



UPPSALA  
UNIVERSITET

*Digital Comprehensive Summaries of Uppsala Dissertations  
from the Faculty of Science and Technology 1709*

# Machine learning with state-space models, Gaussian processes and Monte Carlo methods

ANDREAS SVENSSON



ACTA  
UNIVERSITATIS  
UPSALIENSIS  
UPPSALA  
2018

ISSN 1651-6214  
ISBN 978-91-513-0417-5  
urn:nbn:se:uu:diva-357611

Dissertation presented at Uppsala University to be publicly examined in ITC 2446, Lägerhyddsvägen 2, Uppsala, Friday, 12 October 2018 at 10:15 for the degree of Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Professor Carl Edward Rasmussen (University of Cambridge).

### **Abstract**

Svensson, A. 2018. Machine learning with state-space models, Gaussian processes and Monte Carlo methods. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 1709*. 74 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-0417-5.

Numbers are present everywhere, and when they are collected and recorded we refer to them as data. Machine learning is the science of learning mathematical models from data. Such models, once learned from data, can be used to draw conclusions, understand behavior, predict future evolution, and make decisions. This thesis is mainly concerned with two particular statistical models for this purpose: the state-space model and the Gaussian process model, as well as a combination thereof. To learn these models from data, Monte Carlo methods are used, and in particular sequential Monte Carlo (SMC) or particle filters.

The thesis starts with an introductory background on state-space models, Gaussian processes and Monte Carlo methods. The main contribution lies in seven scientific papers. Several contributions are made on the topic of learning nonlinear state-space models with the use of SMC. An existing SMC method is tailored for learning in state-space models with little or no measurement noise. The SMC-based method particle Gibbs with ancestor sampling (PGAS) is used for learning an approximation of the Gaussian process state-space model. PGAS is also combined with stochastic approximation expectation maximization (EM). This method, which we refer to as particle stochastic approximation EM, is a general method for learning parameters in nonlinear state-space models. It is later applied to the particular problem of maximum likelihood estimation in jump Markov linear models. An alternative and non-standard approach for how to use SMC to estimate parameters in nonlinear state-space models is also presented.

There are also two contributions not related to learning state-space models. One is how SMC can be used also for learning hyperparameters in Gaussian process regression models. The second is a method for assessing consistency between model and data. By using the model to simulate new data, and compare how similar that data is to the observed one, a general criterion is obtained which follows directly from the model specification. All methods are implemented and illustrated, and several are also applied to various real-world examples.

*Keywords:* Machine learning, State-space models, Gaussian processes

*Andreas Svensson, Department of Information Technology, Division of Systems and Control, Box 337, Uppsala University, SE-75105 Uppsala, Sweden. Department of Information Technology, Automatic control, Box 337, Uppsala University, SE-75105 Uppsala, Sweden.*

© Andreas Svensson 2018

ISSN 1651-6214

ISBN 978-91-513-0417-5

urn:nbn:se:uu:diva-357611 (<http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-357611>)

*To everybody who contributes to a  
happier and more sustainable world*



”Saken är den, eller snarare, är det väl mer adekvat, att så att säga säga att saken är biff (trots att det inte handlar om mat). Vad jag vill säga, mera konkret – och nu ska jag gå rakt på sak och vara rak! – är väl helt enkelt att saken är klar. Ja, det vill säga, i sak.”

Claes Eriksson

# Populärvetenskaplig sammanfattning

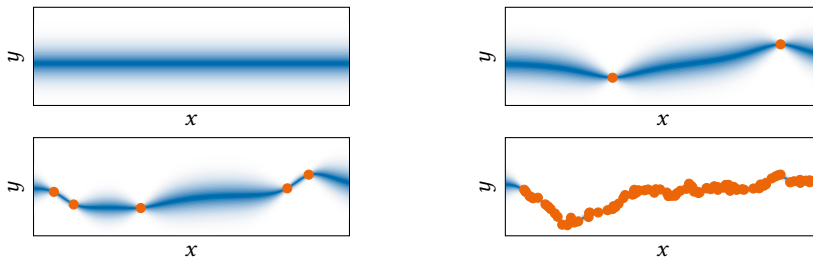
**M**ASKININLÄRNING, eller ofta bara ”machine learning”, handlar om att *automatiskt ta fram och använda matematiska modeller från insamlad statistik*. Statistiken kan vara lite vad som helst, till exempel temperaturer, trafikintensitet, opinionsundersökningar, bilder från övervakningskameror, internettrafik, aktiepriser, röstinspelningar, elektricitetsanvändning eller hur snabbt ett virus sprids. Statistiken, eller datan, kan även ha i stort sett vilket format som helst så länge den bara går att spara i en dator. De matematiska modellerna, som maskininläringen plockar fram ur datan, kan sedan användas för att säga något om ny data som samlas in (”visar den nya bilden från övervakningskameran någon person vi har på bild sedan tidigare?”), förutsäga hur utvecklingen kommer att fortsätta (”kommer viruset att spridas ännu snabbare, eller dö ut?”), eller dra andra slutsatser (”vad skulle hända om trafiken gick längs den här vägen istället?”).

Att ta fram matematiska modeller från den insamlade datan utgör kärnan inom maskininläring. Det första steget för att göra det, är att vi som användare väljer en klass av modeller som vi vill använda. Det finns många olika modellklasser, och olika modellklasser passar olika bra för olika situationer. Ett populärt exempel, som har fått ganska mycket uppmärksamhet även utanför forskningsvärlden, är faltade neurala nätverk som har visat sig fungera bra för att matematiskt beskriva bilder.

I den här avhandlingen tänker vi oss att den insamlade datan består av siffror som är uppmätta och förändrar sig över tid. Det kan till exempel vara mätningar av koldioxidhalter och börsindex, eller dynamiken i hur ett flygplan reagerar på en förändrad rodervinkel eller hur snabbt en bassäng fylls med vatten när strömmen till en pump slås på. För sådan typ av dynamisk data finns det modellklasser som fungerar särskilt bra, och den här avhandlingen handlar om två av dem: tillståndsmodeller och Gaussprocesser.

Idén med tillståndsmodeller är att matematiskt beskriva det väsentliga i en förändring som sker över tid. Det förklaras nog bäst med ett exempel: Om du får ett foto på en bil, kan du då tala om var bilen skulle befinna sig en sekund senare? Nej, det är svårt, eftersom du från ett vanligt foto inte kan avgöra hur snabbt bilen åker. Men om du får en sekvens av bilder tagna med en sekunds mellanrum så kan du jämföra dem med varandra och få en uppfattning om bilens hastighet, och därifrån göra en ganska noggrann förutsägelse om var bilen skulle befinna sig en sekund efter det sista fotot. Problemet med ett ensamt foto är alltså att det inte innehåller all relevant information. Men *om* fotot hade innehållit även en radarmätning av bilens hastighet (eller en bild på hastighetsmätaren), så hade detta enda foto varit tillräckligt för att göra en bra förutsägelse om bilens position i nästa sekund. Det är det här som är poängen med tillståndsmodeller: de är konstruerade för att i varje ögonblick innehålla all matematisk information som är relevant för framtiden. En väldigt kraftfull egenskap

## POPULÄRVETENSKAPLIG SAMMANFATTNING



Den här figuren visar hur en Gaussprocess (blått skuggat område) kan användas för att lära sig ett samband mellan  $x$  (horisontell axel) och  $y$  (vertikal axel). Målet är att kunna leka leken "om du säger  $x$ , så kan jag säga vad  $y$  kommer att bli" (utöver att vara en fänig lek, är det också ett vanligt problem i maskininläring) och till vår hjälp kommer vi att få orangea datapunkter som talar om sanningen  $y$  för vissa värden på  $x$ .

I rutan uppe till vänster finns det ingen data, och Gaussprocessen säger att vi vet väldigt lite, eftersom det blåa skuggade området är förhållandevis brett. I rutan uppe till höger, däremot, finns det två orangea datapunkter och Gaussprocessen har där anpassat sig till dem: i närheten av punkterna är bredden på det blåa skuggade området, det vill säga osäkerheten om vad  $y$  är, liten. Skulle vi däremot be Gaussprocessen om en förutsägelse om vad som händer mitt emellan de två datapunkterna, så är osäkerheten fortfarande ganska stor där.

I den nedre vänstra rutan finns det fem datapunkter, och osäkerheten i Gaussprocessen har minskat ganska mycket på sina håll. En förutsägelse i området mellan de två punkterna längst till vänster innehåller inte mycket osäkerhet alls: eftersom de två punkterna ligger så nära varandra, anser sig Gaussprocessen veta ganska väl vad som händer även i det lilla intervallet emellan dem. Slutligen, i rutan längst ned till höger, har vi fått 100 datapunkter, och Gaussprocessen har anpassat sig till dem alla.

Självva Gaussprocessen är definierad betydligt mer matematiskt än vad som presenteras här, men poängen är att det är en modell som är kapabel att resonera kring osäkerheter om vad  $y$  är, vilket är något som efterfrågas i många moderna maskininläringstillämpningar.

hos tillståndsmodeller är därför att förutsägelser om framtiden inte förbättras ens om det visar sig finnas mer historisk data att tillgå; de har redan sorterat ut allt som är "värt att veta". De här modellerna har visat sig vara användbara och kraftfulla för att beskriva tidsserier och dynamik för många olika tillämpningar, och används flitigt inom så skilda områden som ekonometri och reglerteknik. I den här avhandlingen förekommer tillståndsmodeller många gånger, och vi tittar på och vidareutvecklar metoder att lära sig tillståndsmodeller från insamlade data.

Gaussprocesser är en annan klass av modeller, vars styrka ligger i att kunna interpolera bra mellan olika datapunkter, som illustreras i figuren högst upp på sidan. Med Gaussprocesser beskrivs sannolikheter för olika matematiska funktioner, och på så sätt blir det en modell som kan hantera osäkerheter (att inte veta, en osäkerhet, kan matematiskt beskrivas som att de möjliga alternativen alla har en viss sannolikhet). Gaussprocesser är en relativt ny klass av modeller, och har än så länge kanske haft sin största plats inom maskininlärningsforskningen, men börjar allt mer hitta tillämpningar inom vitt skilda områden där det finns ett behov av att göra förutsägelser om hur stor osäkerhet som finns i ett problem. I den här avhandlingen tittar vi dels på hur Gaussprocesser kan läras från insamlade data, och dels hur de kan kombineras med tillståndsmodeller på ett användbart sätt.

I maskininlärningen, när datan kombineras med en modellklass, används alltså datan för att *lära sig* okända storheter, parametrar, i modellklassen. Att till exempel anpassa en rät linje till några punkter är att lära sig en modell (alltså att lära sig hur mycket linjen ska luta). Principen är densamma även för tillståndsmodeller och Gaussprocesser – att hitta parametrar så att modellen stämmer bra överens med den insamlade datan – men beräkningarna som behöver utföras är ofta mer invecklade.

För att automatiskt lära sig modeller från data, alltså att hitta lämpliga parametervärden, har (kanske lite oväntat?) metoder som bygger på slumpen visat sig vara användbara. Det är nämligen så att de beräkningar som behöver göras för många modellklasser, däribland tillståndsmodeller och Gaussprocesser, är såpass invecklade att de sällan kan göras exakt, och då finns det olika tekniker för att räkna ungefärligt.

Om vi (av någon oklar anledning) har fått i uppgift att slå 3,5 på en vanlig tärning, kan vi välja mellan att leta upp sidan med fyra prickar och säga ”tyvärr, det här var det närmaste jag kunde komma”, eller att kasta tärningen många gånger och säga ”tyvärr, det finns ingen sida som är 3,5, men vi får titta på genomsnittet av många tärningskast istället” (vilket kommer att vara ungefär 3,5). Den första metoden skulle kunna sägas vara en klassisk metod, medan den senare metoden är en metod som bygger på slumpen (att kasta tärningen) och har egenskapen att den ”är i genomsnitt exakt” (vi gör det många gånger); även om tärningen aldrig kommer att visa tre och en halv prick i något enskilt kast, så är det genomsnittet vi tittar på. (Det här exemplet är ju förstås lite fånigt, men liknande situationer kan faktiskt uppstå när man försöker lära sig parametervärden från data.)

Metoder som systematiskt utnyttjar slumpen kallas för Monte Carlo-metoder (efter kasinot i staden med samma namn). Många av Monte Carlo-metoderna<sup>1</sup> har utvecklats just för att lära sig parametrar i matematiska modeller. De är ofta mer beräkningstunga än de klassiska metoderna, men kan ha egenskaper (till exempel att ”göra i genomsnitt rätt”) som gör det värt den extra beräkningskraften, särskilt nuförtiden när det finns snabba datorer som kan användas.

Bidragen i den här avhandlingen handlar till stor del om att utveckla, anpassa och använda sådana här Monte Carlo-metoder för att lära sig parametrar i olika varianter av tillståndsmodeller och Gaussprocesser. I avhandlingen finns det dessutom även ett bidrag som handlar om att utvärdera hur väl en modellklass passar till den insamlade datan.

