$

---

**▼ Definition 2.31 — *likelihood function***


---

The **likelihood function** is defined as:

$$\mathcal{L}(\theta | x) = p(x | \theta). \quad (2.4)$$

The likelihood of the parameters  $\theta$  given observations  $x$  is the probability of these observations given the parameter values.

The definition of the likelihood function allows us to find an optimal set of parameter values given the observations. The probability  $p(x | \theta)$  can be maximized (Aldrich and Others, 1997).

---

**▼ Definition 2.32 — *Maximum Likelihood***


---

**Maximum Likelihood** is defined as:

$$\theta^* \in \operatorname{argmax}_{\theta} \prod_{i=0}^{k-1} p(x_i | \theta). \quad (2.5)$$


---

The Maximum Likelihood method finds the maximum of  $\prod_{i=0}^{k-1} p(x_i | \theta)$  for all possible parameter values  $\theta$ . The maximum in Maximum Likelihood does not need to be unique (Steel, 1994). The notation makes this explicit by writing  $\theta^*$  as a member (denoted by the  $\in$  symbol) of the outcomes of the argmax operation (and does not use the equal sign).

A function  $f(\cdot)$  and the logarithm of a function  $\log f(\cdot)$  have the same maxima. This is due to the fact that the logarithm is a monotonic function (a monotonically increasing function). The log of a product of logarithms is equal to the sum of the individual logarithms.

---

**▼ Definition 2.33 — Maximum log-Likelihood**


---

**Maximum log-Likelihood** is defined as:

$$\theta^* \in \operatorname{argmax}_{\theta} \sum_{i=0}^{k-1} \log p(x_i | \theta). \quad (2.6)$$


---

In the case we have information about the parameters  $\theta$  we can model this with a probability distribution.

---

**▼ Definition 2.34 — prior probability distribution**


---

A **prior probability distribution** defines a probability distribution  $p(\theta)$  to parameters  $\theta$  without a dependency on the observations  $x$ .

---

Given the definition of a prior probability distribution, we can define *Maximum A Posteriori* estimation.

---

**▼ Definition 2.35 — Maximum A Posteriori**


---

**Maximum A Posteriori** estimation:

$$\theta^* \in \operatorname{argmax}_{\theta} \sum_{i=0}^{k-1} \log p(x_i | \theta) + \log p(\theta). \quad (2.7)$$


---

If we are not only interested in the parameter  $\theta^*$  that maximizes  $p(x|\theta)$  and  $p(\theta)$ , but in the complete distribution for  $p(\theta)$  we need Bayes' theorem described by Laplace (1820).

---

**▼ Definition 2.36 — Bayesian inference**


---

**Bayesian inference** uses Bayes' theorem:

$$f(\theta|x) = p(\theta|x) = \frac{\overbrace{p(x|\theta)}^{\text{likelihood}} \overbrace{p(\theta)}^{\text{prior}}}{\underbrace{p(x)}_{\text{normalization constant}}} = \frac{p(x|\theta)p(\theta)}{\int p(x|\theta)p(\theta)d\theta}. \quad (2.8)$$


---

Bayes' theorem describes the posterior probability  $p(\theta|x)$  as the likelihood times the prior probability distribution divided by a normalization constant, also called the evidence. The normalization constant is not a function of the parameters  $\theta$ . If a function is known except for the normalization constant, it is indicated by the "proportional to" symbol  $\propto$ .

$$f(\theta|x) \propto p(x|\theta)p(\theta) \quad (2.9)$$

In Bayesian inference  $p(\theta|x)$  is calculated. In contrast, in Maximum Likelihood and Maximum A Posteriori only parts of Eq. 2.8 are calculated, respectively  $p(x|\theta)$  and  $p(x|\theta)p(\theta)$ . In Section 2.3 inference methods will be described that approximate Bayesian inference. Approximation is required in the case closed-form expressions are not available. If the inference task only requires Maximum A Posteriori, approximation methods are also available (Daume, 2007), but this is outside of the scope of the current thesis.

There are two supervised learning models, a generative model and a discriminative model. Below we provide their definitions and in Fig. 2.2 we give three examples for each model.

---

▼ **Definition 2.37 — generative model**

---

A **generative** model defines the joint probability distribution  $p(x, \theta)$  and uses Bayes rule to define  $p(x|\theta)$ .

---



---

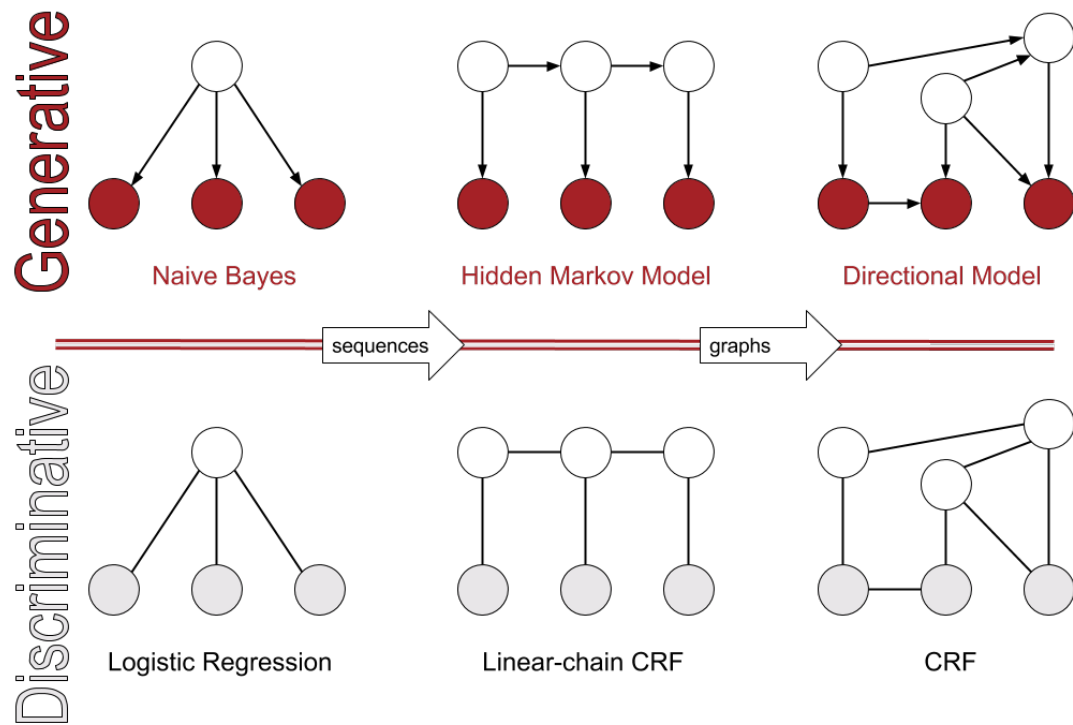
▼ **Definition 2.38 — discriminative model**

---

A **discriminative** model defines the conditional probability distribution  $p(x|\theta)$  directly.

---

Fig. 2.2 shows three generative and three discriminative models. These are chosen because from left to right structure is added between the random variables, first in the form of a sequence structure, next in the form of graph structure. Fig. 2.2 visualizes three generative models: (1) the Naive Bayes Model (Russell et al., 1995), (2) the Hidden Markov Model (Baum and Petrie, 1966), and the Directional Model (Koller and Friedman, 2009). It shows also three discriminative models: (1) Logistic Regression, (2) Linear-chain Conditional Random Fields, and (3) general Conditional Random Fields.



**Figure 2.2:** Generative models: Naive Bayes Model, Hidden Markov Model, and Directional Model. Discriminative models: Logistic Regression, Linear-chain Conditional Random Fields, and general Conditional Random Fields. Figure adapted from Sutton and McCallum (2011).

There is no definitive reason to use a generative model rather than a discriminative model or vice-versa. Here we confine ourselves to two remarks. First, a discriminative model has lower asymptotic error, but a generative model approaches its asymptotic error faster in the case of a Naive Bayes classifier versus Logistic Regression (Jordan, 2002). However, Xue and Titterington (2008) doubt the existence of such precisely defined regimes. According to them the asymptotic error denotes the error with an increasing number of samples. Second, the prior  $p(\theta)$  in the generative model provides a principled way to handle missing information, while the direct modeling of decision boundaries in a discriminative model often leads to better performance in a classification task (Jaakkola et al., 1999). Apart from generative models and discriminative models, there are also hybrid models (Bouchard and Triggs, 2004; Raina et al., 2003; Bosch et al., 2008). In the thesis we will limit ourselves to generative models.

## 2.2 Five Random Processes

In this section we investigate nonparametric Bayesian models. A model can be composed out of a set of probability distributions. We list three of such possible compositions. The Naive Bayes model is a *product* of probability distributions with a prior distribution (def. 2.39). The finite mixture model is a *sum* of probability distribution where each one is weighted

(def. 2.40). The infinite mixture model is a *sum* of probability distribution where each one is weighted (def. 2.41).

---

▼ **Definition 2.39 — Naive Bayes model**

---

The **Naive Bayes model** is a product over a finite number  $k \neq \infty$  of probability distributions  $p(x_i|\theta)$  multiplied by the prior distribution  $p(\theta)$ :

$$p(\theta|x) \propto p(\theta) \prod_{i=0}^{k-1} p(x_i|\theta) \quad (2.10)$$

---



---

A finite mixture model is a sum over a finite number of probability distributions:

---

▼ **Definition 2.40 — finite mixture model**

---

A **finite mixture model** is a sum over a finite number  $k \neq \infty$  of probability distributions  $p(x_i)$ , with each distribution weighted by a factor  $w_i$  with  $\sum_i w_i = 1$ .

$$p(x) = \sum_{i=0}^{k-1} w_i p(x_i) \quad (2.11)$$

---



---

The mixture model is finite in the sense that there are only  $k \neq \infty$  distributions summed up. The weights of the individual distributions  $p(x_i)$  are normalized (sum up to one) such that the weighted sum of probability distributions is itself a probability distribution.

An infinite mixture model is a sum over an infinite number of probability distributions:

---

▼ **Definition 2.41 — infinite mixture model**

---

A **infinite mixture model** is a sum over an infinite number of probability distributions  $p(x_i)$ , with each distribution weighted by a factor  $w_i$  with  $\sum_i w_i = 1$ .

$$p(x) = \sum_{i=0}^{\infty} w_i p(x_i) \quad (2.12)$$

---



---

The infinite mixture model is a sum over an infinite number of probability distributions with weights that sum up to one. In this way it assigns a finite value to a countably infinite set of functions.

In section 2.1.1 random elements were described in general. Random elements can vary from random vectors, random distributions, random clusters (partitions), to random trees.

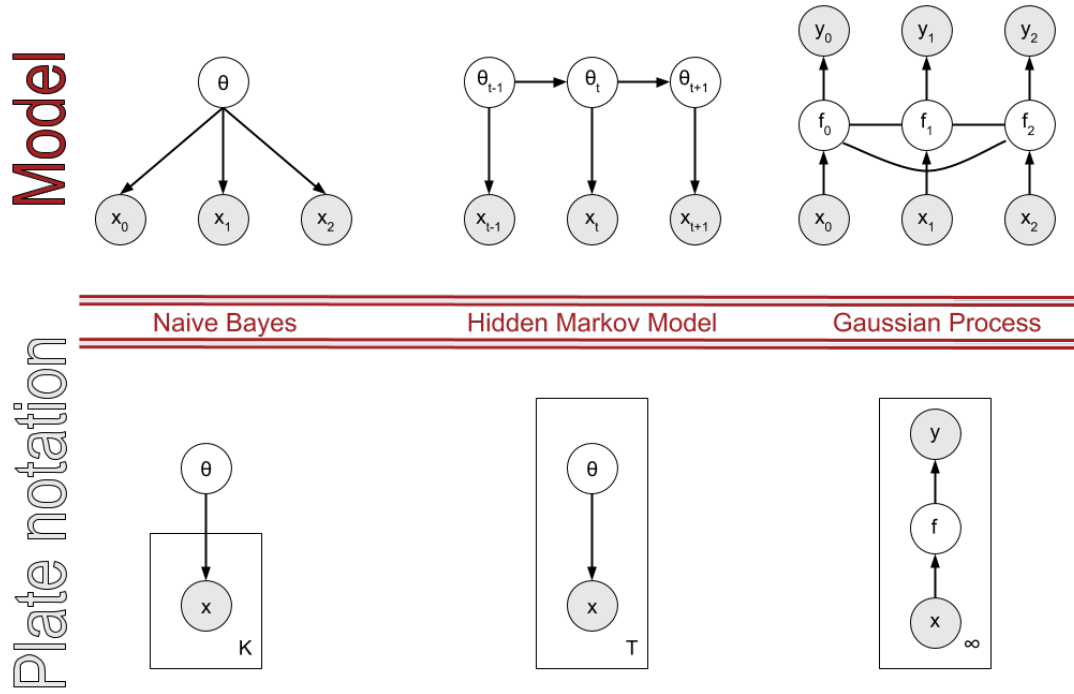
790 Table 2.1 describes the random elements and the corresponding examples of random pro-  
 791 cesses in the literature. Below we mention them with the appropriate references.

**Table 2.1:** A list of mathematical structures and random processes that can be used to generate those mathematical structures.

Structure	Example
Distribution on functions	Gaussian Process
Distribution on distributions	Beta Process
	Gamma Process
	Dirichlet Process
	Polya Tree
Distribution on partitions	Chinese Restaurant Process
	Pitman-Yor Process
Distribution on hierarchical partitions	Dirichlet Diffusion Tree
	Kingman's coalescence
Distribution on sparse binary matrices	Indian Buffet Process
Distribution on integer-valued matrices	Gamma-Poisson Process
Distribution on kd-trees	Mondrian Process

792 The Gaussian Process (Rasmussen and Williams, 2006) describes a distribution on functions.  
 793 The Beta Process (Hjort, 1990), the Gamma Process (Ferguson, 1974), the Dirichlet Process  
 794 and the Polya Tree (Ferguson, 1973) describe a distribution on distributions. The Chinese  
 795 Restaurant Process (Aldous, 1985) and Pitman-Yor Process (Pitman and Yor, 1997) describe  
 796 a distribution on partitions. The Dirichlet Diffusion Tree (Neal, 2001) and Kingman's coales-  
 797 cence (Kingman, 1965) describe a distribution on hierarchical partitions. The Indian Buffet  
 798 Process (Ghahramani and Griffiths, 2005) describes a distribution over sparse binary matri-  
 799 ces. The Gamma-Poisson Process (Titsias, 2008) describes a distribution over integer-valued  
 800 matrices. The Mondrian Process (Roy and Teh, 2009) describes a distribution over kd-trees.

801 Random processes and mixture models are visually represented by a method called *plate*  
 802 *notation* (Koller and Friedman, 2009). Sets of variables are represented in a plate, a rectan-  
 803 gular region (see Fig. 2.3).



**Figure 2.3:** Top: graphical model of a Naive Bayes, Hidden Markov Model, and Gaussian Process. Bottom: corresponding plate notation of the Naive Bayes, Hidden Markov Model, and Gaussian Process. Observed variables are denoted by a circle that is shaded.

804 Plate notation is a representation that does not preserve all dependencies between variables.  
 805 For example, the dependencies between the states in the Hidden Markov Model (e.g., be-  
 806 tween  $\theta_0$  and  $\theta_1$ ) are not represented.

807 Some random process are mathematically represented by a completely random measure  
 808 (Kingman, 1967), which is defined as follows.

809 **▼ Definition 2.42 — completely random measure**

810 A **completely random measure** is a random measure  $\mu : \Omega \times X \rightarrow [0, +\infty]$  from  
 811 probability space  $(\Omega, \mathbb{F}, \mathbb{P})$  to measurable space  $(X, \Sigma)$  with

- 812
- 813 ◦ for any collection of disjoint sets  $A_1, \dots, A_k \in \Sigma$  and  $A_i \cap A_j = \emptyset$  for  $i \neq j$  a mutual  
 814 independency between  $\mu(A_1), \dots, \mu(A_k)$ .
- 815

816 Kingman (1967) shows that a completely random measure can be decomposed into three  
 817 components:

- 818 1. a deterministic function;
- 819 2. a countable set of non-negative random masses at deterministic locations;
- 820 3. a countable set of non-negative random masses at random locations.

The first component is a deterministic function. The second component has non-negative random masses, also called atoms, on deterministic locations. The third component is the one of interest. It has a set of random masses (atoms) that can be represented as a Poisson random measure on  $\mathbb{R}^+ \otimes X$  with mean measure  $\nu$  which is known as the Lévy intensity measure (Favaro et al., 2013).

**Table 2.2:** Lévy measure of the Beta Process (Wang and Carin, 2012), Gamma Process (Knowles et al., 2014), the Dirichlet Process (Lijoi and Prünster, 2010) (indirectly through  $F = 1 - e^{-\nu}$ ).

Random Process	Lévy measure
Beta Process	$\nu(da, dw) = H(da)\alpha w^{-1}(1-w)^{\alpha-1}dw$
Gamma Process	$\nu(da, dw) = H(da)w^{-1}e^{-\alpha w}dw$
Dirichlet Process	$\nu(da, dw) = H(da)\frac{e^{-w\alpha(x, \infty)}}{1-e^{-w}}dw$

For Lévy measure decompositions of other processes such as the Indian Buffet Process, we refer to Wang and Carin (2012).

Here we recall definition 2.30 for exchangeable sequences. De Finetti's theorem states that there is parameter  $\theta$  such that the data  $x_i$  is conditionally independent given this parameter for exchangeable sequences (cf. De Finetti, 1937).

**▼ Definition 2.43 — De Finetti's theorem**

A sequence  $\{x_0, x_1, \dots\}$  of  $(X, \Sigma_X)$ -valued random variables is an infinitely exchangeable sequence if and only if there exist a measure  $\mu(d\theta)$  on  $\theta$  such that

$$p(x_0, \dots, x_{k-1}) = \int_{\Sigma_X(X)} \prod_{i=0}^{k-1} p(x_i|\theta) \mu(d\theta) \quad \forall k \geq 1 \quad (2.13)$$

In words, de Finetti's theorem states that if we have exchangeable data, we have a parameter  $\theta$ , a likelihood  $p(x|\theta)$ , and some measure  $\mu$  on  $\theta$ , such that the data  $(x_0, \dots, x_{k-1})$  is conditionally independent.

The theorem is not limited to exchangeable *sequences*. In contrast, there are similar theorems for other exchangeable objects (Orbanz and Roy, 2015). Five examples (see Table 2.3) of exchangeable structures that have a theorem that describes an underlying measure that can be sampled i.i.d. are: (1) exchangeable sequences (de Finetti, 1930), (2) increments (Bühlmann, 1960), (3) partitions (Kingman, 1978), (4) arrays (Aldous, 1981), and (5) Markov chains (Diaconis and Freedman, 1980).

The *stick-breaking presentation* (Freedman and Diaconis, 1983) or residual allocation model (Hoppe, 1986; Sawyer and Hartl, 1985)



**Table 2.3:** Five exchangeable structures and their theorems.

Mathematical Object	Theorem
Exchangeable Sequence	de Finetti
Exchangeable Increment	Bühlmann
Exchangeable Partition	Kingman
Exchangeable Array	Aldous-Hoover
Exchangeable Markov Chain	Diaconis-Freedman

**▼ Definition 2.44**

An infinite sequence of random variables  $\phi = \{\phi_0, \phi_1, \dots\}$  has a **stick-breaking presentation** with parameters  $\alpha$  and  $\beta$  denoted by  $\phi \sim GEM(\alpha, \beta)$ .

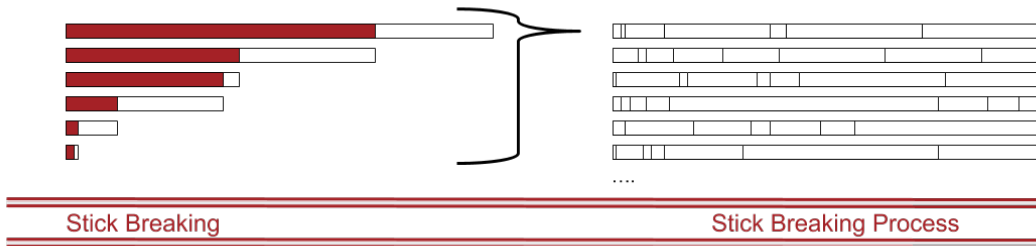
$$w_k \stackrel{i.i.d.}{\sim} Beta(1 - \beta, \alpha + k\beta) \quad k = 1, \dots, K \quad (2.14)$$

$$\phi_k = w_k \prod_{i=1}^{k-1} (1 - w_i) \quad (2.15)$$

848

849

850 The stick-breaking presentation samples repeatedly from a  $Beta(1 - \beta, \alpha + k\beta)$  distribution.  
 851 The result of the process is a vector of  $k$  weights  $\phi_k$ . The abbreviation *GEM* stands for  
 852 Griffiths, Engen, and McCloskey (Ewens, 1990; Ethier, 1990). There is also a variant of  
 853 GEM with a single parameter  $\alpha$  which can be obtained by setting  $\beta = 0$ . In that case  $w_k$  are  
 854 drawn from a  $Beta(1, \alpha)$  distribution.



**Figure 2.4:** The stick-breaking presentation. Left: at the first row, the stick is broken at  $x_0$ , at the next rows the remaining part of the stick is broken  $x_i$  with  $i > 0$ . Only six iterations are shown. Right: samples of a stick-breaking process. The first row shows the stick ratios from the stick-breaking presentation at the left. The next rows show other samples from the same process.

855 Fig. 2.4 visualized the stick-breaking process. A stick of fixed length 1 gets broken at a  
 856 position sampled from a Beta distribution. An infinite number of times it is broken. A stick-  
 857 breaking process generates many of these broken sticks. The stick-breaking presentation is  
 858 convenient for extensions that allow the atoms to vary according to some other stochastic  
 859 process (Dunson et al., 2012) and for approximations by truncating the stick-breaking to a  
 860 limited number of times (Kurihara et al., 2007).

### 861 2.2.1 Beta Process

862 A Beta Process (Hjort, 1990) is a random process with a countably infinite collection of  
863 weighted atoms in a space  $(X, \mathbb{B})$  with weights that are in between  $[0, 1]$  (Hjort, 1990).

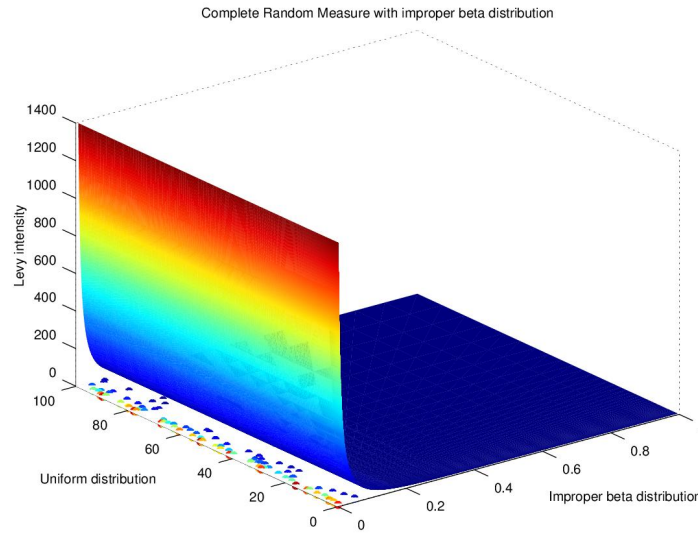
#### 864 ▼ Definition 2.45

865 Let  $(X, \mathbb{B})$  be a Borel space,  $\nu$  a finite measure, and  $\alpha > 0$  a scale parameter, then a  
866 **Beta Process** is a Lévy process on  $(X, \mathbb{B})$  with its Lévy measure  $\nu$  corresponding to the  
867 density:  
868

$$\nu(dw) = \alpha w^{-1}(1-w)^{\alpha-1}dw \quad (2.16)$$

869 with  $w > 0$ .

870



**Figure 2.5:** A Completely Random Measure with a Lévy intensity defined on the product space  $\Omega \otimes (0, 1)$ . Here  $\Omega$  is a bounded interval on which the base measure  $B_0 = U(0, 100)$  is defined. On  $(0, 1)$  we define an improper beta distribution  $\alpha w^{-1}(1-w)^{\alpha-1}$ . In this example  $\alpha = 10$ . This is how a Beta Process can be generated from a nonhomogeneous spatial Poisson point process. This has been visualized before (Jordan, 2010). The image is produced by rejection sampling using a homogeneous Poisson point process at  $\max(\nu)$  over  $w = [0.01, 0.9]$ . For  $w \rightarrow 0$  this maximum would go to  $\infty$  and all points would be rejected. Hence, the points should be denser for  $w$  around 0 and should be seen as an approximation of the actual process.

871 In Fig. 2.5 the Beta Process is generated from a Completely Random Measure with a Lévy  
872 intensity defined on  $\Omega \otimes (0, 1)$  (Thibaux and Jordan, 2007). In this case  $\Omega$  is the so-called base  
873 measure  $B_0$  and is assumed uniform over a bounded region. The  $(0, 1)$  space is equipped  
874 with an improper Beta distribution. It is called improper or degenerate because the scale  
875 parameter of the standard Beta distribution is set to zero. This has the consequence that the  
876 integral is infinite:  $\nu(\Omega \otimes (0, 1)) = \infty$ . It is due to the fact that the density  $w^{-1}(1-w)^{\alpha-1}$   
877 goes to infinity for  $w \rightarrow 0$ . That means that a countable infinite number of points can be  
878 obtained from the Poisson process.

879 The Beta Process has a sequential representation in the form of the Indian Buffet Process:

880 **▼ Definition 2.46**

881 

---

882 An **Indian Buffet Process** is a sequential process that is an exchangeable distribution  
883 over matrices:

$$p(z_{i,j} = k | z_{0,0}, \dots, z_{i-1,K_+}) = \begin{cases} \frac{n_{-i,k}}{i} & \text{if } k \leq K_+ \\ \frac{\lambda^{k_{new}} e^{-\lambda}}{k_{new}!} & \text{if } k > K_+ \end{cases} \quad (2.17)$$

884

885 

---

886 Here  $\lambda = \alpha/i$ ,  $k_{new} = K_+ - k$ . The  $i$ 'th data item samples an existing column with a probability  
887 of the number of times it has been sampled before divided by its index,  $n_{-i,k}/i$ . It samples  
888 a new column with a probability according to a Poisson distribution,  $\lambda^{k_{new}} e^{-\lambda} / k_{new}!$ . The  
889 conditional form of the sequential presentation describes a closed-form solution for Gibbs  
890 sampling, section 2.3.4 (?).

891 The Beta Process (BP) is used in linguistics (He et al., 2013; Vanhainen and Salvi, 2012),  
892 computer vision (Zhou et al., 2011; Gao and Sun, 2013),

893 **2.2.2 Gamma Process**

894 A Gamma process is a random process with independent gamma distributed increments  
895 (Ferguson, 1974), formally:

896 **▼ Definition 2.47**

897 

---

898 Let  $(X, \mathbb{B})$  be a Borel space,  $\nu$  a finite measure, and  $\alpha > 0$  a scale parameter, then a  
899 **Gamma Process** is a Lévy process on  $(X, \mathbb{B})$  with its Lévy measure  $\nu$  corresponding to  
900 the density:

$$\nu(dw) = w^{-1} e^{-\alpha w} dw \quad (2.18)$$

901 with  $w > 0$ .

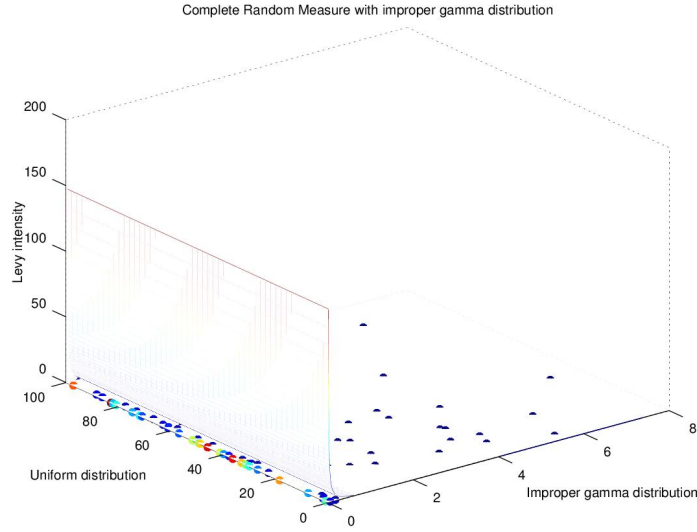
902 

---

903 The Gamma Process is used in risk theory (Dufresne et al., 1991), spatial statistics (Wolpert  
904 and Ickstadt, 1998; Rao and Teh, 2009), erosion (Singpurwalla, 1997; Abdel-Hameed, 2012),  
905 and finance (Madan and Seneta, 1990; Küchler and Tappe, 2008).

906 **2.2.3 Dirichlet Process**

907 The Dirichlet Process (DP) is, just as the Beta Process and the Gamma Process, a distribution  
908 over distributions (Ferguson, 1973).



**Figure 2.6:** A Completely Random Measure with a Lévy intensity defined on the product space  $\Omega \otimes \mathbb{R}$ . Here  $\Omega$  is a bounded interval on which the base measure  $B_0 = U(0, 100)$  is defined. On  $\mathbb{R}$  we define an improper gamma distribution  $w^{-1}e^{-\alpha w}$ . In this example  $\alpha = 1$ . This is how a Gamma Process can be generated from a nonhomogeneous spatial Poisson point process. The image is produced by rejection sampling in the same way as Fig. 2.5.

#### ▼ Definition 2.48

A **Dirichlet process**  $DP$  over a set  $S$  can be used to draw sample paths  $X$ :

$$X \sim DP(\alpha, H)$$

with  $\alpha$  the dispersion parameter and  $H$  a measure on  $S$  and for which any measurable partition  $\{B_0, \dots, B_{n-1}\} \in S$  is drawn from a Dirichlet distribution:

$$(X(B_0), \dots, X(B_{n-1})) \sim \text{Dirichlet}(\alpha H(B_0), \dots, \alpha H(B_{n-1}))$$

909

910

911 The Lévy intensity of the Dirichlet Process is complicated, because it is a so-called normalized  
912 process, see Regazzini et al. (2003).

913 De Finetti's theorem can be used to define the so-called Chinese Restaurant Process (CRP).  
914 This is a *distribution over partitions*. The Dirichlet Process has with the Chinese Restaurant  
915 Process a sequential representation:

916

#### ▼ Definition 2.49

917

918 A **Chinese Restaurant Process** is a sequential process that is an exchangeable distribu-  
919 tion over partitions:

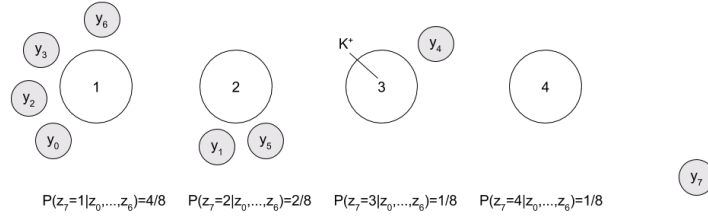
$$p(z_i = k | z_0, \dots, z_{i-1}) = \begin{cases} \frac{n_k}{\alpha + i} & \text{if } k \leq K_+ \\ \frac{\alpha}{\alpha + i} & \text{if } k > K_+ \end{cases} \quad (2.19)$$

920

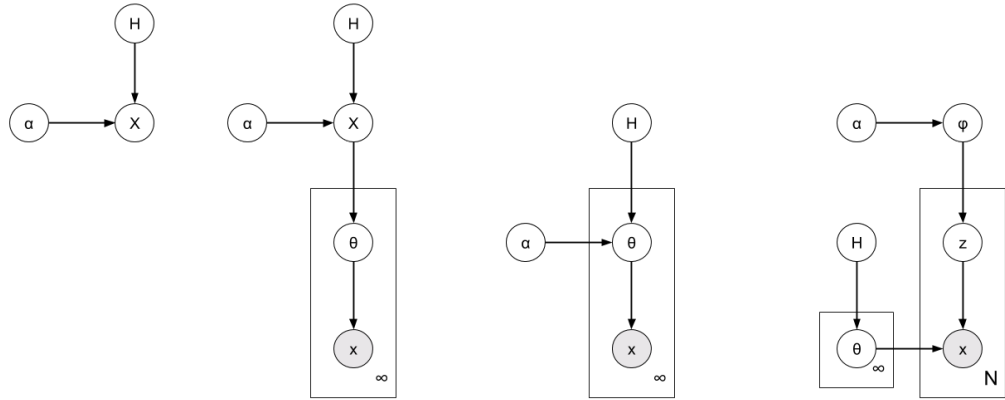
921

922 The conditional probability of a cluster assignment  $z_i$  for data item  $y_i$  given the cluster  
 923 assignments  $z_0, \dots, z_{i-1}$  is proportional to the number of data items  $n_k$  assigned to an existing  
 924 cluster  $k$ , or proportional to  $\alpha$  for a new cluster.

925 The Chinese Restaurant Process is visualized in Fig. 2.8.



**Figure 2.7:** The Chinese Restaurant Process with  $i$  customers already sitting down. A new customer  $y_{i=7}$  arrives and gets assigned,  $z_i$ . This is an existing table  $\{1, 2, 3\}$  with a probability proportional to the number of customers  $n_i$  sitting at that table:  $n_i/(\alpha + i)$ , or a new, empty table 4 with probability  $1/(\alpha + i)$ . In the visualized Chinese Restaurant Process the dispersion factor  $\alpha = 1$ .



Dirichlet Process

Dirichlet Process Mixture

Chinese Restaurant Process

Stick Breaking Process

**Figure 2.8:** The Dirichlet Process  $X \sim DP(\alpha, H)$ . The Dirichlet Mixture Model where  $X$  is a prior for a sum  $\sum_i w_i p(x_i | \theta_i)$ , a Chinese Restaurant Process with  $X$  marginalized out, and the Stick Breaking Process with a distribution over partition sizes  $\pi$  and indicator variables  $z_i$ .

### ▼ Definition 2.50

The **stick-breaking presentation** of the Dirichlet Process states that if

$$\phi_k \sim GEM(\alpha, 0) \quad (2.20)$$

$$\theta_k \sim H \quad (2.21)$$

$$G = \sum_{k=1}^{\infty} \phi_k \delta(\theta, \theta_k) \quad (2.22)$$

926 then  $G \sim DP(\alpha, H)$ .