---

<sup>1</sup>De olika Monte Carlo-metoderna har mer eller mindre lustiga namn, som till exempel avslagsdragning, viktighetsdragning, Markovkedje-Monte Carlo, partikelfilter, partikel-Markovkedje-Monte Carlo, partikel-Gibbs med förfädersdragning, sekventiell Monte Carlo, sekventiell Monte Carlo upphöjd till två, studsigt partikeldragare, och så vidare...



*“Danke schön!”*

Angela Merkel

# Acknowledgments

First of all, I would like to thank professor Thomas Schön. It has been a great pleasure to pursue my doctoral studies under your supervision. I am particularly grateful for your positive attitude, support and coaching whenever I have taken on new (more or less research-related) challenges. It has also been a pleasant journey from starting as your second student here in Uppsala, to now graduating from a vivid and growing machine learning group with a two-digit number of members.

I am also very happy that dr Fredrik Lindsten has served as my co-supervisor. With your deep technical knowledge and willingness to share it, it has been a joy working with you, I have learned a lot!

Thanks also to all my co-authors Dave, Petre, Johan D, Arno, Simo, Manon, Lawrence, Niklas, Dennis and Mahmoud for fruitful (and more or less intense) collaborations. Thanks also to Anna and Calle J who helped with the proofreading of the introductory chapters of this thesis.

This thesis had not been written, had not the Swedish Foundation for Strategic Research (SSF, via the project ENSEMBLE, nr RIT15-0012) and Uppsala University generously funded my position. I hope you find it was worth it. Thanks!

As a part of a Swedish PhD education, I have been undertaking some courses. Some were really good. Thank you Erik Broman, Nicolas Chopin, Omiros Papaspiliopoulos and Henrik Hult for your teaching efforts. A special thanks also goes to the organizers of the Machine Learning Summer School in Tübingen 2015.

The most important part of a workplace is probably the colleagues. Thank you Johan W for many interesting discussions (and sorry for interrupting you all the time you're in your office). Thank you Anna, Niklas, Thomas and Fredrik L for fun times teaching together. In addition to some of you already mentioned, thank you Calle A, Fredrik, Koen and David for nice running sessions. A big thanks also to Calle J, Diana, Guo, Helena, Jack, Johan A, Juozas, Lawrence, Marcus, Maria, Niklas, Pelle, Rubén, Tatiana and Viktor for making it worth to walk all the way to the coffee room three times a day (even though I don't even drink coffee myself). Thanks also to Katarina, Dick and Marina for helping out with all administrative and practical issues.

I would also like to thank some people from my life outside the thesis, Jonathan & Elisabet, Julia & Robert, Lina & Bernhard, David, Diana and Victor, for your nice and more or less regular company during these years!

A big thanks of course also goes to my parents, Bosse & Christina, for your encouragement and support throughout my life. Without you, literally, neither me nor this thesis would exist. My sisters, Brita and Anna, also played an inevitable role in fostering me to the one I am today. Whether that is a good or bad thing, I leave to other to decide, but thank you anyway!

And, finally, thank you Sanna, for your existence, patience and love!

*Andreas*

*Uppsala, August 2018*



“Learn—*that’s a trendy word.*”

Andrew Gelman

# Table of contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Focus of the thesis . . . . .	2
1.2	Outline of the introductory chapters . . . . .	3
1.3	Main contributions . . . . .	3
1.4	Articles included in the thesis . . . . .	4
1.5	Related but not included work . . . . .	5
1.6	A word on notation . . . . .	7
<b>2</b>	<b>Learning models from data</b>	<b>9</b>
2.1	Data $y$ . . . . .	10
2.2	Models $p(y   \theta)$ . . . . .	10
2.3	Two paradigms for deducing unknown parameters . . . . .	11
2.3.1	Finding a point estimate for $\theta$ : $\hat{\theta}$ . . . . .	11
2.3.2	Finding the posterior distribution for $\theta$ : $p(\theta   y)$ . . . . .	12
2.4	Posterior distributions vs. point estimates . . . . .	13
2.5	Priors and regularization . . . . .	15
2.5.1	When the prior does not matter . . . . .	15
2.5.2	When the prior does matter . . . . .	15
2.5.3	Circumventing the prior assumptions? . . . . .	18
2.6	Summary of the chapter . . . . .	19
<b>3</b>	<b>State-space models</b>	<b>21</b>
3.1	The general state-space model . . . . .	21
3.2	Linear Gaussian state-space models . . . . .	22
3.3	Jump-Markov linear state-space models . . . . .	23
3.4	Learning state-space models . . . . .	23
3.4.1	Quantities to learn: states and model parameters . . . . .	23
3.4.2	A Bayesian approach or point estimates? . . . . .	24
3.5	Summary of the chapter . . . . .	26
<b>4</b>	<b>Gaussian processes</b>	<b>27</b>
4.1	Introducing the Gaussian process . . . . .	27
4.2	Noise density, mean and covariance functions . . . . .	32
4.3	Hyperparameter learning . . . . .	33
4.3.1	Empirical Bayes: Finding a point estimate $\hat{\eta}$ . . . . .	33
4.3.2	Hyperpriors: Marginalizing out $\eta$ . . . . .	34
4.4	Computational aspects . . . . .	34
4.5	Two remarks . . . . .	35
4.5.1	A posterior variance independent of observed values? . . . . .	35
4.5.2	What is a typical sample of a GP? . . . . .	35
4.6	Gaussian-process state-space models . . . . .	36

4.7	Summary of the chapter	37
<b>5</b>	<b>Monte Carlo methods for machine learning</b>	<b>39</b>
5.1	The Monte Carlo idea	39
5.2	The bootstrap particle filter	41
5.2.1	Resampling	41
5.2.2	Positive and unbiased estimates of $p(y_{1:T}   \vartheta)$	42
5.3	The Markov chain Monte Carlo sampler	43
5.3.1	The Metropolis-Hastings kernel	44
5.3.2	The Gibbs kernel	44
5.3.3	Convergence	45
5.4	The Sequential Monte Carlo sampler	45
5.4.1	Connection to particle filters	46
5.4.2	Constructing a sequence $\{\pi_p\}_{p=0}^P$	46
5.4.3	Propagating the particles	47
5.4.4	Convergence	47
5.5	Markov Chain or Sequential Monte Carlo?	48
5.6	Monte Carlo for state-space model parameters $\vartheta$	49
5.6.1	MCMC for nonlinear state-space models: PMCMC	49
5.6.2	Particle Gibbs for maximum likelihood estimation	50
5.6.3	SMC for state-space model parameters: SMC <sup>2</sup>	51
5.7	Summary of the chapter	51
<b>6</b>	<b>Conclusions and future work</b>	<b>53</b>
6.1	Conclusions	53
6.2	Future work	54
<b>A</b>	<b>The unbiased estimator <math>\widehat{p}_{N_x}(y_{1:T})</math></b>	<b>55</b>
<b>B</b>	<b>The <math>MNIW</math> distribution in linear regression</b>	<b>59</b>
B.1	The matrix normal and inverse Wishart distributions	59
B.1.1	The scalar case: $NI\mathcal{G}$	60
B.1.2	Generalizing to the matrix case: $MNIW$	61
B.2	Scalar linear regression: $y_t = ax_t + e_t$	63
B.3	Multivariable linear regression: $y_t = Ax_t + e_t$	64
	<b>Notation list</b>	<b>65</b>
	<b>References</b>	<b>67</b>

*“I’m being quoted to introduce something, but I have no idea what it is and certainly don’t endorse it.”*

Randall Munroe

# 1

## Introduction

**D**ATA, in the format of recorded numbers, is literally present everywhere. Examples range from temperature measurements, traffic intensity, polls, internet traffic and stock market prices, to speech recordings, electricity usage and epidemiological data. To process such data in computer systems, mathematical models can be helpful.

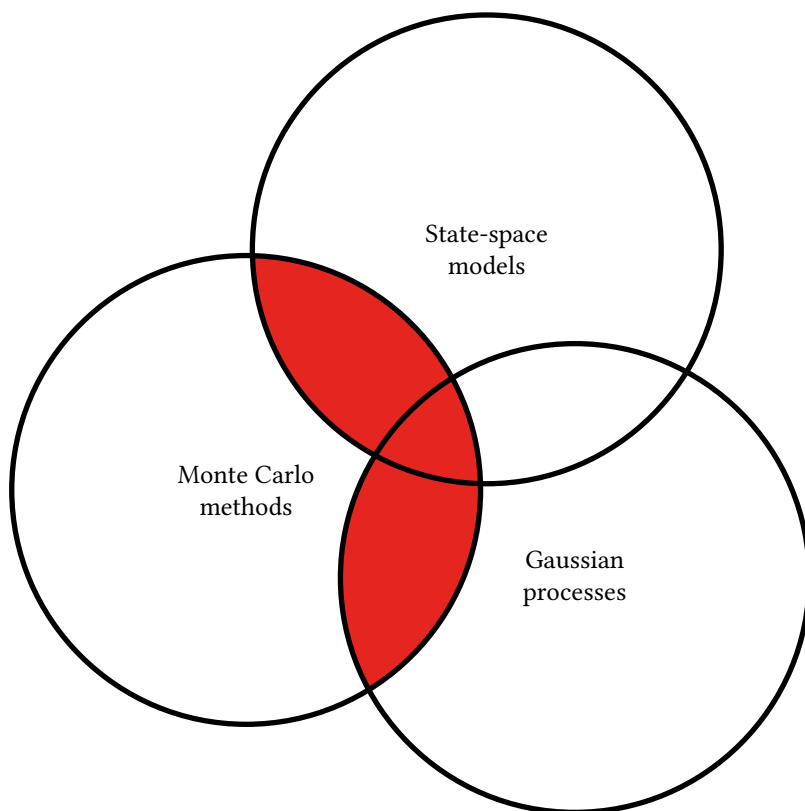
A mathematical model is a set of equations. Those equations can—if carefully chosen—be a compact and powerful tool to describe links and causalities of the data. Thus, by *learning* a mathematical model from recorded data, the learned model can—sometimes—help in explaining the mechanisms behind the data, answer ‘what if’-questions and make predictions for the future evolution. For this reason, *mathematical models are an essential tool for making sense of data*. This thesis has a focus on a subset of mathematical models which are motivated by probability theory and statistics. Such models are referred to as *statistical models*. Statistical models learned from recorded data can, for example, be used to predict future weather, advise on how to drive to avoid congestion, make scenario analysis to decide on future network designs, price assets automatically, translate speech to text, solve energy aggregation problems and predict disease spread.

### Different words for the same thing

In Swedish there is a saying ‘*kårt barn har många namn*’, meaning ‘we have many names for the things we love’. That is definitely true for this topic. Depending on context, names such as *machine learning*, *statistical learning*, *system identification*, *parameter estimation*, *cybernetics* and many more, are used to describe the science of learning mathematical models from data. Somewhat (at least historically) incorrect is also the term *artificial intelligence* used nowadays. Also the term *learning* has many synonyms itself, including *training*, *calibration*, *inference* and *estimation*.

## 1.1 Focus of the thesis

The field of machine learning is vast and ever increasing. Of course, this thesis cannot cover everything, and not even close to. This thesis is primarily about *learning statistical models from data* when *the data has a sequential nature* (such as time series and input-output relationships of dynamical systems). In a technical lingo, the models concerned in this thesis are primarily state-space models (hidden Markov models) and Gaussian processes. The learning, which is done by a computer, is nothing but a set of mathematical computations. There are different methods to perform these computations, and this thesis focuses on a set of methods which makes clever use of randomness, namely Monte Carlo methods. A pictorial view could perhaps look like this, where the shaded red regions are the focus of the thesis:



This thesis, and also the research behind it, is focused on methods rather than applications. That does not mean there are no applications, and examples of applications are given in several of the papers.

## 1.2 Outline of the introductory chapters

The first part of the thesis contains five introductory chapters (including this chapter) and one concluding chapter. The purpose of these introductory chapters is to summarize the background and put the papers into a broader perspective. The concluding chapter, number 6, is meant to be read after the papers.

After this first chapter, we start in Chapter 2 by dissect machine learning into its main pieces *data*, *models* and *how to learn models from data*. Two statistical models are then introduced in detail, the state-space model in Chapter 3 and the Gaussian-process model in Chapter 4. We thereafter devote Chapter 5 to Monte Carlo methods, which are used to make the computations required for the learning. Appendix A contains a central result for sequential Monte Carlo (Chapter 5), and Appendix B contains a derivation of the conjugate prior for Gaussian linear regression; important expressions for Paper I.

## 1.3 Main contributions

The main scientific contributions in this thesis are the following:

- A numerically feasible approximative implementation of the Gaussian process state-space model (Paper I).
- A novel criterion for assessing consistency between data and model (Paper II).
- An introduction to and theoretical analysis of the particle stochastic approximation EM algorithm, as well as its formulation for jump Markov linear models and the empirical Bayes problem (Paper III and IV).
- A novel alternative to SMC<sup>2</sup> for state-space models with highly informative observations (Paper V).
- A novel approach to parameter estimation in non-linear state-space models using particle filters (Paper VI).
- The use of an SMC sampler to learn hyperparameters for the Gaussian process model (Paper VII).