927

928 The weights  $\phi_k$  are sampled from the stick-breaking process  $GEM(\alpha, 0)$ . The parameter  
 929 values  $\theta_k$  are sampled from the base measure  $H$ . To sample from the Dirichlet Process we  
 930 have to sample these parameters with the given weights.

931 If the stick-breaking process is used as a prior for a mixture, then the cluster assignments  $z_i$   
 932 are sampled according to the mixing proportions  $\phi$ :

$$\phi \sim GEM(\alpha, 0) \quad (2.23)$$

$$\theta_k \sim H \quad (2.24)$$

$$G = \sum_{k=1}^{\infty} \phi_k \delta(\theta, \theta_k) \quad (2.25)$$

$$z_i \sim Mult(\phi) \quad (2.26)$$

$$x_i \sim F(\theta_{z_i}) \quad (2.27)$$

933 Here  $\theta_k = \theta_{z_i}$  for observation with index  $i$  and cluster assignment  $k$ :  $z_i = k$ .

#### 934 2.2.4 Pitman-Yor Process

935 The Pitman-Yor Process introduces another parameter  $d$  with respect to the Dirichlet Process.  
 936 It has been developed by Pitman and Yor as the two-parameter Poisson-Dirichlet distribution  
 937 (Pitman and Yor, 1997). The Pitman-Yor Process has the following definition:

##### ▼ Definition 2.51

A Pitman-Yor process  $PY$  over a set  $S$  can be used to draw sample paths  $X$ :

$$X \sim PY(d, \alpha, H)$$

938 with  $\alpha > -d$  a strength parameter,  $0 \leq d < 1$  a discount parameter, and  $H$  a measure  
 939 on  $S$ .

940

941 The Pitman-Yor Process (PYP) generalizes the DP. The Pitman-Yor Process has a stick-  
 942 breaking presentation in which sticks are drawn from  $GEM(\alpha, \beta)$ . The Dirichlet Process  
 943 has a stick-breaking presentation in which sticks are drawn from  $GEM(\alpha, 0)$ , see Def. 2.50.

##### ▼ Definition 2.52

The **stick-breaking presentation** of the PYP states that if

$$\phi_k \sim GEM(\alpha, \beta) \quad (2.28)$$

$$\theta_k \sim H \quad (2.29)$$

$$G = \sum_{k=1}^{\infty} \phi_k \delta(\theta, \theta_k) \quad (2.30)$$

944 then  $G \sim PYP(\alpha, H)$ .

945

---

946 The Pitman-Yor Process is used in quite a few applications, such as language models (Teh  
947 et al., 2006), scene segmentation (Sudderth and Jordan, 2009), speech induction (Blunsom  
948 and Cohn, 2011), and time series (Bassetti et al., 2014).

### 949 2.2.5 Hierarchical Dirichlet Process

950 The Hierarchical Dirichlet Process (HDP) extends the Dirichlet Mixture Model with a hier-  
951 archical structure (Teh et al., 2006).

#### ▼ Definition 2.53

A Hierarchical Dirichlet process  $HDP$  over a set  $S$  can be used to draw sample paths  $X$ :

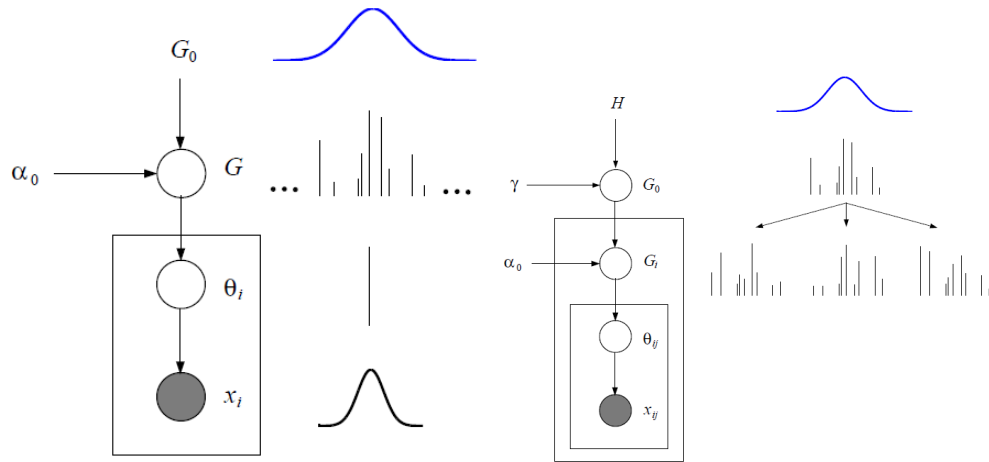
$$G_0 \sim DP(\gamma, H)$$

$$X_i \sim DP(\alpha_0, G_0) \text{ for each group } i$$

952 with a Dirichlet Process with a general  $\gamma$  dispersion parameter and base distribution  $H$   
953 as a measure on  $S$  of which the generated distributions  $G_0$  are used as base distribution  
954 for each group distribution  $X_i$ .

955

---



(a) Dirichlet Process Mixture Model. Each draw from (b) Hierarchical Dirichlet Process. Observe that the local process corresponds to a parameter. Each parameter is associated with a distribution (in this case a Gaussian). The only freedom left to express by  $G_i$  is in the weights of those atoms. This reflects a decomposition in a structural and non-structural component.

**Figure 2.9:** The difference visualized between a Dirichlet Process mixture and a Hierarchical Dirichlet Process. It illustrates also that the input of a Dirichlet process does not have to be a continuous function. If it is a continuous distribution it will become a discrete distributed almost surely. If it is a discrete distribution, it will have atoms at the locations where the discrete distribution had its probability mass concentrated.

956 The Hierarchical Dirichlet Process uses the outcome of a Dirichlet Process as a starting point  
 957 to define multiple distributions with atoms at the same locations, while they come equipped  
 958 with different weights. So, the Dirichlet Process on the lower level uses not a continuous  
 959 distribution as input, but a discrete one, generated by the DP at the top layer. Note, that  
 960 the Dirichlet Process will create an a.s. discrete distribution, but it can be fed a discrete  
 961 distribution as prior just fine.

## 962 2.3 Inference

963 There will be six inference methods described, all sampling methods. Section 2.3.1 describes  
 964 inverse transform sampling. Section 2.3.2 describes rejection sampling. Section 2.3.3 de-  
 965 scribes approximate Bayesian computation. Section 2.3.4 describes Gibbs sampling. Sec-  
 966 tion 2.3.5 describes Metropolis-Hastings. Section 2.3.6 describes Split-Merge MCMC.

### 967 2.3.1 Inverse Transform Sampling

968 Let  $p_f(x)$  and  $p_g(x)$  be two probability distributions with  $p_f(x) + p_g(x) = 1$ . Sample with  
 969 probability  $p_f(x)$  by drawing from a uniform distribution  $u \sim U(0, 1)$ , then if  $u < p_f(x)$ ,  
 970 accept  $x$ , if not reject  $x$ . This can be generalized to more than two probability distributions.

971 Let  $p_f(x)$ ,  $p_g(x)$ ,  $p_h(x)$  be multiple probability distributions. Draw  $u$  from the uniform distri-  
 972 bution  $U(0, 1)$ . If  $u < p_f(x)$  we select  $f$ , else if  $u < p_f(x) + p_g(x)$  we select  $g$ , and otherwise



we select  $g$ . The procedure is called inverse transform sampling. The term “inverse” stems from the fact that we return  $x$  (or  $f(x)$ ) given  $u$ . Inverse transform sampling is common to a lot of sampling methods. Often when there are samples from a uniform distribution this is an inverse transform sampling step.

### 2.3.2 Rejection Sampling

Let  $f(x)$  be a complicated function from which it is hard to take samples. Let  $g(x)$  be a simple function that is easy to sample from. Then we can sample from  $f(x)$  by making sure  $Mg(x) > f(x)$ :

$$X = S(f(x), g(x)) \quad (2.31)$$

The rejection sampling method (Halperin and Burrows, 1960) for  $f(x)$  is described in algorithm 1:

---

**Algorithm 1** Rejection sampling for  $f(x)$

---

```

1: procedure REJECTION SAMPLING( $f(x), g(x)$ )      ▷ Target and proposal distribution.
2:   for  $t = 1 \rightarrow T$  do
3:      $x^t \sim g(x)$                                 ▷ Generate  $x^t$  from  $g(x)$ 
4:      $u \sim U(0, 1)$                                 ▷ Inverse transform sampling
5:      $p_0 = f(x)/(Mg(x))$ 
6:     if  $u < p_0$  then
7:        $X = X \cup x^t$                                 ▷ Accept
8:     end if
9:   end for
10:  return  $X$                                           ▷  $X$  will have the distribution of  $f(x)$ 
11: end procedure

```

---

We can also use rejection sampling to sample from the posterior  $f(\theta|x)$  given that we know the form of the likelihood function and that we can sample from the prior. We know that we can sample from the posterior by sampling from  $p(\theta)p(x|\theta)$ . We also know that the prior  $p(\theta)$  necessarily has to be larger than  $p(\theta)p(x|\theta)$  for any observation, because  $p(x|\theta)$  is a probability, hence smaller than one.

$$\Theta = S(\sim p(\theta), p(x|\theta), x) \quad (2.32)$$

In our notation we make explicit that we need  $p(x|\theta)$  for each combination of observations and parameters, but that we only need to sample  $\sim$  from the prior  $p(\theta)$ .

**Algorithm 2** Rejection sampling for  $f(\theta | x)$ 


---

```

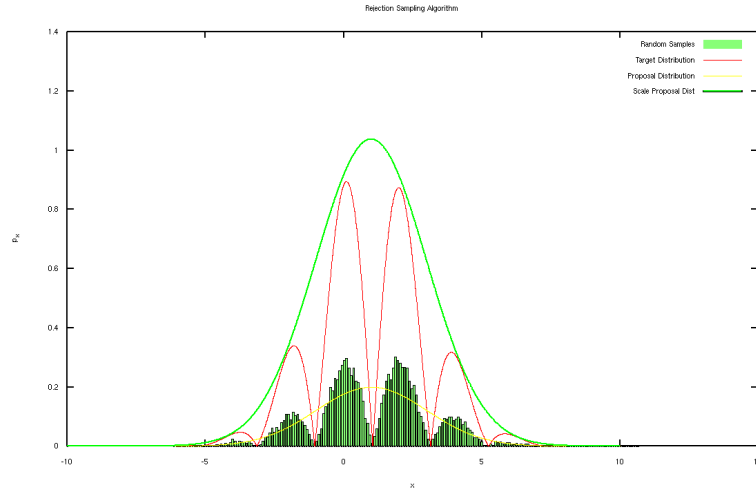
1: procedure REJECTION SAMPLING( $p(\theta), p(x|\theta), x$ )      ▷ Requires prior, likelihood and
   observations.
2:   for  $t = 1 \rightarrow T$  do
3:      $\theta^t \sim p(\theta)$                                 ▷ Generate  $\theta^t$  from prior
4:      $u \sim U(0, 1)$                                     ▷ Inverse transform sampling
5:      $p_0 = p(x | \theta)$ 
6:     if  $u < p_0$  then
7:        $\Theta = \Theta \cup \theta^t$                             ▷ Accept
8:     end if
9:   end for
10:  return  $\Theta$                                           ▷  $\Theta$  will have the distribution of  $f(\theta|x)$ 
11: end procedure

```

---

990 In algorithm 2 the envelope distribution  $p(\theta)$  and the target distribution  $p(\theta)p(x | \theta)$ , cancel  
 991 in such way that only  $p(x | \theta)$  remains.

992 Most examples illustrate rejection sampling by estimating the area of a circle, but let us  
 993 visualize the method in the context of sampling (Fig. 2.10).



**Figure 2.10:** A Gaussian is placed over the complex target probability density function. Subsequently the samples that fall in between these two ‘envelopes’ are rejected. This results in a sampling scheme that follows exactly the more complicated probability density function. Note that if the function is scaled by a factor, the sampling scheme stays the same. Such a scaling factor is only important if we want, for example, to know the area under the graph.

### 994 2.3.3 Approximate Bayesian Computation

995 In Approximate Bayesian Computation (ABC) (Rubin and Others, 1984) the likelihood func-  
 996 tion does not need to be calculated<sup>2</sup> (Sisson and Fan, 2011). In contrast, it is assumed that

<sup>2</sup>ABC is also called likelihood-free computation

997 there is a model available that allows to generate observations given the (searched for) pa-  
 998 rameters. In ABC for each configuration of parameters a set of observations is generated.

$$\Theta = S(\sim p(\theta), X, \sim M(\theta), d(X^t, X), \epsilon) \quad (2.33)$$

999 Approximate Bayesian computation uses many tuning parameters. Its most salient charac-  
 1000 teristic though is that it generates pseudo-observations through  $\sim M(\theta)$ .

---

**Algorithm 3** Approximate Bayesian computation

---

```

1: procedure APPROXIMATE BAYESIAN COMPUTATION( $p(\theta), X, M, d, \epsilon$ )    ▷ Requires prior,
   observations, model, distance function, and threshold.
2:   for  $t = 1 \rightarrow T$  do
3:      $\theta^t \sim p(\theta)$                                               ▷ Generate  $\theta$  from prior
4:      $X^t \sim M(\theta)$                                               ▷ Simulate observations  $X^t$  from model  $M$ 
5:      $\rho = d(X^t, X)$     ▷ Calculate distance between simulated and actual observations
6:     if  $\rho \leq \epsilon$  then
7:        $\Theta = \Theta \cup \theta^t$     ▷ Accept  $\theta^t$  if distance falls under threshold  $\epsilon$ .
8:     end if
9:   end for
10:  return  $\Theta$     ▷  $\Theta$  will have the distribution of  $f(\theta|X)$ 
11: end procedure

```

---

1001 The term Bayesian reflects the fact that a prior is involved. The weight of this prior can be  
 1002 manipulated by the threshold  $\epsilon$ . If this threshold is set very low, the prior plays no role and  
 1003 only observations are taken into account. If  $\epsilon$  is set extremely high, all  $\theta$  coming from the  
 1004 prior will be accepted, and the actual observations are not used in the process. There are  
 1005 several disadvantages to approximate Bayesian computation:

- 1006     ◦ A set of simulated observations has to be compared with the actual observations. This  
 1007       becomes unwieldly if there are many observations.
- 1008     ◦ It is possible to use summary statistics rather than the observations themselves. If  
 1009       these are sufficient statistics there will be no information lost. If not, there will be  
 1010       information loss in practice
- 1011     ◦ The distance function suffers from the curse of dimensionality. In the case that the  
 1012       dimensionality of the individual observations becomes high, or the number of param-  
 1013       eters becomes large, it gets increasingly difficult to come up with a distance function  
 1014       which is efficient and accurate at the same time.

### 1015 2.3.4 Gibbs Sampling

1016 Gibbs sampling (Geman and Geman, 1984) is similar to the *coordinate descent* optimization  
 1017 algorithm (Wright, 2015). In coordinate descent a local minimum of a function is found by  
 1018 iteratively performing a line search along one coordinate direction at a time. Gibbs sampling

1019 is named after the physicist Gibbs and belongs to the family of Markov chain Monte Carlo  
 1020 (MCMC) methods. Gibbs sampling optimizes over one variate in the multivariate probability  
 1021 distribution at a time. The update value is set and fixed. Then, the next variate is chosen in  
 1022 a round-robin like manner.

$$\Theta = S(X, \sim p(\theta_i | \theta_{-i}, X), \sim p(\theta), B) \quad (2.34)$$

1023 The multiple parameters in the multivariate probability distribution are denoted by  $\theta$ . The  
 1024 parameters are denoted individually with  $\theta_i$ . The set of all parameters except for  $i$  is denoted  
 1025 by  $\theta_{-i}$ . If we sample a parameter we write  $\theta^t$  with  $t$  the iteration or sampling round. The  
 1026 set of parameter samples has capital letter  $\Theta$ .

---

**Algorithm 4** Gibbs sampling
 

---

```

1: procedure GIBBS SAMPLING( $p(\theta_i | \theta_{-i}, X), p(\theta), X, B$ )           ▷ Requires parameters,
   observations and burn-in.
2:    $\theta^0 \sim p(\theta)$                                            ▷ Set parameters to some initial value
3:   for  $t = 1 \rightarrow T$  do
4:     for  $i = 1 \rightarrow k$  do
5:        $\theta_i^t \sim p(\theta_i^{t-1} | \theta_{-i}^t, X)$            ▷ Generate  $\theta_i^t$  from the full conditional probability
6:     end for
7:      $\Theta = \Theta \cup \theta^t$ 
8:   end for
9:    $\Theta_{B:T} \in \Theta$                                            ▷ Get  $\Theta_T$  set, from burn-in  $B$  to end of run  $T$ 
10:   $\Theta \sim \Theta_{B:T}$                                            ▷ Sample  $\Theta$  from correlated  $\Theta_{B:T}$ 
11:  return  $\Theta$ 
12: end procedure
  
```

---

1027 Gibbs samples are Markovian, the conditional probability only takes into account values at  
 1028 the previous time step  $t - 1$ . When running the Gibbs sampling algorithm long enough,  
 1029 it will visit all possible states eventually. The Markovian property makes subsequent steps  
 1030 correlated, hence when finally extracting the parameter probabilities, it is important not to  
 1031 consider subsequent steps. It is also important to run the algorithm for a while, so it does not  
 1032 suffer from a bad choice of initial parameter values, this is called burn-in. Gibbs sampling  
 1033 works properly because the time the algorithm spends in parts of the space is directly related  
 1034 to the probability of getting into that part of the space.

1035 In the physics literature Gibbs sampling is known as Glauber dynamics or the heat bath  
 1036 algorithm. Observe that Gibbs sampling does not necessary require an actual calculation of  
 1037 the conditional probability in all cases. The obvious exception is for the observations, which  
 1038 are already known. Another, neat optimization procedure arises when conjugate priors are  
 1039 used. A conjugate prior leads to a posterior distribution that can be described analytically.  
 1040 In such a case it is computationally unnecessary to perform sampling. It is much faster to use  
 1041 the actual available analytic description. This is commonly called collapsed Gibbs sampling.

### 1042 2.3.5 Metropolis-Hastings Sampling

1043 Another MCMC algorithm is Metropolis-Hastings (Metropolis et al., 1953), likewise used for  
 1044 high-dimensional distributions. Metropolis-Hastings calculates an acceptance factor  $\alpha$  which  
 1045 takes into account if a step should be taken according to a predefined proposal distribution.  
 1046 In case this step is not accepted, the current sample is resampled.

$$\Theta = S(X, \theta^0, Q(\theta^{t+1}|\theta^t), f(\theta, X)) \quad (2.35)$$

1047 Here we need  $Q(\theta^{t+1}|\theta^t)$  explicitly as well as samples from it.

---

#### Algorithm 5 Metropolis-Hastings sampling

---

```

1: procedure METROPOLIS-HASTINGS SAMPLING( $\theta^0, X, Q, f$ )  $\triangleright$  Requires initial parameters,
   observations, proposal distribution, and function proportional to desired distribution
2:   for  $t = 1 \rightarrow T$  do
3:      $\theta^{t+1} \sim Q(\theta^{t+1}|\theta^t)$   $\triangleright$  Sample from proposal distribution  $Q$ 
4:      $\alpha = \frac{f(\theta^{t+1}, X^{t+1})Q(\theta^t|\theta^{t+1})}{f(\theta^t, X^t)Q(\theta^{t+1}|\theta^t)}$   $\triangleright$  Calculate acceptance
5:      $u \sim U(0, 1)$   $\triangleright$  Inverse transform sampling
6:     if  $\alpha > u$  then
7:        $\Theta = \Theta \cup \theta^{t+1}$   $\triangleright$  Accept  $\theta^{t+1}$ 
8:     else
9:        $\Theta = \Theta \cup \theta^t$   $\triangleright$  Reuse previous sample (note, different from rejection)
10:    end if
11:  end for
12:  return  $\Theta$   $\triangleright$   $\Theta$  will be samples from the distribution  $f(\theta|x)$ 
13: end procedure

```

---

1048 A particular choice of a Metropolis-Hastings step is that of a proposal distribution that does  
 1049 not depend on the state of the chain. This is already suggested by Hastings and is called the  
 1050 independence sampler.

### 1051 2.3.6 Split-Merge MCMC Sampling

1052 The discussed sampling methods do not assume much structure in the model. This means  
 1053 that in hierarchical models sampling either occurs through updating the to be estimated  
 1054 quantities observation by observation or cluster by cluster. Splitting a cluster in two requires  
 1055 moving data points one by one to this new cluster. Much more efficient sampling methods  
 1056 can be obtained if we would be able to reason about larger chunks of cluster assignments.

1057 Split-merge sampling can update cluster assignments for multiple observations at once. It is  
 1058 an adaptation of acceptance method in the Metropolis-Hastings algorithm.

**Algorithm 6** Split-Merge MCMC sampling

---

```

1: procedure SPLIT-MERGE MCMC SAMPLING( $\theta^0, X, Q, f$ )    ▷ Requires initial parameters,
   observations, proposal distribution, and function proportional to desired distribution
2:   for  $t = 1 \rightarrow T$  do
3:      $i \sim D(0, N - 1)$                                 ▷ Sample observation  $i$  discretely
4:      $j \sim D(0, N - 1)$                                 ▷ Sample observation  $j$  discretely
5:     if  $c_i == c_j$  then
6:        $c_{old} = c_i$ 
7:        $\theta_{c_{new}}^{t+1} \sim Q(\theta^{t+1} | \theta^t)$           ▷ Sample from proposal distribution  $Q$ 
8:       for  $k \in c_{old}$  do
9:          $c_k \sim C(c_{old}, c_{new})$                     ▷ Assign to new cluster categorically
10:      end for
11:     else
12:        $c_{merge} = c_i$ 
13:       for  $k \in c_j$  do
14:          $c_k = c_{merge}$                                 ▷ Assign all to first cluster
15:       end for
16:     end if
17:      $\alpha = \frac{f(\theta^{t+1}, X^{t+1})Q(\theta^{t+1} | \theta^t)}{f(\theta^t, X^t)Q(\theta^t | \theta^{t+1})}$           ▷ Calculate acceptance
18:      $u \sim U(0, 1)$                                     ▷ Inverse transform sampling
19:     if  $\alpha > u$  then
20:        $\Theta = \Theta \cup \theta^{t+1}$                         ▷ Accept  $\theta^{t+1}$ 
21:     else
22:        $\Theta = \Theta \cup \theta^t$                             ▷ Reuse previous sample (note, different from rejection)
23:     end if
24:   end for
25:   return  $\Theta$                                           ▷  $\Theta$  will be samples from the distribution  $f(\theta|x)$ 
26: end procedure

```

---

1059 The exact acceptance probability depends on the model. For the mixture model with a  
 1060 Dirichlet Process as prior, its performance is further improved by adjusting the assignment  
 1061 process from random to observation-supported by introducing intermediate restricted Gibbs  
 1062 sampling steps (Jain and Neal, 2004, 2007). Similarly, there are other variants that incor-  
 1063 porate data fit to the splitting step. Labels can for example be calculated sequentially (Dahl,  
 1064 2003) or methods can be used that postulate subcluster structure within clusters to stream-  
 1065 line reasoning over split and merge sets (?).

## 1066 2.4 Chapter Conclusions

1067 Chapter 3 describes Gibbs sampling to perform inference over an infinite set of lines. Gibbs  
 1068 sampling requires conditional probabilities. These are given in closed-form because there  
 1069 is a conjugate description of the line parameters given the points that form the lines in this  
 1070 application.

1071 Chapter 4 describes Split-Merge MCMC sampling to perform inference over an infinite set  
1072 of line segments. The parameters for line segments do not have a conjugate description.  
1073 Metropolis-Hastings can be used to perform inference over the line segments, but the search  
1074 space is quite large. The Split-Merge MCMC method performs faster inference than Metropolis-  
1075 Hastings because it is able to split and merge line segments (with multiple points ascribed  
1076 to them) at once.





# NONPARAMETRIC BAYESIAN LINE DETECTION

## Contents

The nonparametric Bayesian models from the literature (Chapter 2) can be applied to perform inference over point clouds. An example of a point cloud are points distributed over lines in a two-dimensional space. Traditionally, RANSAC and the Hough transform have been used to perform inference over such lines. This chapter uses a nonparametric Bayesian model to perform inference over a countably infinite number of lines. Given a prior with respect to the noise and distribution of points over the lines, Bayesian inference describes the optimal procedure to perform line fitting.

## Published in

A.C. van Rossum, H.X. Lin, J. Dubbeldam, and H.J. van den Herik. Nonparametric Bayesian Line Detection. *International Conference on Pattern Recognition and Methods*, ICPRAM 2016, Rome, Italy, February 24-26, 2016. Best paper award in theory and methods track.

A.C. van Rossum, H.X. Lin, J. Dubbeldam, and H.J. van den Herik. *Fundamentals of Nonparametric Bayesian Line Detection*. Springer, 2017.

## Outline

The infinite line model describes a collection of lines with a Dirichlet Process as prior (Sect. 3.1). Inference in the infinite line model is performed through Gibbs sampling. (Sect. 3.2). Gibbs sampling over parameters converges slowly. It can be accelerated through sampling over clusters (Sect. 3.3). The inference method results are measured using clustering performance measures (Sect. 3.4). The chapter summarizes the findings (Sect. 3.5) and introduces extensions which will be handled in the next chapters.

In computer vision and particularly in robotics, traditionally the task of line detection has been performed through sophisticated, but ad-hoc methods. We will give two examples

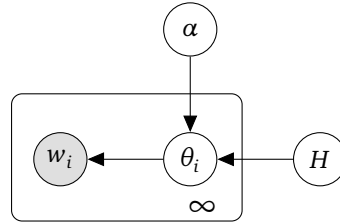
of such methods. RANSAC (Bolles and Fischler, 1981) is a method that iteratively tests a hypothesis. A line is fitted through a subset of points. Then other points that are in consensus with this line (according to a certain loss function) are added to the subset. This procedure is repeated till a certain performance level is obtained. The Hough transform (Hough, 1962) is a deterministic approach which maps points in the image space to curves in the so-called Hough space of slopes and intercepts. A line is extracted by getting the maximum in the Hough space.

There are four main problems with these methods. First, the extension of RANSAC or Hough to the detection of multiple lines is nontrivial (Zhang and Kősecká, 2007; Gallo et al., 2011; Chen et al., 2001). Second, the noise level is hardcoded into model parameters and it is not possible to incorporate knowledge about the nature of the noise. Third, it is hard to extend the model to hierarchical forms, for example, to lines that form more complicated structures such as squares or volumetric forms. Fourth, there are no results known with respect to any form of optimality of the mentioned algorithms.

In this chapter we postulate a method to perform inference over the number of lines and over the fitting of points on that line using the nonparametric Bayesian methods from chapter 2.

### 3.1 Infinite Line Model

The Dirichlet Process described as prior for a mixture distribution (Fig. 2.8 in section 2.2.3) can be used in this particular case as a prior for the distribution of points over a countably infinite set of lines.



**Figure 3.1:** The Infinite Line Model in the Chinese Restaurant Process representation (compare with Fig. 2.8). Top:  $\alpha$ , the concentration parameter of the Dirichlet Process. Bottom, left to right:  $w_i$ , the observation, an individual point in a 2D space;  $\theta_i$ , the parameters (intercept, slope) of the line belonging to observation  $w_i$ ;  $H$ , the base distribution from which line parameter values are sampled.

The infinite line model is visualized (Fig. 3.2) through plate notation (Buntine, 1994). In section 3.1.1 it is described how  $\theta_i$  is sampled from  $H$  and  $\alpha$ . In section 3.1.2 it is described how  $w_i$  is sampled from  $\theta_i$ . In section 3.1.3 the prior  $H$  for  $\theta_i$  is described.

#### 3.1.1 Posterior Predictive for a Line given Other Lines

Let us reiterate the definition of the Dirichlet Process. Let  $H$  be a distribution over  $\Theta$ , let  $\alpha$  be scalar. The Dirichlet Process generates a distribution  $G \sim DP(\alpha, H)$ :

$$G(\theta_1, \dots, \theta_\infty) \sim DP(\alpha, H(\theta_1, \dots, \theta_\infty)) \quad (3.1)$$

1131 A Dirichlet Process assigns a Dirichlet distribution to every parameter partition  $\Theta_1, \dots, \Theta_r$ :

$$(G(\Theta_1), \dots, G(\Theta_r)) \sim Dir(\alpha H(\Theta_1), \dots, \alpha H(\Theta_r)) \quad (3.2)$$

1132 The Dirichlet is conjugate to the categorical:

$$(G(\Theta_1), \dots, G(\Theta_r)) \mid \theta_1, \dots, \theta_n \sim Dir(\alpha H(\Theta_1) + n_1, \dots, \alpha H(\Theta_r) + n_r) \quad (3.3)$$

$$n_k = \sum_{j=1}^n \delta_{\theta_j}(\Theta_k)$$

1133 In the above notation,  $\delta_{\theta_j}(\Theta_k)$  is a Dirac measure (a generalization of the Dirac delta func-  
 1134 tion), also known as an indicator function. Given a set  $\Theta_k$  with a  $\sigma$ -algebra over subsets of  
 1135  $\Theta$ :

$$\delta_{\theta_j}(\Theta_k) = 1_{\Theta_k}(\theta_j) = \begin{cases} 1 & \text{if } \theta_j \in \Theta_k \\ 0 & \text{if } \theta_j \notin \Theta_k \end{cases} \quad (3.4)$$

1136 The posterior for the Dirichlet Process base distribution and dispersion parameter is a Dirich-  
 1137 let Process with adjusted parameters:

$$G(\cdot) \mid \theta_1, \dots, \theta_n \sim DP\left(\alpha + n, \frac{\alpha}{\alpha + n} H(\cdot) + \frac{n}{\alpha + n} \frac{\sum_{j=1}^n \delta_{\theta_j}(\cdot)}{n}\right) \quad (3.5)$$

1138 The posterior base distribution  $G$  is a weighted average between the prior base distribution  
 1139  $H$  and the empirical distribution  $n^{-1} \sum_{j=1}^n \delta_{\theta_j}$  with the weights respectively  $\alpha$  and  $n$  (nor-  
 1140 malized). The dispersion parameter  $\alpha$  is updated to  $\alpha + n$ . Note that  $\delta_{\theta_j}(\cdot)$  is a distribution,  
 1141 the Dirac measure Eq. 3.4.

1142 The posterior predictive for a new parameter  $\theta_{n+1}$  has the form:

$$P(\theta_{n+1} \in \Theta_k \mid \theta_1, \dots, \theta_n) = \frac{1}{\alpha + n} \left( \alpha H(\Theta_k) + \sum_{j=1}^n \delta_{\theta_j}(\Theta_k) \right) \quad (3.6)$$

1143 In other words, the posterior predictive of  $\theta_{n+1}$  given the parameters  $\theta_1, \dots, \theta_n$  in Eq. 3.6 has  
 1144 exactly the same form as the posterior base distribution  $G$  given the parameters  $\theta_1, \dots, \theta_n$   
 1145 (Blackwell and MacQueen, 1973) in Eq. 3.5, namely:

$$\theta_{n+1} \mid \theta_1, \dots, \theta_n \sim \frac{1}{\alpha + n} \left( \alpha H(\theta_{n+1}) + \sum_{j=1}^n \delta(\theta_j - \theta_{n+1}) \right) \quad (3.7)$$

1146 A normal Dirac delta function  $\delta(\theta_j - \theta_{n+1})$  can be used here, which is only non-zero when  
 1147  $\theta_j$  is equal to  $\theta_{n+1}$ .

1148 Equivalently, if we describe  $\theta_n$  conditioned on  $\theta_1, \dots, \theta_{n-1}$  we have to run over  $n - 1$  rather  
 1149 than  $n$  parameters:

$$\theta_n \mid \theta_1, \dots, \theta_{n-1} \sim \frac{1}{\alpha + n - 1} \left( \alpha H(\theta_n) + \sum_{j=1}^{n-1} \delta(\theta_j - \theta_n) \right) \quad (3.8)$$

1150 Due to the exchangeability property we can also consider any other parameter update (Neal,  
 1151 2000):

$$\theta_i \mid \theta_{-i} \sim \frac{1}{\alpha + n - 1} \left( \alpha H(\theta_i) + \sum_{j \neq i} \delta(\theta_j - \theta_i) \right) \quad (3.9)$$

1152 The notation  $\theta_{-i}$  means every parameter  $\theta$  except for the one equal to  $\theta_i$ .

### 1153 3.1.2 Likelihood of Data given a Line

1154 The likelihood of data given line parameters is defined to be according to the **Bayesian linear**  
 1155 **regression** model. The Bayesian linear regression model for a single line (Box and Tiao,  
 1156 2011) assumes a linear relationship between the independent  $x_i$  and dependent variables  
 1157  $y_i$  with Gaussian noise added in the  $y$ -direction. The individual points  $i$  are drawn from a  
 1158 Normal distribution:

$$y_i \sim \mathcal{N}(x_i \beta, \sigma^2) \quad (3.10)$$

1159 The (column) vector  $\beta$  maps the (row) vector with independent variables  $x_i$  to the depen-  
 1160 dent variable  $y_i$ . The noise is normally distributed with standard deviation  $\sigma$  along the  
 1161 dimension of the dependent variable.

1162 In a 2D point cloud the point  $p$  is represented by  $(x_p, y_p)$ . The points are mapped into an  
 1163 intercept-slope representation through defining  $X_i = [1, x_p]$  and  $y_i = y_p$ . The vector  $\beta$  will  
 1164 then contain the  $y$ -intercept as the first value, the slope as the second value.