## 1.4 Articles included in the thesis

The second part of this thesis contains scientific articles. The articles are listed below, together with a brief summary and statement of my (the thesis author's) contributions.

### Paper I

Andreas Svensson and Thomas B. Schön (2017). “A flexible state-space model for learning nonlinear dynamical systems”. In: *Automatica* 80, pp. 189–199.

*A conceptually interesting combination of the state-space model and the Gaussian process model is provided by the Gaussian process state-space model (Section 4.6). This article presents a numerically feasible approximation of that combination, from a system identification perspective. My contributions to this paper are the mathematical derivations, implementation and experiments. I have written the major part, but also Thomas B. Schön, to whom the original idea should be attributed, has contributed to the writing.*

### Paper II

Andreas Svensson, Dave Zachariah, Petre Stoica, and Thomas B. Schön (2018). “Data consistency approach to model validation”. Submitted for publication.

*This paper presents a criterion to assess if a given data set and a model class are consistent, in the sense that the given data should be ‘similar’ to data artificially generated from the model. The writing, as well as the coining of the technical idea, was done jointly with Dave Zachariah and Petre Stoica. The implementations are mine.*

### Paper III

Andreas Svensson and Fredrik Lindsten (2018). “Learning dynamical systems with particle stochastic approximation EM”. Submitted for publication.

*The particle stochastic approximation EM (PSAEM) is a method for learning state-space models, based on particle filters and the Expectation-Maximization (EM) algorithm. I have written the major part of the paper and performed the experiments. The idea was originally coined by Fredrik Lindsten, who has also done most of the theoretical analysis.*

### Paper IV

Andreas Svensson, Thomas B. Schön, and Fredrik Lindsten (2014). “Identification of jump Markov linear models using particle filters”. In: *Proceedings of the 53<sup>rd</sup> IEEE Conference on Decision and Control (CDC)*. Los Angeles, CA, USA, pp. 6504–6509.

*The PSAEM method, which is the topic of Paper III, is here adapted to jump Markov linear models, a special class of state-space models. The original idea is due to Thomas B. Schön and Fredrik Lindsten, whereas I have done the major part of the writing and all implementations.*

## Paper V

Andreas Svensson, Thomas B. Schön, and Fredrik Lindsten (2018). “Learning of state-space models with highly informative observations: a tempered Sequential Monte Carlo solution”. In: *Mechanical Systems and Signal Processing* 104, pp. 915–928.

*State-space models with very little or no measurement noise turn out, perhaps surprisingly, to be very hard to learn with methods based on the particle filter. To this end, a scheme is proposed where artificial measurement noise is introduced and gradually decreased, in a consistent way. The original idea, analysis and implementation are all my work, and I have also done the major part of the writing.*

## Paper VI

Andreas Svensson, Fredrik Lindsten, and Thomas B. Schön (2018). “Learning nonlinear state-space models using smooth particle-filter-based likelihood approximations”. In: *Proceedings of the 18<sup>th</sup> IFAC symposium on system identification (SYSID)*. Stockholm, Sweden, pp. 652–657.

*This paper is a novel approach to maximum likelihood estimation of unknown parameters in non-linear state-space models. By scrutinizing the particle-filter algorithm, a slightly different interpretation of it can be found, which can be used to formulate a maximization problem to which conventional optimization methods can be applied. The original idea and implementation are all my work, and I have also written the major part of the paper. The analysis was done jointly with Fredrik Lindsten and Thomas B. Schön.*

## Paper VII

Andreas Svensson, Johan Dahlin, and Thomas B. Schön (2015). “Marginalizing Gaussian process hyperparameters using sequential Monte Carlo”. In: *Proceedings of the 6<sup>th</sup> IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. Cancún, Mexico, pp. 489–492.

*The hyperparameters in the Gaussian process model can, if unknown, either be estimated or marginalized. This paper suggests the use of the sequential Monte Carlo sampler for the latter. The original idea should be attributed to Johan Dahlin, whereas I have implemented the examples and written the major part of the paper.*

## 1.5 Related but not included work

Beyond the articles included in the thesis, the following research (to which I have contributed) is also relevant to this thesis:

- [A] Andreas Svensson, Dave Zachariah, and Thomas B. Schön (2018). “How consistent is my model with the data? Information-theoretic model check”. In: *Proceedings of the 18<sup>th</sup> IFAC symposium on system identification (SYSID)*. Stockholm, Sweden, pp. 407–412.

- [B] Thomas B. Schön, Andreas Svensson, Lawrence M. Murray, and Fredrik Lindsten (2018). “Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo”. In: *Mechanical Systems and Signal Processing* 104, pp. 866–883.
- [C] Dennis W. van der Meer, Mahmoud Shepero, Andreas Svensson, Joakim Widén, and Joakim Munkhammar (2018). “Probabilistic forecasting of electricity consumption, photovoltaic power generation and net demand of an individual building using Gaussian Processes”. In: *Applied Energy* 213, pp. 195–207.
- [D] Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas B. Schön (2016). “Computationally efficient Bayesian learning of Gaussian process state space models”. In: *Proceedings of the 19<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS)*. Cádiz, Spain, pp. 213–221.
- [E] Andreas Svensson, Thomas B. Schön, Arno Solin, and Simo Särkkä (2015). “Nonlinear state space model identification using a regularized basis function expansion”. In: *Proceedings of the 6<sup>th</sup> IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. Cancún, Mexico, pp. 493–496.
- [F] Thomas B. Schön, Fredrik Lindsten, Johan Dahlin, Johan Wågberg, Christian A. Naesseth, Andreas Svensson, and Liang Dai (2015). “Sequential Monte Carlo methods for system identification”. In: *Proceedings of the 17<sup>th</sup> IFAC Symposium on System Identification (SYSID)*. Beijing, China, pp. 775–786.
- [G] Andreas Svensson and Thomas B. Schön (2016). *Comparing two recent particle filter implementations of Bayesian system identification*. Tech. rep. 2016-008. (Presented at Reglermöte 2016, Gothenburg, Sweden). Department of Information Technology, Uppsala University.
- [H] Andreas Svensson, Thomas B. Schön, and Manon Kok (2015). “Nonlinear state space smoothing using the conditional particle filter”. In: *Proceedings of the 17<sup>th</sup> IFAC Symposium on System Identification (SYSID)*. Beijing, China, pp. 975–980.
- [I] Andreas Svensson (2016). “Learning probabilistic models of dynamical phenomena using particle filters”. Licentiate thesis. Department of Information Technology, Uppsala University.

In addition, I have also contributed to some pedagogic work also of relevance to this thesis:

- [J] Fredrik Lindsten, Andreas Svensson, Niklas Wahlström, and Thomas B. Schön (2018). *Statistical Machine Learning. Lecture notes on linear regression, logistic regression, deep learning & boosting*. Department of Information Technology, Uppsala University.
- [K] Fredrik Lindsten, Thomas B. Schön, Andreas Svensson, and Niklas Wahlström (2017). *Probabilistic modeling—linear regression & Gaussian processes*. Department of Information Technology, Uppsala University.
- [L] Andreas Svensson (2013). *Particle Filter Explained without Equations*. URL: <https://www.youtube.com/watch?v=aUkBa1zMKv4>.

## 1.6 A word on notation

The first part of the thesis (Chapter 1–5) are meant to have a consistent notation (a complete list can be found on page 65). The notation in the included articles is, however, slightly different, and introduced separately for each article.

In general we use a probabilistic language and notation, and use the word ‘model’ mainly to refer to some (possibly complicated) probability distribution. We work in the first place with probability distributions in terms of their densities (or mass)  $p(\cdot)$ , and thereby we implicitly assume its existence. The generalization to the case when the density does not exist is often possible, but not covered. Also the existence of a  $\sigma$ -algebra and dominating measure is implicit. Different densities are distinguished by their arguments, and  $p(\cdot | \cdot)$  denotes a conditional density.

All random variables are written with lowercase letters  $x$ ,  $\theta$ , etc., and no distinction between random variables and their realizations is made in the notation. Integrals without any explicit limits are over the entire domain of the integration variable.

## CHAPTER 1. INTRODUCTION

*“All models are wrong  
but some are useful.”*

George E. P. Box

# 2

## Learning models from data

**M**ACHINE learning is a multifaceted term, but in this thesis we will understand it as the *processing of observed data into a statistical model*. The key elements in this process are

- (i) the recorded *data*  $y$  (Section 2.1),
- (ii) the statistical *model*  $p(y | \theta)$ , which has some degrees of freedom expressed via a parameter  $\theta$  (Section 2.2),
- (iii) the *learning* which links the model to the data by drawing conclusions about  $\theta$  from  $y$  (Sections 2.3–2.5).

Loosely speaking, one could say that learning is to extract the essence of the data  $y$ , and put that information into the parameters  $\theta$ . In this chapter, we think in the first place of  $\theta$  as being finite-dimensional and real-valued. In Chapter 4, we consider a model where  $\theta$  is infinite-dimensional.

The purpose of this chapter is to give an introduction to the way we think about data and models, and also to cover some background on the learning side. We use a statistical perspective, and sometimes use the more technical term *statistical inference* for learning. We cover two different paradigms for how to perform the inference, namely the point estimation approach and the Bayesian approach.

We remain on a rather general level in this chapter. Particular examples of data and models will be introduced first in Chapter 3 and 4, as well as in some of the papers. We also leave integrals and optimization problems hanging in midair without attempting to compute them for now. Methods for performing these computations are introduced in Chapter 5 and in some of the papers.

## 2.1 Data $y$

The first and foremost thing in machine learning is the data  $y$ . The data could in principle be anything that can be recorded, but we limit ourselves to numbers, typically (but not necessarily) recorded sequentially during some period of time, as<sup>1</sup>  $y = \{y_1, \dots, y_T\}$ . The data could be artificially generated by a computer, but in most (if not all) cases of interest for the broader society, the data is recorded from some real phenomenon which is not yet completely understood. Examples of typical data could be logs of outdoor temperatures, measured forces in a mechanical system, or stock prices. We make no assumptions on the data, other than that it exists and has a certain format (such as  $y \in \mathbb{R}^{n_y \times T}$  or similar). Throughout this thesis, we will be in the position that the data is already recorded and available to us, meaning that questions on how to record the data or how to design experiments to ‘reveal as much information as possible’ falls outside of the scope of this thesis (see, e.g., Chaloner and Verdinelli 1995; Hjalmarsson 2009; Pukelsheim 1993).

Throughout the first part of this chapter, we use a toy example to illustrate the concepts. Let us therefore say that we have data which consists of two observations  $y_1 = 1.54$  and  $y_2 = 3.72$  ( $n_y = 1, T = 2$ ).

## 2.2 Models $p(y | \theta)$

Next, we introduce a model for the data  $y$ , which we denote by  $p(y | \theta)$ . The notation  $p(y | \theta)$  encodes a probability distribution which is assumed to be able to describe  $y$  in some respect. Crucially, the model might depend on some unknown parameter  $\theta$ . This unknown parameter will later be learned, and for that result to be as relevant as possible, it is good if the present knowledge (and ignorance) about the problem is included in the model: if the data exhibits strong saturation effects, the learned parameters in a linear model might carry very little insights.

The model can be derived from first principles (such as Newton’s laws of motions, Leavitt’s law or the ideal gas law) where the unknown parameters have some physical interpretation. The model can also be of a more generic flexible type, where the parameters carry no direct physical interpretation. The latter is perhaps more of a typical machine-learning case.

Let us demystify the abstract notion of a model by using the toy example. Say that we decide to model the data points as independent draws from the same Gaussian distribution. The unknown parameters are then the mean  $\mu$  and the variance  $\sigma^2$  of the Gaussian distribution, i.e.,  $\theta \triangleq \{\mu, \sigma^2\}$ , and we write the model as

$$p(y | \theta) = \mathcal{N}(y_1; \mu, \sigma^2) \cdot \mathcal{N}(y_2; \mu, \sigma^2). \quad (2.1)$$

Note that we have not limited ourselves to data actually generated by  $p(y | \theta)$ : the model is only an *assumption* within the learning procedure. We are, in fact, free to make arbitrary model assumptions, perhaps in the interest of feasible computations! This means that inference results should *always be read with the model assumptions in mind*. To validate a model assumption  $p(y | \theta)$  for some data  $y$ , we present a new method in Paper II.

---

<sup>1</sup>We later use the notation  $y_{1:t} = \{y_1, \dots, y_t\}$  when there is a need to emphasize exactly which data points are under consideration. For now, we settle with only  $y$  to denote all the available data.

## 2.3 Two paradigms for deducing unknown parameters

In our notation, we prepared for the learning by including the possible dependence on a parameter  $\theta$  in the model  $p(y | \theta)$ . The big question throughout the rest of this chapter is the following: if an unknown parameter  $\theta$  is present in the model, how should the data  $y$  and the model  $p(y | \theta)$  be used for drawing conclusions about  $\theta$ ?

This question is at the core of statistical inference, and some textbook references are Casella and R. L. Berger (2002), Gelman et al. (2014), and Schervish (1995). The field has traditionally been divided into several paradigms, ultimately differing perhaps in their interpretation of probabilities. We will pursue two alternative ways of handling the unknown parameters  $\theta$  throughout the thesis. The two following questions are meant to reflect the underlying alternative reasoning about how to learn  $\theta$ ,

- (i) which estimate  $\hat{\theta}$  fits the data  $y$  the best?
- (ii) after digesting the information brought to us by the data  $y$ , what degree of belief  $p(\theta | y)$  do we have in different values of  $\theta$ ?

We will refer to these alternatives as (i) the *point estimation* and (ii) the *Bayesian* approach, respectively. The distinction between the different paradigms is not always entirely clear in the literature, but below we give an explanation of the way in which we use the terms. Some texts on the major paradigms in statistical inference, in addition to the discussion below, are Efron (1986), Efron (2013), Efron and Hastie (2016), and Lindley (1990).

### 2.3.1 Finding a point estimate for $\theta$ : $\hat{\theta}$

The first learning approach we review, is that of finding a *point estimate*  $\hat{\theta}$  that fits the observed data as well as possible. Which particular point estimate  $\hat{\theta}$  to choose, i.e., the meaning of ‘fit’ in the rhetoric question above, might however vary. One choice (common in this thesis) is to maximize the likelihood function

$$\mathcal{L}(\theta) \triangleq p(y | \theta), \tag{2.2}$$

an alternative that we will refer to as *maximum likelihood*. Note that even though the likelihood function  $\mathcal{L}(\theta)$  is a function of  $\theta$ , the object  $p(y | \theta)$  describes how ‘likely’  $y$  (not  $\theta$ ) is. Also note that in (2.2), the data  $y$  is fixed<sup>2</sup>, in contrast to when we talk about the model  $p(y | \theta)$ . Two alternatives to maximum likelihood are to either optimize the predictive capabilities of the model, or to maximize the likelihood function subject to some additional constraints, such as keeping the numerical values close to zero or promote sparsity in  $\hat{\theta}$ . The latter alternatives are often referred to as *regularization*, a theme we will return to in Section 2.5.2. The choice of which point estimate to use can be formalized mathematically using decision theory, a topic not considered in this thesis (see, e.g., Schervish 1995, Chapter 3).

Computing point estimates  $\theta$  is often at the heart of the classical/frequentist/Neyman-Pearson-Wald school in the literature, whereas inference based on the likelihood function historically is separated into the Fisherian tradition. In the statistical

<sup>2</sup>In the traditional notation with uppercase letter for random variables, and lowercase for their realizations, we could write (2.2) as  $\mathcal{L}(\theta) \triangleq p(Y = y | \theta)$ , whereas the term ‘model’ refers to  $p(Y | \theta)$ .

literature, it is also common to introduce confidence regions expressing uncertainty about  $\hat{\theta}$  (not  $\theta$ ). In this thesis, we group these schools together as the point estimation approach. We refrain from putting focus on confidence regions, because of the tradition and available computational tools for the models to be presented later on.

In the toy example from above, a maximum likelihood point estimate  $\hat{\theta} \triangleq \{\hat{\mu}, \hat{\sigma}^2\}$  is found by solving the problem

$$\hat{\theta} = \arg \max_{\theta} \mathcal{L}(\theta) = \arg \max_{\mu, \sigma^2} \mathcal{N}(1.54; \mu, \sigma^2) \cdot \mathcal{N}(3.72; \mu, \sigma^2). \quad (2.3)$$

The solution turns out to be  $\hat{\mu} = 2.63$  and  $\hat{\sigma}^2 = 1.09$ , i.e., some numbers  $\hat{\theta}$  that can be used to analyze the data, making subsequent predictions, etc.

### 2.3.2 Finding the posterior distribution for $\theta$ : $p(\theta | y)$

The second approach relates to the interpretation of probabilities as degrees of belief, and makes use of Bayes' theorem (named after Thomas Bayes 1763)

$$p(\theta | y) = \frac{p(y | \theta)p(\theta)}{p(y)} \quad (2.4)$$

to update the *prior* belief  $p(\theta)$  into the *posterior* belief  $p(\theta | y)$ . Going from the prior to the posterior can be understood as *conditioning the belief on data*. The right hand side of (2.4) contains, apart from the prior  $p(\theta)$ , the *likelihood* (or *data density*)  $p(y | \theta)$  and<sup>3</sup>  $p(y)$ .

*Bayesian* inference is all about computing the posterior  $p(\theta | y)$ . The central role of Bayes' theorem is the obvious reason behind the name of the paradigm. However, also the name of Pierre-Simon de Laplace (1820) (Stigler 1986) and Bruno de Finetti (1992) occurs in the literature.

There is nothing conceptually different between the prior  $p(\theta)$  and the posterior  $p(\theta | y)$ : they both reflect a degree of belief about  $\theta$ , before and after observing the data  $y$ , respectively. If more data is observed subsequently, Bayes' theorem may be applied repeatedly to incorporate the new observations into the belief. However, Bayes' theorem only provides a mechanism for *updating* beliefs, not creating beliefs from nothing. Therefore, the prior  $p(\theta)$  has to be *chosen*<sup>4</sup>. To obtain a useful result, the choice of prior should preferably reflect present ignorance and knowledge, in the very same way as the model  $p(y | \theta)$  should be chosen carefully.

<sup>3</sup>Note that  $p(y)$ , the denominator, can be written as an integral over the numerator with respect to  $\theta$ ,  $p(y) = \int p(y | \theta)p(\theta)d\theta$ .  $p(y)$  can thus be thought of as a normalization to ensure that  $\int p(\theta | y) = 1$ , and can be ignored if it is sufficient to compute (2.4) up to proportionality.

<sup>4</sup>The (inevitably subjective) prior choice is, according to the authors personal experience, one of the main criticisms that is brought up towards the Bayesian paradigm. It is, however, unclear to the author why the prior choice should be subject to criticism, whereas the equally subjective model choice (present in both paradigms) mostly is kept out of the discussion. J. O. Berger (2006) comments to this concern as follows: '[The model choice] will typically have a much greater effect on the answer than will such things as choice of prior distributions for model parameters. Model-building is not typically part of the objective/subjective debate, however—in part because of the historical success of using models, in part because all the major philosophical approaches to statistics use models and, in part, because models are viewed as "testable," and hence subject to objective scrutiny. It is quite debatable whether these arguments are sufficient to remove model choice from the objective/subjective debate, but I will simply follow statistical (and scientific) tradition and do so.'

It should, however, be remembered that all degree of belief in  $\theta$  is still conditional on the choice of model  $p(y | \theta)$ . A popular sales pitch for the Bayesian approach is that the posterior distribution provides *uncertainty* about the parameters. That is indeed true, but note that the use of Bayes' theorem does not imply a question whether  $p(y | \theta)$  is relevant for modeling  $y$ ; if no choice of the unknown  $\theta$  gives a reasonable model for  $y$ , the posterior belief  $p(\theta | y)$  will only concentrate around the 'least bad' parameter value.

It is also possible to extract point estimates of  $\theta$  from the posterior  $p(\theta | y)$ . Popular estimates of that kind are the posterior mean and the posterior mode, where the latter usually is referred to as maximum a posteriori (MAP) estimation. However, since a point estimate does not represent a degree of belief (which is a core component in the Bayesian approach), we do not consider it to be a Bayesian method here. It does, however, bear some resemblances to the regularized maximum likelihood approach, as we will see in Section 2.5.2.

Let us have a look at the little toy example again, now from the Bayesian point of view. First, we have to append our assumption  $p(y | \theta)$  also with assumptions about  $\mu$  and  $\sigma^2$ . Let us assume a normal-inverse-gamma prior distribution (Appendix B),  $p(\mu, \sigma^2) = \text{NIG}(\mu, \sigma^2; 0, 1, 1, 1)$ . By inserting all expressions into Bayes' theorem (2.4) and performing some algebraic manipulations, we find the posterior  $p(\mu, \sigma^2 | y) = \text{NIG}(\mu, \sigma^2; 1.75, 3, 2, 4.49)$ . This is a distribution, which we may use subsequently to analyze the data, do predictions, etc.

The particular choice of prior in the toy example was a so-called *conjugate prior* since the prior, a *NIG* distribution, together with the likelihood model (2.1), a Gaussian distribution with unknown mean and variance, yields another *NIG* distribution as the posterior. For some priors, the posterior may not admit a closed form, and conjugate priors only exist for a limited set of models.

## 2.4 Posterior distributions vs. point estimates

The most striking difference between the point estimates and the Bayesian paradigm for a user, is perhaps not the different underlying philosophies about the meaning of probabilities, nor the presence or absence of priors. Instead, the major difference for a user is that point estimates  $\hat{\theta}$  and distributions  $p(\theta | y)$  are very different objects: A point estimate  $\hat{\theta}$  is a number, whereas  $p(\theta | y)$  is, well, a distribution. If the user interest is, for example, to predict a future observation  $y^*$ , the point estimation approach is to put  $\hat{\theta}$  into the model and take the mean

$$\hat{y}^* = \mathbb{E} [p(y^* | \hat{\theta})] \quad (2.5)$$

as the (point) prediction  $y^*$ . For the Bayesian case on the contrary<sup>5</sup>, the prediction of  $y^*$  is the predictive distribution

$$p(y^* | y) = \int p(y^* | \theta) p(\theta | y) d\theta. \quad (2.6)$$

<sup>5</sup>Indeed,  $p(y^* | \hat{\theta})$  is also a distribution. However, as it bears no meaning akin to (2.6), and the point estimation approach is more concerned with point estimates, the entire distribution  $p(y^* | \hat{\theta})$  is typically not considered, but only its mean (2.5), or similar.

In many cases, the predictive distribution (and often also the posterior) admits no closed form expression. Instead, those distributions have to be approximated. Two such approximative alternatives are provided by the variational approach (e.g., Blei et al. 2016) and the Monte Carlo approach (Chapter 5).

Whether to take the point estimate or the Bayesian approach, may depend on several aspects. Often, but not always, point estimation can be less computationally intensive compared to the Bayesian approach, an argument for preferring the former. However, if the computational aspect allows a choice, one may consider questions such as

- What is the intended use of the obtained results: does a posterior distribution  $p(\theta | y)$  provide valuable information in the solution, which is not preserved by a single point estimate  $\hat{\theta}$ ?
- Is it sensible, or even crucial, to include prior beliefs about  $\theta$  into the solution? (See Section 2.5.2)

Personal preferences may of course also influence the choice: point estimates have, for example, traditionally dominated the system identification community (an interesting uphill struggling paper arguing for the Bayesian approach is Peterka (1981)).

If the data is highly informative about the parameters  $\theta$ , the differences between the two paradigms may be small. Consider a toy example with  $T$  observations  $\{y_t\}_{t=1}^T$  of a one-dimensional parameter  $\theta \triangleq \mu$ . We model the observations to be exchangeable (see, e.g., Section 1.2 in Schervish 1995) and all have a Gaussian distribution with mean  $\mu$  and variance 1, and we assume a prior  $p(\mu) = \mathcal{N}(\mu; 0, 1)$ . This yields the posterior

$$p(\mu | y) \propto \underbrace{\mathcal{N}(\mu; 0, 1)}_{p(\mu)} \underbrace{\prod_{t=1}^T \mathcal{N}(\mu; y_t, 1)}_{p(y | \mu)} \quad (2.7a)$$

which after some algebraic manipulations can be written

$$p(\mu | y) = \mathcal{N}\left(\mu; \frac{\sum_{t=1}^T y_t}{T+1}, \frac{1}{T+1}\right). \quad (2.7b)$$

That is, the posterior variance tends to 0 as the number of observations  $T \rightarrow \infty$ . *Thus, with a large number of observations  $T$ , it may (from a practical point of view) suffice to represent the (Bayesian) posterior (2.7b) with a single point estimate!*

By this argument, one may catch a sight of a bridge between the two paradigms. The argument is often relevant when  $T \rightarrow \infty$ , not only for the toy case (2.7). It is, however, not completely generally applicable, for instance not if

- (i) the number of parameters is large, so that the ‘information per parameter’ is still low despite a large number of observations  $T$ ,
- (ii) the data cannot determine the parameters uniquely, e.g.,  $\theta = \{\alpha, \beta\}$ , but only information about the product  $\alpha \cdot \beta$  is observed (a problem sometimes referred to as non-identifiability),
- (iii) the variance in the example model would have been proportional to  $T$  instead of 1, which would yield a posterior variance that does not decrease with  $T$ .

## 2.5 Priors and regularization

Let us now consider the role of the prior. The prior has a central role in the Bayesian approach, and is not present at all when computing maximum likelihood point estimates. Its presence may therefore appear as a major difference between the two approaches. The role of the prior is, however, not always crucial when it comes to the practical aspects, as we will discuss in this section.