1165 All observations that belong to the same single line lead to a likelihood function that corre-  
 1166 sponds to a normally distributed random variable with  $y$  and  $X$  as parameters:

$$p(y \mid X, \beta, \sigma^2) \propto \sigma^{-n} \exp \left( -\frac{1}{2\sigma^2} (y - X\beta)^T (y - X\beta) \right) \quad (3.11)$$

1167 The dependent variable is now a column vector of values  $y$  and each observation has a row  
 1168 of independent variables in  $X$ . The vector  $\beta$  and the standard deviation  $\sigma$  are shared across  
 1169 all observations. The term  $y - X\beta$  is written out like this:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} - \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix} \quad (3.12)$$

1170 Note that Eq. 3.11 has exactly the same form for a single point or multiple points that belong  
 1171 to the same line. Hence, we have the probability of a point  $w_i$  given the line parameters  
 1172  $\theta_k = (\beta_k, \sigma_k)$ :

$$F(w_i, \theta_k) = p(w_i | \theta_k) = p(w_i | \beta_k, \sigma_k^2) = p(y_i | X_i, \beta_k, \sigma_k^2) \quad (3.13)$$

1173 To get the full distribution  $p(w_i, \beta, \sigma^2)$  we will need also  $p(\beta, \sigma^2)$ .

### 1174 3.1.3 Conjugate Prior for a Line

1175 The conjugate prior for the likelihood in Eq. 3.11 is a product of a prior for the standard  
 1176 deviation  $p(\sigma)$  and the conditional probability of the line coefficients given the standard  
 1177 deviation  $p(\beta | \sigma^2)$ .

$$p(\sigma^2, \beta) = p(\sigma^2)p(\beta | \sigma^2) \quad (3.14)$$

1178 The standard deviation  $\sigma$  is sampled from an Inverse-Gamma (IG) distribution:

$$p(\sigma) \propto (\sigma^2)^{-(\nu_0/2+1)} \exp\left(-\frac{1}{2\sigma^2} \nu_0 s_0^2\right) \quad (3.15)$$

1179 This is an  $IG(a_0, b_0)$  with  $a_0 = \nu_0/2$  and  $b_0 = 1/2 \nu_0 s_0^2$ . The conditional with respect to the  
 1180 line coefficients has a normal distribution as prior:

$$p(\beta | \sigma^2) \propto \sigma^{-n} \exp\left(-\frac{1}{2\sigma^2} (\beta - \mu_0)^T \Lambda_0 (\beta - \mu_0)\right) \quad (3.16)$$

1181 Let us collect  $\Lambda_0, \mu_0, a_0, b_0$  into  $\lambda_0$ , we have now a description of our base distribution  $H$ :

$$H(\theta_k) = NIG(\theta_k; \lambda_0) \quad (3.17)$$

1182 NIG is an abbreviation of the Normal-Inverse-Gamma distribution. The standard deviation  
 1183 is sampled from the Gamma distribution with  $a_0$  and  $b_0$  as hyperparameters and the line  
 1184 coefficients from a Normal distribution:

$$\begin{aligned} \sigma_k &= \tau_k^{-1/2} & \tau_k &\sim \mathcal{G}(a_0, b_0) \\ \mu_k &\sim \mathcal{N}(\mu_0, \sigma^2 \Lambda_0^{-1}) \end{aligned} \quad (3.18)$$

### 1185 3.1.4 Posterior Predictive for a Line given Data

1186 Due to the fact that it is a conjugate distribution we have a simplified description for updating  
 1187 the parameters at once, given a set of observations. The sufficient statistics are updated  
 1188 (Minka, 2000) according to:

$$\begin{aligned}\Lambda_n &= (X^T X + \Lambda_0) \\ \mu_n &= \Lambda_n^{-1}(\Lambda_0 \mu_0 + X^T y) \\ a_n &= a_0 + n/2 \\ b_n &= b_0 + 1/2(y^T y + \mu_0^T \Lambda_0 \mu_0 - \mu_n^T \Lambda_n \mu_n)\end{aligned}\tag{3.19}$$

1189 Let us collect  $\Lambda_0, \mu_0, a_0, b_0$  into  $\lambda_0$  and  $\Lambda_n, \mu_n, a_n, b_n$  into  $\lambda_n$ . Let us collect a set of our obser-  
 1190 vations and  $(X, y)_k$  into  $w_k$ . The update for the sufficient statistics can then be summarized  
 1191 as:

$$\lambda_n = U_{ss}(\lambda_0, w)\tag{3.20}$$

1192 If we combine this update with sampling  $\theta_k$  from  $\lambda_n$  according to Eq. 3.17, then we obtain:

$$p(\theta_k | \lambda_0, w_k) \propto F(w_k, \theta_k) H(\theta_k; \lambda_0) = p(\theta_k | \lambda_n) = NIG(\theta_k; \lambda_n)\tag{3.21}$$

1193 Sampling of  $NIG(\theta_k; \lambda_n)$  is as in Eq. 3.18, but with  $\lambda_n$  rather than  $\lambda_0$ .

1194 Let us integrate over  $\theta$  (through the function  $H$ ):

$$Q(w_k, \lambda_0) = \int_{\Theta} F(w_k, \theta) dH(\theta; \lambda_0)\tag{3.22}$$

## 1195 3.2 Inference for the Infinite Line Model

1196 The posterior predictive for parameters (see Eq. 3.9) combined with observations  $w_i$  is de-  
 1197 scribed by:

$$p(\theta_i | \theta_{-i}, w_i) \propto r_i H_i(\theta_i) + \sum_{j \neq i} L_{i,j} \delta(\theta_j - \theta_i)\tag{3.23}$$

1198 The posterior is proportional (indicated by  $\propto$ ) to three terms. First, the  $\alpha$  weighted posterior  
 1199 predictive  $r_i$  for a new cluster. Second, the posterior to sample from  $H_i(\theta_i)$  with probability  
 1200  $r_i$ . Third, the likelihood of an observation given a line  $L_{i,j}$ :

$$r_i = \alpha Q(w_i, \lambda_0) = \alpha \int_{\Theta} F(w_i, \theta) dH(\theta)\tag{3.24}$$

1201 The posterior  $H_i(\theta)$  is the normalized product of the prior distribution  $H(\theta_i)$  with the like-  
 1202 lihood  $F(w_i, \theta_i)$  for a single observation  $w_i$ .

$$H_i(\theta_i) \propto H(\theta_i)F(w_i, \theta_i) \quad (3.25)$$

$$L_{i,j} = F(w_i, \theta_j) \quad (3.26)$$

1203 Sampling a new cluster parameter from  $H_i(\theta_i)$  is done with probability:

$$p(\theta_{new}) = \frac{r_i}{r_i + \sum_{j \neq i} L_{i,j}} \quad (3.27)$$

1204 We can use this to derive the parameters  $\theta_i$ .

---

**Algorithm 7** Gibbs sampling over parameters  $\theta_i$

---

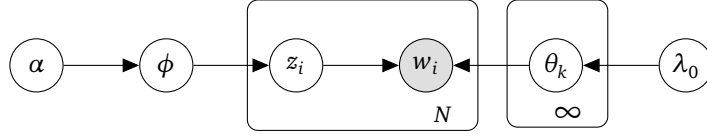
```

1: procedure GIBBS ALGORITHM 1( $w, \lambda_0, \alpha$ )           ▷ Accepts points  $w$ , hyperparameters  $\lambda_0, \alpha$  and
   returns  $k$  line coordinates
2:   for all  $t = 1 : T$  do
3:     for all  $i = 1 : N$  do
4:        $r_i = \alpha Q(w_i, \lambda_0)$            ▷ Posterior predictive of  $w_i$  given hyper parameters (Eq. 3.24)
5:       for all  $j = 1 : N, j \neq i$  do
6:          $L_{i,j} = F(w_i, \theta_j)$            ▷ Likelihood for a line given observation (Eq. 3.26)
7:       end for
8:        $p(\theta_{new}) = \frac{r_i}{r_i + \sum_{j \neq i} L_{i,j}}$            ▷ Probability of sampling a new parameter (Eq. 3.27)
9:        $u \sim U(0, 1)$ 
10:      if  $p(\theta_{new}) > u$  then                               ▷ Sample with probability  $p(\theta_{new})$ 
11:         $\lambda_n = U_{ss}(w_i, \lambda_0)$            ▷ Update sufficient statistics with  $w_i$  (Eq. 3.20)
12:         $\theta_i \sim NIG(\theta_i; \lambda_n)$            ▷ Sample  $\theta_i$  from NIG (Eq. ??)
13:      else
14:         $\theta_i$  sampled from existing clusters           ▷ Sample old cluster
15:      end if
16:    end for
17:  end for
18:  return summary on  $\theta_k$  for  $k$  lines
19: end procedure

```

---

1205 This Gibbs algorithm is described in its general form before (Neal, 2000) (Algorithm 1). We  
 1206 perform a loop in which for  $T$  iterations each  $\theta_i$  belonging to observation  $w_i$  is updated  
 1207 in sequence. First, the posterior predictive for  $w_i$  given the hyperparameters  $p(w_i | \lambda_0)$  is  
 1208 calculated. Second, the likelihood  $L_{i,j}$  for all  $\theta_j$  given  $w_i$  (with  $j \neq i$ ) is calculated. Third,  
 1209 the fraction with  $r_i$  defines if  $\theta_i$  will be sampled from a new cluster or if one of the existing  
 1210 clusters will be sampled. Fourth, a new cluster is sampled, the sufficient statistics are up-  
 1211 dated with information on  $w_i$  and thereafter  $\theta$  is sampled from a Normal-Inverse-Gamma  
 1212 distribution with the updated hyperparameters. That, or an existing cluster will be sampled.



**Figure 3.2:** The Infinite Line Model in the stick-breaking representation (compare with Fig. 2.8). From left to right:  $\alpha$ , the concentration parameter of the Dirichlet Process;  $(\phi_1, \dots, \phi_k)$ , the partition of points over lines;  $z_i$ , the assignment parameters that link observation  $w_i$  with line  $k$ ;  $w_i$ , the observation, an individual point with  $x$  and  $y$  coordinates;  $\theta_k$ , the parameters of line  $k$ ;  $\lambda_0$ , the base measure from which the line parameter values are sampled.

### 1213 3.3 Accelerating Inference for the Infinite Line Model

1214 It is also possible to iterate only over the clusters. The derivation takes a few steps (Neal,  
1215 2000) but leads to a simple update for the component indices that only depends on the  
1216 number of data items per cluster, the parameter  $\alpha$ , and the data.

---

#### Algorithm 8 Gibbs sampling over clusters $c_k$

---

```

1: procedure GIBBS ALGORITHM 2( $w, \lambda_0, \alpha$ )    ▷ Accepts points  $w$  and hyperparameters  $\lambda_0$  and  $\alpha$ ,
   returns  $k$  line coordinates
2:   for all  $t = 1 : T$  do
3:     for all  $i = 1 : N$  do
4:        $c = \text{cluster}(w_i)$                     ▷ Get cluster  $c$  currently assigned to observation  $w_i$ 
5:        $\lambda_c = \text{downdate}(w_i, \lambda_c)$         ▷ Adjust sufficient statistics for cluster  $c$  by removing
   observation  $w_i$ 
6:        $m_c = m_c - 1$                         ▷ Adjust cluster size  $m_c$  (observation  $w_i$  removed reduces it with one)
7:       for all  $k = 1 : K$  do
8:          $L_k = m_k F(w_i, \theta_k)$             ▷ Update likelihood for cluster  $k$  given observation  $w_i$ 
9:       end for
10:       $r_i = Q(w_i, \lambda_0)$                   ▷ Posterior predictive of  $w_i$  given hyper parameters
11:       $p(\text{new}) = \frac{r_i}{r_i + \sum_k L_k}$           ▷ Sample new or old?
12:      if  $p(\text{new})$  then
13:         $\lambda_k = U_{ss}(w_i, \lambda_0)$           ▷ Update sufficient statistics with observation  $w_i$ 
14:         $\theta_i \sim \text{NIG}(\lambda)$                 ▷ Sample  $\theta_i$  from NIG
15:      else
16:         $k$  sampled from existing clusters
17:         $\lambda_k = \text{update}(w_i, \lambda_k)$           ▷ Restore sufficient statistics with observation  $w_i$ 
18:      end if
19:       $m_k = m_k + 1$                         ▷ Increment cluster size  $m_k$ 
20:    end for
21:    for all  $k = 1 : K$  do
22:       $\theta_k \sim \text{NIG}(\lambda_k)$                 ▷ Sample  $\theta_k$  from NIG
23:    end for
24:  end for
25:  return summary on  $\theta_k$  for  $k$  lines
26: end procedure

```

---

1217 The probability to sample from a cluster depends on the number of items in that cluster (the  
1218 current data item excluded). This is expressed in equation 3.28.

$$p(c_i = c \text{ and } c_i = c_j \text{ and } i \neq j \mid c_{-i}, w_i, \alpha, \theta) \propto \frac{n_{c,-i}}{\alpha + n - 1} F(w_i \mid \theta_i) \quad (3.28)$$



1219 The probability to sample a new cluster only depends on  $\alpha$  and the total number of data  
 1220 items. This is described in equation 3.29.

$$p(c_i \in \Omega(c) \text{ and } c_i \neq c_j \text{ and } i \neq j \mid c_{-i}, \alpha) \propto \frac{\alpha}{\alpha + n - 1} \int F(w_i \mid \theta_i) dH(\theta) \quad (3.29)$$

1221 Here  $\Omega(c)$  denotes all admitted values for  $c_i$ .

1222 The importance of conjugacy is obvious from Eq. 3.29, it will lead to an analytic form of the  
 1223 integral. The inference method using equations 3.28 and 3.29 is described in section ??.

1224 Directly sampling over the clusters is described in its general form (Neal, 2000) (Algo-  
 1225 rithm 2).

1226 Rather than updating each  $\theta_i$  per observation  $w_i$ , an entire cluster  $\theta_k$  is updated. In Algo-  
 1227 rithm 7 the update of a cluster would require a first observation to generate a new cluster  
 1228 at  $\theta_j$  and then moving all observations of the old cluster  $\theta_i$  to  $\theta_j$ .

1229 Algorithm 8 follows the same procedure in excluding  $w_i$  from calculating the likelihood. This  
 1230 requires the previously mentioned “downdate” from the corresponding sufficient statistics.  
 1231 In Algorithm 8 after all observations have been iterated over and assigned the corresponding  
 1232 cluster  $k$ , an outer loop iterates over all clusters to obtain new parameters  $\theta$  from the NIG  
 1233 prior.

## 1234 3.4 Performance of the Infinite Line Model

1235 The Infinite Line Mixture Model (see section ??) is able to fit an infinite number of lines  
 1236 through a point cloud in two dimensions. These lines are no line segments, but infinite  
 1237 lines. However, to test the model a variable number of lines are generated of a length that is  
 1238 considerably larger compared to the spread caused by the standard deviation of points from  
 1239 that line.

1240 As described before, Gibbs sampling leads to correlated samples. We choose to get the  
 1241 Maximum A Posterior estimates for our clusters by picking the median values for all the  
 1242 parameters involved.

### 1243 3.4.1 Clustering Performance

1244 The results are measured using conventional metrics for clustering performance. For exam-  
 1245 ple the Rand Index describes the accuracy of cluster assignments (Rand, 1971):

$$R = \frac{a + b}{a + b + c + d} \quad (3.30)$$

1246 Here  $a$  numbers the pair of points that belong to the same cluster, both at ground truth as  
1247 well as after the inference procedure. Likewise  $b$  numbers the pair of points that belong to  
1248 different clusters in both sets. The values  $c$  and  $d$  describe discrepancies between the ground  
1249 truth and the results after inference. A Rand Index of one means that there have been no  
1250 mistakes.

1251 The clustering performance is separate from the line estimation performance. If the points  
1252 are not properly assigned, the line will not be estimated correctly. Due to the fact that line  
1253 estimation has this secondary effect, this performance is not taken into account. Moreover,  
1254 from lines that generated only a single, or very few points, we can extract point assignments,  
1255 but line coefficients are impossible to derive. This would lead to introducing a threshold  
1256 for the number of points per cluster. Moreover, the performance would then need to be  
1257 measured by weighting the fitting versus the assignment.

1258 The performance of Algorithm 1 can be seen in Fig. 3.3 and is rather disappointing. On  
1259 average the inference procedure agrees upon the ground truth for 75% of the cases consid-  
1260 ering the Rand Index. Moreover, if we adjust for chance as with the Adjusted Rand Index,  
1261 the performance drops to only having 25% correct!

1262 Algorithm 2 leads to stellar performance measures (Fig. 3.4). Apparently updating entire  
1263 clusters at once with respect to their parameter values leads at times to perfect clustering,  
1264 bringing the performance metrics close to their optimal values.

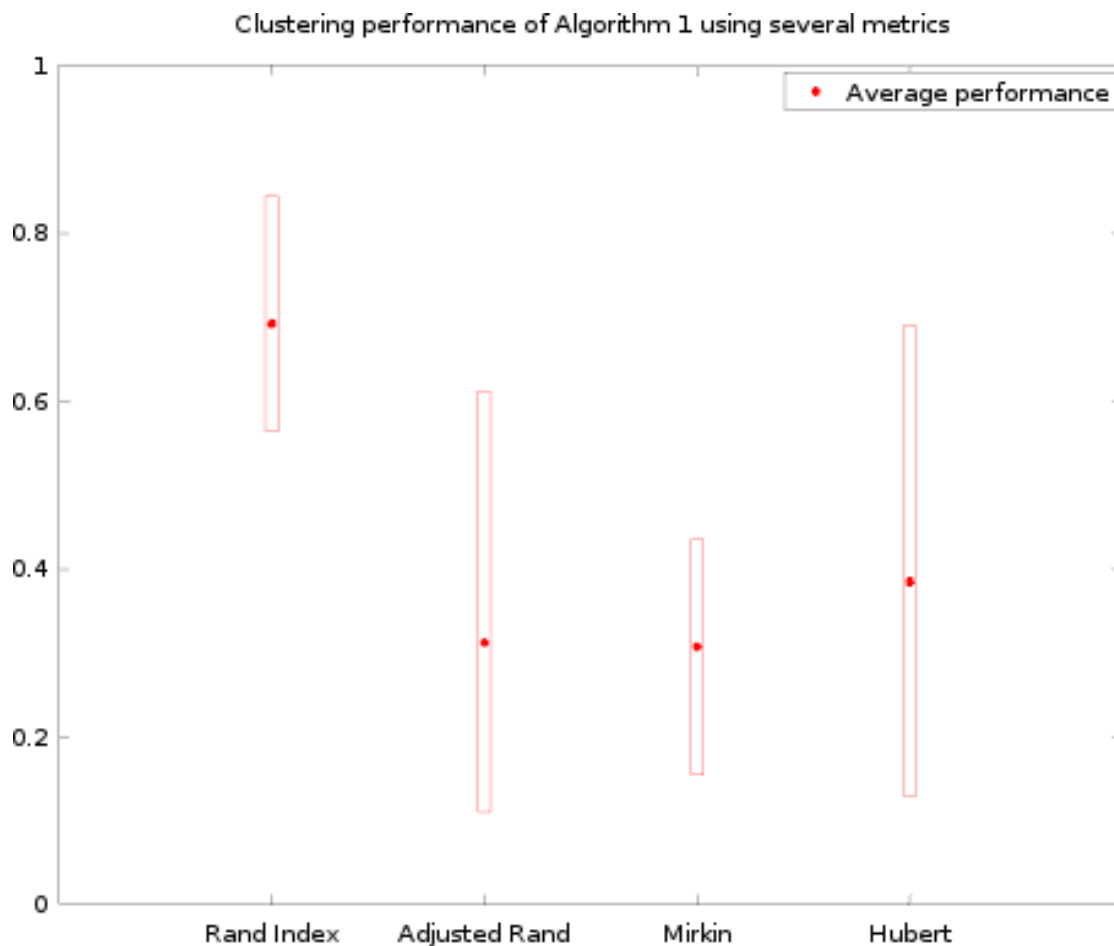
1265 The lack of performance of Algorithm 1 is not only caused by slower mixing (time required to  
1266 reach the steady state distribution). Also when allowing it ten times the number of iterations  
1267 of Algorithm 2, it still does not reach the same performance levels. A line seems to form  
1268 local regions of high probability making it difficult for points to postulate slightly changed  
1269 line coordinates.

### 1270 3.4.2 Two Examples

1271 In the following we show two examples to understand the inference process better. Fig-  
1272 ure 3.5 shows the assignment after a single Gibbs step in Algorithm 1. There is a single line  
1273 that is represented by two clusters. Algorithm 1 does not have merge or split steps to group  
1274 these clusters at once, it thus has to move each data point one by one. By the way, there  
1275 are split-merge algorithms that take these more sophisticated Gibbs steps into account (Jain  
1276 and Neal, 2004).

1277 The example in Fig. 3.6 shows that a single point as an outlier is not a problem for our  
1278 method. A single point might throw off Bayesian linear regression, but because there are  
1279 multiple lines to be estimated in our Infinite Line Mixture Model, this single point is assigned  
1280 its own line.

1281 The extension to more points as outliers would of course require us to postulate a distribution  
1282 for these outlier points as well. A uniform distribution might for example be used in tandem  
1283 with the proposed model. This however would lead to a non-conjugate model and hence  
1284 different inference methods.

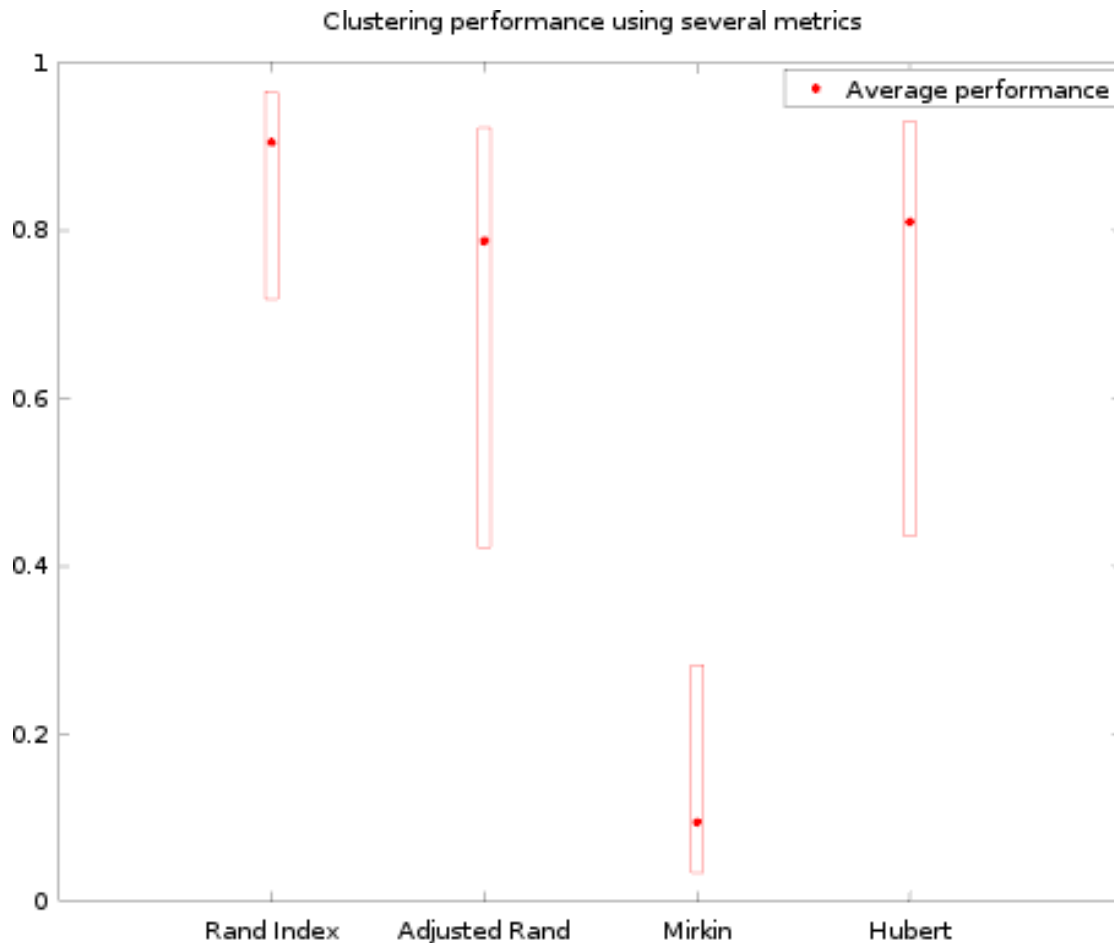


**Figure 3.3:** The performance of Algorithm 1 with respect to clustering is measured using the Rand Index, the Adjusted Rand Index, the Mirkin metric, and the Hubert metric. A figure of 1 means perfect clustering for all metrics, except Mirkin's where 0 denotes perfect clustering.

## 1285 3.5 Chapter Conclusions

1286 The Infinite Line Mixture Model that is proposed extends the familiar Bayesian linear regres-  
 1287 sion model to an infinite number of lines using a Dirichlet Process as prior. The model is a  
 1288 full Bayesian method to detect multiple lines. A full Bayesian method, in contrast to ad-hoc  
 1289 methods such as the Hough transform or RANSAC, means optimal inference (Zellner, 1988)  
 1290 given the model and noise definition.

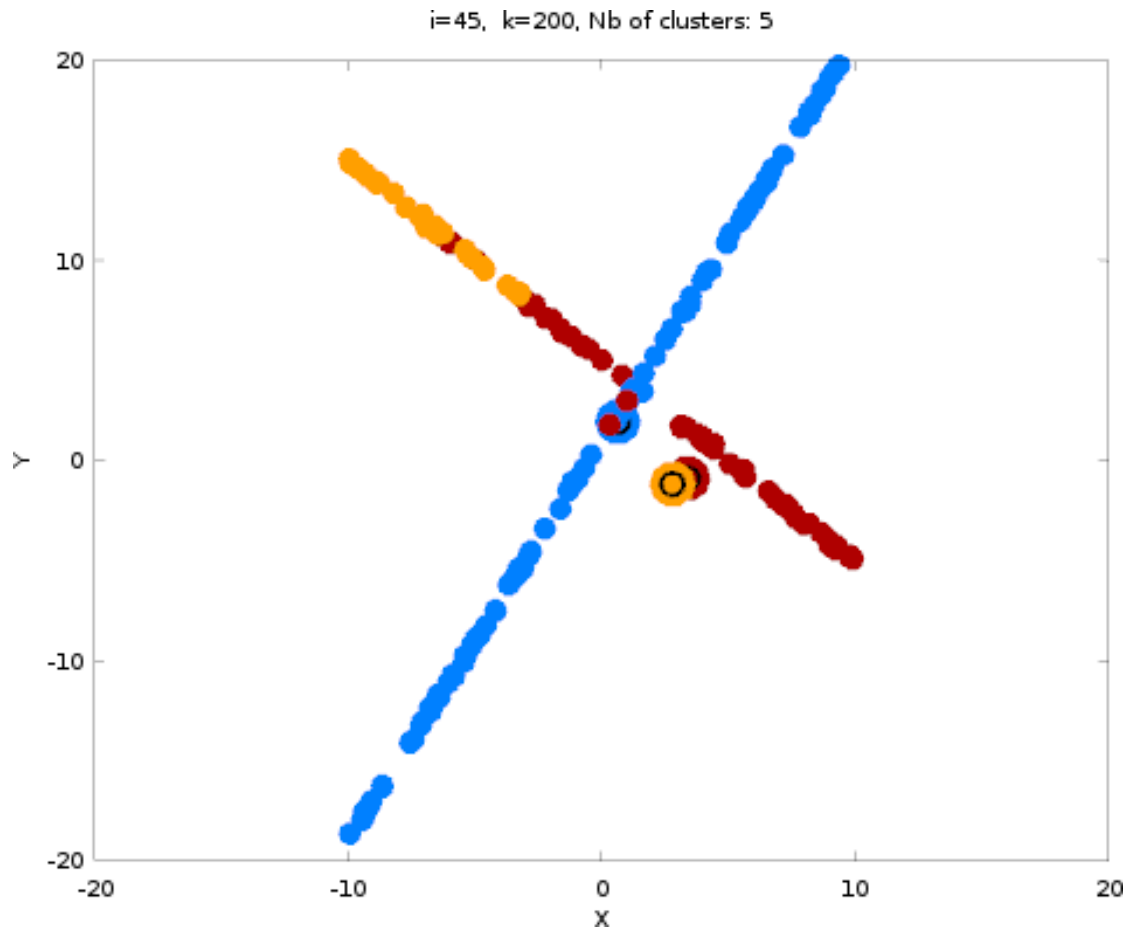
1291 Results in section ?? show high values for difference performance metrics for clustering,  
 1292 such as the Rand Index, the Adjusted Rand Index, and other metrics. The Bayesian model  
 1293 is solved through two types of algorithms. Algorithm 7 iterates over all observations and  
 1294 suffers from slow mixing. The individual updates makes it hard to reassign large number of  
 1295 points at the same time. Algorithm 8 iterates over entire clusters. This allows updates for  
 1296 groups of points leading to much faster mixing. Note, that even optimal inference results  
 1297 in occasional misclassifications. The dataset is generated by a random process. Hence,



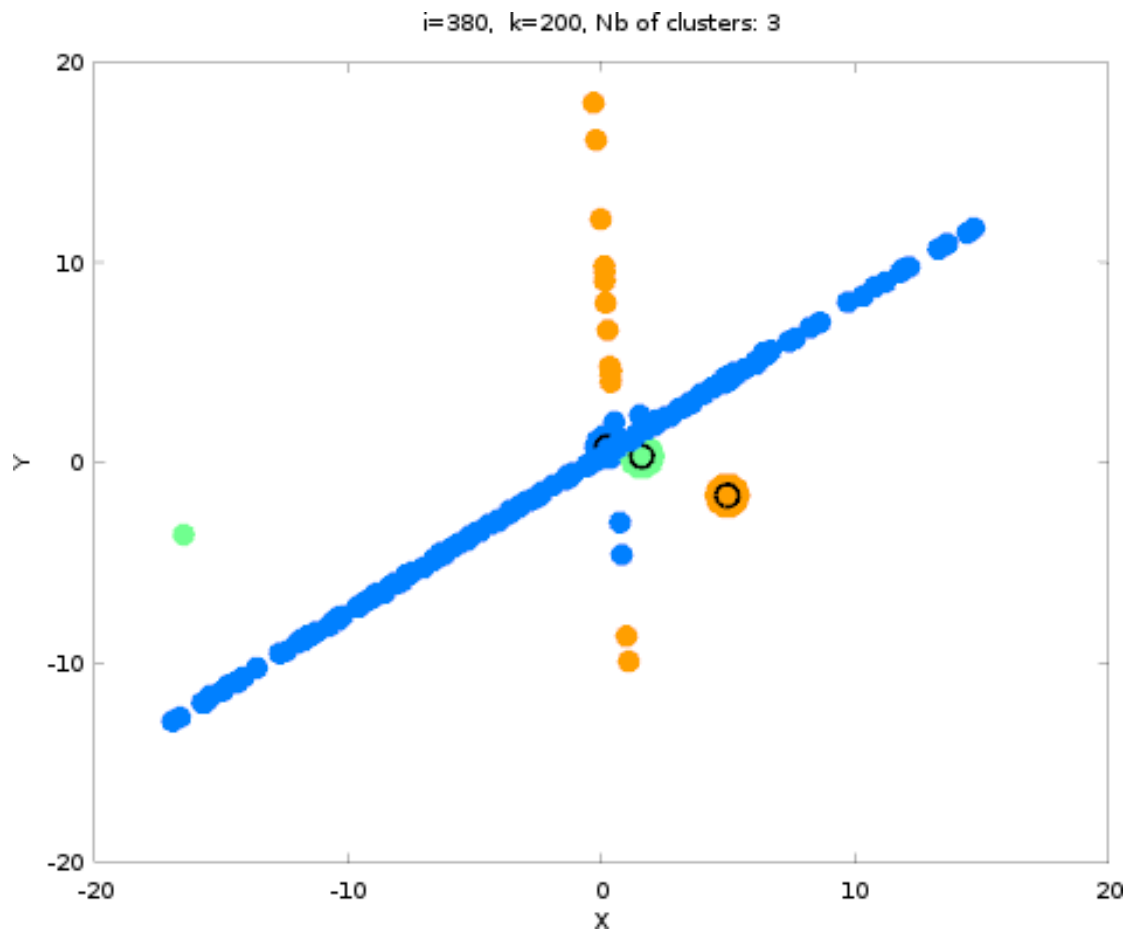
**Figure 3.4:** The performance of Algorithm 2 with respect to clustering is measured using the Rand Index, the Adjusted Rand Index, the Mirvin metric, and the Hubert metric. A figure of 1 means perfect clustering for all metrics, except Mirvin's where 0 denotes perfect clustering.

occasionally two lines are generated with almost the same slope and intercept. Points on these lines are impossible to assign to the proper line.

The essential contribution in this chapter is the introduction of a fully Bayesian method to infer lines and there are two ways in which the postulated model can to be extended for full-fledged inference in computer vision as required in robotics. First, the extension of lines in 2D to planes in 3D. This is quite a trivial extension that does not change anything of the model except for the dimension of the data points. Second, somehow a prior needs to be incorporated to limit the lines of infinite length, to line segments. To restrict points on the lines to a uniform distribution of points over a line segment, a symmetric Pareto distribution can be used as prior (for the end points). This would subsequently allow for a hierarchical model in which these end points are in their turn part of more complicated objects. Hence, the Infinite Line Mixture Model is an essential step towards the use of Bayesian methods (and thus properly formulated priors) for robotic computer vision.



**Figure 3.5:** One of the Gibbs steps in the inference of two particular lines. The points are more or less distributed according to the lines, but one line exists out of two large clusters. The line coordinates are visualized by a double circle. The x-coordinate is the y-intercept of the line, the y-coordinate is the slope.



**Figure 3.6:** The assignment of a line to a single point. There are three clusters found, rather than only the obvious two.

1311

1312

1313

# NONPARAMETRIC BAYESIAN SEGMENT ESTIMATION

1314

## Contents

1315

1316

1317

1318

1319

1320

The nonparametric Bayesian model for line estimation (Chapter 3) does not take into account lines that are of finite length. In this chapter, we introduce a Bayesian method to perform inference over such line segments. In this model our prior for the extend of the line segment is a symmetric Pareto distribution. Due to the fact that the prior and likelihood is not a conjugate pair a more general inference method is used, namely Gibbs sampling with auxiliary variables.

1321

## Published in

1322

1323

1324

A.C. van Rossum, H.X. Lin, J. Dubbeldam, and H.J. van den Herik. Non-parametric Segment Detection. *Proceedings of the Eighth European Starting AI Researcher Symposium, STAIRS 2016*, The Hague, Netherlands, August 26-September 2, 2016.