### 2.5.1 When the prior does not matter

From the previous section, we have the example of  $T$  exchangeable observations of  $\mu$  with Gaussian noise, where we also could write (cf. (2.7b))

$$p(\mu | y) = \mathcal{N}\left(\mu; \frac{\sum_{t=1}^T y_t}{T+1}, \frac{1}{T+1}\right) \approx \mathcal{N}\left(\mu; \frac{\sum_{t=1}^T y_t}{T}, \frac{1}{T}\right) = p(y | \mu) = \mathcal{L}(\theta), \quad (2.8)$$

i.e., the posterior and the likelihood function are approximately equal, and the mode of the posterior is approximately the same as the maximum likelihood solution when there is a large amount of data available ( $T$  large). One may say that ‘the prior is swamped by the data’ or refer to the situation as ‘stable estimation’ (J. O. Berger 1985, Section 4.7.8; Vaart 1998, Section 10.2). It is, however, possible to construct counterexamples, such as pathological cases with Dirac priors etc.

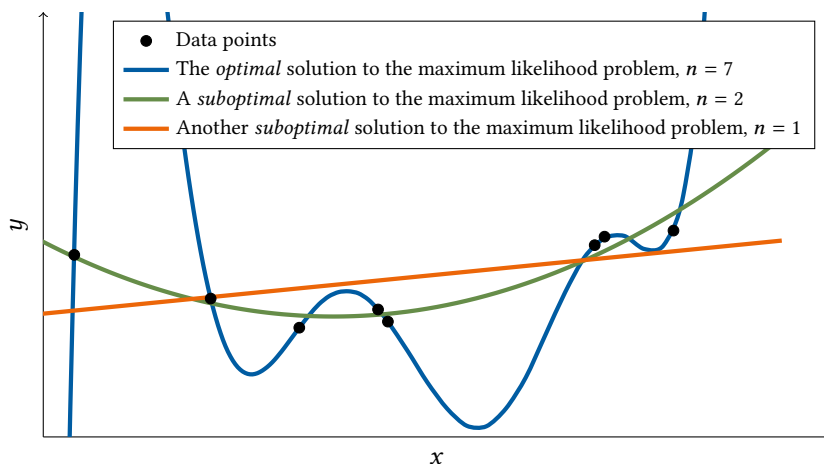
### 2.5.2 When the prior does matter

The point estimation, and in particular the maximum likelihood approach, might seem intuitively appealing: ‘finding the parameter  $\theta$  for which the data  $y$  is as likely as possible’ sounds very reasonable. It is, however, important to realize that this is *not* equivalent to ‘finding the most likely parameter  $\theta$  given the data  $y$ ’. The latter statement is related to the posterior  $p(\theta | y)$ , whereas the former is related to the likelihood function  $\mathcal{L}$ . Failing to distinguish between these is sometimes referred to as ‘the fallacy of the transposed conditional’. We illustrate this by the toy example in Figure 2.1: Consider 8 data points on the form  $(x, y)$ . We make the decision to model the data using an  $n$ th order polynomial and Gaussian measurement noise as

$$p(y | \theta) = \mathcal{N}(y; c_0 + c_1x + c_2x^2 + \dots + c_nx^n, \sigma_n^2). \quad (2.9)$$

We let the polynomial order be undecided, meaning that  $\theta = \{n, c_0, \dots, c_n, \sigma_n^2\}$ . This is arguably a very flexible model, which is able to take many different shapes: a feature that might be desired by the user who wishes not to make too many restrictions beforehand. The maximum likelihood solution is  $n = 7$  (i.e., as many degrees of freedoms as data points),  $\sigma_n^2 = 0$  (i.e., no noise) and  $c_0, \dots, c_7$  chosen to fit the data perfectly. This is illustrated by the solid blue line in Figure 2.1. Two suboptimal solutions, *not* maximizing the likelihood function for this flexible model of polynomials with arbitrary orders, are  $n = 2$  (green) and  $n = 1$  (orange), also shown in Figure 2.1.

Studying Figure 2.1, we may ask ourselves if the 7th order polynomial, the maximum likelihood solution, actually is able to capture and generalize the data well? Indeed all data points are exactly on the blue line, but the behavior in between the data points is not very appealing to our intuition—instead the 2nd or perhaps even



**Figure 2.1:** Eight data points marked with black dots, modeled using  $n$ th order polynomials and Gaussian noise, where the polynomial order  $n$  is undecided. The optimal maximum likelihood solution is  $n = 7$ , with the 8 polynomial coefficients chosen such that it (blue curve) fits the 8 data points perfectly. Two suboptimal solutions are  $n = 2$  (green curve) and  $n = 1$  (orange curve), which—despite their suboptimality in a maximum likelihood sense—both might appear to be more sensible models, in terms of inter- and extrapolating the behavior seen in the data. The key aspect here is that the maximum likelihood solution is explaining the data the best exactly as it is seen; indeed, the blue curve fits the data perfectly. There is, however, no claim that the blue curve is the ‘most likely solution’ (cf. the Bayesian approach). The green and orange curves could, however, have been obtained as regularized maximum likelihood estimates, if a regularization term penalizing large values of  $n$  had been added to the objective function (2.2).

the 1st order polynomial would be more reasonable, even though none of them fit the data exactly. The problem with the blue line, the maximum likelihood solution, is often referred to as *overfitting*. Overfitting occurs when the parameter estimate is adapted to some behavior in the data which we do not believe should be considered as useful information, but rather as stochastic noise.

There are several solutions proposed for how to avoid overfitting, such as aborting the optimization procedure prematurely (early stopping: e.g., Duvenaud, Maclaurin, et al. 2016; Sjöberg and Ljung 1995), some ‘information criteria’ (e.g., the Akaike information criterion, AIC: Akaike 1974, or the Bayesian information criterion, BIC: Schwarz 1978) or the use of cross-validation (Hastie et al. 2009, Section 7.10). We will, however, try to understand the overfit problem as an unfortunate ignorance during the modeling process: From Figure 2.1, we realize that we may actually have a preference for a lower order polynomial, and our mistake is that we have considered the maximum likelihood approach when we actually had different prior beliefs in different parameter values: we prefer the predictable behavior of a low order polynomial to avoid the strange behavior of a higher order polynomial.<sup>6</sup>

<sup>6</sup>The related philosophical question whether simpler models (in this case, a 1st or 2nd order polynomial) should be preferred over more advanced models (the 7th order polynomial) is often referred to as Occam’s razor or the principle of parsimony, a discussion we leave aside.

In the Bayesian framework, on the other hand, the prior  $p(\theta)$  is also taken into consideration using Bayes' theorem (2.4). Via Bayes' theorem, it is (on the contrary to maximum likelihood) possible to reason about likely parameters. A sensibly chosen prior would in the example describe a preference for low order polynomials, and the posterior would then dismiss the 7th order polynomial solution (unless it had fitted the data significantly better than a low order polynomial). Hence, there is no Bayesian counterpart to the overfit problem, an advantage that comes at the price of instead having to choose a prior and working with probability distributions rather than point estimates.<sup>7</sup>

Either inspired by the Bayesian approach or heuristically motivated, a popular modification of the maximum likelihood approach is *regularized* maximum likelihood, which appends the likelihood function with a regularization term  $R(\cdot)$ . The regularization plays a role akin to that of the prior, by 'favoring' solutions of, e.g., low orders. There are a few popular choices of  $R(\cdot)$  with a variety of names, such as the  $\|\cdot\|_1$  norm (Lasso or  $L_1$  regularization: Tibshirani 1996), the  $\|\cdot\|_2$  norm ( $L_2$  or Tikhonov regularization, ridge regression: Hoerl and Kennard 1970; Phillips 1962), or a combination thereof (elastic net regularization: Zou and Hastie 2005).

The connection between regularization and the Bayesian approach can be detailed as follows: If having a scalar  $\theta$  with prior  $\mathcal{N}(\theta; 0, \sigma^2)$ , the logarithm of the posterior becomes

$$\log p(\theta | y) = \log p(y | \theta) + \log p(\theta) - \log p(y) = C + \log p(y | \theta) - |\theta|^2, \quad (2.10)$$

which apart from the constant  $C$  is equivalent to the regularized (log) likelihood function

$$\mathcal{L}^r(\theta) = \log p(y | \theta) - R(\theta), \quad (2.11)$$

if  $R(\cdot) = \|\cdot\|_2$ , i.e.,  $L_2$  regularization. The same equivalence can be shown for  $L_1$  and the use of a Laplace prior. Thus, regularization gives another connection between the point estimation and the Bayesian approach.

In 1960, Bertil Matérn wrote in his thesis on stochastic models that *'needless to say, a model must often be almost grotesquely oversimplified in comparison with the actual phenomenon studied'* (Matérn 1960, p. 28). As long as the statement by Matérn holds true and the model is rigid and much less complicated than the behavior of the data (which perhaps was the case for most computationally feasible models in 1960), regularization is probably of limited interest. However, if the model class under consideration is more complex and contains a huge number of parameters, overfitting may be an actual problem. In such cases, additional information encoded in priors or regularization has in several areas proven to be of great importance, such as compressed sensing (Eldar and Kutyniok 2012) with applications in, e.g., MRI (Lustig et al. 2007) and face recognition (Wright et al. 2009), machine learning (Hastie et al. 2009, Chapter 5) and system identification (T. Chen et al. 2012, Paper I). The increased access to cheap computational power during the last decades might therefore explain the massive recent interest in regularization.

<sup>7</sup>There are two different perspective one can take when understanding the absence of overfitting in the Bayesian paradigm: Pragmatically seen, any sensible prior will (as argued in the text) have a regularizing effect. From a more philosophical point of view, there is no overfitting since the posterior *by definition* represents our (subjective) beliefs about the situation, and therefore contains nothing but useful information (and hence no overfitting to non-informative noise).

### 2.5.3 Circumventing the prior assumptions?

Sometimes the user of the Bayesian approach might feel uncomfortable making prior assumptions, perhaps in the interest of avoiding another subjective choice (in addition to the model choice  $p(y | \theta)$ ). Several alternatives for avoiding, or at least minimizing the influence of the prior choice, have therefore been investigated.

#### ‘Noninformative’ priors

Attempts to formulate ‘noninformative’ priors containing ‘no’ prior knowledge have been made. In the toy example above, a ‘noninformative’ prior for  $\sigma^2$  would intuitively perhaps be a flat prior  $p(\sigma^2) \propto 1$  for  $\sigma^2 > 0$ , since it puts equal mass on all feasible values for  $\sigma^2$ . Apart from the obvious fact that such a density would not integrate to 1, there is also a more subtle and disturbing issue: why should the variance  $\sigma^2$ , and not the standard deviation  $\sigma$ , have a flat prior? In fact, if the prior for the variance  $\sigma^2$  is  $p(\sigma^2) \propto 1$  for  $\sigma^2 > 0$ , it implies that the prior for the standard deviation  $\sigma$  is  $p(\sigma) \propto \sigma$  for  $\sigma > 0$ , which does not appear very ‘noninformative’.

To avoid this undesired effect, a prior that is invariant under re-parametrizations has been proposed, the so-called Jeffreys prior. Jeffreys prior is, however, not always ‘noninformative’ in the sense that a flat prior intuitively is: Efron (2013) provides a simple example where the Jeffreys prior has a clear and perhaps unwanted influence on the posterior. On this topic, Peterka (1981) writes ‘*However, it turns out that it is impossible to give a satisfactory definition of “knowing nothing” and that a model of an “absolute ignorant”, in fact, does not exist. (Perhaps, for the reason that an ignorant has no problems to solve.)*’.

J. O. Berger (2006) argues, on the other hand, that the process of translating expert knowledge into prior assumptions are typically costly (and not always very crucial to the final result), and ‘standard’ priors (such as Jeffreys) should for this reason be considered by the practitioner: it is still far more useful than abandoning the Bayesian approach entirely.

#### Hyperparameters and empirical Bayes

Another alternative is to choose a prior  $p(\theta | \eta)$  with some undecided *hyperparameters*  $\eta$ , and choose a point estimate  $\hat{\eta}$  which fits the data. This is commonly referred to as *empirical Bayes* or maximum likelihood type II. This combination of point estimation and Bayesian inference is perhaps more pragmatic than faithful to any of the paradigms, but can be seen as a promising combination of them, indeed proven to work well in many situations (see, e.g., Paper III; Bishop 2006; Efron 2013 and references therein).

Since empirical Bayes involves point estimation, overfitting may occur, in that the prior becomes overly adapted to the data. In many situations, this only has minor practical implications (typically not as severe as the situation in Figure 2.1), but the user should be aware of the risk.