1325

## Outline

1326

1327

1328

1329

1330

1331

1332

The model is using a Normal-Inverse-Gamma and a Pareto prior for an individual line segment (Sect. 4.1). These line segments are generated using a Dirichlet Process (Sect. 4.2). This generative process is used to perform inference using Gibbs sampling over auxiliary variables (Sect. 4.3). The results for inference over line segments are compared with these for lines. (Sect. 4.4). Finally, weak aspects of the current MCMC method are established (Sect. 4.5) which will form the basis for new inference methods in the next chapters.

1333

## 4.1 Pareto Pairs

1334

.

1335

Lines in a two-dimensional space are mathematical objects that can be described by two parameters. To limit a line to a line segment, four parameters are required. There are

1336

two parametrizations that come to mind. First, a center-point parametrization, in which parameters describe the center of a line segment, the slope of the line through the center, and the size of the line segment. Second, an endpoint parametrization, in which parameters describe the locations of the two endpoints. These parametrizations are equivalent, but generalizations can be intuitive or cumbersome. The generalization to a line segment from a two-dimensional space to a three-dimensional space, requires the endpoints to be positions in a 3D space. The center-point parametrization would require a nonintuitive description of the angles in particular directions. The generalization to squares and rectangles or shapes with many endpoints, might benefit from the center-point parametrization.

There seems to be no statistical description of data points distributed over a line segment that has a conjugate prior form. A line segment itself, however, has a conjugate form! Suppose that we have a prior for the location of endpoints on the x-axis. Given the data we want to update the location of the endpoints. By leaving out the spread of the data over the segment, we can do this using a conjugate Bayesian construction.

The data is distributed according to a symmetric uniform distribution. Hence the likelihood is given by:

$$p(x | a) \sim \mathcal{U}(-a, a) = \begin{cases} \frac{1}{2a} & \text{for } x \leq |a| \\ 0 & \text{otherwise} \end{cases} \quad (4.1)$$

Here the uniform distribution is centered around 0 and extends with size  $a$  in both directions. It is possible to shift the entire distribution with  $b$ . For now, let's continue with one endpoint at  $a$  and one endpoint at  $-a$ .

A prior for the (endpoints of a) symmetric uniform distribution is a symmetric Pareto distribution:

$$p(a) \sim \mathcal{P}_s(\lambda, k) = \begin{cases} \frac{1}{2} k \lambda^k |a|^{-k-1} & |a| \geq \lambda \\ 0 & \text{otherwise} \end{cases} \quad (4.2)$$

The factor  $\frac{1}{2}$  stems from the fact that the symmetric Pareto distribution is now mirrored across the y-axis. Hence, the probability density is half of that of the normal Pareto distribution for the positive x-axis.

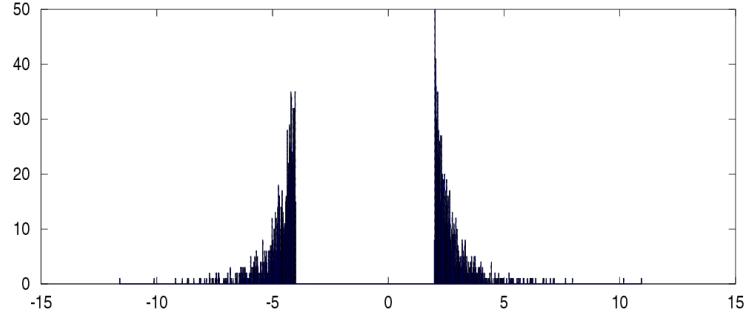
If we would just sample from a symmetric Pareto distribution, we can sample multiple times from the positive x-axis. To actually sample endpoints of segments we have to sample pairs of points.

$$p(a, b) \sim \mathcal{P}_p(\lambda_m, \lambda_n, k) \quad (4.3)$$



1364 We can describe this process as first sampling  $a$  and  $b$  from a categorical distribution to  
 1365 decide which one will be the left endpoint and which one the right endpoint. Then we  
 1366 sample the right endpoint from a normal Pareto distribution and the left endpoint from a  
 1367 mirrored Pareto distribution.

1368 The sampling of Pareto pairs is visualized in Fig. 4.1.



**Figure 4.1:** Sampling of Pareto pairs. The parameters are  $\lambda_m = 2$ ,  $\lambda_n = -2$ ,  $k = 5$ , and we have sampled  $N = 1000$  pairs.

1369 The Pareto distribution is a conjugate prior for the uniform distribution, with updated hy-  
 1370 perparameters:

$$p(a \mid D) = \mathcal{P}(c, N + k) \quad (4.4)$$

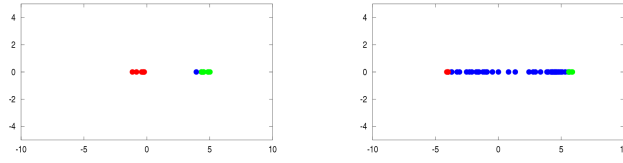
1371 The data is denoted by  $D = \{x_0, \dots, x_{N-1}\}$ , the parameter  $k$  is adjusted with the number of  
 1372 data points  $N$ , and the parameter  $c$  is the maximum of  $\{m, \lambda\}$  with  $m$  the maximum value  
 1373 in  $D$ .

1374 Given this description for the posterior for a single point, the posterior for a Pareto pair  
 1375 can be found by sampling in parallel from a Pareto distribution  $\mathcal{P}(c_n, N + k_n)$  with  $c_n$  the  
 1376 maximum of the data points  $D$  and  $\lambda_n$ ,  $N$  the number of Pareto pairs, and  $k_n$  the hyperprior  
 1377 for the endpoint at the right. Plus sampling from a Pareto distribution  $\mathcal{P}(c_m, N + k_m)$  with  
 1378  $c_m$  the minimum of the data points  $D$  and  $\lambda_m$ ,  $N$  the same, and  $k_m$  the hyperprior for the  
 1379 endpoint at the left.

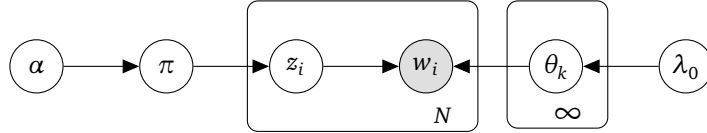
1380 If  $k_n \neq -k_m$  the distribution is shifted such that  $k'_n = -k'_m$ . This makes the form of the  
 1381 probability distribution symmetric with respect to the  $y$  axis. In the end the results are  
 1382 shifted back. This transformation makes sense for pairs of points. We do not want the  
 1383 two scale parameters of the Pareto distribution to influence the symmetry of the overall  
 1384 distribution.

1385 Sampling from the Pareto distribution is through inverse transform sampling. By sampling  
 1386 from  $U(0, 1)$  with 1 included, we transform according to  $k/U^{1/a}$ .

1387 Fig. 4.2 shows how the endpoints are updated given the data. An uninformative prior is  
 1388 used. In this case the hyperparameters  $k_n$  and  $k_m$  are set close to 0, thus the data will wash  
 1389 out the prior immediately. Naturally, it is possible to set them quite large. In that case it



**Figure 4.2:** Consider the data uniformly distributed on a line segment and a symmetric Pareto prior for both the endpoints, then we can update the estimate for the endpoints given the data as visualized. Each subfigure shows an adjustment of the endpoints given more data points (1, 3, 10, and 100 data points).



**Figure 4.3:** The Bayesian linear regression model for multiple line segments in plate notation is the same as for the Infinite Line Model. The Dirichlet process is defined at the left with concentration parameter  $\alpha$ . It generates the partitions  $(\pi_1, \dots, \pi_k)$  with assignment parameters  $z_i$  that denote which observation  $w_i$  belongs to which cluster  $k$ . The cluster is summarized through the parameter set  $\theta_k$  and has  $\lambda_0$  as its hyperparameter. The parameter set  $\theta_k$  includes parameters that signify the line itself such as slope and y-intercept, plus the parameters that denote the extend of the segment.

1390 must be noted that the data will never be able to “correct for” this prior. Note also that the  
 1391 maximum and minimum operators are quite sensitive to outliers as well.

## 1392 4.2 Generative Process to Create a Line Segment

1393 To be able to perform inference over a line segment in a two-dimensional space, we’ll have  
 1394 to map somehow these points to a one-dimensional space.

1395 In the case of a line we can sample  $\theta_i$  from a Normal-Inverse-Gamma with hyperparameter  
 1396  $\lambda_{temp}$ . The latter we have in closed form given observations through a single update.

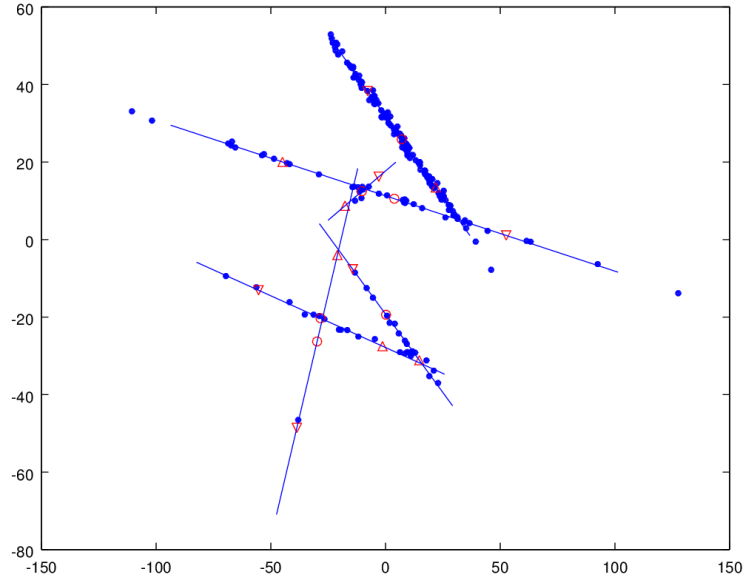
1397 In the case of a line segment there is no known conjugate prior available. Let’s reiterate the  
 1398 Dirichlet Process basis for our nonparametric model:

$$\begin{aligned} G &\sim DP(\alpha, H) \\ \theta_i &| G \sim G \\ w_i &| \theta_i \sim F(\theta_i) \end{aligned} \tag{4.5}$$

1399 Again  $F$  describes the mapping from parameters  $\theta_i$  to observations  $w_i$ . As described, for line  
 1400 segments this mapping is different from that of lines.

$$\begin{aligned} F(\theta_i) &= \mathcal{N}(\mu_i + H(v_i), \Sigma_i) \\ H(v_i) &= \mathcal{N}(v_i, 1)\gamma_i \\ \gamma_i &\sim \mathcal{N}(0, 1) \end{aligned} \tag{4.6}$$

1401 The probability density  $F$  is a Gaussian with a mean that is additively distributed according  
 1402 to another distribution  $H$ . The latter distribution originates from the product of a normal  
 1403 distribution with a value sampled from a normal distribution. Fig. 4.4 shows how points are  
 1404 generated from the described distribution.



**Figure 4.4:** Line segments generated through a Dirichlet Process. The Dirichlet Process itself is again the same. But now four parameters are generated. A normal-inverse-Wishart distribution is used to generate the center of the line segment, and an inverse-Wishart distribution to generate one of the endpoints of the line segments (the other end point is mirrored through its center). Points are generated normally over the line segments, with an additional Gaussian component to indicate the deviation from the line segment from the normal-inverse-Wishart.

1405 To generate lines uniformly, only  $\gamma_i$  needs adjustment:

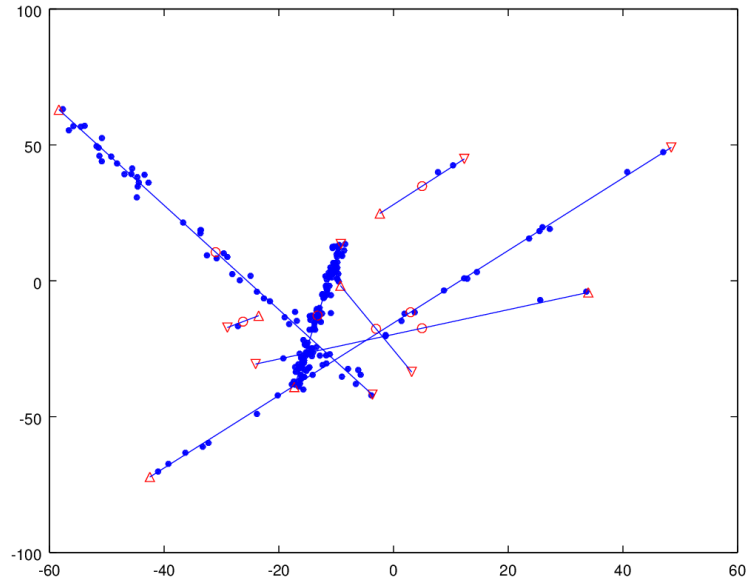
$$\begin{aligned}
 F(z_i) &= \mathcal{N}(\mu_i + H_i(v_i), \Sigma_i) \\
 H(v_i) &= \mathcal{N}(v_i, 1)\gamma_i \\
 \gamma_i &\sim \mathcal{U}(-1, 1)
 \end{aligned}
 \tag{4.7}$$

1406 Fig. 4.5 displays the adjustment with points generated uniformly over the line segment.

1407 The descriptions in Eq. 4.6 and 4.7 are clearly not conjugate setups. This means that infer-  
 1408 ence over line segments requires more complicated sampling strategies.

### 1409 4.3 Inference over a Line Segment

1410 To perform inference over a line segment our model is not conjugate anymore. This re-  
 1411 quires a sampling algorithm that does not make use of conjugacy. An algorithm that does



**Figure 4.5:** Line segments generated through a Dirichlet Process. Compared to Fig. 4.4 the points are generated uniformly over the line segments: points are not generated outside of the line segments.

1412 not assume conjugacy is described in its general form before (Neal, 2000) (Algorithm 8). The  
 1413 sampling process proposes  $m$  new values for the parameters directly from the hyperparam-  
 1414 eters. These are called auxiliary parameters. Now, to establish to which cluster a certain  
 1415 observation  $w_i$  need to be assigned, the likelihood of each existing and new clusters alike  
 1416 are compared. The weight of an old cluster is defined through the number of data points  
 1417 assigned to it. The weight of a new cluster is defined through  $\alpha/m$ .

1418 After every data item is assigned a cluster, the cluster parameters themselves are updated  
 1419 given the assigned data items. In a conjugate model the sufficient statistics can be updated  
 1420 at once, given such observations. In a nonconjugate model we will need to update  $\theta_j$  by  
 1421 sampling from  $p(\theta_j | y)$ .

**Algorithm 9** Gibbs sampling over auxiliary variables (a  $\theta_i$ )

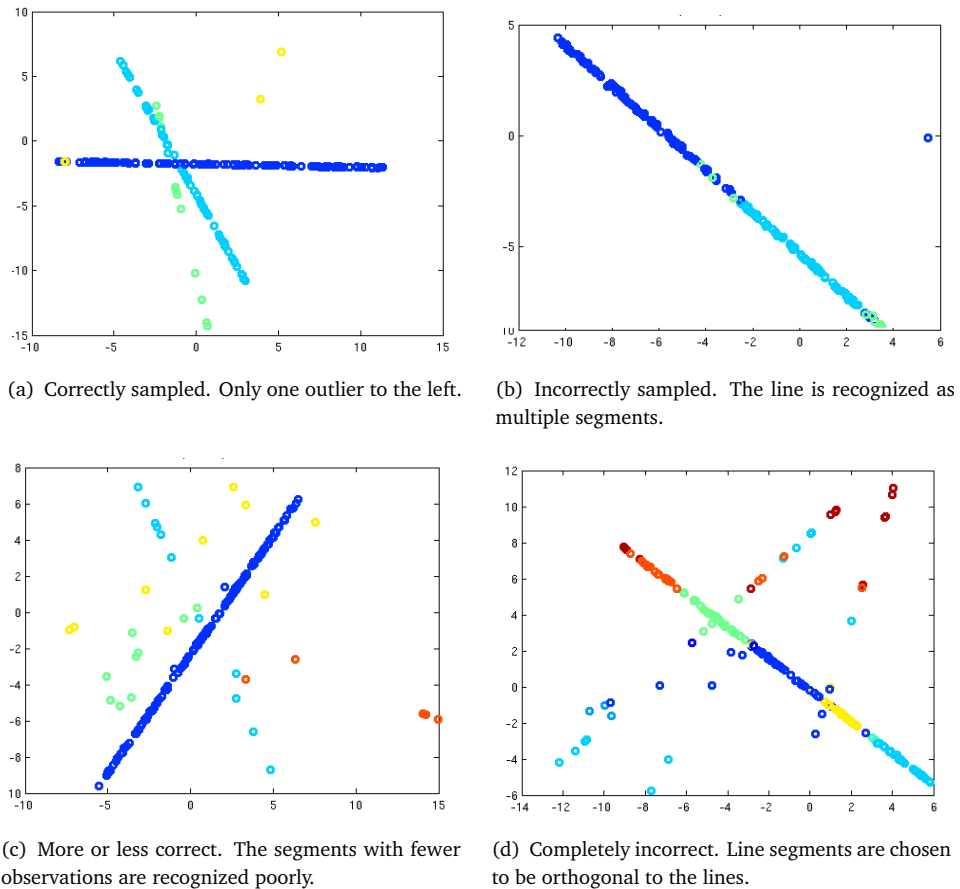
---

```

1: procedure GIBBS ALGORITHM WITH AUXILIARY VARIABLES( $w, \lambda_0, \alpha$ )  $\triangleright$  Accepts points  $w$ ,
   hyperparameters  $\lambda_0, \alpha$ , number of auxiliary variables  $m$ , and returns  $k$  line coordinates
2:   for all  $t = 1 : T$  do
3:     for all  $i = 1 : N$  do
4:       for all  $j = 1 : m$  do
5:          $\theta_j \sim NIG(\lambda_0)$   $\triangleright$  Sample  $\theta_j$  from NIG
6:       end for
7:       for all  $j = 1 : K + m, j \neq i$  do
8:          $L_j = \text{likelihood}(w_i, \theta_j)$   $\triangleright$  Update likelihood for all theta (except  $\theta_i$ ) given
           observation  $w_i$ 
9:       end for
10:       $P_{-i=1:K} = b \sum_{-i} L_{-i}$   $\triangleright$  Calculate probability of existing cluster
11:       $P_{-i=K:K+m} = b\alpha / m L_m L_{-i}$   $\triangleright$  Calculate probability of new cluster
12:       $\theta_i = \theta_j$  according to above  $P_{-i}$   $\triangleright$  Sample  $\theta_i$  accord. to above prob
13:      Remove unused clusters
14:    end for
15:    for all  $j = 1 : K$  do
16:       $\theta_j \sim p(\theta_j | y)$   $\triangleright$  Update  $\theta_j$ 
17:    end for
18:  end for
19:  return summary on  $\theta_k$  for  $k$  line segments
20: end procedure

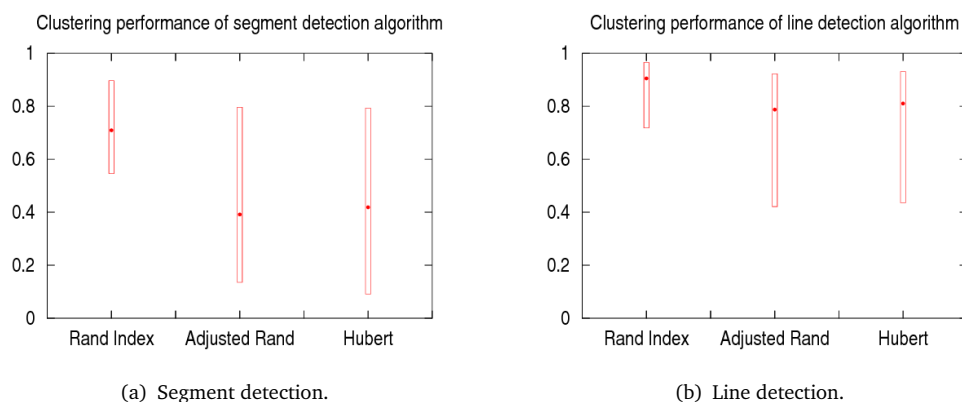
```

---

1422 **4.4 Results****Figure 4.6:** Bayesian point estimates of the sampling process with varying outcomes.

There is one phenomenon that is very noticeable in Fig. 4.6. Line segments that form a larger line segment are not recognized as such by the inference method.

The results over a larger dataset can be measured with clustering metrics as visualized in Fig. 4.7. The Rand Index, Adjusted Rand Index, and Hubert metrics show all reduced performance compared to line detection where there are no constraints on segment size.



**Figure 4.7:** Segment detection performs much worse than line detection across all three clustering performance indicators. Perfect clustering is indicated by 1.0 for Rand Index, Adjusted Rand Index, and Hubert.

## 4.5 Chapter Conclusions

Segment estimation is a much harder problem than line estimation (Chap. 3). In this chapter we used an advanced method, namely MCMC sampling with auxiliary variables to perform inference over an infinite set of line segments. The parameters for line segments do not have a conjugate description. Metropolis-Hastings has been used to perform inference over the line segments, but the search space is quite large. The auxiliary variable MCMC method is indeed faster than ordinary Metropolis-Hastings thanks to postulating multiple new lines than only one.

However, the segment estimation problem is a challenge for the current inference methods. The target probability density has a lot of modes that each needs to be found and are separated by very low probability regions. In Chapter 5 we will introduce new sampling methods that will cope with these challenges.



1440

1441

## TRIADIC SPLIT-MERGE SAMPLER

### Contents

1443

1444

1445

1446

1447

The nonparametric Bayesian model for line estimation, the infinite line model (Chapter 3) thanks to its conjugate properties has been solved with moderately straightforward sampling methods. The additional constraints that limit lines to line segments (Chapter 4) reduced convergence of the underlying MCMC sampling method (a Gibbs method with auxiliary variables) to sub-par results.

1448

1449

This chapter introduces a new sampling method called the triadic split-merge sampler.

### Published in

1451

1452

A.C. van Rossum, H.X. Lin, J. Dubbeldam, and H.J. van den Herik. Triadic Split-Merge Sampler. *The 10th International Conference on Machine Vision, ICMV 2017, Vienna, Austria, November 13-November 15, 2017.*

### Outline

1454

1455

1456

1457

1458

1459

1460

1461

The class of split-merge samplers, part of MCMC samplers, are introduced (Sect. 5.1). A conventional split-merge sampler, labeled the dyadic split-merge sampler is detailed (Sect. 5.2). The new split-merge sampler, the triadic split-merge sampler is introduced (Sect. 5.3). The results for inference over lines is compared between the conventional and the new sampler (Sect. 5.4). Finally, although this sampler already improves on the state-of-the-art we see in the chapter conclusions (Sect. 5.5) how we further improve the inference procedure, which will be the basis of the next chapter.

## 5.1 The Class of Split-Merge Samplers

1462

We will consider a Dirichlet process as a prior on the distribution over parameters  $G$ . The form of this model is:

1464



$$\begin{aligned}
y_i | \theta_i &\sim F(\theta_i) \\
\theta_i | G &\sim G \\
G &\sim DP(H, \alpha)
\end{aligned} \tag{5.1}$$

## 1465 5.2 Conventional Split-Merge Sampler

1466 The conventional split-merge sampler Jain and Neal (2004) splits a single cluster into two  
 1467 clusters, and merges two clusters into a single cluster. Hence, this split-merge sampler oper-  
 1468 ates on two clusters at each time step, for which reason we will call it a dyadic split-merge  
 1469 sampler in constrast with our approach.

---

### Algorithm 10 Dyadic split-merge sampler

---

```

1: procedure DYADIC SPLIT-MERGE SAMPLER( $c$ )                                ▷ Accepts cluster assignments
    $c$  of length  $N$  (besides Metropolis-Hastings acceptance factors  $a(c', c)$  and a split procedure e.g.
   SIMPLERANDOMSPLIT) and returns a (potentially) updated cluster assignment vector  $c'$ .
2:    $i \sim U(1, N)$                                                          ▷ Sample  $i$  random uniformly over cluster assignments.
3:    $j \sim U(1, N) \cap i$                                                  ▷ Sample  $j$  also random uniformly, but with  $j \neq i$ .
4:    $S_R = \{c_i, c_j\}$                                                      ▷ Sampled clusters  $c_i, c_j$ .
5:    $S_I = \{c_x\}$  with  $c_x \in S_R$  for  $x \in \{1, \dots, N\}$                 ▷ All data in clusters  $c_i, c_j$ .
6:    $S_E = S \cap S_R$                                                      ▷ All data in clusters  $c_i, c_j$  excluding  $S_R$ .
7:    $N_S = \text{unique}(S_R)$ 
8:   if  $N_S = 1$  then                                                     ▷ Case:  $i, j$  belong to the same cluster.
9:      $c_i^{(2)} = c_k$  with  $c_k \notin \{c_1, \dots, c_N\}$                      ▷ Sample new cluster for  $c_i^{(2)}$ .
10:     $c_j^{(2)} = c_j^{(1)}$                                                    ▷ Keep  $c_j$  the same.
11:     $c_e^{(2)} = \text{SPLITPROCEDURE}(S_E, c_i^{(2)}, c_j^{(2)})$                  ▷ After  $c_i^{(2)}, c_j^{(2)}$  assign  $S_E$ .
12:    for all  $m \notin S_I$  do
13:       $c_m^{(2)} = c_m^{(1)}$                                                  ▷ Data points in clusters other than  $c_i, c_j$  are not adjusted.
14:    end for
15:     $c' = \{c_i^{(2)}, c_j^{(2)}, c_e^{(2)}, c_m^{(2)}\}$ 
16:     $a = a_{\text{split}}(c', c)$  according to Eq. 5.3                         ▷ MH acceptance for a split.
17:  else                                                                 ▷ Case:  $i, j$  belong to different clusters  $c_i \neq c_j$  ( $N_S = 2$ ).
18:    for all  $q \in S_I$  do
19:       $c_q^{(1)} = c_j^{(2)}$                                                  ▷ Assign all data points in  $c_i$  and  $c_j$  to  $c_j$ .
20:    end for
21:    for all  $m \notin S_I$  do
22:       $c_m^{(1)} = c_m^{(2)}$                                                  ▷ Data points in clusters other than  $c_i, c_j$  are not adjusted.
23:    end for
24:     $c' = \{c_q^{(1)}, c_m^{(1)}\}$ 
25:     $a = a_{\text{merge}}(c', c)$  according to Eq. 5.10                         ▷ MH acceptance for a merge.
26:  end if
27:   $u \sim U(0, 1)$                                                          ▷ Sample  $u$  between 0 or 1 uniformly.
28:  if  $a < u$  then
29:     $c' = c$                                                              ▷ Reject  $c'$  by setting it to  $c$ 
30:  end if
31:  return  $c'$ , the (updated) cluster assignment vector:  $c \rightarrow c'$ .
32: end procedure

```

---

1470 In algorithm 10 the notation  $c_i^{(2)}$  is used to signify that the cluster assignment  $c_i$  has 2 clusters  
 1471 under consideration. In the dyadic algorithm we could have used  $c_i^{merge}$  and  $c_i^{split}$ , however  
 1472 in the triadic algorithm (see algorithm 13) with multiple split and merge operations the  
 1473 latter notation would become confusing.

---

**Algorithm 11** Simple random split
 

---

```

1: procedure SIMPLERANDOMSPLIT( $S, c_0, c_1$ )  $\triangleright$  Accepts unassigned set  $S$  and cluster indices  $c_0, c_1$ ,
   returns cluster assignment  $c'_m$ .
2:   for all  $m \in S$  do
3:      $c'_m \sim Cat(c_0, c_1)$  with equiprobable  $p(c_0) = p(c_1) = \frac{1}{2}$ .
4:   end for
5:   return  $c'_m$ , the cluster assignment for  $S$ .
6: end procedure
  
```

---

1474 The dyadic split-merge sampler in Algorithm 10 samples two distinct data items. If the data  
 1475 items belong to the same cluster a split step is attempted. If the data items belong to different  
 1476 clusters a merge step is attempted. The split procedure itself is the so-called simple random  
 1477 split (Algorithm 11) that assigns data items with the same probability to one of the parts of  
 1478 the splitted cluster without consideration for data fit.

### 1479 5.2.1 Acceptance for the Split Step

1480 The acceptance ratio contains the Metropolis ratio to step from  $c$  to  $c'$ :

$$\frac{P(c')L(c'|y)}{P(c)L(c|y)} \quad (5.2)$$

1481 Additionally, the Hastings correction is applied because of the asymmetry of the proposal  
 1482 distribution in the form of  $q(c|c')/q(c'|c)$ :

$$a_{split}(c^{(2)}, c^{(1)}) = \min \left[ 1, \frac{q(c^{(1)}|c^{(2)})}{q(c^{(2)}|c^{(1)})} \frac{P(c^{(2)})}{P(c^{(1)})} \frac{L(c^{(2)}|y)}{L(c^{(1)}|y)} \right] \quad (5.3)$$

1483 The notation  $c^{(2)}$  is used to indicate that the cluster index vector is referencing 2 unique  
 1484 clusters (in this case after the split step).

1485 The prior distribution is represented by a Chinese Restaurant Process with concentration  
 1486 parameter  $\alpha$  and no discount factor. Data not yet assigned is assigned with probability  
 1487  $\alpha/(n + \alpha)$  to a new cluster and with probability  $n_c/(n + \alpha)$  to an existing cluster  $c$ . Here  $n$   
 1488 are the total number of assigned data points,  $n_c$  are the number of data points assigned to  
 1489 cluster  $c$ . There are  $D$  clusters. Hence, the prior over clusters:

$$P(c) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + n)} \alpha^D \prod_{c_l} \Gamma(n_{c_l}) = \alpha^D \frac{\prod_{c_l} (n_{c_l} - 1)!}{\prod_{k=1}^n (\alpha + k - 1)} \quad (5.4)$$

1490 In the prior distribution ratio before and after the split step many of the factors drop out.  
 1491 There is one factor  $\alpha$  remaining and the number of data points in the splitted cluster is part

of the equation. There is no dependency on other clusters or the total number of data points and we can simplify the formula using the beta function  $B(a, b)$ :

$$\frac{P(c^{(2)})}{P(c^{(1)})} = \alpha \frac{(n_{c_i^{(2)}} - 1)!(n_{c_j^{(2)}} - 1)!}{(n_{c_i^{(1)}} - 1)!} = \alpha B(n_{c_i^{(2)}}, n_{c_j^{(2)}}) \quad (5.5)$$

The likelihood can be written as:

$$L(c|y) = \prod_{c=1}^D \prod_{k:c_k=c} p(y_k|\phi) \quad (5.6)$$

Here we assume no conjugacy between  $F(y_k, \phi)$  and prior distribution  $H(\phi)$  and hence write  $p(y_k|\phi)$  rather than the conjugate construction  $\int F(y_k, \phi) dH_{k,c}(\phi)$  (see Dahl (2005)). The likelihood ratio becomes:

$$\frac{L(c^{(2)}|y)}{L(c^{(1)}|y)} = \frac{\prod_{k:c_k^{(2)}=c_i^{(2)}} p(y_k|\phi) \prod_{k:c_k^{(2)}=c_j^{(2)}} p(y_k|\phi)}{\prod_{k:c_k^{(1)}=c_i^{(1)}} p(y_k|\phi)} \quad (5.7)$$

The split step determines the probability of a particular split. Given that already two data points are assigned to distinct clusters, only the remaining ones have to be assigned with equal probability to  $c_i^{(2)}$  and  $c_j^{(2)}$ :

$$q(c^{(2)}|c^{(1)}) = \left(\frac{1}{2}\right)^{-2+n_{c_i^{(2)}}+n_{c_j^{(2)}}} = \left(\frac{1}{2}\right)^{-2+n_{c_i^{(1)}}} \quad (5.8)$$

The probability of the reverse of the split operation is exactly 1. There is only one way in which a single cluster could have risen from a split cluster, hence:

$$\frac{q(c^{(1)}|c^{(2)})}{q(c^{(2)}|c^{(1)})} = \frac{1}{\left(\frac{1}{2}\right)^{n_{c_i^{(2)}}+n_{c_j^{(2)}}-2}} = 2^{-2+n_{c_i^{(1)}}} \quad (5.9)$$

### 5.2.2 Acceptance for the Merge Step

Acceptance of a merge step consists of the same components as that of the split step.

$$a_{merge}(c^{(1)}, c^{(2)}) = \min \left[ 1, \frac{q(c^{(2)}|c^{(1)})}{q(c^{(1)}|c^{(2)})} \frac{P(c^{(1)})}{P(c^{(2)})} \frac{L(c^{(1)}|y)}{L(c^{(2)}|y)} \right] \quad (5.10)$$

$$\frac{P(c^{(1)})}{P(c^{(2)})} = \alpha^{-1} \frac{(n_{c_i^{(1)}} - 1)!}{(n_{c_i^{(2)}} - 1)!(n_{c_j^{(2)}} - 1)!} = \frac{1}{\alpha B(n_{c_i^{(2)}}, n_{c_j^{(2)}})} \quad (5.11)$$

$$\frac{L(c^{(1)}|y)}{L(c^{(2)}|y)} = \frac{\prod_{k:c_k^{(1)}=c_i^{(1)}} p(y_k|\phi)}{\prod_{k:c_k^{(2)}=c_i^{(2)}} p(y_k|\phi) \prod_{k:c_k^{(2)}=c_j^{(2)}} p(y_k|\phi)} \quad (5.12)$$

1507

$$\frac{q(c^{(2)}|c^{(1)})}{q(c^{(1)}|c^{(2)})} = \left(\frac{1}{2}\right)^{-2+n_{c_i^{(1)}}} = 2^{2-n_{c_i^{(1)}}} \quad (5.13)$$

1508 The ratios of the merge step are the inverse of the ratios of the split step.

### 1509 5.2.3 Sequentially-Allocated Merge-Split Sampler

1510 A variant on the conventional split-merge sampler is the Sequentially Allocated Merge-Split<sup>1</sup>  
 1511 (SAMS) sampler Dahl (2003). The simple random split procedure of Algorithm 11 is re-  
 1512 placed by a procedure that sequentially assigns observations to clusters rather than splitting  
 1513 the data random uniformly over the splitted clusters.

---

#### Algorithm 12 Sequentially Allocated Merge-Split

---

```

1: procedure SAMS( $S, c_0, c_1$ )    ▷ Accepts unassigned set  $S$ , cluster indices  $c_i$ , and  $p(y_k|\theta_{c_i})$  with
    $i = 0, 1$ , returns cluster assignment  $c'_m$ .
2:    $T = \text{random\_shuffle}(S)$ 
3:   for all  $m \in T$  do
4:      $p(c_m = c_0 | c_0, c_1, \theta_{c_0}, \theta_{c_1}) = \frac{N_0 p(y_k | \theta_0)}{N_0 p(y_k | \theta_0) + N_1 p(y_k | \theta_1)}$ 
5:      $p(c_m = c_1 | c_0, c_1, \theta_{c_0}, \theta_{c_1}) = 1 - p(c_m = c_0 | c_0, c_1, \theta_{c_0}, \theta_{c_1})$ 
6:      $c'_m \sim p(c_m | c_0, c_1, \theta_{c_0}, \theta_{c_1})$ 
7:   end for
8:   return  $c'_m$ , the cluster assignment for  $S$ .
9: end procedure

```

---

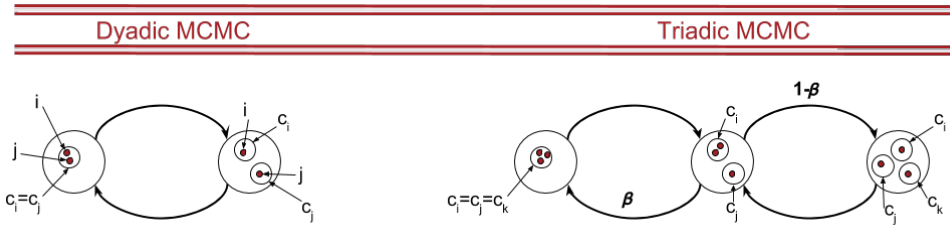
1514 In contrast to the simple random split, observations  $y_k$  are used in the SAMS to obtain cluster  
 1515 assignments that correspond with the data rather than cluster assignments independent of  
 1516 the data.

## 1517 5.3 Triadic split-merge sampler

1518 The triadic split-merge sampler uses up to three clusters for a split or merge step (Fig. 5.1).

---

<sup>1</sup>In the naming of split-merge or merge-split samplers, the order of merge split does not bear any significance.



**Figure 5.1:** Right: dyadic MCMC picks two data items  $i, j$  random uniformly. If both are in the same cluster a split towards two clusters is attempted. If both are in distinct clusters a merge towards one cluster is attempted. Left: triadic MCMC picks three data items  $i, j, k$  random uniformly. If all three are in the same cluster a split towards two clusters is attempted. If the three items are in two clusters either a split into three (with probability  $1 - \beta$ ) or a merge into a single cluster (with probability  $\beta$ ) is attempted. If the three data items are in three distinct clusters a merge is attempted. There are no direct transitions from a single cluster to three clusters or the other way around.

1519 The intuition behind the triadic split-merge sampler is twofold:

- 1520     ◦ In the dyadic sampler there is a large asymmetry between split and merge steps. There  
 1521         is only one way in which two clusters can be merged into one single cluster, while there  
 1522         are many ways in which one single cluster can be split into two clusters. This asym-  
 1523         metry is reduced by transitioning between two and three clusters. This is a straight-  
 1524         forward improvement in balancing split and merge steps (for alternatives, see Wang  
 1525         and Russell (2015)).
- 1526     ◦ In practical optimization problems it might be useful to form a third cluster out of  
 1527         subsets of two other clusters. The dyadic MCMC sampler requires immediate steps in  
 1528         which (1) one of these clusters is split into two, (2) the other is split into two, and  
 1529         (3) the two new clusters are merged. This means that (a) mixing and hence conver-  
 1530         gence will be slow and (b) the intermediate steps might have very low probability and  
 1531         function as an unnecessary barrier between high probable states.

1532 Sampling random uniformly for three unique items is implemented through a random shuf-  
 1533 fle algorithm, in particular the modern version of the Fisher-Yates shuffle introduced by  
 1534 Durstenfeld Durstenfeld (1964) and picking the first three items.

### 1535 5.3.1 Acceptance for the Split Step

1536 In the triadic split-merge sampler there are two splitting steps. It is possible to split according  
 1537 to the dyadic split-merge sampler. However, given two clusters there are (split) jumps to  
 1538 three states as well as (merge) jumps to single states again. To account for this asymmetry  
 1539 another Hastings correction is applied to establish detailed balance.

$$a_{split}(c^{(2)}, c^{(1)}) = \min \left[ 1, \frac{r(c^{(1)}|c^{(2)}) q(c^{(1)}|c^{(2)}) P(c^{(2)}) L(c^{(2)}|y)}{r(c^{(2)}|c^{(1)}) q(c^{(2)}|c^{(1)}) P(c^{(1)}) L(c^{(1)}|y)} \right] \quad (5.14)$$

**Algorithm 13** Triadic split-merge sampler

---

```

1: procedure TRIADIC SPLIT-MERGE SAMPLER( $c$ )                                ▷ Accepts
   cluster assignments  $c$  of length  $N$  (besides Metropolis-Hastings acceptance factors  $a(c', c)$  and a
   split procedure) and returns a (potentially) updated cluster assignment vector  $c'$ .
2:    $i \sim U(1, N)$                                 ▷ Sample  $i$  random uniformly over cluster assignments.
3:    $j \sim U(1, N) \cap i$                             ▷ Sample  $j$  also random uniformly, but with  $j \neq i$ .
4:    $k \sim U(1, N) \cap \{i, j\}$                         ▷ Sample  $k$  random uniformly, but with  $k \neq j, k \neq i$ .
5:    $S_R = \{c_i, c_j, c_k\}$                                 ▷ Sampled clusters  $c_i, c_j, c_k$ .
6:    $S_I = \{c_x\}$  with  $c_x \in S_R$  for  $x \in \{1, \dots, N\}$         ▷ All data in clusters  $c_i, c_j, c_k$ .
7:    $S_E = S_I \cap S_R$                                 ▷ All data in clusters  $c_i, c_j, c_k$  excluding  $S_R$ .
8:    $N_S = \text{unique}(S_R)$ 
9:    $u \sim U(0, 1)$                                 ▷ Sample  $u$  between 0 or 1 uniformly.
10:  if  $N_S = 1$  then                                ▷ Case:  $i, j, k$  belong to the same cluster.
11:    return  $c' = \text{DYADIC SPLIT-MERGE SAMPLER}(c)$ 
12:  else if  $N_S = 2$  and  $u < \beta$  then                ▷ Case: a cluster with one item and one with two items and
    $u < \beta$ .
13:    return  $c' = \text{DYADIC SPLIT-MERGE SAMPLER}(c)$ 
14:  else if  $N_S = 2$  and  $u \geq \beta$  then                ▷ Case: a cluster with one item and one with two items and
    $u \geq \beta$ .
15:     $c_i^{(3)} = c_k$  with  $c_k \notin \{c_1, \dots, c_N\}$         ▷ Sample new cluster for  $c_i^{(3)}$ .
16:     $c_j^{(3)} = c_j^{(2)}$                                 ▷ Keep  $c_j$  the same.
17:     $c_e^{(3)} = \text{SPLITPROCEDURE}(S_E, c_i^{(3)}, c_j^{(3)})$         ▷ After  $c_i^{(3)}, c_j^{(3)}$  assign  $S_E$ .
18:    for all  $m \notin S_I$  do
19:       $c_m^{(3)} = c_m^{(2)}$                                 ▷ Data points in clusters other than  $c_i, c_j$  are not adjusted.
20:    end for
21:     $c' = \{c_i^{(3)}, c_j^{(3)}, c_e^{(3)}, c_m^{(3)}\}$ 
22:     $a = a_{\text{split}}(c', c)$  according to Eq. 5.14        ▷ MH acceptance for a split.
23:  else                                ▷ Case:  $i, j, k$  belong to three different clusters  $c_i \neq c_j \neq c_k$  ( $N_S = 3$ ).
24:     $S_L = S_I \cap \{c_i^{(3)}, c_j^{(3)}\}$                 ▷ Data in clusters  $c_i, c_j, c_k$  except for  $i$  and  $j$  itself.
25:     $\{c_i^{(2)}, c_j^{(2)}\} = \text{SAMS}(S_L, c_i^{(3)}, c_j^{(3)})$         ▷ Assign data points in  $c_i, c_j, c_k$  to  $c_i, c_j$ .
26:    for all  $m \notin S_L$  do
27:       $c_m^{(2)} = c_m^{(3)}$                                 ▷ Data points in clusters other than  $S_L$  are not adjusted.
28:    end for
29:     $c' = \{c_i^{(2)}, c_j^{(2)}, c_m^{(2)}\}$ 
30:     $a = a_{\text{merge}}(c', c)$  according to Eq. 5.21        ▷ MH acceptance for a merge.
31:  end if
32:   $u \sim U(0, 1)$                                 ▷ Sample  $u$  between 0 or 1 uniformly.
33:  if  $a < u$  then
34:     $c' = c$                                 ▷ Reject  $c'$  by setting it to  $c$ 
35:  end if
36:  return  $c'$ , the (updated) cluster assignment vector:  $c \rightarrow c'$ .
37: end procedure

```

---

1540 Here we have one additional term compared to the split step from one cluster to two clusters:

$$\frac{r(c^{(1)}|c^{(2)})}{r(c^{(2)}|c^{(1)})} = \frac{\beta}{1} \quad (5.15)$$

1541 The parameter  $\beta$  is free to control, as long as  $0 < \beta < 1$  (to maintain ergodicity). The  
1542 transition from two states to three states is another split step:

$$a_{split}(c^{(3)}, c^{(2)}) = \min \left[ 1, \frac{r(c^{(2)}|c^{(3)})}{r(c^{(3)}|c^{(2)})} \frac{q(c^{(2)}|c^{(3)})}{q(c^{(3)}|c^{(2)})} \frac{P(c^{(3)})}{P(c^{(2)})} \frac{L(c^{(3)}|y)}{L(c^{(2)}|y)} \right] \quad (5.16)$$

1543 The fraction with  $r$ :

$$\frac{r(c^{(2)}|c^{(3)})}{r(c^{(3)}|c^{(2)})} = \frac{1}{1 - \beta} \quad (5.17)$$

1544 The fraction with  $q$  uses the total number of data points  $n_c$  in the clusters:

$$\frac{q(c^{(2)}|c^{(3)})}{q(c^{(3)}|c^{(2)})} = \frac{\left(\frac{1}{2}\right)^{n_c-2}}{\left(\frac{1}{3}\right)^{n_c-3}} = (3^{n_c-3})(2^{2-n_c}) = \left(\frac{3}{2}\right)^{n_c} \frac{2^2}{3^3} \quad (5.18)$$

1545 To move from 2 clusters to 3 clusters the probability is a  $1/3$  for each cluster index in vector  
1546  $c$  (except for the three data items already selected randomly, hence  $n_c - 3$ ). To move back,  
1547 the probability is a  $1/2$  and there are only two data items randomly assigned beforehand.  
1548 The fraction with  $P$  uses the number of data points in each of the clusters before and after  
1549 the step:

$$\frac{P(c^{(3)})}{P(c^{(2)})} = \alpha \frac{(n_{c_i^{(3)}} - 1)!(n_{c_j^{(3)}} - 1)!(n_{c_k^{(3)}} - 1)!}{(n_{c_i^{(2)}} - 1)!(n_{c_j^{(2)}} - 1)!} = \alpha \frac{B(n_{c_i^{(3)}}, n_{c_j^{(3)}}, n_{c_k^{(3)}})}{B(n_{c_i^{(2)}}, n_{c_j^{(2)}})} \quad (5.19)$$

1550 Here we introduced a generalized Beta function  $B(a, b, c) = \Gamma(a)\Gamma(b)\Gamma(c)/\Gamma(a + b + c)$  with  
1551  $\Gamma(x) = (x - 1)!$  the Gamma function. The likelihood ratio becomes:

$$\frac{L(c^{(3)}|y)}{L(c^{(2)}|y)} = \frac{\prod_{m:c_m^{(3)}=c_i^{(3)}} P(y_m|\phi) \prod_{m:c_m^{(3)}=c_j^{(3)}} P(y_m|\phi) \prod_{m:c_m^{(3)}=c_k^{(3)}} P(y_m|\phi)}{\prod_{m:c_m^{(2)}=c_i^{(2)}} P(y_m|\phi) \prod_{m:c_m^{(2)}=c_j^{(2)}} P(y_m|\phi)} \quad (5.20)$$

### 1552 5.3.2 Acceptance for the Merge Step

1553 The merge step from two to one cluster is analogous to the split step:

$$a_{merge}(c^{(1)}, c^{(2)}) = \min \left[ 1, \frac{r(c^{(2)}|c^{(1)})}{r(c^{(1)}|c^{(2)})} \frac{q(c^{(2)}|c^{(1)})}{q(c^{(1)}|c^{(2)})} \frac{P(c^{(1)})}{P(c^{(2)})} \frac{L(c^{(1)}|y)}{L(c^{(2)}|y)} \right] \quad (5.21)$$

1554 The merge step from three clusters to two clusters is:

$$a_{merge}(c^{(2)}, c^{(3)}) = \min \left[ 1, \frac{r(c^{(3)}|c^{(2)})}{r(c^{(2)}|c^{(3)})} \frac{q(c^{(3)}|c^{(2)})}{q(c^{(2)}|c^{(3)})} \frac{P(c^{(2)})}{P(c^{(3)})} \frac{L(c^{(2)}|y)}{L(c^{(3)}|y)} \right] \quad (5.22)$$

1555 Note that all the fractions in Eq. 5.22 are the reverse of the fractions in Eq. 5.16. Inverting  
1556 Eq. 5.17–5.20 will be left to the reader.

One additional issue we have to consider. When merging three clusters into two we can (1) distribute the data over all three clusters or (2) alternatively, keep the data in two clusters assigned to these clusters and only distribute the data in the third cluster over the other two clusters. The second and alternative option however would introduce unnecessary asymmetry with the merge step. In other words, Eq. 5.23 is not the inverse of Eq. 5.18. In contrast, the equation is similar to splitting one cluster across two as in Eq. 5.9:

$$\frac{q_{alt}(c^{(3)}|c^{(2)})}{q_{alt}(c^{(2)}|c^{(3)})} = 2^{-2+n_c} \quad (5.23)$$

Hence the first option is entertained and the  $q$ -fraction is exactly the inverse of Eq. 5.18.

Another choice has been made, namely to exclude direct operations between a single cluster and three clusters. This is because factors like:

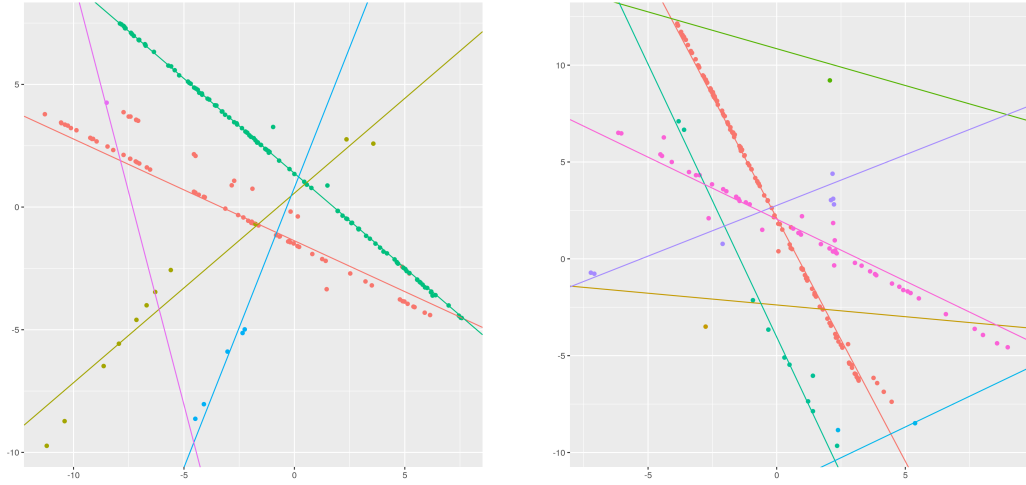
$$\frac{P(c^{(3)})}{P(c^{(1)})} = \alpha^2 \frac{(n_{c_i^{(3)}} - 1)!(n_{c_j^{(3)}} - 1)!(n_{c_k^{(3)}} - 1)!}{(n_{c_i^{(1)}} - 1)!} \quad (5.24)$$

become very small and although compensated by a large  $q$  fraction, remain further away from an acceptance factor of 1. Note that by the ability to split a single cluster into two and then into three, there is no ergodic argument to introduce also the immediate step.

## 5.4 Results

The problem we use to test our sampler is a well-known problem in computer vision, namely that of the inference of line parameters (slope and intercept) given data points. Rather than ordinary linear regression, in computer vision there is a mixture of lines that have to be estimated. Moreover, the number of lines is not known beforehand. To solve this problem we use the Dirichlet process mixture (Eq. 5.1) with a normal distribution  $N(0, \sigma_0)$  to generate the line parameters and a likelihood function that defines points to be uniformly distributed across a line of length 20 and deviating from the line according to a normal distribution  $N(0, \sigma_1)$ .





**Figure 5.2:** Two examples of fitting a mixture of lines to data items scattered over a two-dimensional space. The lines drawn are inferred using one of the methods in this chapter. The lines are not the ground truth, but are meant to demonstrate the typical errors made by fitting methods. Note for example that there are mistakes in both the assignment of points to lines as well as the line parameters (slope and intercept).

#### 1578 5.4.1 Implementation

1579 The sampler is open-source<sup>2</sup> implemented in C++ which means that (a) it is computationally  
 1580 fast, (b) it can be run on embedded devices if a cross-compiler is available and the Eigen3  
 1581 library is ported. Note, that due to the fact that the simulator uses a lot of random numbers  
 1582 the system should use a modern compiler (g++-6 or newer) and should have enough entropy  
 1583 available<sup>3</sup>. Rather than a random scan, the implementation uses a fixed scan as advocated  
 1584 in the literature MacEachern (2007).

1585 To speed up the sampler most calculations are done in log-space. Consider  $v = u + 1$ . The  
 1586 ratio with probabilities (Eq. 5.5 and 5.19) becomes:

$$\log \frac{P(c^{(v)})}{P(c^{(u)})} = \log(\alpha) + \sum_i \log \Gamma(n_{c_i^{(v)}}) - \sum_i \log \Gamma(n_{c_i^{(u)}}) \quad (5.25)$$

1587 The fraction with  $q(\cdot)$  (Eq. 5.9 and 5.18) becomes:

$$\log \frac{q(c^{(v-1)}|c^{(v)})}{q(c^{(v)}|c^{(v-1)})} = (v - n_c - 1) \log(v - 1) - (v - n_c) \log(v) \quad (5.26)$$

1588 The fraction with  $r$  becomes for example (Eq. 5.17):

$$\log \frac{r(c^{(2)}|c^{(3)})}{r(c^{(3)}|c^{(2)})} = -\log(1 - \beta) \quad (5.27)$$

<sup>2</sup>Code can be found at <https://code.annevanrossum.nl/noparama>.

<sup>3</sup>On Linux this can be checked in `/proc/sys/kernel/random/entropy_avail`.

1589 The log-probability to calculate the likelihood given by a multivariate Normal distribution is  
 1590 well-known.

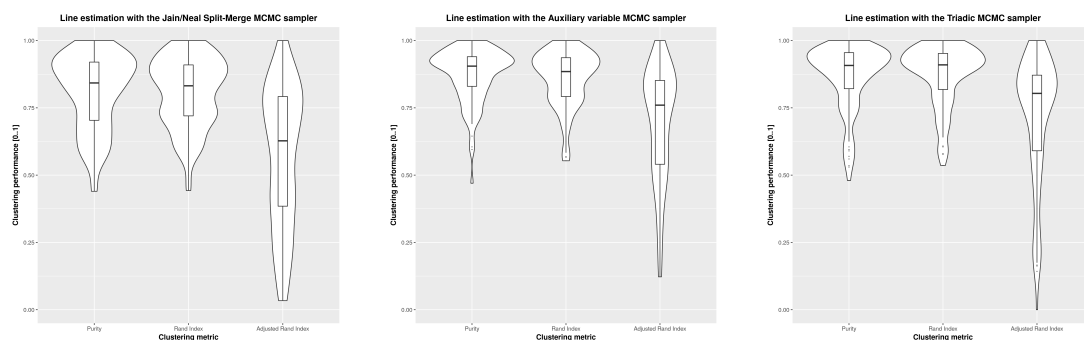
## 1591 5.4.2 Comparison

1592 The Triadic sampler using SAMS is compared with the Jain-Neal Dyadic sampler using SAMS  
 1593 and an auxiliary variable sampler with  $m = 3$  (see algorithm 8 in Neal (2000)).

Method	Purity	Rand Index	Adjusted Rand Index
Dyadic sampler	0.80960	0.80580	0.56382
Auxiliary variables	0.87235	0.85879	0.68224
Triadic sampler	0.86405	0.87188	0.71067

**Table 5.1:** The purity, rand index, and adjusted rand index establishing the quality of the clustering method. The closer the values to one, the better the method performed. The purity metric assigns high values to clusters that do not have data points from other clusters (but does not penalize the number of clusters). The rand index computes similarity between clusters taking false negatives and false positives into account. The adjusted rand index accounts for chance. The adjusted rand index is most useful in our comparison.

1594 In Table 5.1 the line estimation problem is compared for the dyadic sampler, an auxiliary  
 1595 variables sampler, and the proposed triadic sampler. The simulation is run with  $\beta = 0.1$  so  
 1596 that a significant number of steps are tried between two and three clusters (rather than only  
 1597 between one and two clusters).



**Figure 5.3:** The same results as in Table 5.1, but visualized in a violin plot. The distribution over metric values are displayed in a vertical fashion. From left to right the distribution shifts to one, signifying better clustering performance.

1598 In Fig. 5.3 the different metrics are visualized in the form of violin plots. The improvement  
 1599 in clustering is especially visible with the adjusted rand index.

## 1600 **5.5 Chapter Conclusions**

1601 A new split-merge sampler has been introduced, implemented, and applied to the computer  
1602 vision problem of line estimation. The sampler outperforms existing samplers, such as the  
1603 ordinary (dyadic) split-merge sampler Jain and Neal (2004) and auxiliary variable sampler  
1604 Neal (2000).

1605 Although the proposed split-merge sampler is able to mix considerably faster through a mix-  
1606 ture model, it does not use global jumps directly based on the data. It is reasonable to suggest  
1607 that MCMC methods benefit from combining the local jumps with global jumps, for example  
1608 by a mixture of the local Metropolis-Hastings sampler with a Metropolized independence  
1609 sampler Jampani et al. (2015). We will introduce such a sampler in chapter 6.



# ADVERSARIALLY TRAINED MCMC KERNELS

1613	<b>Contents</b>	To use MCMC for volumetric inference it is necessary to be able to accelerate the algorithms even further. Volumetric objects exhibit more structure, which is reflected by symmetry.
1616	<b>Outline</b>	We describe MCMC methods that cope with symmetric objects.

## 6.1 Data-Driven Inference

There are three aspects we would like to address in our inference engine.

The first aspect aims to have structure within our inference engine. The proposal distribution in a Markov chain, although moderately complex in the previous chapter, does not have much knowledge about the model. An artificial border is maintained that does not allow the inference engine to have knowledge about the model. The purpose of this is never articulated in particular. However, it is logical from a separation of concern. Such an inference engine (1) does not need to receive any information about the model and (2) is guaranteed to be general in the sense that it is not tailored to a particular model. This is nicely articulated by Tran et al. (2017) from which we quote.

“Many existing probabilistic programming languages treat the inference engine as a black box, abstracted away from the model. These cannot capture probabilistic inferences that reuse the model’s representation - a key idea in recent advances in variational inference, generative adversarial networks, and also in more classic inference.

The second aspect concerns the data. In MCMC the position for the chain is driven by (1) the prior, (2) the prior and the likelihood, (3) a sequence of priors and likelihood, (4) a

sequence of priors, likelihood and proposal distributions, basically anything, except for the data itself. Data-driven approaches would namely destroy the convergence of the Markov chain. To start an MCMC sampler in a data-driven manner and continue in a data-oblivious manner is a possible solution (Zhang and Perez-Cruz, 2017). Even better, it is possible to use a Metropolized independence sampler Jampani et al. (2015). Such a sampler samples independently from the previous state and uses global information. However, to work well its proposal distribution needs to match the target distribution quite well. Although, when combined with a local sampler, it might be sufficient to just be able to match the modes of the target distribution well.

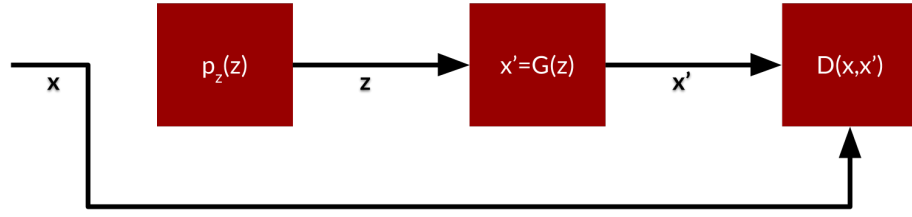
The third aspect concerns the way we build our MCMC engine. The split-merge sampler of the previous chapter has been meticulously designed. If we admit a data-driven approach, we might as well adjust our MCMC engine using training samples. Note, that this training will be across a set of line, box, or scenes mixtures. The MCMC engine will not be able to learn just the parameters of a particular visual object. It will learn how to jump around (optionally, adaptively) from one visual object to the next or from one cluster configuration to the next. In other words, it will be able to teach itself to become a Triadic Split-Merge sampler if that happens to be a good engine. Is it possible to constrain the search through MCMC kernels such that its result is always converging in an MCMC sense? If we aim to learn the transition operator of our Markov chains, there is new literature that makes use of deep nets.

## 6.2 Learning the Transition Operator

There are multiple methods that can be used in a generative setting. We will discuss the three most prominent ones: (1) generative adversarial networks, (2) variational autoencoders, and (3) infusion training. This is far from an extensive categorization, worth studying are variational walkback (Goyal et al., 2017), stacked generative adversarial networks (Huang et al., 2016), generative latent optimization (Bojanowski et al., 2017), deep learning through the use of non-equilibrium thermodynamics (Sohl-Dickstein et al., 2015), denoising autoencoders, or generative stochastic networks, to name just a few.

### 6.2.1 Adversarial Training

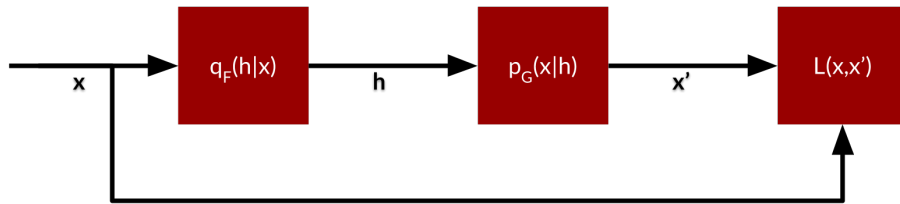
Adversarial training has been extensively studied since the article on generative adversarial networks by Goodfellow et al. (2014). A particular adversarial setup for training an MCMC has been suggested as well (Song et al., 2017). The generator samples from a Markov chain. A discriminator subsequently needs to judge if its incoming data comes from the generator or if it is sampled from the actual data set. To start the process, the generator can run the chain from the model as well as from the data.



**Figure 6.1:** Left:  $p_z(z)$  is a (prior) random distribution that generates random variables  $z$ . Middle: the generator  $G$  maps the random variables  $z$  to simulated data points  $x'$ . Right: the discriminator  $D(x, x')$  compares the simulated data  $x'$  with the real data  $x$ . The generator tries to generate samples in such way that the discriminator has difficulties distinguishing the simulated from the real data.

### 6.2.2 Variational Autoencoders

Variational autoencoders (Kingma and Welling, 2013; Rezende et al., 2014) are ordinary autoencoders with additional constraints on the latent variables. The latent variables in autoencoder parlance are called the code. In a variational autoencoder the latent variables are forced to approximately describe a unit Gaussian distribution. The autoencoder is trained using a loss function that is composed out of (1) a generative loss, a mean squared error that measures how accurately the network reconstructs its input, and (2) a latent loss, a KL-divergence that measures how closely the latent variables match a unit Gaussian. To optimize the KL divergence a reparameterization trick is applied. The encoder does not generate a vector with real values, but generates a vector with means and standard deviations instead.



**Figure 6.2:** Left:  $q_F(h|x)$  maps the data  $x$  to (hidden) random variables  $h$ . Middle:  $p_G(x|h)$  maps the hidden random variables to reconstructed data  $x'$ . Right:  $L(x, x')$  measures the similarity between  $x$  and  $x'$ .

### 6.2.3 Infusion Training

The transition operator can also be learned directly through infusion training. In infusion training we gradually adjust totally unstructured noise to a target distribution as well. In this method a particular data point is ‘infused’ into the Markov chain to bias the model sampling to move towards this particular data point (and not another). In contrast to a generative loss that is a mean squared error, this promises to have less blurry reconstructions.

### 1686 6.3 Volumetric Models

1687 In image processing autoencoders have been used for 2D shape recognition. To apply the  
1688 same type of models to 3D point clouds, these point clouds are represented through voxels.  
1689 The application of data-driven deep learning techniques, be it autoencoders, generative ad-  
1690 versarial networks, or adversarial autoencoders promises similar good results in these 3D  
1691 settings than in the current computer vision tasks.

1692 The 3D ShapeNet model (Wu et al., 2015) exists of 3D voxel input that is piped through  
1693 several stages with an increasing number of filters. The used voxel representation is a bi-  
1694 nary tensor. It assigns a value of 1 to each voxel that is inside the 3D object mesh and a  
1695 value 0 to each voxel outside the mesh (empty space). The voxel sizes are fixed as well as  
1696 the grid size (in this particular model the grid exists of 30x30x30 voxels). The inference  
1697 model is a Deep Belief Network (DBN). Convolution operators, in the form of filters over  
1698 small neighbourhoods, are used to reduce the number of model parameters (30x30x30 fully  
1699 connected would be really many weights). The DBN is used in a supervised setting where  
1700 shapes are trained with object labels. The model subsequently learns to generate shapes  
1701 given an object label.

1702 An unsupervised method in the form of a convolutional (volumetric) autoencoder (Sharma  
1703 et al., 2016) has been applied to the same type of data. This (denoising) autoencoder,  
1704 coined VConv-DAE maps from an entire voxel grid to another voxel grid. This work uses a  
1705 combination of standard techniques, a dropout layer, a deconvolution layer, ReLu as well as  
1706 sigmoid activation, but it is not in particular tailored to 3D point clouds.

1707 Other representations than voxels are used. For example collections of 2D views and trans-  
1708 formation parameters (Dosovitskiy et al., 2017). The most interesting are methods that  
1709 work with raw data, the point cloud themselves. This alleviate the need to process the data  
1710 and does not inadvertently increase the data dimensions, for example by artificially introduce  
1711 voxels where there is no object present.

1712 PointNet (?) directly operates on point clouds. To handle the input as a set of points (un-  
1713 ordered), it uses a symmetric function over  $n$  input vectors and outputs a vector that is  
1714 invariant to the input order. Typically sum and multiplication operators are such symmetric  
1715 functions. After input and feature transforms by multi-layer perceptrons, a max pooling op-  
1716 erator is used to map the input to a global feature. In the ModelNet40 shape classification  
1717 benchmark there are more than 12000 CAD models from 40 object categories. PointNet  
1718 achieves state of the art results compared to volumetric methods for a fraction of the com-  
1719 putational costs.

1720 Point clouds are also directly used in so-called deep kd-networks (Klokov and Lempitsky,  
1721 2017). A kd-tree is constructed by recursively picking the coordinate axis with the largest  
1722 range of point coordinates and splitting the set of points into two subsets of equal size.  
1723 These subsets are recursed into successively. The recursion stops at a particular level, depth  
1724  $D$ . The kd-networks are purported to outperform for example PointNet amongst other model  
1725 architectures.



1726 A deep permutation equivariant (for semisupervised learning) and permutation invariant  
1727 (for supervised learning) network has also been directly applied to point clouds (Ravanbakhsh  
1728 et al., 2016). It does not reach the ModelNet40 accuracy levels from PointNet or the kd-  
1729 networks though.

1730 PointNet++ Qi et al. (2017) introduces hierarchical structure to PointNet. This fits better  
1731 non-uniform point distributions and seems to surpass kd-nets again on the ModelNet40 task.



1732

1733

## RECOMMENDER ENGINE

### Contents

1735

1736

1737

The described nonparametric Bayesian models (Chapter 3, 4, 5, and 6) are not limited to computer vision tasks. This chapter describes a recommender engine in which groups of runners are extracted from data collected from social media.

### Outline

1739

1740

1741

1742

1743

We (1) introduce the form of the data at hand, (2) describe a multi-modal Von Mises-Uniform distribution to model the individual runners, (3) use a Dirichlet Process prior to group people, (4) use the previously described MCMC methods to perform inference, (5) show the results on an artificial and real-world data set, and (6) discuss ways with which the model can be expanded.

## 7.1 Application

1744

The data of people exercising can be considered binary (someone is either exercising or not in a particular timeslot). We do have however more information available. We know how often people have been exercising in a timeslot. This data has the form as visualized in Table 7.1.

1748

**Table 7.1:** Example of the type of data about the timing of exercising. A person is represented by row, her preferences by column. There is not a predefined number of users or groups of users.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	0	4	23	38	9	12	6	2	7	2	3	2	7	5	3	2	0	3	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	11	4	4	2	1	2	0	0	0
0	0	0	0	0	1	0	0	1	2	0	0	0	1	0	2	3	1	1	3	2	0	0	0
0	0	0	0	0	0	5	4	1	3	11	3	7	3	3	4	3	10	23	3	0	0	0	0
0	0	0	0	0	0	8	35	23	12	4	41	14	11	8	7	14	38	36	6	5	1	2	0
0	0	0	0	0	0	0	1	9	2	0	3	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	29	9	8	7	5	3	0	2	0	1	8	6	1	4	1	2	0	0
0	0	0	0	0	2	14	12	7	1	0	2	0	2	0	3	4	6	4	3	9	2	0	1
0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	3	3	2	0	1	0	0	0
0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	2	1	2	3	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	4	5	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	3	2	1	0	1	1	0	0	0	0	3	11	12	0	0	0
0	0	0	0	0	0	0	0	0	2	1	0	0	0	1	0	3	5	4	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	18	4	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	2	0	0	1	9	14	14	2	1	4	15	4	0	0
0	0	0	0	0	0	0	0	2	4	2	1	3	5	2	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	1	0	0	0	1	1	3	3	1	3	2	3	0	0
0	0	0	0	0	0	0	1	2	1	3	2	0	1	2	0	0	0	0	1	2	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	8	4	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	4	6	2	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	1	1	6	2	1	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	2	2	3	12	18	12	12	2	4	3	5	6	43	37	16	1	2	0

1749 The first row defines the time of day, starting at midnight till the last timeslot from 23.00 till  
 1750 00.00. In this particular case it shows that nobody is running from midnight to four o'clock  
 1751 in the morning, understandably so.