#### Hyperpriors

A third option on the topic of circumventing the explicit formulation of prior assumptions, is to take a Bayesian (rather than a point estimation) approach to hyperpa-

rameters, and formulate *hyperpriors* on the hyperparameters  $\eta$ . Then, the inference amounts to inferring

$$p(\eta | y) = \int \frac{p(y | \theta)p(\theta | \eta)p(\eta)}{p(y)} d\theta \quad (2.12)$$

rather than  $p(\theta | y)$ . For a subsequent prediction, the prediction  $p(y^* | y)$  would instead of (2.6) be

$$p(y^* | y) = \iint p(y^* | \theta)p(\theta | \eta)p(\eta | y) d\theta d\eta. \quad (2.13)$$

Obviously such a nested construction does not avoid the choice of a prior, but only defers it to the level of  $p(\eta)$  instead of  $p(\theta)$ , and also adds to the computational complexity of the sometimes already involved computations needed. However, in cases shown to be computationally feasible, interesting and promising results have been obtained, e.g., for the Gaussian process (Chapter 4) model, even with relatively simple choices of hyperpriors: Heinonen et al. (2016), Shah et al. (2014) and Paper VII. An insight from these developments is perhaps that the introduction of a hyperprior  $p(\eta)$  may in some models open up for a significantly more flexible modeling process compared to directly choosing a prior  $p(\theta)$ .

## 2.6 Summary of the chapter

We have discussed three cornerstones of machine learning from a statistical perspective: data  $y$ , models  $p(y | \theta)$  and two approaches for learning (or, equivalently, inference). The data is given, whereas the model is chosen by us. The choice of model is important in that it heavily influences which results we obtain. With data and model in place, there are still different options for how to learn the parameters  $\theta$  from the data, either the point estimation or the Bayesian approach, or possibly something in between (regularization etc). We have, however, only discussed different high-level approaches for learning, and we return to some methods for performing the actual computations later on in Chapter 5.



“Models are to be used, not believed.”

Henri Theil

# 3

## State-space models

STATE-space models, or hidden Markov models, is a popular family of models. In this chapter, we introduce the general state-space model, and thereafter also introduce two important special cases, namely the linear and the jump-Markov linear state-space models. (In the next chapter, also a third special case is introduced, namely the Gaussian process state-space model.) We also devote a section to discuss learning for state-space models.

### 3.1 The general state-space model

At the core of the state-space model we have a Markov process  $\dots, x_{t-1}, x_t, x_{t+1}, \dots$ , which evolves as  $p(x_{t+1} | x_t) = f(x_{t+1} | x_t)$ , where  $f(\cdot | \cdot)$  is called the state transition density. We refer to  $x_t \in \mathbb{R}^{n_x}$  as the *state*, and  $t = 0, \dots, T$  is an index typically representing time in time-series data, but other interpretations are also possible.

The state  $x_t$  may represent the physical state of an object under study, such as the position, speed, heading and acceleration of a vehicle, but it can also be an abstract representation without any clear physical interpretation. The Markov property means that once  $x_t$  is known, the previous states  $\dots, x_{t-1}$  do not add any information about the later states  $x_{t+1}, \dots$ , i.e.,

$$p(x_{t+1} | \dots, x_{t-1}, x_t) = p(x_{t+1} | x_t). \quad (3.1)$$

This Markov assumption is the key for efficiency when learning state-space models.

The state-space model also includes the observation density  $g(\cdot | \cdot)$ , which models the relation between the state  $x_t$  and the observation, or output,  $y_t \in \mathbb{R}^{n_y}$ , as  $p(y_t | x_t) = g(y_t | x_t)$ . Note that the Markov property (3.1) does *not* necessarily hold for the observations  $\dots, y_{t-1}, y_t, y_{t+1}, \dots$ !

To summarize the state-space model, we write

$$p(x_{t+1} | x_t) = f(x_{t+1} | x_t), \quad (3.2a)$$

$$p(y_t | x_t) = g(y_t | x_t). \quad (3.2b)$$

For completeness, the model also has to include a density  $p(x_0)$  for the initial state  $x_0$ .

An alternative naming of (3.2) is a hidden Markov model, where ‘hidden Markov’ refers to the unobserved states  $x_t$  that obey the Markov assumption (3.1). The term is, however, also (and perhaps more often) used for models where  $x_t$  lives in a discrete space rather than in  $\mathbb{R}^{n_x}$ .

In the automatic control literature, state-space models are often used with the addition of an exogenous (and known) input signal  $u_t \in \mathbb{R}^v$ . Another flavor of (3.2) is the time-varying state-space model, where  $f$  and  $g$  (and possibly also  $n_x$ ) explicitly depends on  $t$ .

State-space models are typically used to model time-series data  $\{y_1, \dots, y_T\}$  which exhibits some dynamical behavior, i.e., there is a non-trivial correlation between different data points. In a common user case the data  $\{y_1, \dots, y_T\}$  is far from obeying the Markov assumption, but thanks to the state-space model a state sequence  $\{x_1, \dots, x_T\}$  can be (re)constructed. If the model describes the data well, the state sequence will follow the Markov assumption. The reasons for using a state-space model may, at least, be twofold:

- The states bear a physical meaning (e.g., the position and speed of a vehicle) which is of interest.
- In the interest of making predictions, the states  $x_t$  provide a compact summary of all relevant history: rather than storing and processing all data  $y_1, \dots, y_t$ , it suffices to consider  $x_t$  for predicting the future observations  $y_{t+1}, \dots$ , provided that the Markov assumption for the states  $x_t$  holds.

A relevant question is whether a state-space model, which accurately describes any data set recorded from the same process, always exists? The answer is no; several practically relevant counterexamples exist (e.g., Ljung and Glad 2004, Chapter 7) where the state-space model is insufficient. Nevertheless, the state-space model has proven a practically useful model for many cases.

## 3.2 Linear Gaussian state-space models

The perhaps most well-studied version of the state-space model is the *linear* state-space model with additive Gaussian noise,

$$x_{t+1} = Ax_t + Bu_t + w_t, \quad w_t \sim \mathcal{N}(0, Q), \quad (3.3a)$$

$$y_t = Cx_t + Du_t + e_t, \quad e_t \sim \mathcal{N}(0, R). \quad (3.3b)$$

Here,  $A, B, C, D, Q$  and  $R$  are matrices of appropriate sizes, and  $w_t$  and  $e_t$  are stochastic noise, i.i.d. with respect to time. In (3.3) we have deviated from the probabilistic notation, and also included an exogenous input signal  $u_t$ , in order to conform with the standard notation in the system identification literature.

Entire books (e.g., Kailath 1980; Rugh 1993) have been written on models of the type (3.3) and its almost equivalent alternative formulation as a transfer function. We make no attempt on covering that literature here.

The linear Gaussian state-space model (3.3) has the advantage that many learning problems can be carried out relatively easy, if not in closed form at least with relatively

efficient algorithms. The downside, however, is its limited expressiveness (even though it has turned out to be very useful, judging from its widespread use) compared to the much more general model (3.2).

A compromise between the expressiveness of the nonlinear state-space model and the analytical tractability of the linear Gaussian state-space model is to keep the linear state transition (i.e.,  $x_{t+1} = Ax_t + Bu_t + w_t$ ), but also append (3.3) with some nonlinear feature. Two such examples are the jump-Markov linear state-space models (which we discuss in the next section) and the Wiener and Hammerstein models.

### 3.3 Jump-Markov linear state-space models

To obtain an expressiveness beyond the linear state-space model (3.3), the *jump-Markov* linear state-space model augments (3.3) with another Markov process (in addition to  $x_t$ ), namely the mode sequence  $\dots, s_{t-1}, s_t, s_{t+1}, \dots$ . The sequence takes values on the finite discrete space  $\{1, 2, \dots, K\}$ , and is defined via its transitions probabilities

$$p(s_{t+1} | s_t) = \pi_{s_t, s_{t+1}}. \quad (3.4a)$$

One linear state-space model belongs to each mode (all with the same state dimensions  $n_x$ ), whose corresponding matrices we denote by a subscript. Conditioned on the mode sequence, the states evolve as (cf. (3.3))

$$x_{t+1} = A_{s_t} x_t + B_{s_t} u_t + w_t, \quad w_t \sim \mathcal{N}(0, Q_{s_t}), \quad (3.4b)$$

$$y_t = C_{s_t} x_t + D_{s_t} u_t + e_t, \quad e_t \sim \mathcal{N}(0, R_{s_t}). \quad (3.4c)$$

Clearly, (3.4) is a more general model than (3.3) (if  $k > 1$ ), but it is still just a special case of the general state-space model (3.2). Paper IV develops a particular inference algorithm tailored for models on the form (3.4).

### 3.4 Learning state-space models

Due to the Markov structure of the state-space model (3.2), most inference problems in state-space models take a particular form. We give an introduction here, and Paper I, IV, III, V and VI are all concerned with particular aspects of learning state-space models.

#### 3.4.1 Quantities to learn: states and model parameters

When we discussed learning in Chapter 2, we talked about parameters  $\theta$ , referring to some unknown numerical quantities in the model that remains to be determined using observed data  $\{y_1, \dots, y_T\}$  (and also inputs  $\{u_1, \dots, u_T\}$  if applicable). It has, however, not yet been said what  $\theta$  correspond to in the state-space model: By construction, the states  $x_t$  are not observed and might be of interest to learn, but there might also be unknown quantities in the model itself, i.e.,  $f(\cdot | \cdot)$  and  $g(\cdot | \cdot)$  might be parameterized by some unknown *model parameters*  $\vartheta$  as  $f_{\vartheta}(\cdot | \cdot)$  and  $g_{\vartheta}(\cdot | \cdot)$ .

There is no inherent difference between the states  $x_t$  and the model parameters  $\vartheta$  from a learning perspective: they are both unknown quantities in the state-space model. However, depending on the user's case, different settings are of interest:

- (i) The state-space model (i.e.,  $f(\cdot | \cdot)$  and  $g(\cdot | \cdot)$  in (3.2)) is completely known, and only the state sequence  $\{x_1, \dots, x_T\}$  remains to be determined. We refer to this problem as *state inference*, a problem typically appearing if the model is derived from first principles, implying that the states bear a physical meaning (e.g., the position and velocity of a vehicle).
- (ii) Only limited knowledge about the state-space model is present, and we have to infer a set of unknown model parameters  $\vartheta$  (the states are not available either<sup>1</sup>). We refer to this case as *model parameter learning*, typically occurring if the physical insight about the real process (from which the data is recorded) is limited.

It should be noted that while the model parameters  $\vartheta$  typically are of a rather low dimension (say<sup>2</sup>, 1-20), the entire state sequence  $\{x_1, \dots, x_T\}$  is of dimension  $T \cdot n_x$ , where  $T > 100\,000$  is not unrealistic. For this reason, the state inference and the model parameter learning algorithms have to be designed differently, in order to gain computationally feasible solutions.

The model parameter learning problem contains a spectrum of settings, ranging from learning a single parameter value to determining the entire functional forms of  $f(\cdot | \cdot)$  or  $g(\cdot | \cdot)$ . In this thesis, Paper IV, Paper III and Paper VI represent the former problem (in particular, inference of the numerical values in (3.4)), whereas paper I deals with the latter case where no parametric form of  $f(\cdot | \cdot)$  nor  $g(\cdot | \cdot)$  is known a priori. A very well studied case is inference of the matrices  $A, B, C, D, Q, R$  in (3.3), referred to as *linear system identification* (Ljung 1999; Söderström and Stoica 1989).

We assume the state dimension  $n_x$  is known. Learning  $n_x$  is another problem, not considered in this thesis.

### 3.4.2 A Bayesian approach or point estimates?

Given the two learning problems in the state-space model, state and model parameter inference respectively, we now turn to the next question: what inference paradigm to use, the Bayesian or the point estimation approach?

The learning approach for the model parameter may vary with the amount of data, properties of the model, intended use, etc., as discussed in Section 2.4. The point estimation approach has historically been favored (e.g., Ljung 1999; Söderström and Stoica 1989), but a discussion in favor of the Bayesian approach is given by Peterka (1981). If the dimension of  $\vartheta$  is low, a large amount of data is available ( $T$  is large), and  $\vartheta$  is identifiable (Söderström and Stoica 1989, Section 6.4), the maximum likelihood and the Bayesian solution can often be expected to provide similar results in practice (cf. Section 2.4). Also other point estimates than maximum likelihood are popular in the literature, such as the one minimizing the simulation error of the model.

<sup>1</sup>For this reason, the case (i) can be seen as a subproblem of (ii).

<sup>2</sup>We explore much larger cases in Paper I.

For the state inference problem (i.e., finding  $x_{1:T}$  when given  $y_{1:T}$  and  $\vartheta$ ), we may once again refer back to the discussion in Section 2.4, and note that the problem is of the peculiar form that with more data (i.e., larger  $T$ ), the dimension of the state sequence  $\{x_1, \dots, x_T\}$  also grows. Thus, the argument from Section 2.4 about concentration of the posterior towards a point as the data record grows is not applicable<sup>3</sup>, and we should for this reason be cautious about applying a point estimation approach: we may ignore important uncertainty information if we do so. Perhaps for this reason, the state inference problem is almost exclusively approached by the Bayesian paradigm in the literature, which we will review now.