## 1752 7.2 Model of Individuals

### 1753 7.2.1 Multi-modal Normal-Uniform Distribution Model

1754 We first postulate the likelihood function for the moments at which people exercise through  
 1755 the day as defined in Eq. 7.1.

$$f(x|\theta) = w_0 \mathcal{U}(a, b) + \sum_{i=1}^2 w_i \mathcal{N}(\mu_i, \sigma_i) \quad (7.1)$$

1756 The likelihood (Eq. 7.1) is built up out of three probability density functions: one Uniform dis-  
 1757 tribution  $\mathcal{U}(a, b)$  with  $a$  and  $b$  as parameters and two Normal distributions  $\mathcal{N}(\mu_i, \sigma_i)$  with  
 1758 mean  $\mu_i$  and  $\sigma_i$ . The distributions are weighted by the factors  $w_0, w_1, w_2$ . The collection of  
 1759 parameters for the likelihood function is referred to by  $\theta = \{a, b, \mu_1, \sigma_1, \mu_2, \sigma_2, w_0, w_1, w_2\}$ .

1760 This probability density function  $f(x|\theta)$  will have the form as in Fig. 7.1. The uniform  
 1761 distribution generates values here between 00:00 and 24:00. There are on top of that the  
 1762 two Normal distributions that form peaks at certain moments during the day.



**Figure 7.1:** The likelihood function for the moments at which people decide to exercise during the day. On the horizontal axis time, on the vertical axis the frequency of exercising.

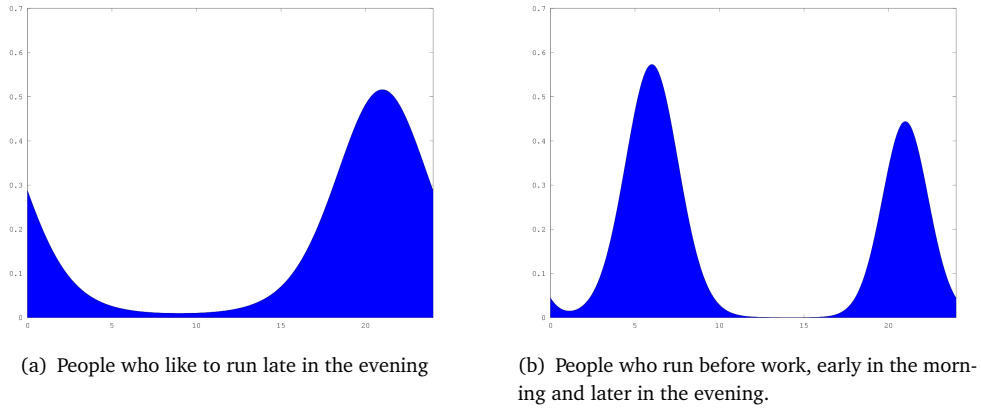
1763 There is something noticable in Fig. 7.1, namely that the 24 hours of a day cause the Nor-  
 1764 mal distribution to be cut off. Especially in Fig. 7.1 (a) there should be some considerable  
 1765 likelihood of running in the wee hours of the morning between 00:00 and 01:00.

## 1766 7.2.2 Multi-modal Von-Mises-Uniform Distribution Model

1767 There are several options to define a distribution over a limited range  $T$ . A so-called wrapped  
 1768 distribution is a distribution defined over the unity circle. By just multiplying it with  $T/(2\pi i)$   
 1769 it can be used to define a probability density function over a day ( $T = 24$ ).

$$f(x|\theta) = w_0 \mathcal{U}(a, b) + \sum_{i=1}^2 w_i \mathcal{VM}(\mu_i, \kappa_i) \quad (7.2)$$

1770 The likelihood (Eq. 7.2) is again built up out of three probability density functions: one  
 1771 Uniform distribution  $\mathcal{U}(a, b)$  with  $a$  and  $b$  as parameters and two Von Mises distributions  
 1772  $\mathcal{VM}(\mu_i, \kappa_i)$  with mean  $\mu_i$  and  $\kappa_i$ . The parameters  $\mu_i$  will be scaled and shifted with  $[a, b]$  so  
 1773 all variables within this range fall on the unity circle. The parameter  $\kappa_i$  plays the same role as  
 1774  $\sigma_i$  for the Normal distribution. The distributions are weighted by the factors  $w_0, w_1, w_2$ . The  
 1775 collection of parameters for the likelihood function is referred to by  $\theta = \{a, b, \mu_1, \kappa_1, \mu_2, \kappa_2, w_0, w_1, w_2\}$ .



**Figure 7.2:** The improved likelihood function for the moments at which people decide to exercise during the day using the Von Mises distribution. On the horizontal axis time, on the vertical axis the frequency of exercising.

The likelihood with Von Mises distributions rather than Normal distributions is visualized in Fig. 7.2. The Von Mises distributions capture behavior in deviations from a standard exercise time. It does not take into account structured deviations, for example running late from work every Thursday, or weekends, or a person either running in the morning or in the evening, but never both at the same day.

### 7.2.3 Hyperparameters

The parameters  $\theta$  for each user  $j$  are either fixed or generated from prior distributions. The hyperparameters  $a$  and  $b$  for the Uniform distribution are set to 0 and 24. People can run potentially any time of the day. The hyperparameters  $\mu_i, \kappa_i$  for the Von Mises distributions are generated from a Uniform-Exponential distribution (Eq. 7.3). The Uniform distribution reflects the fact that if people run on regular times, this time can be any time of the day. The Exponential distribution defines a prior on how much people deviate from such a regular time to exercise.

$$\begin{aligned} f(\mu_i|a, b) &= \mathcal{U}(a, b) \\ f(\kappa_i|\lambda) &= \mathcal{E}(\lambda) \end{aligned} \tag{7.3}$$

The Uniform distribution generates  $\mu_i$  between  $a$  and  $b$ . The parameter  $\kappa_i$  is generated from an Exponential distribution with hyperparameter  $\lambda$ . If  $\lambda$  is set to be small ( $< 0.5$ ) we have a high likelihood that  $\kappa$  can be large and we have pronounced peaks. In contrary, if  $\lambda$  is set to be large, the Von Mises distribution likely approaches the Uniform distribution due to a higher chance of sampling a small value for  $\kappa$ .

The weights we sample from a normalized product of a zero-deflated Bernoulli distribution and a Dirichlet distribution (Eq. 7.4).

$$f(w_i|p, \alpha_i) = \mathcal{B}(p)\mathcal{D}(\alpha_i)/Z \quad (7.4)$$

1796 The Bernoulli distribution samples zeros and ones with probability of  $p = 0.5$ , leading to  
 1797 3-vectors like 001, 101, etc. The distribution is corrected in such way that the chance to  
 1798 sample 000 is zero. The Bernoulli distribution only gives weights  $w_i = 0$  or  $w_i = 1$ , hence  
 1799 it is multiplied with a Dirichlet distribution. The Dirichlet samples a 3-vector with weights  
 1800 between zero and one where the weights  $\sum_i w_i = 1$ . A symmetric Dirichlet distribution  
 1801 with  $\alpha = 1$  is similar to the Uniform distribution over the simplex. It is set slightly more  
 1802 towards favoring particular distributions with  $\alpha_i = 1/3$  (we assume that people sample one  
 1803 or two distributions, and rarely from all three). The product with the zero-deflated Bernoulli  
 1804 distribution is made up to sum up to one again by normalizing the result.

1805 We can combine Eq. 7.3 and Eq. 7.4 in Eq. 7.5:

$$\theta \sim \mathcal{U}(a, b)\mathcal{E}(\lambda)\mathcal{B}(p)\mathcal{D}(\alpha_i)/Z \quad (7.5)$$

1806 To sample the parameters  $\theta$  this is the base distribution we will encounter in the next sec-  
 1807 tion 7.3.

## 1808 7.3 Model of Groups

1809 Each person's exercise schedule is represented by a Von-Mises-Uniform distribution. People  
 1810 that are similar do have exercise schedules that can be represented by the same Von-Mises-  
 1811 Uniform distribution. To group similar schedules we define a nonparametric discrete distri-  
 1812 bution over a potentially infinite number of groups with each person assigned to a group.

1813 A Dirichlet Process (Eq. 7.6) is a distribution over distributions that can be used as a prior  
 1814 for such a nonparametric discrete distribution.

$$DP(\alpha, H) \quad (7.6)$$

1815 The Dirichlet Process has (1) a hyperparameter  $\alpha$ , which defines the likelihood that there are  
 1816 many clusters versus few clusters (although it doesn't say anything about its actual count),  
 1817 and (2) a base distribution  $H$ , the distribution that generates  $\theta$  (Eq.7.5).

## 1818 7.4 Inference

1819 The implementation of the model makes use of Gibbs sampling with auxiliary variables Jain  
 1820 and Neal (2007).

1821 Details on this algorithm can be found in Jain and Neal (2007) and previous work of the  
 1822 authors van Rossum et al. (2016a,b).

**Algorithm 14** Gibbs sampling over auxiliary variables

---

```

1: procedure GIBBS ALGORITHM WITH AUXILIARY VARIABLES( $w, \lambda_0, \alpha$ )      ▷ Accepts schedule  $w$ ,
   hyperparameters  $\lambda_0, \alpha$ , number of auxiliary variables  $m$ , and returns  $k$  groups
2:   for all  $t = 1 : T$  do
3:     for all  $i = 1 : N$  do
4:       for all  $j = 1 : m$  do
5:          $\theta_j \sim H(\lambda_0)$                                           ▷ Sample  $\theta_j$  from base distribution  $H$  in Eq. 7.5
6:       end for
7:       for all  $j = 1 : K + m, j \neq i$  do
8:          $L_j = \text{likelihood}(w_i, \theta_j)$       ▷ Update likelihood for all theta (except  $\theta_i$ ) given  $w_i$ 
9:       end for
10:       $P_{-i=1:K} = b \sum_{-i} L_{-i}$                                           ▷ Calculate probability of existing cluster
11:       $P_{-i=K:K+m} = b\alpha/mL_mL_{-i}$       ▷ Calculate probability of new cluster
12:       $\theta_i = \theta_j$  according to above  $P_{-i}$       ▷ Sample  $\theta_i$  accord. to above prob
13:      Remove unused clusters
14:    end for
15:    for all  $j = 1 : K$  do
16:       $\theta_j \sim p(\theta_j | y)$                                           ▷ Update  $\theta_j$ 
17:    end for
18:  end for
19:  return summary on  $\theta_k$  for  $k$  groups of runners
20: end procedure

```

---

## 1823 7.5 Results

1824 The algorithm is run first on an artificial dataset of which we know the ground truth (Sect. 7.5.1)  
 1825 and next on the real-world dataset from Twitter (Sect. 7.5.2).

### 1826 7.5.1 Artificial Dataset

1827 The algorithm has been used on a self-generated dataset (generated from the probability  
 1828 density function as in Fig. 7.2). In this case we have the ground truth that establishes which  
 1829 exercise schedule comes from which probability density function. Using this ground truth  
 1830 we can calculate how often our algorithm makes a mistake, grouping a person with people  
 1831 that belong to another group. The results with this dataset are perfect (Rand Index equal to  
 1832 one: perfect clustering). The results are of the form of Table 7.2 (except by a permutation  
 1833 of indices), hence are not shown (the indices have no intrinsic meaning).

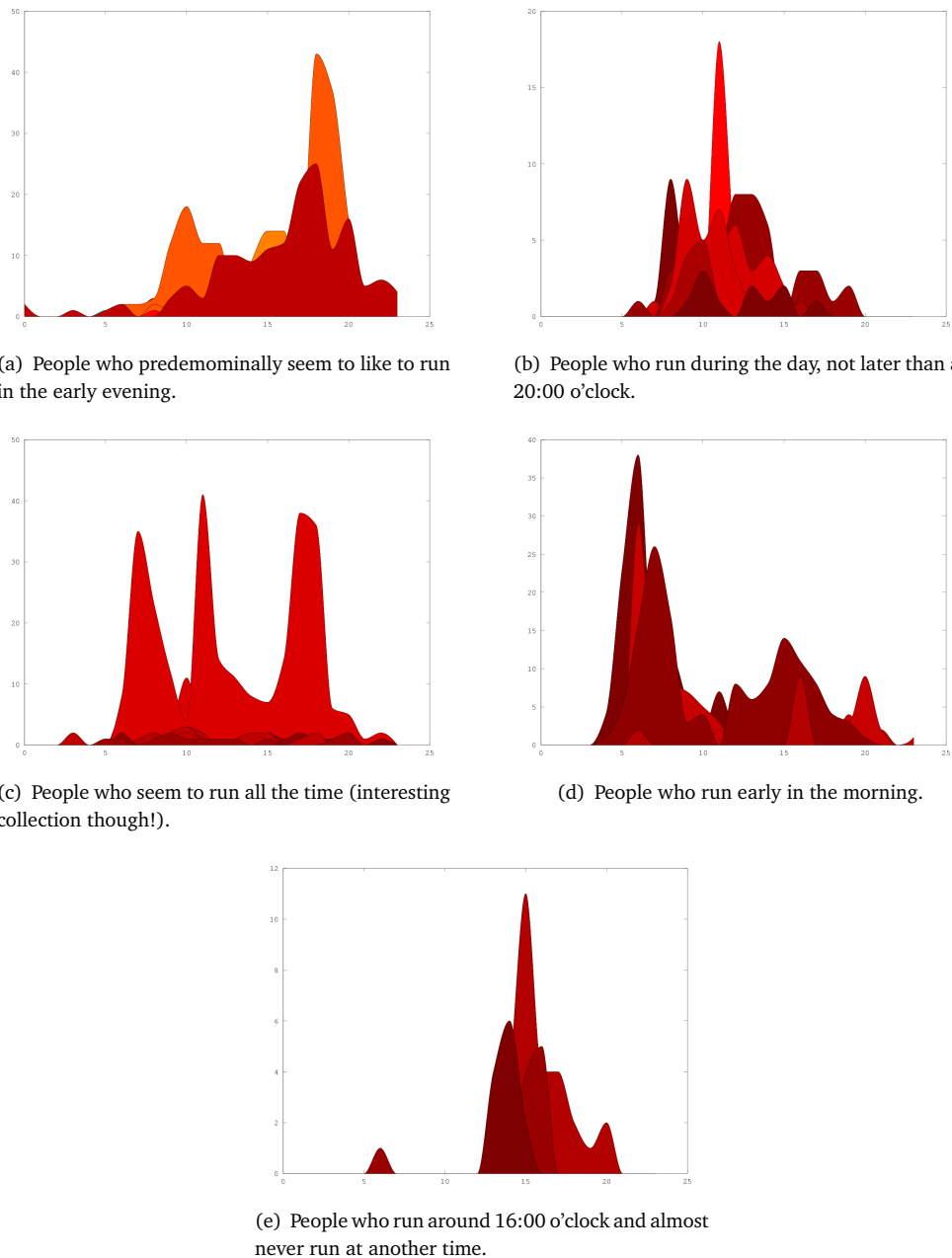
**Table 7.2:** Top row: A sequence of cluster indices indicates the ground truth. Each cluster index represents a multi-modal Von-Mises-Uniform distribution with different parameters  $\theta$ . Bottom row: A sequence of cluster indices that are the result of the described algorithm. Each cluster index represents again a multi-modal Von-Mises distribution. Errors would be represented by an inconsistent mapping from the top row to the bottom row.

1	4	4	5	2	2	2	1	3	2	1	5	1	1	4	4	3	1	3	1	2	...
4	5	5	1	2	2	2	4	3	2	4	1	4	4	5	5	3	4	3	4	2	...



### 1834 7.5.2 Real-world Dataset

1835 The results when we actually use the collected Twitter dataset can be best visualized (Fig. 7.3).  
 1836 This dataset consists of around 4000 moments at which people decide to run. The dataset  
 1837 is subsequently filtered on regulars, people that at least have run a few times.



**Figure 7.3:** After running the algorithm, the above figures show the different categories of runners that have been found.

1838 The results in Fig. 7.3 show that there is a categorization of people indeed. After the al-  
 1839 gorithm has done its work it is possible to assign a certain label in a post-hoc manner, e.g.  
 1840 “People who run around 16:00 o'clock and almost never run at another time.”.

## 1841 7.6 Discussion

1842 The postulated model allows to reason about groups of people performing exercising, with-  
1843 out predefining what these groups constitute apart from very general characteristics such  
1844 as that there might be preferred times of day to exercise. A Dirichlet process is used as a  
1845 nonparametric Bayesian model in which both the number of groups and the assignment of  
1846 people exercising are learnt from the data.

1847 There are several directions in which this research can be extended. First, more data would  
1848 be very helpful. Only a limited number of people are posting consistently their training  
1849 data online. Tapping into the data of current fitness promoting companies would allow the  
1850 model to wash out the prior a bit more and adjust to the data. Second, we also collected  
1851 weather data over this time period and also expect the day of the week to have significant  
1852 influence. When there will be more data available these are logically dimensions to include  
1853 in the dataset. Third, it would be interesting to study if people relate to the group of people  
1854 they have been categorized with. Can this help or support their exercise regime?



1855

CHAPTER  
8

1856

## DISCUSSION AND CONCLUSIONS



## REFERENCES

1857

1858

- 1859 M Abdel-Hameed. Optimal replacement policies for devices subject to a gamma wear process. *The*  
1860 *theory and applications of reliability*, pages 397–412, 2012.
- 1861 David J Aldous. Representations for partially exchangeable arrays of random variables. *Journal of*  
1862 *Multivariate Analysis*, 11(4):581–598, 1981.
- 1863 David J Aldous. *Exchangeability and related topics*. Springer, 1985.
- 1864 John Aldrich and Others. R.A. Fisher and the making of Maximum Likelihood 1912-1922. *Statistical*  
1865 *Science*, 12(3):162–176, 1997.
- 1866 Charles E Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric  
1867 problems. *The annals of statistics*, pages 1152–1174, 1974.
- 1868 Stefan Banach and Alfred Tarski. Sur la décomposition des ensembles de points en parties respec-  
1869 tivement congruentes. *Fund. math*, 6(1):924, 1924.
- 1870 Federico Bassetti, Roberto Casarin, and Fabrizio Leisen. Beta-product dependent Pitman–Yor pro-  
1871 cesses for Bayesian inference. *Journal of Econometrics*, 180(1):49–72, 2014.
- 1872 Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov  
1873 chains. *The annals of mathematical statistics*, 37(6):1554–1563, 1966.
- 1874 Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European*  
1875 *conference on computer vision*, pages 404–417. Springer, 2006.
- 1876 David Blackwell and James B MacQueen. Ferguson distributions via Pólya urn schemes. *The annals*  
1877 *of statistics*, pages 353–355, 1973.
- 1878 Phil Blunsom and Trevor Cohn. A hierarchical Pitman-Yor process HMM for unsupervised part of  
1879 speech induction. In *Proceedings of the 49th Annual Meeting of the Association for Computational*  
1880 *Linguistics: Human Language Technologies-Volume 1*, pages 865–874. Association for Computa-  
1881 tional Linguistics, 2011.
- 1882 Piotr Bojanowski, Armand Joulin, David Lopez-Paz, and Arthur Szlam. Optimizing the latent space  
1883 of generative networks. *arXiv preprint arXiv:1707.05776*, 2017.
- 1884 Robert C Bolles and Martin A Fischler. A RANSAC-Based Approach to Model Fitting and Its Application  
1885 to Finding Cylinders in Range Data. In *IJCAI*, volume 1981, pages 637–643, 1981.
- 1886 Anna Bosch, Andrew Zisserman, and Xavier Muñoz. Scene classification using a hybrid generative/dis-  
1887 criminative approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):712–  
1888 727, 2008.
- 1889 Guillaume Bouchard and Bill Triggs. The tradeoff between generative and discriminative classifiers.  
1890 In *16th IASC International Symposium on Computational Statistics (COMPSTAT'04)*, pages 721–  
1891 728, 2004.

- George E P Box and George C Tiao. *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons, 2011.
- H Bühlmann. *Austauschbare stochastische Variablen und ihre Grenzwertsätze*. PhD thesis, ETH Zürich, 1960.
- Wray L Buntine. Operations for learning with graphical models. *JAIR*, 2:159–225, 1994.
- John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698, 1986.
- Haifeng Chen, Peter Meer, and David E Tyler. Robust regression for data with multiple structures. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I—1069. IEEE, 2001.
- David B Dahl. An Improved Merge-Split Sampler for Conjugate Dirichlet Process Mixture Models. Technical report, University of Wisconsin–Madison, November 2003.
- David B Dahl. Sequentially-Allocated Merge-Split Sampler for Conjugate and Nonconjugate Dirichlet Process Mixture Models. Technical report, Texas A&M University, November 2005.
- Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 1, pages 886–893. IEEE, 2005.
- Hal Daume. Fast search for Dirichlet process mixture models. In *International Conference on Artificial Intelligence and Statistics*, pages 83–90, 2007.
- B de Finetti. Funzione caratteristica di un fenomeno aleatorio. *Atti Reale Accademia Nazionale dei Lincei*, VI:86–133, 1930.
- Bruno De Finetti. La prévision: ses lois logiques, ses sources subjectives. In *Annales de l’institut Henri Poincaré*, volume 7, pages 1–68, 1937.
- Persi Diaconis and David Freedman. de Finetti’s theorem for Markov chains. *The Annals of Probability*, pages 115–130, 1980.
- Alexey Dosovitskiy, Jost Tobias Springenberg, Maxim Tatarchenko, and Thomas Brox. Learning to generate chairs, tables and cars with convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):692–705, 2017.
- François Dufresne, Hans U Gerber, and Elias S W Shiu. Risk theory with the gamma process. *Astin Bulletin*, 21(02):177–192, 1991.
- David B Dunson, Ya Xue, and Lawrence Carin. The matrix stick-breaking process. *Journal of the American Statistical Association*, 2012.
- Richard Durstenfeld. Algorithm 235: Random Permutation. *Communications of the ACM*, 7(7):420, July 1964. ISSN 0001-0782. doi: 10.1145/364520.364540. URL <http://doi.acm.org/10.1145/364520.364540>.
- Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the american statistical association*, 90(430):577–588, 1995.
- Stewart N Ethier. The distribution of the frequencies of age-ordered alleles in a diffusion model. *Advances in Applied Probability*, pages 519–532, 1990.
- Warren John Ewens. Population genetics theory-the past and the future. In *Mathematical and statistical developments of evolutionary theory*, pages 177–227. Springer, 1990.
- Stefano Favaro, Yee Whye Teh, and Others. MCMC for normalized random measure mixture models. *Statistical Science*, 28(3):335–359, 2013.
- William Feller. An introduction to probability theory and its applications. Vol. I. 1950.
- Thomas S Ferguson. A Bayesian analysis of some nonparametric problems. *The annals of statistics*, pages 209–230, 1973.
- Thomas S Ferguson. Prior distributions on spaces of probability measures. *The annals of statistics*, pages 615–629, 1974.

- 1940 E.W. Forgy. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications.  
1941 *Biometrics*, 21:768–769, 1965.
- 1942 Wolfgang Förstner and Eberhard Gülch. A fast operator for detection and precise location of distinct  
1943 points, corners and centres of circular features. In *Proc. ISPRS intercommission conference on fast*  
1944 *processing of photogrammetric data*, pages 281–305, 1987.
- 1945 David Freedman and Persi Diaconis. On inconsistent Bayes estimates in the discrete case. *The Annals*  
1946 *of Statistics*, pages 1109–1118, 1983.
- 1947 David Heaver Fremlin. *Measure theory*, volume 4. Torres Fremlin, 2000.
- 1948 Orazio Gallo, Roberto Manduchi, and Abbas Rafii. CC-RANSAC: Fitting planes in the presence of  
1949 multiple surfaces in range data. *Pattern Recognition Letters*, 32(3):403–410, 2011.
- 1950 Qing-Bin Gao and Shi-Liang Sun. Human activity recognition with beta process hidden Markov  
1951 models. In *Machine Learning and Cybernetics (ICMLC), 2013 International Conference on*, volume 2,  
1952 pages 549–554. IEEE, 2013.
- 1953 Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian  
1954 restoration of images. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):721–  
1955 741, 1984.
- 1956 Zoubin Ghahramani and Thomas L Griffiths. Infinite latent feature models and the Indian buffet  
1957 process. In *Advances in neural information processing systems*, pages 475–482, 2005.
- 1958 Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron  
1959 Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information*  
1960 *processing systems*, pages 2672–2680, 2014.
- 1961 Anirudh Goyal, Nan Rosemary Ke, Surya Ganguli, and Yoshua Bengio. Variational walkback: Learning  
1962 a transition operator as a stochastic recurrent net. *arXiv preprint arXiv:1711.02282*, 2017.
- 1963 Max Halperin and G L Burrows. The Effect of Sequential Batching for Acceptance–Rejection Sam-  
1964 pling Upon Sample Assurance of Total Product Quality. *Technometrics*, 2(1):19–26, 1960.
- 1965 Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*,  
1966 volume 15, page 50. Citeseer, 1988.
- 1967 Li He, Hairong Qi, and Russell Zaretzki. Beta process joint dictionary learning for coupled feature  
1968 spaces with application to single image super-resolution. In *Proceedings of the IEEE Conference on*  
1969 *Computer Vision and Pattern Recognition*, pages 345–352, 2013.
- 1970 Nils Lid Hjort. Nonparametric Bayes estimators based on beta processes in models for life history  
1971 data. *The Annals of Statistics*, pages 1259–1294, 1990.
- 1972 Fred M Hoppe. Size-biased filtering of Poisson-Dirichlet samples with an application to partition  
1973 structures in genetics. *Journal of Applied Probability*, pages 1008–1012, 1986.
- 1974 Paul V.C. Hough. Method and Means for Recognizing Complex Patterns, Dec 1962. URL <https://www.google.com/patents/US3069654>. Patent US 3069654 A.
- 1976 Xun Huang, Yixuan Li, Omid Poursaeed, John Hopcroft, and Serge Belongie. Stacked generative  
1977 adversarial networks. *arXiv preprint arXiv:1612.04357*, 2016.
- 1978 Tommi S Jaakkola, David Haussler, and Others. Exploiting generative models in discriminative clas-  
1979 sifiers. *Advances in neural information processing systems*, pages 487–493, 1999.
- 1980 Sonia Jain and Radford M. Neal. A Split-Merge Markov Chain Monte Carlo Procedure for the Dirichlet  
1981 Process Mixture Model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004.  
1982 ISSN 10618600. URL <http://www.jstor.org/stable/1391150>.
- 1983 Sonia Jain and Radford M Neal. Splitting and Merging Components of a Nonconjugate Dirichlet  
1984 Process Mixture Model. *Bayesian Analysis*, 2(3):445–472, 2007.
- 1985 Varun Jampani, Sebastian Nowozin, Matthew Loper, and Peter V Gehler. The informed sampler: A  
1986 discriminative approach to bayesian inference in generative computer vision models. *Computer*  
1987 *Vision and Image Understanding*, 136:32–44, 2015.
- 1988 Edwin T Jaynes. *Probability theory: The logic of science*. Cambridge university press, 2003.



- 1989 Dominik Joho, Gian Diego Tipaldi, Nikolas Engelhard, Cyrill Stachniss, Wolfram Burgard, Mar-  
1990 tin Senk, Felix Faber, Maren Bennewitz, Clemens Eppner, Attila Görög, and Others. Unsuper-  
1991 visued Scene Analysis and Reconstruction Using Nonparametric Bayesian Models. *Robotics and*  
1992 *Autonomous Systems (RAS)*, 59(5):319–328, 2011.
- 1993 A Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive  
1994 Bayes. *Advances in neural information processing systems*, 14:841, 2002.
- 1995 Michael I Jordan. Hierarchical models, nested models and completely random measures. *Frontiers of*  
1996 *Statistical Decision Making and Bayesian Analysis: in Honor of James O. Berger*. New York: Springer,  
1997 2010.
- 1998 Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*  
1999 *arXiv:1312.6114*, 2013.
- 2000 J F C Kingman. *Poisson processes*, volume 3 of *Oxford Studies in Probability*. The Clarendon Press  
2001 Oxford University Press, New York, 1993. ISBN 0-19-853693-3.
- 2002 J F C Kingman. Some further analytical results in the theory of regenerative events. *Journal of*  
2003 *Mathematical Analysis and Applications*, 11:422–433, 1965.
- 2004 J F C Kingman. Random partitions in population genetics. In *Proceedings of the Royal Society of*  
2005 *London A: Mathematical, Physical and Engineering Sciences*, volume 361, pages 1–20. The Royal  
2006 Society, 1978.
- 2007 John Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.
- 2008 Roman Klovov and Victor Lempitsky. Escape from cells: Deep kd-networks for the recognition of  
2009 3d point cloud models. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages  
2010 863–872. IEEE, 2017.
- 2011 David Knowles, Zoubin Ghahramani, and Konstantina Palla. A reversible infinite HMM using nor-  
2012 malised random measures. In *Proceedings of the 31st International Conference on Machine Learning*  
2013 *(ICML-14)*, pages 1998–2006, 2014.
- 2014 Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press,  
2015 2009.
- 2016 A Kolmogorov. *Grundbegriffe der wahrscheinlichkeitsrechnung*, volume 2 of *Ergebnisse der Mathematik*  
2017 *und ihrer Grenzgebiete*. Springer-Verlag, 1933.
- 2018 Uwe Küchler and Stefan Tappe. Bilateral Gamma distributions and processes in financial mathemat-  
2019 ics. *Stochastic Processes and their Applications*, 118(2):261–283, 2008.
- 2020 Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed Variational Dirichlet Process Mixture  
2021 Models. In *IJCAI*, volume 7, pages 2796–2801, 2007.
- 2022 Pierre-Simon Laplace. *Théorie analytique des probabilités*. V. Courcier, 1820.
- 2023 Henri Lebesgue. Intégrale, longueur, aire. *Annali di Matematica Pura ed Applicata (1898-1922)*, 7  
2024 (1):231–359, 1902.
- 2025 Antonio Lijoi and Igor Prünster. Models beyond the Dirichlet process. *Bayesian nonparametrics*, 28:  
2026 80, 2010.
- 2027 S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inform. Theory*, 28:129–137, 1982. Originally  
2028 as an unpublished Bell laboratories Technical Note (1957).
- 2029 David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999.*  
2030 *The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee,  
2031 1999.
- 2032 Steven N MacEachern. Comment on "Splitting and Merging Components of a Nonconjugate Dirichlet  
2033 Process Mixture Model" by Jain and Neal. *Bayesian Analysis*, 2(3):483–494, 2007.
- 2034 Steven N MacEachern and Peter Müller. Estimating mixture of Dirichlet process models. *Journal of*  
2035 *Computational and Graphical Statistics*, 7(2):223–238, 1998.
- 2036 Dilip B Madan and Eugene Seneta. The variance gamma (VG) model for share market returns. *Journal*  
2037 *of business*, pages 511–524, 1990.

- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. doi: 10.1063/1.1699114.
- Thomas Minka. Bayesian linear regression. Technical report, Citeseer, 2000.
- Radford M Neal. Defining Priors for Distributions Using Dirichlet Diffusion Trees. Technical report, University of Toronto, Toronto, Ontario, Canada, 2001.
- Radford M Neal. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.
- Peter Orbanz and Daniel M Roy. Bayesian models of graphs, arrays and other exchangeable random structures. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(2):437–461, 2015.
- Jim Pitman and Marc Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, pages 855–900, 1997.
- Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5105–5114, 2017.
- Rajat Raina, Yirong Shen, Andrew McCallum, and Andrew Y Ng. Classification with hybrid generative/discriminative models. In *Advances in neural information processing systems*, page None, 2003.
- William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.
- Vinayak Rao and Yee W Teh. Spatial normalized gamma processes. In *Advances in neural information processing systems*, pages 1554–1562, 2009.
- Carl Edward Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*, volume 2. 2006.
- Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. Deep learning with sets and point clouds. *arXiv preprint arXiv:1611.04500*, 2016.
- Eugenio Regazzini, Antonio Lijoi, and Igor Prünster. Distributional results for means of normalized random measures with independent increments. *Annals of Statistics*, pages 560–585, 2003.
- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.
- Daniel M Roy and Yee W Teh. The mondrian process. In *Advances in neural information processing systems*, pages 1377–1384, 2009.
- Donald B Rubin and Others. Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4):1151–1172, 1984.
- Stuart Russell, Peter Norvig, and Artificial Intelligence. Artificial Intelligence: A modern approach. *Artificial Intelligence. Prentice-Hall, Englewood Cliffs*, 25:27, 1995.
- Leonard J Savage. *The foundations of statistics*. Courier Corporation, 1972.
- Stanley Sawyer and Daniel Hartl. A sampling theory for local selection. *Journal of Genetics*, 64(1): 21–29, 1985.
- René L Schilling. *Measures, integrals and martingales*, volume 13. Cambridge University Press, 2005.
- Abhishek Sharma, Oliver Grau, and Mario Fritz. Vconv-dae: Deep volumetric shape learning without object labels. In *European Conference on Computer Vision*, pages 236–250. Springer, 2016.
- Jianbo Shi and Carlo Tomasi. Good features to track. In *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94., 1994 IEEE Computer Society Conference on*, pages 593–600. IEEE, 1994.
- Nozer Singpurwalla. Gamma processes and their generalizations: an overview. In *Engineering probabilistic design and maintenance for flood protection*, pages 67–75. Springer, 1997.