### Bayesian filtering

To alleviate the notation, we use the shorthand symbol  $x_{1:t} \triangleq \{x_1, \dots, x_t\}$ , and similar for  $y_{1:t}$ . The state inference in the Bayesian paradigm (2.4) can be written as

$$p(x_{1:T} | y_{1:T}) = \frac{p(y_{1:T} | x_{1:T})p(x_{1:T})}{p(y_{1:T})}. \quad (3.5)$$

We may interpret this as (3.2a) providing the prior for the states  $p(x_{1:T}) = \prod_{t=1}^{T-1} f(x_{t+1} | x_t)$ , and (3.2b) giving the model<sup>4</sup> for the data as  $p(y_{1:T} | x_{1:T}) = \prod_{t=1}^T g(y_t | x_t)$ . In a computational perspective, however, (3.5) is of very limited use. Instead the recursion (see, e.g., Särkkä 2013)

$$p(x_t | y_{1:t}) = \frac{1}{p(y_t | y_{1:t-1})} g(y_t | x_t) \int f(x_t | x_{t-1}) p(x_{t-1} | y_{1:t-1}) dx_{t-1} \quad (3.6)$$

has proven useful for computing the (marginal) posterior distributions  $p(x_t | y_{1:t})$ . The denominator in (3.6) only serves the purpose of normalization (and may in some computational schemes be omitted), and the remaining quantities are known. The Kalman filter (below) as well as the particle filter (Chapter 5) are direct implementations of (3.6). We refer to (3.6) as the *Bayesian filtering recursion*, a name commonly used<sup>5</sup>. The term *filtering* refers to the distributions  $p(x_1 | y_1), p(x_2 | y_{1:2}), \dots, p(x_T | y_{1:T})$ , as opposed to the (marginal) *smoothing* distributions  $p(x_1 | y_{1:T}), p(x_2 | y_{1:T}), \dots, p(x_T | y_{1:T})$  (note the different conditioning). For computing the smoothing distributions, there is a variety of popular recursions used, for which we refer to the literature, e.g., Lindsten and Schön 2013; Särkkä 2013; Svensson, Schön, et al. 2015.

### The Kalman filter

Without doubt, the most popular implementation of the Bayesian filtering recursion is the Kalman filter, named after Rudolf Kálmán (1960). The Kalman filter is nothing but (3.6) written down for the special case of the linear Gaussian state-space

<sup>3</sup>From a time-series perspective, we may use the argument that a data point  $y_t$  does not necessarily provide more information about the state  $x_\tau$  if  $t \gg \tau$  or  $t \ll \tau$ .

<sup>4</sup>For consistency, we should thus refer to (3.2b) as the model and (3.2a) as the prior. Maximum likelihood estimation of some unknown parameters  $\vartheta$  in  $f(\cdot | \cdot)$  should then be termed empirical Bayes. Such a terminology would perhaps provide some additional insight, but would probably cause more confusion than clarity in the end.

<sup>5</sup>The Bayesian filtering recursion is commonly also named ‘optimal’ filtering, where ‘optimal’ only reflects that it is the Bayesian solution.

model<sup>6</sup> (3.3). We refer to, e.g., Peterka (1981) and Schön and Lindsten (2011) for the derivation and the final equations.

The Kalman filter is often applied also to more general state-space models not exactly on the linear Gaussian form (3.3), due to its relative simplicity. Often modifications are made to approximately handle more general formulations than (3.3), e.g., the extended Kalman filter, the unscented Kalman filter, etc. (Särkkä 2013).

The likelihood for the state space model

We also introduce the probability density for  $y_{1:T}$  given  $\vartheta$ , i.e.,  $p(y_{1:T} | \vartheta)$ . When we in Chapter 5 discuss numerical methods for model parameter learning, this expression is at the center of attention:

$$p(y_{1:T} | \vartheta) = \prod_{t=1}^T p(y_t | y_{1:t-1}, \vartheta) = \prod_{t=1}^T \int p(y_t | x_{t-1}, \vartheta) p(x_{t-1} | y_{1:t-1}, \vartheta) dx_{t-1}, \quad (3.7)$$

where we have factorized the expression in such a way that we can see that finding  $p(x_t | y_{1:t})$  might help in computing  $p(y_{1:T} | \vartheta)$ . Thus, solving the state inference problem in the Bayesian paradigm, i.e., finding  $p(x_t | y_{1:t})$ , may help also when a maximum likelihood point estimate of  $\vartheta$  is sought!

### 3.5 Summary of the chapter

This chapter has introduced the general state-space model, as well as some special cases of it; the linear state-space model and the jump-Markov linear state-space model. We have also introduced and discussed two different learning problems for state-space models, the state inference problem and the model parameter learning problem.

---

<sup>6</sup>The Kalman filter can alternatively also be derived as the optimal (in mean-square-error sense) linear estimator for a more wide class than (3.3).

*“I think it is much more interesting to live with uncertainty than to live with answers that might be wrong.”*

Richard Feynman

# 4

## Gaussian processes

THE Gaussian process (GP) defines a probability distribution over functions  $f$ , and is commonly used as a model for functional relationships between variables. The GP is tightly connected with the Bayesian paradigm, and conditioning on data  $y$ , i.e., updating the prior  $p(f)$  into the posterior  $p(f | y)$ , will be our most common usage of the GP model.

The GP is a so-called *nonparametric* model, in that it does not rely on a finite set of parameters  $\theta$ . A parametric model for  $f$  (e.g., a polynomial of finite order) involves a set of parameters  $\theta$  acting as a ‘mid-layer’ between the data and the posterior over  $f$ . In a parametric model, finding the posterior  $p(f | y)$  amounts to first find  $p(\theta | y)$  and then compute  $p(f | y) = \int p(f | \theta)p(\theta | y)d\theta$  (cf. (2.6)). In a nonparametric model, however, the distribution  $p(f | y)$  is computed directly without (explicitly) involving any parameters  $\theta$ . One may alternatively understand this as the data (in a nonparametric model) takes the role of the parameters (in a parametric model). The main advantage of a nonparametric model is perhaps that there is no upper limit on ‘how much information the model can contain’, in contrast to a parametric model.

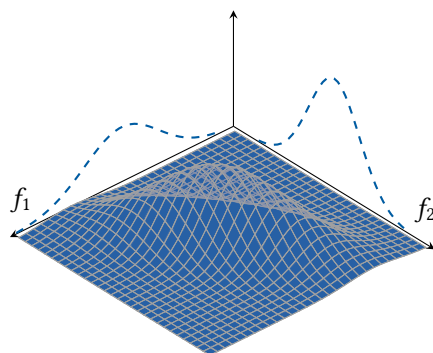
### 4.1 Introducing the Gaussian process

The nonparametric GP can be understood as a limit of the  $k$ -dimensional multivariate Gaussian distribution as  $k$  tends to infinity. We will try to follow the intuition behind this limit, in order to develop an understanding for the connections between the Gaussian distribution and the GP. All technical details can be found in the literature (MacKay 1998; Rasmussen and Williams 2006).

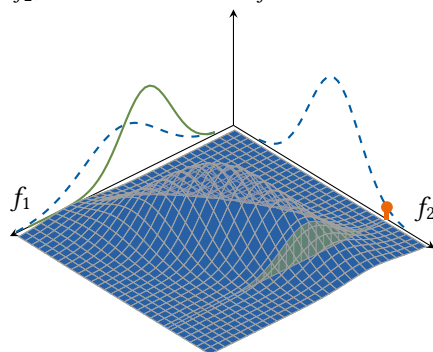
The density for the  $k$ -dimensional multivariate Gaussian distribution is

$$\mathcal{N}(\bar{f}; \mu, \Sigma) = (2\pi)^{-\frac{k}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\bar{f} - \mu)^\top \Sigma^{-1}(\bar{f} - \mu)\right), \quad (4.1)$$

where  $\bar{f} = [f_1 \cdots f_k]^\top$  is a  $k$ -dimensional vector with random scalar elements  $f_1, \dots, f_k$ ,  $\mu \in \mathbb{R}^k$  is the mean, and  $\Sigma \in \mathbb{R}^{k \times k}$  is the (positive semidefinite) covariance



(a) A two-dimensional Gaussian distribution for the random variables  $f_1$  and  $f_2$ , with a blue surface plot for the density, and the marginal distribution for each component sketched using dashed blue lines along each axis. Note that the marginal distributions do not contain all information about the distribution of  $f_1$  and  $f_2$ , since the covariance information is lacking in that representation.



(b) The conditional distribution of  $f_1$  (green line), when  $f_2$  is observed (orange dot). The conditional distribution of  $f_1$  is given by (4.3), which (apart from a normalizing constant) in this graphical representation also is the green 'slice' of the joint distribution (blue surface). The marginals of the joint distribution from Figure 4.1a are kept for reference (blue dashed lines).

**Figure 4.1:** A two-dimensional multivariate Gaussian distribution for  $f_1$  and  $f_2$  in (a), and the conditional distribution for  $f_1$ , when a particular value of  $f_2$  is observed, in (b).

matrix, which means that it has  $k + \frac{k(k+1)}{2}$  parameters. In the limit  $k \rightarrow \infty$ , the number of parameters tends to infinity, which can be intuitively understood as the transition from the parametric Gaussian distribution to the nonparametric GP. (The technical challenge in this limit, which we will not fully address here, is the generalization from countable to measurable infinity.)

Considering the Gaussian distribution (4.1), we can partition  $\bar{f}$  into  $\begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix}$ , and  $\mu$  and  $\Sigma$  similarly, and then write

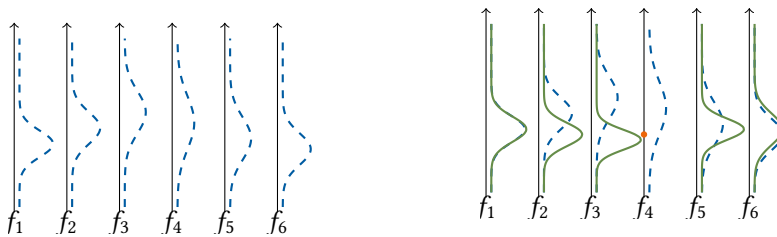
$$p\left(\begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} \bar{f}_1 \\ \bar{f}_2 \end{bmatrix}; \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right). \quad (4.2)$$



(a) The marginal distributions for  $f_1$  and  $f_2$  from Figure 4.1a.

(b) The distribution for  $f_1$  (green line) when  $f_2$  is observed (orange dot), as in Figure 4.1b.

**Figure 4.2:** The marginals of the distributions in Figure 4.1, here plotted slightly differently. Note that this more compact plot comes with the cost of missing the information about the covariance between  $f_1$  and  $f_2$ .



(a) A 6-dimensional Gaussian distribution, plotted in the same way as Figure 4.2a, i.e., only its marginals are illustrated.

(b) The conditional distribution  $f_1, f_2, f_3, f_5$  and  $f_6$  when  $f_4$  is observed (orange dot), illustrated by its marginals (green lines), cf Figure 4.2b.

**Figure 4.3:** A 6-dimensional Gaussian distribution, illustrated in the same fashion as Figure 4.2.

If some elements of  $\bar{f}$ , let us say the ones in  $\bar{f}_2$ , are observed, the conditional distribution for  $\bar{f}_1$  given the observation of  $\bar{f}_2$  is

$$p(\bar{f}_1 | \bar{f}_2) = \mathcal{N}(\bar{f}_1; \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(\bar{f}_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}). \quad (4.3)$$

The conditional distribution is nothing but another Gaussian distribution with closed-form expressions for the mean and covariance. This is particularly useful.

Figure 4.1 shows a 2-dimensional example where a multivariate Gaussian distribution is conditioned on data. In Figure 4.2, we have plotted the marginal distributions from Figure 4.1, to prepare for the generalization to GP. It is also straightforward to plot a 6-dimensional multivariate Gaussian distribution by its margins, akin to Figure 4.2, as we do in Figure 4.3. Bear in mind that to fully illustrate the joint distribution for  $f_1, \dots, f_6$ , a 6-dimensional surface plot would be needed, whereas Figure 4.3a only contains the marginal distributions for each component. As earlier, we may also condition the 6-dimensional distribution underlying Figure 4.3a on an observation of, e.g.,  $f_4$ . Once again, the conditional distribution is another Gaussian distribution, and the marginals of the 5-dimensional distribution are plotted in Figure 4.3b.

In Figure 4.2 and 4.3, we had a distribution over a finite set of discrete points. If we were to study a phenomenon taking values on a finite set of discrete points, like  $\{1, 2, 3, 4, 5, 6\}$  in Figure 4.3, we could use this as a probabilistic model. However, our aim is the GP, a probabilistic model for functions on a *continuous* space.