- 2085 Scott A Sisson and Yanan Fan. Likelihood-free MCMC. *Handbook of Monte Carlo*, ed. Brooks, A.,  
2086 Gelman, A., Jones, GL, Meng XL, pages 313–335, 2011.
- 2087 I Sobel. *Camera Models and Perception*. PhD thesis, Stanford University, Stanford, CA, 1970.
- 2088 Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised  
2089 learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.
- 2090 Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-nice-mc: Adversarial training for mcmc. *arXiv*  
2091 *preprint arXiv:1706.07561*, 2017.
- 2092 Mike Steel. The maximum likelihood point for a phylogenetic tree is not unique. *Systematic Biology*,  
2093 43(4):560–564, 1994.
- 2094 Erik B Sudderth and Michael I Jordan. Shared segmentation of natural scenes using dependent  
2095 Pitman-Yor processes. In *Advances in Neural Information Processing Systems*, pages 1585–1592,  
2096 2009.
- 2097 Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Machine Learn-*  
2098 *ing*, 4(4):267–373, 2011.
- 2099 Terence Tao. *An introduction to measure theory*, volume 126. American Mathematical Soc., 2011.
- 2100 Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet processes.  
2101 *Journal of the American statistical association*, 101(476), 2006.
- 2102 Romain Thibaux and Michael I Jordan. Hierarchical beta processes and the Indian buffet process. In  
2103 *International Conference on Artificial Intelligence and Statistics*, pages 564–571, 2007.
- 2104 Michalis K Titsias. The infinite gamma-Poisson feature model. In *Advances in Neural Information*  
2105 *Processing Systems*, pages 1513–1520, 2008.
- 2106 Dustin Tran, Matthew D Hoffman, Rif A Saurous, Eugene Brevdo, Kevin Murphy, and David M Blei.  
2107 Deep probabilistic programming. *arXiv preprint arXiv:1701.03757*, 2017.
- 2108 Anne C. van Rossum, Hai Xiang Lin, Johan Dubbeldam, and H. Jaap van den Herik. Nonparametric  
2109 Bayesian Line Detection - Towards Proper Priors for Robotic Computer Vision. In *Proceedings of*  
2110 *the 5th International Conference on Pattern Recognition Applications and Methods*, pages 119–127,  
2111 Feb 2016a. ISBN 978-989-758-173-1. doi: 10.5220/0005673301190127.
- 2112 Anne C. van Rossum, Hai Xiang Lin, Johan Dubbeldam, and H. Jaap van den Herik. Nonparametric  
2113 Segment Detection. In *Proceedings of the 8th European Starting AI Researcher Symposium (STAIRS)*,  
2114 pages 203–208, Aug 2016b. ISBN 978-989-758-173-1. doi: 10.5220/0005673301190127.
- 2115 Niklas Vanhainen and Giampiero Salvi. Word Discovery with Beta Process Factor Analysis. In *INTER-*  
2116 *SPEECH*, pages 799–802, 2012.
- 2117 Wei Wang and Stuart Russell. A Smart-dumb/Dumb-smart Algorithm for Efficient Split-merge MCMC.  
2118 In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, UAI15, pages  
2119 902–911, Arlington, Virginia, United States, 2015. AUAI Press. ISBN 978-0-9966431-0-8. URL  
2120 <http://dl.acm.org/citation.cfm?id=3020847.3020940>.
- 2121 Yingjian Wang and Lawrence Carin. Levy measure decompositions for the beta and gamma processes.  
2122 *arXiv preprint arXiv:1206.4615*, 2012.
- 2123 Larry Wasserman. Asymptotic properties of nonparametric Bayesian procedures. In *Practical non-*  
2124 *parametric and semiparametric Bayesian statistics*, pages 293–304. Springer, 1998.
- 2125 Robert L Wolpert and Katja Ickstadt. Poisson/gamma random field models for spatial statistics.  
2126 *Biometrika*, 85(2):251–267, 1998.
- 2127 Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.
- 2128 Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong  
2129 Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE*  
2130 *conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- 2131 Jing-Hao Xue and D Michael Titterton. Comment on “A discriminative vs. generative clas-  
2132 sifiers: A comparison of logistic regression and naive Bayes”. *Neural processing letters*, 28(3):  
2133 169–187, 2008.

- 2134 Arnold Zellner. Optimal information processing and Bayes's theorem. *The American Statistician*, 42  
2135 (4):278–280, 1988.
- 2136 Michael M Zhang and Fernando Perez-Cruz. Accelerated inference for latent variable models. *arXiv*  
2137 *preprint arXiv:1705.07178*, 2017.
- 2138 Wei Zhang and Jana Křsecká. Nonparametric estimation of multiple structures with outliers. In  
2139 *Dynamical Vision*, pages 60–74. Springer, 2007.
- 2140 Mingyuan Zhou, Hongxia Yang, Guillermo Sapiro, David B Dunson, and Lawrence Carin. Dependent  
2141 hierarchical beta process for image interpolation and denoising. In *International conference on*  
2142 *artificial intelligence and statistics*, pages 883–891, 2011.





## PROBABILISTIC CONCEPTS

**TODO:** What follows down here is old and has to be adapted to the measure-theoretic realizations.

Let us introduce the notation of expectation for random variable  $X$ :

$$E_p[X] = \sum_i^k p(X = x_i)x_i \quad (\text{A.1})$$

For the continuous case, if  $f_X(x)$  is a properly defined probability density function, the expected value becomes:

$$E_f[X] = \int_{-\infty}^{\infty} x f_X(x) dx \quad (\text{A.2})$$

From the context it can be seen that the object  $X$  at the left is a different object than  $x$  at the right. The former is random variable (a one-dimensional function, an (in)finite vector), the latter is a value of a random variable (a scalar). The term  $dx$  is a measure, in this case it assigns volume to *subsets of random variables values*. A proper notation would incorporate this aspect, but not much will be gained by creating such complex notations.

Let us also introduce the conditional probability:

$$p(X|Y) = \frac{p(X, Y)}{p(Y)} \quad (\text{A.3})$$

Suppose  $X$  is a discrete random variable, then the object  $p(X)$  is a vector of finite size  $k$ ,  $p(Y)$  is a vector of finite size  $l$ , and  $p(X|Y)$  as well as  $p(X, Y)$  are matrices of size  $k \times l$ . A conditional probability hence ‘divides’ a matrix by a vector. It is a proper measure if  $p(Y) \neq 0$  (for all values of - or events in -  $Y$ ).

2159 In for example importance sampling (Sect. ??), the expectation is taken over a function of  
 2160 a random variable. A function, if measurable, can be taken the expectation over using the  
 2161 so-called law of the unconscious statistician:

$$E_f[g(X)] = \int_{-\infty}^{\infty} g(x)f_X(x)dx \quad (\text{A.4})$$

2162 Here  $g(X)$  is a general measurable function, and not restricted to a probability density func-  
 2163 tion.

2164 The notation above is an indefinite integral. We can approach this integral by Monte Carlo  
 2165 integration (Sect. ??).

$$E_f[g(X)] = \frac{1}{k} \sum_{i=1}^k g(x_i) \quad \text{with} \quad x_i \sim f_X(x) \quad (\text{A.5})$$

2166 Rather than summing over  $f_X(x_i)$ , we now sample  $x_i \sim f_X(x)$ .

## 2167 **A.1 Common Inequalities**

### 2168 **A.1.1 Markov's Inequality**

2169 Markov's inequality comes up with an upper bound for the probability that an non-negative  
 2170 random variable  $X$  exceeds some constant positive threshold  $a$ .

$$p(X \geq a) \leq \frac{E[X]}{a} \quad (\text{A.6})$$

2171 The proof in classical probability theory uses an indicator variable:

$$aI(X \geq a) \leq X \quad (\text{A.7})$$

2172 Here  $I(X \geq a) = 1$  if the event  $X \geq a$  occurs, setting the left-hand side to  $a$  (which is of  
 2173 course smaller than  $X$ ). And  $I(X \geq a) = 0$  on the event  $X < a$ , which is naturally smaller  
 2174 than the non-negative  $X$ .

2175 Expectations obey the inequality: if  $(X \leq Y)$ , then  $E[X] \leq E[Y]$ , hence:

$$E[aI(X \geq a)] \leq E[X] \quad (\text{A.8})$$

2176 And because expectations add up linearly:

$$E[aI(X \geq a)] = aE[I(X \geq a)] = a(1 \cdot p(X \geq a) + 0 \cdot p(X < a)) = a \cdot p(X \geq a) \quad (\text{A.9})$$

2177 So, we have Markov's inequality combining Eq. A.8 and A.9:

$$a \cdot p(X \geq a) \leq E[X] \quad (\text{A.10})$$

### 2178 A.1.2 Chebyshev's Inequality

2179 Now, Chebyshev's inequality defines in a similar way (Chebyshev was a teacher of Markov<sup>1</sup>)  
 2180 an upper bound on the deviation from the mean for a random variable. Recall the definition  
 2181 of the variance of  $X$  and assume it is finite:

$$\text{Var}(X) = E[(X - E[X])^2] = \sigma^2 \quad (\text{A.11})$$

2182 Consider now the random variable  $(X - E[X])^2$  and constant  $a = (\sigma k)^2$  and write down  
 2183 Markov's inequality:

$$p((X - E[X])^2 \geq (\sigma k)^2) \leq \frac{E[(X - E[X])^2]}{(\sigma k)^2} \quad (\text{A.12})$$

2184 Taking the square root of the inequality at the left, and using the definition of  $\sigma^2$  at the  
 2185 right, leads to:

$$p(|X - E[X]| \geq \sigma k) \leq \frac{1}{k^2} \quad (\text{A.13})$$

### 2186 A.1.3 Weak Law of Large Numbers

2187 Chebyshev's inequality can be used to prove the weak law of large numbers. Given that we  
 2188 have a series of random variables, all with the same finite expectation,  $E[X_i] = \mu$ , then this  
 2189 law states that the sample average  $\bar{X} = \frac{1}{n}(X_1 + \dots + X_n)$  converges in probability towards  
 2190 the expected value:

$$\bar{X} \xrightarrow{P} \mu \quad \text{for} \quad n \rightarrow \infty \quad (\text{A.14})$$

2191 We can use the independence assumption between variables  $X_i$  to write down the variance  
 2192 and expectation of  $\bar{X}$ :

$$\text{Var}(\bar{X}) = \text{Var}\left(\frac{1}{n}(X_1 + \dots + X_n)\right) = \frac{1}{n^2} \text{Var}(X_1 + \dots + X_n) = \frac{\sigma^2}{n} E[\bar{X}] = \mu \quad (\text{A.15})$$

---

<sup>1</sup>There were many mathematically gifted Markov's. This is Andrey Andreyevich Markov Sr., known from the Markov chains and Markov processes. Jr. is known from Markov's principle, Markov's rule and the Markov algorithm.



2193 And now we can apply Chebyshev's inequality on  $\bar{X}$ :

$$p(|\bar{X} - \mu| \geq \epsilon) \leq \frac{\sigma^2}{n\epsilon^2} \quad (\text{A.16})$$

2194 Convergence in probability towards  $X$  is the case if for all  $\epsilon$ :

$$\lim_{n \rightarrow \infty} p(|\bar{X} - X| \geq \epsilon) = 0 \quad (\text{A.17})$$

2195 This is the case for  $n \rightarrow \infty$  indeed.

#### 2196 **A.1.4 Strong Law of Large Numbers**

2197 The strong law incorporates the weak law. Rather than convergence *in probability*, it states  
2198 convergence *almost surely* towards the expected value.

$$\bar{X} \xrightarrow{a.s.} \mu \quad \text{for} \quad n \rightarrow \infty \quad (\text{A.18})$$

2199 The strong law states that with probability 1, for any  $\epsilon > 0$ , the inequality  $|X - \mu| < \epsilon$   
2200 holds for large enough  $n$ . The weak law states only that the average  $\bar{X}$  is likely near  $\mu$ , but  
2201  $|X - \mu| \geq \epsilon$  can still happen, even for large  $n$ .

#### 2202 **A.1.5 Common Distributions**

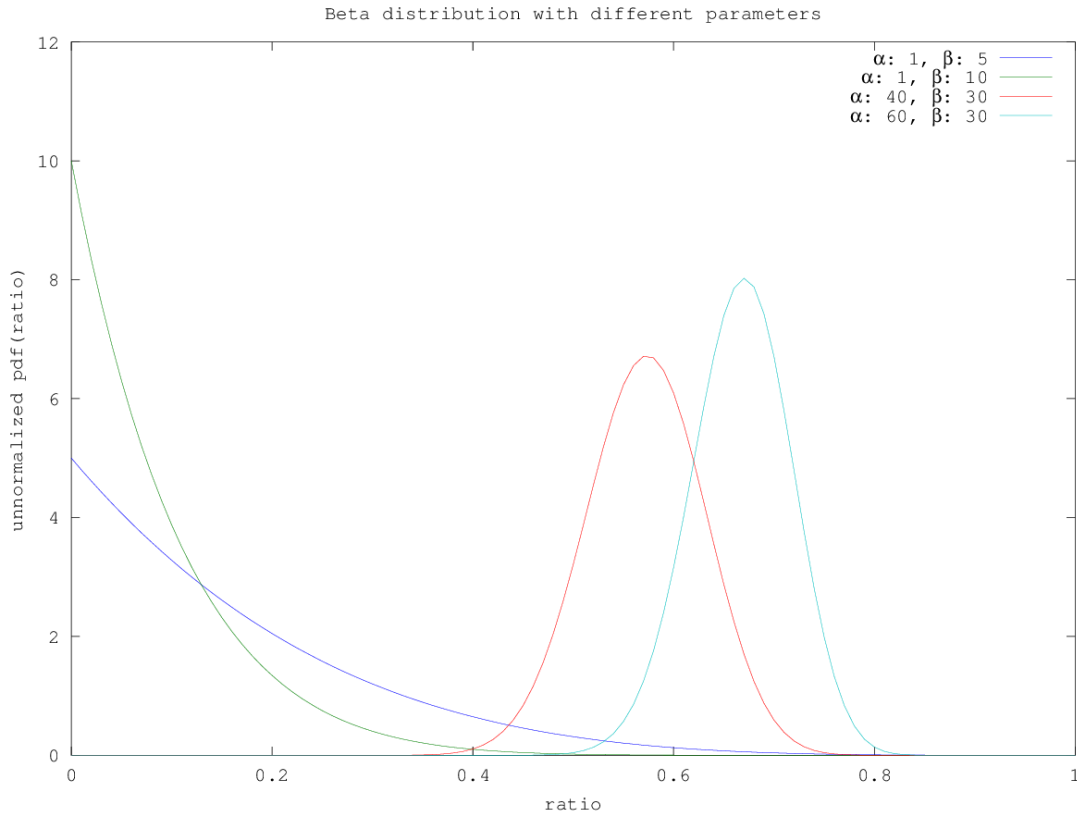
2203 One of the probability distributions that is interesting to us is the beta-distribution. A normal  
2204 distribution might be a reasonable prior for a continuous variable such as human heights in  
2205 a population. If this variable however is itself a probability, a reasonable prior is the beta-  
2206 distribution. The beta-distribution can be described as:

$$f(x, \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1} = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1} \quad (\text{A.19})$$

2207 Here  $B$  is the beta function and  $\Gamma$  is the gamma function, the continuous extension of the  
2208 factorial function:  $\Gamma(n) = (n-1)!$ . Naturally, there are many of such extensions. The  
2209 gamma function extends the factorial in a specific sense. It obeys the recurrence relation  
2210  $f(x+1) = xf(x)$  with  $f(1) = 1$ . Its description is defined with an improper integral:

$$\Gamma(t) = \int_0^\infty x^{t-1} e^{-x} dx \quad (\text{A.20})$$

2211 The expected value of a random variable  $X$  with a beta-distribution:



**Figure A.1:** The beta-distribution with different parameters. The x-axis is the quantity modelled, for example a ratio between wins and losses in a soccer season. The y-axis is the corresponding unnormalized density function. Setting  $\alpha = 1$  shows a monotonically decreasing density function. Having a variable  $\alpha$  allows probability mass to shift to the end.

$$E_f[X] = \int_0^1 x f(x, \alpha, \beta) dx = \int_0^1 \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^\alpha (1-x)^{\beta-1} dx = \frac{\alpha}{\alpha + \beta} \quad (\text{A.21})$$

2212 Let us illustrate the effects of the parameters of the beta-distribution.

2213 The stick-breaking presentation shows that a Dirichlet process consists of beta processes with  
 2214  $\alpha = 1$ . The Pitman-Yor process has  $\alpha$  left variable. Informally, the location where the stick  
 2215 will be broken (iteratively) for the Dirichlet process is ‘quite close to the beginning’ with high  
 2216 probability. In the case  $\alpha$  becomes larger, the breaking can also occur likely at the end of the  
 2217 stick. This means for breaking a stick, say 20 times, the distribution of stick lengths for the  
 2218 Pitman-Yor process has a much larger support. The difference between the large and small  
 2219 sticks is much larger.

## DIRICHLET-MULTINOMIAL

There is confusion in the literature with respect to the Dirichlet-multinomial. There are three ways to interpret this.

### B.1 Categorical Distribution

Often, the Dirichlet-multinomial is actually not a compound Dirichlet and a multinomial, but a compound Dirichlet and **categorical** distribution:

$$p(z|\theta) = \prod_i \theta_i^{z_i} \quad (\text{B.1})$$

This means that this is about only one categorical variable, not a set. The notation of above would for dice assign the vector  $[1, 0, 0, 0, 0, 0]$  to the face with one pip,  $[0, 1, 0, 0, 0, 0]$  to the face with two pips, etc. Naturally, this means that  $\sum_i z_i = 1$ .

This gets rid off the  $\frac{n!}{\prod_i z_i!}$  factor and leads to the much shorter:

$$p(z|\theta)p(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_i \theta_i^{z_i + \alpha_i - 1} \quad (\text{B.2})$$

To subsequently derive at the Dirichlet-multinomial, you'll have to integrate over:

$$\int p(z|\theta)p(\theta|\alpha)d\theta = \frac{1}{B(\alpha)} \int \prod_i \theta_i^{z_i + \alpha_i - 1} d\theta \quad (\text{B.3})$$

Now, the Dirichlet didn't come from nowhere... The factor  $B(\alpha)$  is a normalization factor:

$$p(\theta|\alpha) = \frac{1}{B(\alpha)} \prod_i \theta_i^{\alpha_i-1} = \frac{\prod_i \theta_i^{\alpha_i-1}}{\int_{\Delta^n} \prod_i \theta_i^{\alpha_i-1} d\theta} \quad (\text{B.4})$$

2233 with  $\int_{\Delta^n}$  corresponding to the condition  $\sum_i \theta_i = 1$ .

2234 In other words, the multivariate Beta function is actually this integral directly from the def-  
2235 inition:

$$\int_{\Delta^n} \prod_i \theta_i^{\alpha_i-1} d\theta = B(\alpha) \quad (\text{B.5})$$

2236 And hence the integral:

$$\int \prod_i \theta_i^{z_i+\alpha_i-1} d\theta = B(\alpha + z) \quad (\text{B.6})$$

2237 Hence:

$$\int p(z|\theta)p(\theta|\alpha)d\theta = \frac{B(\alpha + z)}{B(\alpha)} \quad (\text{B.7})$$

2238 Or to end up with something commonly stated as the Dirichlet-multinomial:

$$\int p(z|\theta)p(\theta|\alpha)d\theta = \frac{\prod_i \Gamma(\alpha_i + z_i)}{\Gamma(\sum_i (\alpha_i + z_i))} \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \quad (\text{B.8})$$

2239 Collecting terms:

$$\int p(z|\theta)p(\theta|\alpha)d\theta = \frac{\Gamma(\sum_i \alpha_i)}{\Gamma(\sum_i \alpha_i + \sum_i z_i)} \prod_i \frac{\Gamma(\alpha_i + z_i)}{\Gamma(\alpha_i)} \quad (\text{B.9})$$

2240 Note, however that we run  $i$  here over the entries in our categorical variable  $z$  represented  
2241 as a vector! This is very different from a multinomial distribution over a set of variables!

## 2242 **B.2 Multinomial Distribution**

2243 In case of an actual multinomial distribution, counts of  $z$ , let's write them  $n(z)$  are actually  
2244 the topic of consideration, not  $z$  itself.

$$p(z|\theta) = \frac{(\sum_k n(z_k))!}{\prod_k (n(z_k)!) } \prod_k \theta_k^{n(z_k)} \quad (\text{B.10})$$

2245 We now run over  $k$  unique variables, not over a vectorized categorical variable.

2246 Of course, we can now again multiply with a Dirichlet distribution and the derivation is  
 2247 along the lines as described before. The result:

$$\int p(z|\theta)p(\theta|\alpha)d\theta = \frac{(\sum_k n(z_k))!}{\prod_k (n(z_k))!} \frac{\Gamma(\sum_k \alpha_k)}{\Gamma(\sum_k \alpha_k + \sum_k n(z_k))} \prod_k \frac{\Gamma(\alpha_k + n(z_k))}{\Gamma(\alpha_k)} \quad (\text{B.11})$$

2248 This might not be pretty, but it is the actual full Dirichlet-multinomial.

## 2249 B.3 N Categorical Distributions

2250 The third option, and this is meant most times is the distribution of a **sequence** of categorical  
 2251 variables. Recall that the multinomial assigns probabilities to the **number** of extracted balls  
 2252 (in an experiment getting n balls out of a bag with k ball types). A sequence of categorical  
 2253 variables assigns a probability to a **sequence** and has a form without the normalization  
 2254 factor:

$$p(z|\theta) = \prod_k \theta_k^{z_k} \quad (\text{B.12})$$

2255 Here  $k$  runs over the categories. We can now follow the derivation as with the single cate-  
 2256 gorical variable.



## GIBBS SAMPLING

2257

2258

2259 Notation:

$$\int dF(x) = F(x) \quad (\text{C.1})$$

2260 A mixture model:

$$L(x) = \int dF(x) \mu(x) \quad (\text{C.2})$$

2261 If we have a particular form of  $F(x)$ , namely it admits a decomposition of a sum of individual  
 2262 values  $x_i$ :

$$F(x) = \sum_i \delta_{x_i} = \delta(x = x_0) + \delta(x = x_1) + \dots \quad (\text{C.3})$$

2263 Then our mixture model can be written as:

$$L(x) = \int dF(x) \mu(x) = \sum_i \mu(x_i) \quad (\text{C.4})$$

2264 Let  $x_0 = 3$ ,  $x_1 = 4$ ,  $\mu(x) = x^2$ , then  $L(x) = 3^2 + 4^2 = 25$ .2265 Walker with  $P = F$ ,  $\mu(x) = N(y|\theta)$ ,  $i = j$  and giving each  $\theta_j$  a weight  $\omega_j$ :

$$f_P(y) = \int dP(\theta) N(y|\theta) \quad P = \sum_j \omega_j \delta_{\theta_j} \quad (\text{C.5})$$

2266 Then:

$$f_{\omega,j}(y) = \sum_j \omega_j N(y|\theta_j) \quad (\text{C.6})$$

2267 The likelihood:

$$L(w | \alpha, \lambda_0) = p(\phi | \alpha) \prod_{i=0}^{N-1} \int p(w_i | \theta_i, \phi) dG_0(\theta_i) \quad (\text{C.7})$$

2268 Here the index runs over all data points  $w_i$ . Each data point corresponds to a line with  
 2269 parameters  $\theta_i$ . Here the parameters  $\theta_i$  and  $\theta_j$  for data point  $w_i$  and  $w_j$  can be the same and  
 2270 thus reflect the same line. The index for  $\theta$  runs over the  $N$  data points, not over the  $K$  lines.

2271 The distribution  $G_0$  does have hyperparameters  $\lambda_0$ .

2272 We can also group all data points that belong to the same line  $k$  together by reordering the  
 2273 product terms:

$$L(w | \alpha, \lambda_0) = p(\phi | \alpha) \prod_k \prod_{i:z_i=k} \int p(w_i | \theta_i, \phi) dG_0(\theta_i) \quad (\text{C.8})$$

2274 Here the factors that belong to line  $k$  are multiplied. The index still runs over the data points.

2275 It is also possible not to limit the second product to only the data points  $i$  that are assigned  
 2276 to line  $k$ .

$$L(w | \alpha, \lambda_0) = p(\phi | \alpha) \prod_k \prod_i p(z_i | \phi) \int p(w_i | \theta_i, \phi) dG_0(\theta_i) \quad (\text{C.9})$$

2277 Now, we are gonna introduce the stick-breaking sum, which turns our integral into a discrete  
 2278 sum.

2279  $G = \sum_l p_l \delta_{Z_l}$  with  $Z_l$  iid from  $G_0$  and  $p_l$  defined as a product of beta distributions.

$$L(w | \alpha, \lambda_0) = p(\phi | \alpha) \sum_k \prod_i p(z_i | \phi) p(w_i | \theta_k) p(\theta_k | \lambda_0) \quad (\text{C.10})$$

2280 The first term at the right hand side,  $p(\phi | \alpha)$ , generates the partition  $\phi$  by the Dirichlet  
 2281 Process with concentration parameter  $\alpha$ . The second term  $p(z_i | \phi)$  defines indices  $z_0, \dots, z_N$   
 2282 to link observations  $w_0, \dots, w_N$  with the parameters  $\theta_0, \dots, \theta_K$ . The probability  $p(w_i | \theta_k)$   
 2283 corresponds to the likelihood equations 3.10 and 3.11 with  $w_i$  the tuple of  $x_i$  and  $y_i$  and  
 2284  $\theta_k$  the line parameters  $\sigma_k^2$  and  $\beta_k$ . The probability  $p(\theta_k | \lambda_0)$  corresponds to the prior from  
 2285 equation 3.14. The parameters  $\theta_k$  (that is,  $\sigma_k^2$  and  $\beta_k$ ) are generated from hyperparameters  
 2286  $\lambda_0$ . The hyperparameters  $\lambda_0 = \{\mu_0, \Lambda_0, a, b\}$  are the parameters from the Normal-Inverse-  
 2287 Gamma prior.

2288 The Dirichlet process can be used as a mixture model (Antoniak, 1974; Escobar and West,  
 2289 1995; MacEachern and Müller, 1998) in which it generates (non-unique) parameters that  
 2290 subsequently generate observations:

$$\begin{aligned} G &\sim DP(\alpha, G_0) \\ \theta_i &| G \sim G \\ w_i &| \theta_i \sim F(\theta_i) \end{aligned} \tag{C.11}$$

2291 Here  $F$  describes the mapping from parameters  $\theta_i$  to observations  $w_i$ . It is possible to inte-  
 2292 grate over  $G$  and sample the parameters directly from the base distribution  $G_0$ .

2293 It is possible to integrate over  $G$  and get a description in the form of conditionals over the  
 2294 parameters (Blackwell and MacQueen, 1973):

$$\theta_{n+1} | \theta_1 \dots \theta_{n-1} \sim \frac{1}{\alpha + n} (\alpha G_0 + \sum_{i=1}^n \delta_{\theta_i}) \tag{C.12}$$

## 2295 C.1 Gibbs Sampling of Parameters

2296 Algorithm, we will draw  $\theta_i | \theta_{-i}, y_i$  for all  $i$ .

2297 And that continuously. So, that's how we get theta.

2298 Gibbs sampling requires the conditional probabilities of all entities involved (Geman and  
 2299 Geman, 1984). Gibbs sampling just as other Markov chain Monte Carlo methods generates  
 2300 a sequence of correlated samples. Subsequently, if necessary, the Maximum A Posteriori  
 2301 estimation of a value can be found through picking the mode (most common occurring  
 2302 value) of a parameter.

2303 The derivation of the conditional probabilities of parameters with respect to the remain-  
 2304 ing parameters has been described in the literature (Neal, 2000). Such a derivation uses  
 2305 an important property of the Dirichlet process, namely that it is the conjugate prior of the  
 2306 multinomial distribution. Thanks to conjugacy the following equations have closed-form  
 2307 descriptions. The conditional probabilities are sampled from the base distribution  $G_0$  and  
 2308 the other parameters  $\theta_i$  in the following way:

$$\theta_{n+1} | \theta_1 \dots \theta_{n-1} \sim \frac{1}{\alpha + n} (\alpha G_0 + \sum_{i=1}^n \delta_{\theta_i}) \tag{C.13}$$

2309 If we include the observations themselves, we need to include the likelihood as well:

$$\theta_i | \theta_{-i}, w_i \sim C \left\{ \sum_{j, j \neq i} F(w_i, \theta_j) \delta_{\theta_j} + \alpha H_i \int F(w_i, \theta) dG_0(\theta) \right\} \tag{C.14}$$



2310 The constant  $C$  is a normalization factor to make the above a proper probability density  
2311 (summing to one). The entity  $H_i$  is the posterior density of  $\theta$  given  $G_0$  as prior and  $y_i$  as  
2312 observation. The notation  $\theta_{-i}$  describes the set of all parameters  $\Theta$  with  $\theta_i$  excluded. The  
2313 integral over  $dG_0(\theta)$  is a Lebesgue-Stieltjes integral that weighs the contribution of  $F(w_i, \theta)$   
2314 with the base distribution  $G_0(\theta)$ .

2315 Equation C.14 can be used to perform inference directly with all (non-unique) parameters  
2316  $\theta_i$  tied to observations  $w_i$ . Details on inference will be provided in Sect. ??.





## GLOSSARY

2317

2318

2319

2320 **burn-in** running a Markov chain Monte Carlo for a while before starting to sample from it,  
2321 so the results are not depending on its initial random starting position. 38

2322 **collapsed Gibbs sampling** Rao-Blackwellized Gibbs sampling. Certain sampling steps are  
2323 replaced by steps where one or more variables are integrated out. This is can be thanks  
2324 to analytic descriptions that arise from the use of conjugate priors. 38

## LIST OF FIGURES

2326	2.1	Probability measure . . . . .	12
2327	2.2	Generative vs Discriminative . . . . .	22
2328	2.3	Plate notation . . . . .	25
2329	2.4	Stick-breaking presentation . . . . .	27
2330	2.5	Beta Process . . . . .	28
2331	2.6	Gamma Process . . . . .	30
2332	2.7	Chinese Restaurant Process . . . . .	31
2333	2.8	Dirichlet Process . . . . .	31
2334	2.9	The difference visualized between a Dirichlet Process mixture and a Hierarchical Dirichlet Process. It illustrates also that the input of a Dirichlet process does not have to be a continuous function. If it is a continuous distribution it will become a discrete distributed almost surely. If it is a discrete distribution, it will have atoms at the locations where the discrete distribution had its probability mass concentrated. . . . .	34
2340	2.10	Rejection Sampling . . . . .	36
2341	3.1	The Infinite Line Model in the Chinese Restaurant Process representation (compare with Fig. 2.8). Top: $\alpha$ , the concentration parameter of the Dirichlet Process. Bottom, left to right: $w_i$ , the observation, an individual point in a 2D space; $\theta_i$ , the parameters (intercept, slope) of the line belonging to observation $w_i$ ; $H$ , the base distribution from which line parameter values are sampled. . . . .	44
2347	3.2	The Infinite Line Model in the stick-breaking representation (compare with Fig. 2.8). From left to right: $\alpha$ , the concentration parameter of the Dirichlet Process; $(\phi_1, \dots, \phi_k)$ , the partition of points over lines; $z_i$ , the assignment parameters that link observation $w_i$ with line $k$ ; $w_i$ , the observation, an individual point with $x$ and $y$ coordinates; $\theta_k$ , the parameters of line $k$ ; $\lambda_0$ , the base measure from which the line parameter values are sampled. . . . .	50
2353	3.3	The performance of Algorithm 1 with respect to clustering is measured using the Rand Index, the Adjusted Rand Index, the Mirvin metric, and the Hubert metric. A figure of 1 means perfect clustering for all metrics, except Mirvin's where 0 denotes perfect clustering. . . . .	53
2357	3.4	The performance of Algorithm 2 with respect to clustering is measured using the Rand Index, the Adjusted Rand Index, the Mirvin metric, and the Hubert metric. A figure of 1 means perfect clustering for all metrics, except Mirvin's where 0 denotes perfect clustering. . . . .	54

2361	3.5	One of the Gibbs steps in the inference of two particular lines. The points are more or less distributed according to the lines, but one line exists out of two large clusters. The line coordinates are visualized by a double circle. The x-coordinate is the y-intercept of the line, the y-coordinate is the slope. . . . .	55
2362			
2363			
2364			
2365	3.6	The assignment of a line to a single point. There are three clusters found, rather than only the obvious two. . . . .	56
2366			
2367	4.1	Sampling of Pareto pairs. The parameters are $\lambda_m = 2$ , $\lambda_n = -2$ , $k = 5$ , and we have sampled $N = 1000$ pairs. . . . .	59
2368			
2369	4.2	Consider the data uniformly distributed on a line segment and a symmetric Pareto prior for both the endpoints, then we can update the estimate for the endpoints given the data as visualized. Each subfigure shows an adjustment of the endpoints given more data points (1, 3, 10, and 100 data points). . .	60
2370			
2371			
2372			
2373	4.3	The Bayesian linear regression model for multiple line segments in plate notation is the same as for the Infinite Line Model. The Dirichlet process is defined at the left with concentration parameter $\alpha$ . It generates the partitions $(\pi_1, \dots, \pi_k)$ with assignment parameters $z_i$ that denote which observation $w_i$ belongs to which cluster $k$ . The cluster is summarized through the parameter set $\theta_k$ and has $\lambda_0$ as its hyperparameter. The parameter set $\theta_k$ includes parameters that signify the line itself such as slope and y-intercept, plus the parameters that denote the extend of the segment. . . . .	60
2374			
2375			
2376			
2377			
2378			
2379			
2380			
2381	4.4	Line segments generated through a Dirichlet Process. The Dirichlet Process itself is again the same. But now four parameters are generated. A normal-inverse-Wishart distribution is used to generate the center of the line segment, and an inverse-Wishart distribution to generate one of the endpoints of the line segments (the other end point is mirrored through its center). Points are generated normally over the line segments, with an additional Gaussian component to indicate the deviation from the line segment from the normal-inverse-Wishart. . . . .	61
2382			
2383			
2384			
2385			
2386			
2387			
2388			
2389	4.5	Line segments generated through a Dirichlet Process. Compared to Fig. 4.4 the points are generated uniformly over the line segments: points are not generated outside of the line segments. . . . .	62
2390			
2391			
2392	4.6	Bayesian point estimates of the sampling process with varying outcomes. . .	63
2393	4.7	Segment detection performs much worse than line detection across all three clustering performance indicators. Perfect clustering is indicated by 1.0 for Rand Index, Adjusted Rand Index, and Hubert. . . . .	64
2394			
2395			
2396	5.1	Dyadic vs triadic MCMC . . . . .	71
2397	5.2	Two examples of fitting a mixture of lines to data items scattered over a two-dimensional space. The lines drawn are inferred using one of the methods in this chapter. The lines are not the ground truth, but are meant to demonstrate the typical errors made by fitting methods. Note for example that there are mistakes in both the assignment of points to lines as well as the line parameters (slope and intercept). . . . .	75
2398			
2399			
2400			
2401			
2402			
2403	5.3	The same results as in Table 5.1, but visualized in a violin plot. The distribution over metric values are displayed in a vertical fashion. From left to right the distribution shifts to one, signifying better clustering performance. . . . .	76
2404			
2405			
2406	6.1	Generative Adversarial Network . . . . .	81
2407	6.2	Varational Autoencoder . . . . .	81

2408	7.1	The likelihood function for the moments at which people decide to exercise during the day. On the horizontal axis time, on the vertical axis the frequency of exercising. . . . .	87
2409			
2410			
2411	7.2	The improved likelihood function for the moments at which people decide to exercise during the day using the Von Mises distribution. On the horizontal axis time, on the vertical axis the frequency of exercising. . . . .	88
2412			
2413			
2414	7.3	After running the algorithm, the above figures show the different categories of runners that have been found. . . . .	91
2415			
2416	A.1	Beta distribution . . . . .	108

## LIST OF TABLES

2418	2.1 Structures and Processes . . . . .	24
2419	2.2 Levy measure . . . . .	26
2420	2.3 Exchangeable structures . . . . .	27
2421	5.1 The purity, rand index, and adjusted rand index establishing the quality of	
2422	the clustering method. The closer the values to one, the better the method	
2423	performed. The purity metric assigns high values to clusters that do not have	
2424	data points from other clusters (but does not penalize the number of clus-	
2425	ters). The rand index index computes similarity between clusters taking false	
2426	negatives and false positives into account. The adjusted rand index accounts	
2427	for chance. The adjusted rand index is most useful in our comparison. . . . .	76
2428	7.1 Example of the type of data about the timing of exercising. A person is repre-	
2429	sented by row, her preferences by column. There is not a predefined number	
2430	of users or groups of users. . . . .	86
2431	7.2 Top row: A sequence of cluster indices indicates the ground truth. Each clus-	
2432	ter index represents a multi-modal Von-Mises-Uniform distribution with dif-	
2433	ferent parameters $\theta$ . Bottom row: A sequence of cluster indices that are the	
2434	result of the described algorithm. Each cluster index represents again a multi-	
2435	modal Von-Mises distribution. Errors would be represented by an inconsistent	
2436	mapping from the top row to the bottom row. . . . .	90

## LIST OF ABBREVIATIONS

<b>AIC</b>	<b>A</b> kaike <b>I</b> nformation <b>C</b> riterion
<b>BIC</b>	<b>B</b> ayesian <b>I</b> nformation <b>C</b> riterion
<b>DP</b>	<b>D</b> irichlet <b>P</b> rocess





24 38

## SUMMARY

24 39 Summary...



2440

## SAMENVATTING

2441 Samenvatting...



2442

## ACKNOWLEDGMENTS

2443 I want to thank...

2444

## CURRICULUM VITAE

2445 Anne van Rossum ....

2446

## PUBLICATIONS

2447 The investigations performed during my Ph.D. research resulted in the following publica-  
2448 tions.

2449     ◦ A.C. van Rossum. ....



# SIKS DISSERTATION SERIES

2451 **1998**

- 2452 1 Johan van den Akker (CWI<sup>1</sup>) *DEGAS - An Active,* 2483  
 2453 *Temporal Database of Autonomous Objects* 2484  
 2454 2 Floris Wiesman (UM) *Information Retrieval by* 2485  
 2455 *Graphically Browsing Meta-Information* 2486  
 2456 3 Ans Steuten (TUD) *A Contribution to the Linguistic* 2487  
 2457 *Analysis of Business Conversations within the Lan-* 2488  
 2458 *guage/Action Perspective* 2489  
 2459 4 Dennis Breuker (UM) *Memory versus Search in* 2490  
 2460 *Games* 2491  
 2461 5 Eduard W. Oskamp (RUL) *Computerondersteuning* 2492  
 2462 *bij Straftoemeting* 2493

2463 **1999**

- 2464 1 Mark Sloof (VU) *Physiology of Quality Change Mod-* 2497  
 2465 *elling; Automated Modelling of Quality Change of* 2498  
 2466 *Agricultural Products* 2499  
 2467 2 Rob Potharst (EUR) *Classification using Decision* 2500  
 2468 *Trees and Neural Nets* 2501  
 2469 3 Don Beal (UM) *The Nature of Minimax Search* 2502  
 2470 4 Jacques Penders (UM) *The Practical Art of Moving* 2503  
 2471 *Physical Objects* 2504  
 2472 5 Aldo de Moor (KUB) *Empowering Communities: A* 2505  
 2473 *Method for the Legitimate User-Driven Specification* 2506  
 2474 *of Network Information Systems*  
 2475 6 Niek J.E. Wijngaards (VU) *Re-Design of Composi-* 2507  
 2476 *tional Systems*  
 2477 7 David Spelt (UT) *Verification Support for Object* 2508  
 2478 *Database Design* 2509  
 2479 8 Jacques H.J. Lenting (UM) *Informed Gambling:* 2510  
 2480 *Conception and Analysis of a Multi-Agent Mecha-* 2511  
 2481 *nism for Discrete Reallocation*

2482 **2000**

- 1 Frank Niessink (VU) *Perspectives on Improving Soft-*  
*ware Maintenance*  
 2 Koen Holtman (TU/e) *Prototyping of CMS Storage*  
*Management*  
 3 Carolien M.T. Metselaar (UvA) *Sociaal-*  
*organisatorische Gevolgen van Kennistechnologie;*  
*een Procesbenadering en Actorperspectief*  
 4 Geert de Haan (VU) *ETAG, A Formal Model of Com-*  
*petence Knowledge for User Interface Design*  
 5 Ruud van der Pol (UM) *Knowledge-Based Query*  
*Formulation in Information Retrieval*  
 6 Rogier van Eijk (UU) *Programming Languages for*  
*Agent Communication*  
 7 Niels Peek (UU) *Decision-Theoretic Planning of Clin-*  
*ical Patient Management*  
 8 Veerle Coupé (EUR) *Sensitivity Analysis of Decision-*  
*Theoretic Networks*  
 9 Florian Waas (CWI) *Principles of Probabilistic*  
*Query Optimization*  
 10 Niels Nes (CWI) *Image Database Management Sys-*  
*tem Design Considerations, Algorithms and Archite-*  
*cture*  
 11 Jonas Karlsson (CWI) *Scalable Distributed Data*  
*Structures for Database Management*

**2001**

- 1 Silja Renooij (UU) *Qualitative Approaches to Quan-*  
*tifying Probabilistic Networks*  
 2 Koen Hindriks (UU) *Agent Programming Lan-*  
*guages: Programming with Mental Models*

<sup>1</sup>Abbreviations: SIKS - Dutch Research School for Information and Knowledge Systems; CWI - Centrum voor Wiskunde en Informatica, Amsterdam; EUR - Erasmus Universiteit, Rotterdam; KUB - Katholieke Universiteit Brabant, Tilburg; KUN - Katholieke Universiteit Nijmegen; OU - Open Universiteit; RUL - Rijksuniversiteit Leiden; RUN - Radboud Universiteit Nijmegen; TUD - Technische Universiteit Delft; TU/e - Technische Universiteit Eindhoven; UL - Universiteit Leiden; UM - Universiteit Maastricht; UT - Universiteit Twente, Enschede; UU - Universiteit Utrecht; UvA - Universiteit van Amsterdam; UvT - Universiteit van Tilburg; VU - Vrije Universiteit, Amsterdam.

- 2512 3 Maarten van Someren (UvA) *Learning as Problem Solving* 2568  
2513 2569
- 2514 4 Evgueni Smirnov (UM) *Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets* 2570  
2515 2571
- 2516 5 Jacco van Ossenbruggen (VU) *Processing Structured Hypermedia: A Matter of Style* 2572  
2517 2573
- 2518 6 Martijn van Welie (VU) *Task-Based User Interface Design* 2574  
2519 2575
- 2520 7 Bastiaan Schonhage (VU) *Diva: Architectural Perspectives on Information Visualization* 2576  
2521 2577
- 2522 8 Pascal van Eck (VU) *A Compositional Semantic Structure for Multi-Agent Systems Dynamics* 2578  
2523 2579
- 2524 9 Pieter Jan 't Hoen (RUL) *Towards Distributed Development of Large Object-Oriented Models, Views of Packages as Classes* 2580  
2525 2581
- 2526 10 Maarten Sierhuis (UvA) *Modeling and Simulating Work Practice BRAHMS: a Multiagent Modeling and Simulation Language for Work Practice Analysis and Design* 2582  
2527 2583
- 2528 11 Tom M. van Engers (VU) *Knowledge Management: The Role of Mental Models in Business Systems Design* 2584  
2529 2585
- 2530 2586  
2531 2587  
2532 2588  
2533 2589
- 2534 **2002** 2590
- 2535 1 Nico Lassing (VU) *Architecture-Level Modifiability Analysis* 2591  
2536 2592
- 2537 2 Roelof van Zwol (UT) *Modelling and Searching Web-based Document Collections* 2593  
2538 2594
- 2539 3 Henk Ernst Blok (UT) *Database Optimization Aspects for Information Retrieval* 2595  
2540 2596
- 2541 4 Juan Roberto Castelo Valdueza (UU) *The Discrete Acyclic Digraph Markov Model in Data Mining* 2597  
2542 2598
- 2543 5 Radu Serban (VU) *The Private Cyberspace Modeling Electronic Environments Inhabited by Privacy-Concerned Agents* 2599  
2544 2600
- 2545 2601
- 2546 6 Laurens Mommers (UL) *Applied Legal Epistemology; Building a Knowledge-based Ontology of the Legal Domain* 2602  
2547 2603
- 2548 2604
- 2549 7 Peter Boncz (CWI) *Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications* 2605  
2550 2606
- 2551 8 Jaap Gordijn (VU) *Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas* 2607  
2552 2608
- 2553 9 Willem-Jan van den Heuvel (KUB) *Integrating Modern Business Applications with Objectified Legacy Systems* 2609  
2554 2610
- 2555 2611
- 2556 10 Brian Sheppard (UM) *Towards Perfect Play of Scrabble* 2612  
2557 2613
- 2558 11 Wouter C.A. Wijngaards (VU) *Agent Based Modelling of Dynamics: Biological and Organisational Applications* 2614  
2559 2615
- 2560 2616
- 2561 12 Albrecht Schmidt (UvA) *Processing XML in Database Systems* 2617  
2562 2618
- 2563 13 Hongjing Wu (TU/e) *A Reference Architecture for Adaptive Hypermedia Applications* 2619  
2564 2620
- 2565 14 Wieke de Vries (UU) *Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems* 2621  
2566 2622  
2567 2623
- 15 Rik Eshuis (UT) *Semantics and Verification of UML Activity Diagrams for Workflow Modelling*
- 16 Pieter van Langen (VU) *The Anatomy of Design: Foundations, Models and Applications*
- 17 Stefan Manegold (UvA) *Understanding, Modeling, and Improving Main-Memory Database Performance*
- 2003**
- 1 Heiner Stuckenschmidt (VU) *Ontology-Based Information Sharing in Weakly Structured Environments*
- 2 Jan Broersen (VU) *Modal Action Logics for Reasoning About Reactive Systems*
- 3 Martijn Schuemie (TUD) *Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy*
- 4 Milan Petkovic (UT) *Content-Based Video Retrieval Supported by Database Technology*
- 5 Jos Lehmann (UvA) *Causation in Artificial Intelligence and Law – A Modelling Approach*
- 6 Boris van Schooten (UT) *Development and Specification of Virtual Environments*
- 7 Machiel Jansen (UvA) *Formal Explorations of Knowledge Intensive Tasks*
- 8 Yong-Ping Ran (UM) *Repair-Based Scheduling*
- 9 Rens Kortmann (UM) *The Resolution of Visually Guided Behaviour*
- 10 Andreas Lincke (UT) *Electronic Business Negotiation: Some Experimental Studies on the Interaction between Medium, Innovation Context and Cult*
- 11 Simon Keizer (UT) *Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks*
- 12 Roeland Ordelman (UT) *Dutch Speech Recognition in Multimedia Information Retrieval*
- 13 Jeroen Donkers (UM) *Nosce Hostem – Searching with Opponent Models*
- 14 Stijn Hoppenbrouwers (KUN) *Freezing Language: Conceptualisation Processes across ICT-Supported Organisations*
- 15 Mathijs de Weerd (TUD) *Plan Merging in Multi-Agent Systems*
- 16 Menzo Windhouwer (CWI) *Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouse*
- 17 David Jansen (UT) *Extensions of Statecharts with Probability, Time, and Stochastic Timing*
- 18 Levente Kocsis (UM) *Learning Search Decisions*
- 2004**
- 1 Virginia Dignum (UU) *A Model for Organizational Interaction: Based on Agents, Founded in Logic*
- 2 Lai Xu (UvT) *Monitoring Multi-party Contracts for E-business*
- 3 Perry Groot (VU) *A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving*

- 2622 4 Chris van Aart (UvA) *Organizational Principles for* 2676  
2623 *Multi-Agent Architectures* 2677
- 2624 5 Viara Popova (EUR) *Knowledge Discovery and* 2678  
2625 *Monotonicity* 2679
- 2626 6 Bart-Jan Hommes (TUD) *The Evaluation of Busi-* 2680  
2627 *ness Process Modeling Techniques* 2681
- 2628 7 Elise Boltjes (UM) *Voorbeeld<sub>IG</sub> Onderwijs; Voor-* 2682  
2629 *beeldgestuurd Onderwijs, een Opstap naar Abstracte* 2683  
2630 *Denken, vooral voor Meisjes* 2684
- 2631 8 Joop Verbeek (UM) *Politie en de Nieuwe Inter-* 2685  
2632 *nationale Informatiemarkt, Grensregionale Politie* 2686  
2633 *Gegevensuitwisseling en Digitale Expertise* 2687
- 2634 9 Martin Caminada (VU) *For the Sake of the Argu-* 2688  
2635 *ment; Explorations into Argument-based Reasoning* 2689
- 2636 10 Suzanne Kabel (UvA) *Knowledge-rich Indexing of* 2690  
2637 *Learning-objects* 2691
- 2638 11 Michel Klein (VU) *Change Management for Dis-* 2692  
2639 *tributed Ontologies* 2693
- 2640 12 The Duy Bui (UT) *Creating Emotions and Facial Ex-* 2694  
2641 *pressions for Embodied Agents* 2695
- 2642 13 Wojciech Jamroga (UT) *Using Multiple Models of* 2696  
2643 *Reality: On Agents who Know how to Play* 2697
- 2644 14 Paul Harrenstein (UU) *Logic in Conflict. Logical Ex-* 2698  
2645 *plorations in Strategic Equilibrium* 2699
- 2646 15 Arno Knobbe (UU) *Multi-Relational Data Mining* 2700  
2647 16 Federico Divina (VU) *Hybrid Genetic Relational* 2701  
2648 *Search for Inductive Learning*
- 2649 17 Mark Winands (UM) *Informed Search in Complex* 2702  
2650 *Games*
- 2651 18 Vania Bessa Machado (UvA) *Supporting the Con-* 2703  
2652 *struction of Qualitative Knowledge Models* 2704
- 2653 19 Thijs Westerveld (UT) *Using generative probabilis-* 2705  
2654 *tic models for multimedia retrieval* 2706
- 2655 20 Madelon Evers (Nyenrode) *Learning from Design* 2707  
2656 *facilitating multidisciplinary design teams* 2708  
2709
- 2657 **2005** 2710
- 2658 1 Floor Verdenius (UvA) *Methodological Aspects of* 2713  
2659 *Designing Induction-Based Applications* 2714
- 2660 2 Erik van der Werf (UM) *AI techniques for the game* 2715  
2661 *of Go* 2716
- 2662 3 Franc Grootjen (RUN) *A Pragmatic Approach to the* 2717  
2663 *Conceptualisation of Language* 2718
- 2664 4 Nirvana Meratnia (UT) *Towards Database Support* 2719  
2665 *for Moving Object data* 2720
- 2666 5 Gabriel Infante-Lopez (UvA) *Two-Level Probabilis-* 2721  
2667 *tic Grammars for Natural Language Parsing* 2722
- 2668 6 Pieter Spronck (UM) *Adaptive Game AI* 2723
- 2669 7 Flavius Frasinca (TU/e) *Hypermedia Presentation* 2724  
2670 *Generation for Semantic Web Information Systems* 2725
- 2671 8 Richard Vdovjak (TU/e) *A Model-driven Approach* 2726  
2672 *for Building Distributed Ontology-based Web Appli-* 2727  
2673 *cations* 2728
- 2674 9 Jeen Broekstra (VU) *Storage, Querying and Infer-* 2729  
2675 *encing for Semantic Web Languages* 2730  
2731
- 2676 10 Anders Bouwer (UvA) *Explaining Behaviour: Using*  
2677 *Qualitative Simulation in Interactive Learning Envi-*  
2678 *ronments*
- 2679 11 Elth Ogston (VU) *Agent Based Matchmaking and*  
2680 *Clustering - A Decentralized Approach to Search*
- 2681 12 Csaba Boer (EUR) *Distributed Simulation in Indus-*  
2682 *try*
- 2683 13 Fred Hamburg (UL) *Een Computermodel voor het*  
2684 *Ondersteunen van Euthanasiebeslissingen*
- 2685 14 Borys Omelayenko (VU) *Web-Service configuration*  
2686 *on the Semantic Web; Exploring how semantics*  
2687 *meets pragmatics*
- 2688 15 Tibor Bosse (VU) *Analysis of the Dynamics of Cog-*  
2689 *nitive Processes*
- 2690 16 Joris Graaumanns (UU) *Usability of XML Query Lan-*  
2691 *guages*
- 2692 17 Boris Shishkov (TUD) *Software Specification Based*  
2693 *on Re-usable Business Components*
- 2694 18 Danielle Sent (UU) *Test-selection strategies for prob-*  
2695 *abilistic networks*
- 2696 19 Michel van Dartel (UM) *Situated Representation*
- 2697 20 Cristina Coteanu (UL) *Cyber Consumer Law, State*  
2698 *of the Art and Perspectives*
- 2699 21 Wijnand Derks (UT) *Improving Concurrency and*  
2700 *Recovery in Database Systems by Exploiting Appli-*  
2701 *cation Semantics*
- 2006**
- 1 Samuil Angelov (TU/e) *Foundations of B2B Elec-*  
2702 *tronic Contracting*
- 2703 2 Cristina Chisalita (VU) *Contextual issues in the de-*  
2704 *sign and use of information technology in organiza-*  
2705 *tions*
- 2706 3 Noor Christoph (UvA) *The role of metacognitive*  
2707 *skills in learning to solve problems*
- 2708 4 Marta Sabou (VU) *Building Web Service Ontologies*
- 2709 5 Cees Pierik (UU) *Validation Techniques for Object-*  
2710 *Oriented Proof Outlines*
- 2711 6 Ziv Baida (VU) *Software-aided Service Bundling -*  
2712 *Intelligent Methods & Tools for Graphical Service*  
2713 *Modeling*
- 2714 7 Marko Smiljanic (UT) *XML schema matching – bal-*  
2715 *ancing efficiency and effectiveness by means of clus-*  
2716 *tering*
- 2717 8 Eelco Herder (UT) *Forward, Back and Home Again*  
2718 *- Analyzing User Behavior on the Web*
- 2719 9 Mohamed Wahdan (UM) *Automatic Formulation of*  
2720 *the Auditor's Opinion*
- 2721 10 Ronny Siebes (VU) *Semantic Routing in Peer-to-Peer*  
2722 *Systems*
- 2723 11 Joeri van Ruth (UT) *Flattening Queries over Nested*  
2724 *Data Types*
- 2725 12 Bert Bongers (VU) *Interactivation - Towards an e-*  
2726 *cology of people, our technological environment, and*  
2727 *the arts*
- 2728 13 Henk-Jan Lebbink (UU) *Dialogue and Decision*  
2729 *Games for Information Exchanging Agents*

- 2732 14 Johan Hoorn (VU) *Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change* 2786  
2733 2787  
2734 2788
- 2735 15 Rainer Malik (UU) *CONAN: Text Mining in the Biomedical Domain* 2789  
2736 2790
- 2737 16 Carsten Riggelsen (UU) *Approximation Methods for Efficient Learning of Bayesian Networks* 2791  
2738 2792
- 2739 17 Stacey Nagata (UU) *User Assistance for Multitasking with Interruptions on a Mobile Device* 2793  
2740 2794
- 2741 18 Valentin Zhizhkun (UvA) *Graph transformation for Natural Language Processing* 2795  
2742 2796
- 2743 19 Birna van Riemsdijk (UU) *Cognitive Agent Programming: A Semantic Approach* 2797  
2744 2798
- 2745 20 Marina Velikova (UvT) *Monotone models for prediction in data mining* 2799  
2746 2800
- 2747 21 Bas van Gils (RUN) *Aptness on the Web* 2801
- 2748 22 Paul de Vrieze (RUN) *Fundamentals of Adaptive Personalisation* 2802  
2749 2803
- 2750 23 Ion Juvina (UU) *Development of Cognitive Model for Navigating on the Web* 2804  
2751 2805
- 2752 24 Laura Hollink (VU) *Semantic Annotation for Retrieval of Visual Resources* 2806  
2753 2807
- 2754 25 Madalina Drugan (UU) *Conditional log-likelihood MDL and Evolutionary MCMC* 2808  
2755 2809
- 2756 26 Vojkan Mihajlovic (UT) *Score Region Algebra: A Flexible Framework for Structured Information Retrieval* 2810  
2757 2811  
2758 2812  
2813
- 2759 27 Stefano Bocconi (CWI) *Vox Populi: generating video documentaries from semantically annotated media repositories* 2814  
2760 2815  
2761 2816
- 2762 28 Borkur Sigurbjornsson (UvA) *Focused Information Access using XML Element Retrieval* 2817  
2763 2818  
2819
- 2764 **2007** 2820
- 2765 1 Kees Leune (UvT) *Access Control and Service-Oriented Architectures* 2821  
2766 2822
- 2767 2 Wouter Teepe (RUG) *Reconciling Information Exchange and Confidentiality: A Formal Approach* 2823  
2768 2824
- 2769 3 Peter Mika (VU) *Social Networks and the Semantic Web* 2825  
2770 2826
- 2771 4 Jurriaan van Diggelen (UU) *Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach* 2827  
2772 2828  
2773 2829
- 2774 5 Bart Schermer (UL) *Software Agents, Surveillance and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance* 2830  
2775 2831  
2776 2832
- 2777 6 Gilad Mishne (UvA) *Applied Text Analytics for Blogs* 2833  
2778 2834
- 2779 7 Natasa Jovanovic' (UT) *To Whom It May Concern Addressee Identification in Face-to-Face Meetings* 2835  
2780 2836
- 2781 8 Mark Hoogendoorn (VU) *Modeling of Change in Multi-Agent Organizations* 2837  
2782 2838
- 2783 9 David Mobach (VU) *Agent-Based Mediated Service Negotiation* 2839  
2784 2840
- 2785 10 Huib Aldewereld (UU) *Autonomy vs. Conformity an Institutional Perspective on Norms and Protocols* 2841
- 11 Natalia Stash (TU/e) *Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System*
- 12 Marcel van Gerven (RUN) *Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty*
- 13 Rutger Rienks (UT) *Meetings in Smart Environments; Implications of Progressing Technology*
- 14 Niek Bergboer (UM) *Context-Based Image Analysis*
- 15 Joyca Lacroix (UM) *NIM: a Situated Computational Memory Model*
- 16 Davide Grossi (UU) *Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems*
- 17 Theodore Charitos (UU) *Reasoning with Dynamic Networks in Practice*
- 18 Bart Orriens (UvT) *On the development and management of adaptive business collaborations*
- 19 David Levy (UM) *Intimate relationships with artificial partners*
- 20 Slinger Jansen (UU) *Customer Configuration Updating in a Software Supply Network*
- 21 Karianne Vermaas (UU) *Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005*
- 22 Zlatko Zlatev (UT) *Goal-oriented design of value and process models from patterns*
- 23 Peter Barna (TU/e) *Specification of Application Logic in Web Information Systems*
- 24 Georgina Ramírez Camps (CWI) *Structural Features in XML Retrieval*
- 25 Joost Schalken (VU) *Empirical Investigations in Software Process Improvement*
- 2008**
- 1 Katalin Boer-Sorbán (EUR) *Agent-Based Simulation of Financial Markets: A modular, continuous-time approach*
- 2 Alexei Sharpanskykh (VU) *On Computer-Aided Methods for Modeling and Analysis of Organizations*
- 3 Vera Hollink (UvA) *Optimizing hierarchical menus: a usage-based approach*
- 4 Ander de Keijzer (UT) *Management of Uncertain Data - towards unattended integration*
- 5 Bela Mutschler (UT) *Modeling and simulating causal dependencies on process-aware information systems from a cost perspective*
- 6 Arjen Hommersom (RUN) *On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective*
- 7 Peter van Rosmalen (OU) *Supporting the tutor in the design and support of adaptive e-learning*
- 8 Janneke Bolt (UU) *Bayesian Networks: Aspects of Approximate Inference*
- 9 Christof van Nimwegen (UU) *The paradox of the guided user: assistance can be counter-effective*

- 2842 10 Wauter Bosma (UT) *Discourse oriented Summariza-*2900  
2843 *tion* 2901
- 2844 11 Vera Kartseva (VU) *Designing Controls for Network*  
2845 *Organizations: a Value-Based Approach* 2902
- 2846 12 Jozsef Farkas (RUN) *A Semiotically oriented Cogni-*  
2847 *tive Model of Knowledge Representation* 2903
- 2848 13 Caterina Carraciolo (UvA) *Topic Driven Access to*2904  
2849 *Scientific Handbooks* 2905
- 2850 14 Arthur van Bunningen (UT) *Context-Aware Query-*2906  
2851 *ing; Better Answers with Less Effort* 2907
- 2852 15 Martijn van Otterlo (UT) *The Logic of Adaptive Be-*2908  
2853 *havior: Knowledge Representation and Algorithms*2909  
2854 *for the Markov Decision Process Framework in First-*2910  
2855 *Order Domains* 2911
- 2856 16 Henriette van Vugt (VU) *Embodied Agents from a*2912  
2857 *User's Perspective* 2913
- 2858 17 Martin Op't Land (TUD) *Applying Architecture and*2914  
2859 *Ontology to the Splitting and Allying of Enterprises*2915
- 2860 18 Guido de Croon (UM) *Adaptive Active Vision* 2916
- 2861 19 Henning Rode (UT) *From document to entity re-*2917  
2862 *trieval: improving precision and performance of fo-*2918  
2863 *cused text search* 2919
- 2864 20 Rex Arendsen (UvA) *Geen bericht, goed bericht. Een*2920  
2865 *onderzoek naar de effecten van de introductie van*2921  
2866 *elektronisch berichtenverkeer met een overheid op de*2922  
2867 *administratieve lasten van bedrijven* 2923
- 2868 21 Krisztian Balog (UvA) *People search in the enter-*2924  
2869 *prise* 2925
- 2870 22 Henk Koning (UU) *Communication of IT*2926  
2871 *architecture* 2927
- 2872 23 Stefan Visscher (UU) *Bayesian network models for*2928  
2873 *the management of ventilator-associated pneumonia*2929
- 2874 24 Zharko Aleksovski (VU) *Using background knowl-*2930  
2875 *edge in ontology matching* 2931
- 2876 25 Geert Jonker (UU) *Efficient and Equitable exchange*2932  
2877 *in air traffic management plan repair using spender-*2933  
2878 *signed currency* 2934
- 2879 26 Marijn Huijbregts (UT) *Segmentation, diarization*2935  
2880 *and speech transcription: surprise data unraveled* 2936
- 2881 27 Hubert Vogten (OU) *Design and implementation*2937  
2882 *strategies for IMS learning design* 2938
- 2883 28 Ildiko Flesh (RUN) *On the use of independence re-*2939  
2884 *lations in Bayesian networks* 2940
- 2885 29 Dennis Reidsma (UT) *Annotations and subjective*2941  
2886 *machines- Of annotators, embodied agents, users,*2942  
2887 *and other humans* 2943
- 2888 30 Wouter van Atteveldt (VU) *Semantic network anal-*2944  
2889 *ysis: techniques for extracting, representing and*2945  
2890 *querying media content* 2946
- 2891 31 Loes Braun (UM) *Pro-active medical information re-*2947  
2892 *trieval* 2948
- 2893 32 Trung B. Hui (UT) *Toward affective dialogue man-*2949  
2894 *agement using partially observable markov decision*2950  
2895 *processes* 2951
- 2896 33 Frank Terpstra (UvA) *Scientific workflow design,*2952  
2897 *theoretical and practical issues* 2953
- 2898 34 Jeroen de Knijf (UU) *Studies in Frequent Tree Min-*2954  
2899 *ing* 2955
- 35 Benjamin Torben-Nielsen (UvT) *Dendritic mor-*  
*phology: function shapes structure*
- 2009**
- 1 Rasa Jurgelenaite (RUN) *Symmetric Causal Inde-*  
*pendence Models*
- 2 Willem Robert van Hage (VU) *Evaluating Ontology-*  
*Alignment Techniques*
- 3 Hans Stol (UvT) *A Framework for Evidence-based*  
*Policy Making Using IT*
- 4 Josephine Nabukenya (RUN) *Improving the Qual-*  
*ity of Organisational Policy Making using Collabo-*  
*ration Engineering*
- 5 Sietse Overbeek (RUN) *Bridging Supply and De-*  
*mand for Knowledge Intensive Tasks - Based on*  
*Knowledge, Cognition, and Quality*
- 6 Muhammad Subianto (UU) *Understanding Classi-*  
*fication*
- 7 Ronald Poppe (UT) *Discriminative Vision-Based Re-*  
*covery and Recognition of Human Motion*
- 8 Volker Nannen (VU) *Evolutionary Agent-Based Pol-*  
*icy Analysis in Dynamic Environments*
- 9 Benjamin Kanagwa (RUN) *Design, Discovery and*  
*Construction of Service-oriented Systems*
- 10 Jan Wielemaker (UvA) *Logic programming for*  
*knowledge-intensive interactive applications*
- 11 Alexander Boer (UvA) *Legal Theory, Sources of Law*  
*& the Semantic Web*
- 12 Peter Massuthe (TU/e, Humboldt-Universität zu  
Berlin) *Operating Guidelines for Services*
- 13 Steven de Jong (UM) *Fairness in Multi-Agent Sys-*  
*tems*
- 14 Maksym Korotkiy (VU) *From ontology-enabled ser-*  
*vices to service-enabled ontologies (making onto-*  
*logies work in e-science with ONTO-SOA)*
- 15 Rinke Hoekstra (UvA) *Ontology Representation -*  
*Design Patterns and Ontologies that Make Sense*
- 16 Fritz Reul (UvT) *New Architectures in Computer*  
*Chess*
- 17 Laurens van der Maaten (UvT) *Feature Extraction*  
*from Visual Data*
- 18 Fabian Groffen (CWI) *Armada, An Evolving*  
*Database System*
- 19 Valentin Robu (CWI) *Modeling Preferences, Strate-*  
*gic Reasoning and Collaboration in Agent-Mediated*  
*Electronic Markets*
- 20 Bob van der Vecht (UU) *Adjustable Autonomy: Con-*  
*trolling Influences on Decision Making*
- 21 Stijn Vanderlooy (UM) *Ranking and Reliable Clas-*  
*sification*
- 22 Pavel Serdyukov (UT) *Search For Expertise: Going*  
*beyond direct evidence*
- 23 Peter Hofgesang (VU) *Modelling Web Usage in a*  
*Changing Environment*
- 24 Annerieke Heuvelink (VU) *Cognitive Models for*  
*Training Simulations*
- 25 Alex van Ballegooij (CWI) *"RAM: Array Database*  
*Management through Relational Mapping"*

- 2957 26 Fernando Koch (UU) *An Agent-Based Model for the* 2977 35 Wouter Koelewijn (UL) *Privacy en Politiegegevens;*  
 2958 *Development of Intelligent Mobile Services* 2978 *Over geautomatiseerde normatieve informatie-*  
 2959 27 Christian Glahn (OU) *Contextual Support of social* 2979 *uitwisseling*  
 2960 *Engagement and Reflection on the Web* 2980 36 Marco Kalz (OUN) *Placement Support for Learners*  
 2961 28 Sander Evers (UT) *Sensor Data Management with* 2981 *in Learning Networks*  
 2962 *Probabilistic Models* 2982 37 Hendrik Drachsler (OUN) *Navigation Support for*  
 2963 29 Stanislav Pokraev (UT) *Model-Driven Semantic In-* 2983 *Learners in Informal Learning Networks*  
 2964 *tegration of Service-Oriented Applications* 2984 38 Riina Vuorikari (OU) *Tags and self-organisation: a*  
 2965 30 Marcin Zukowski (CWI) *Balancing vectorized query* 2985 *metadata ecology for learning resources in a multi-*  
 2966 *execution with bandwidth-optimized storage* 2986 *lingual context*  
 2967 31 Sofiya Katrenko (UvA) *A Closer Look at Learning* 2987 39 Christian Stahl (TUE, Humboldt-Universität zu  
 2968 *Relations from Text* 2988 Berlin) *Service Substitution – A Behavioral Ap-*  
 2969 32 Rik Farenhorst and Remco de Boer (VU) *Architec-* 2989 *proach Based on Petri Nets*  
 2970 *tural Knowledge Management: Supporting Archi-* 2990 40 Stephan Raaijmakers (UvT) *Multinomial Language*  
 2971 *tects and Auditors* 2991 *Learning: Investigations into the Geometry of Lan-*  
 2972 33 Khiet Truong (UT) *How Does Real Affect Affect Affect* 2992 *guage*  
 2973 *Recognition In Speech?* 2993 41 Igor Berezhnyy (UvT) *Digital Analysis of Paintings*  
 2974 34 Inge van de Weerd (UU) *Advancing in Software* 2994 42 Toine Bogers (UvT) *Recommender Systems for So-*  
 2975 *Product Management: An Incremental Method En-* 2995 *cial Bookmarking*  
 2976 *gineering Approach*