The extension of the Gaussian distribution (defined on a finite set) to the GP (defined on a continuous space) is achieved by replacing the index set  $\{1, 2, 3, 4, 5, 6\}$  in Figure 4.3 by a variable  $x$  taking values on the continuous real line. In the Gaussian distribution,  $\mu$  is a vector with  $k$  components (e.g.,  $\mu \in \mathbb{R}^2$  in Figure 4.2, and  $\mu \in \mathbb{R}^6$  in Figure 4.3), and similarly for the covariance matrices. In the GP, we replace  $\mu$  by a mean *function*  $\mu(x)$  parameterized by  $x$ , and the covariance matrix  $\Sigma$  by a covariance *function*  $\kappa(x, x')$  parameterized by  $x$  and  $x'$ . The GP is then defined as:

**Definition** (the Gaussian process). *Let  $\{x_1, \dots, x_n\}$  be any finite set of points for which  $\mu(x_i)$  and  $\kappa(x_i, x_j)$  are defined. Then,*

$$p \left( \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \right) = \mathcal{N} \left( \begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix}; \begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} \kappa(x_1, x_1) & \cdots & \kappa(x_1, x_n) \\ \vdots & & \vdots \\ \kappa(x_n, x_1) & \cdots & \kappa(x_n, x_n) \end{bmatrix} \right). \quad (4.4)$$

That is, for any choice of  $\{x_1, \dots, x_n\}$ , we have a multivariate Gaussian distribution, just like the one in Figure 4.3. Since  $\{x_1, \dots, x_n\}$  can be chosen arbitrarily on the continuous line, this implicitly defines a distribution for *all* points on that line. Of course, for this definition to make sense,  $\kappa(\cdot, \cdot)$  has to be such that a positive semidefinite covariance matrix is obtained for any choice of  $\{x_1, \dots, x_n\}$ .

We will use the notation

$$f \sim \mathcal{GP}(\mu(\cdot), \kappa(\cdot, \cdot)) \quad (4.5)$$

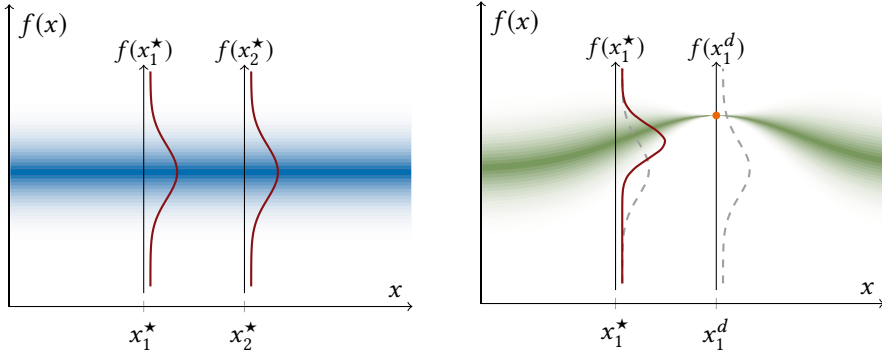
to express that the function  $f$  is distributed according to a GP with mean function  $\mu(\cdot)$  and covariance function  $\kappa(\cdot, \cdot)$ . If we want to plot the GP, which we do in Figure 4.4, we may choose  $\{x_1, \dots, x_n\}$  to correspond to the pixels on the screen or the printer dots on the paper, so that it appears as a continuous line to the eye (despite that we actually can access the distribution only in a finite, however arbitrary, set of points).

The perhaps most interesting procedure is the calculation of the conditional distribution given some observations  $\{f(x_1^d), \dots, f(x_m^d)\}$ , the GP counterpart to Figure 4.1b, 4.2b and 4.3b. We start by introducing the following more compact notation,

$$x^* \triangleq \begin{bmatrix} x_1^* \\ \vdots \\ x_n^* \end{bmatrix}, \quad K^{**} \triangleq \begin{bmatrix} \kappa(x_1^*, x_1^*) & \cdots & \kappa(x_1^*, x_n^*) \\ \vdots & & \vdots \\ \kappa(x_n^*, x_1^*) & \cdots & \kappa(x_n^*, x_n^*) \end{bmatrix}, \quad (4.6a)$$

$$x^d \triangleq \begin{bmatrix} x_1^d \\ \vdots \\ x_m^d \end{bmatrix}, \quad K^{dd} \triangleq \begin{bmatrix} \kappa(x_1^d, x_1^d) & \cdots & \kappa(x_1^d, x_m^d) \\ \vdots & & \vdots \\ \kappa(x_m^d, x_1^d) & \cdots & \kappa(x_m^d, x_m^d) \end{bmatrix}, \quad (4.6b)$$

$$K^{*d} \triangleq \begin{bmatrix} \kappa(x_1^*, x_1^d) & \cdots & \kappa(x_1^*, x_m^d) \\ \vdots & & \vdots \\ \kappa(x_n^*, x_1^d) & \cdots & \kappa(x_n^*, x_m^d) \end{bmatrix} = (K^{d*})^\top. \quad (4.6c)$$



(a) A GP defined on the real line parameterized by  $x$ , not conditioned on any observations. The intensity of the blue color is proportional to the (marginal) density, and the marginal distributions for some  $x_1^*$  and  $x_2^*$  are pictured in red. Akin to Figure 4.3, we only plot the marginal distribution for each  $x^*$ , but the GP defines a full joint distribution for all points on the  $x$ -axis, even though it is hard to illustrate.

(b) The conditional GP distribution given the observation of  $f(x_1^d)$  in the point  $x_1^d$  corresponding to  $x_2^*$  in (a). The prior distribution from Figure (a) is dashed gray. Note how the conditional distribution adjusts to the observation, both in terms of mean (closer to the observation) and (marginal) variance (smaller in the proximity of the observation, but it remains unchanged in areas distant from it).

**Figure 4.4:** A GP. Figure (a) shows the prior distribution (shaded blue), whereas (b) shows the posterior distribution (shaded green) after conditioning on one observation (orange dot).

We can use this notation and the definition to write the joint distribution between the values  $f(x^d)$  in the points  $x^d$ , and the value  $f(x^*)$  in some other points  $x^*$  as

$$p\left(\begin{bmatrix} f(x^*) \\ f(x^d) \end{bmatrix}\right) = \mathcal{N}\left(\begin{bmatrix} f(x^*) \\ f(x^d) \end{bmatrix}; \begin{bmatrix} \mu(x^*) \\ \mu(x^d) \end{bmatrix}, \begin{bmatrix} K^{**} & K^{*d} \\ K^{d*} & K^{dd} \end{bmatrix}\right). \quad (4.7)$$

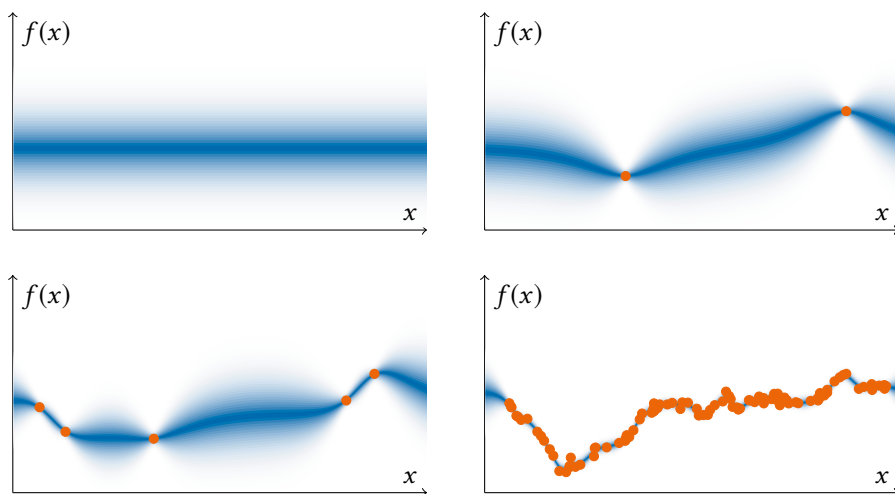
Now, as we have observed  $f(x^d)$ , we can express the posterior distribution for  $f(x^*)$  conditional on the observations as

$$p\left(f(x^*) \mid f(x^d)\right) = \mathcal{N}\left(f(x^*); \mu(x^*) + K^{*d}(K^{dd})^{-1}\left(f(x^d) - \mu(x^d)\right), K^{**} - K^{*d}(K^{dd})^{-1}K^{d*}\right), \quad (4.8)$$

i.e., nothing but another multivariate Gaussian distribution for any finite set  $x^*$ . We illustrate this by Figure 4.4.

The GP, and in particular (4.8), now provides a way to probabilistically inter- and extrapolate observations under the assumption that the observations are drawn from a Gaussian process. In most practical cases this assumption is most likely not true, but it has nevertheless proven to be a useful model. The typical use of the GP as a modeling tool is illustrated in Figure 4.5.

The Gaussian process can alternatively also be introduced as a nonlinear and nonparametric generalization of linear regression, which perhaps is more standard in the literature (cf. Bishop 2006, Section 6.4; MacKay 1998; Rasmussen and Williams 2006).



**Figure 4.5:** The GP as a modeling tool: the conditional distribution (shaded blue) for  $f(x)$  after 0, 2, 5 and 100 observations (orange dots) of  $y = f(x) + \text{noise}$ . (We have now left our earlier convention of plotting the posterior distribution after conditioning on data in green, since the prior–posterior notion becomes entangled when we sequentially condition on more and more data.)

## 4.2 Noise density, mean and covariance functions

We have in the previous section assumed the existence of a mean  $\mu(\cdot)$  and covariance function<sup>1</sup>  $\kappa(\cdot, \cdot)$ . When using the GP to model observed data, these functions somehow have to be chosen by the user. If there is detailed domain knowledge present, it can be incorporated into the covariance function: one such example is Solin, Kok, et al. (2018), where the magnetic field is modeled using a GP covariance function tailored to obey Maxwell’s equations (see also Jidling et al. 2018 for general constructions along these lines). In many situations, however, such detailed knowledge is not present, and one has to make a less informed choice of covariance function. Two common choices are the exponentiated quadratic and the Matérn class<sup>2</sup> of covariance functions; their expressions are found in Table 4.1, and their properties have been widely discussed in the literature (e.g., Rasmussen and Williams 2006, Section 4.2) and will not be repeated here. There are also ways to combine different covariance functions into new ones, creating, e.g., periodic covariance functions (Rasmussen and Williams 2006, Section 4.2.4; Duvenaud, Lloyd, et al. 2013). A standard terminology is that if  $\kappa(x, x')$  is a function of only  $x - x'$ , it is referred to as *stationary*, and if it is only a function of  $\|x - x'\|$ , it is called *isotropic*.

A common choice for the mean function is  $\mu(x) = 0$ , which at a first glance may seem very restrictive. However, already by inspection of (4.8) or Figure 4.4b, it is clear that the posterior mean (i.e., after conditioning on observations) may be non-zero even though the prior is 0. In fact,  $\mu(x) = 0$  appears to work well in many situations.

<sup>1</sup>The covariance function is often referred to as a *kernel* in the literature. We refrain from that terminology here to avoid confusion with the MCMC kernel in the next chapter.

<sup>2</sup>Named after the Swedish statistician Bertil Matérn (1960).

Function	Meaning	Limitations	Examples
Mean $\mu(x)$	Prior assumption about mean	-	$C$ (constant) $a \cdot x$ (linear)
Covariance $\kappa(x, x')$	Assumption on how correlated two $x$ -values are	Must be positive semidefinite	$\exp\left(-\frac{\ x-x'\ ^2}{2\ell^2}\right)$ (exponentiated quadratic) $\frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\ x-x'\ }{\ell}\right)^\nu K_\nu\left(\frac{\sqrt{2\nu}\ x-x'\ }{\ell}\right)$ (Matérn class)
Observation noise	Assumption about noise level in observed data	Analytically tractable only if Gaussian distribution	$\varepsilon = 0$ (noiseless) $\mathcal{N}(\varepsilon; 0, \sigma_n^2)$ (Gaussian distribution)

**Table 4.1:** A summary and some examples of functions involved in the GP model.

In addition to a mean and covariance function, also a third function can be introduced: if  $f(x^d)$  is not observed directly, but corrupted by some additive noise  $\varepsilon$ , as  $y^d = f(x^d) + \varepsilon$ , the distribution for  $\varepsilon$  also has to be modeled. In effect, the noise model determines how much the observed data should be ‘trusted’. If the noise model is chosen as a Gaussian distribution, it can be incorporated into the covariance function, and (4.8) is still valid. Other alternatives are possible, but gives no closed-form expressions à la (4.8). All functions discussed here, including some typical examples, are summarized in Table 4.1.

## 4.3 Hyperparameter learning

Most mean functions, covariance functions, and noise distributions contain some parameters, such as the length scale parameter  $\ell$  in the exponentiated quadratic covariance function, or the noise variance  $\sigma_n^2$  in the Gaussian noise model. We will refer to these as hyperparameters, denoted by  $\eta$ . The hyperparameters are often interpretable (such as length scale or noise level, Rasmussen and Williams 2006, Section 2.3), but due to ignorance (such as limited physical insight) when using the GP as a model, the hyperparameters might effectively be unknown.

As discussed in the learning chapter in Section 2.5.3, there are two common alternatives for how to learn unknown hyperparameters: empirical Bayes or hyperpriors.

### 4.3.1 Empirical Bayes: Finding a point estimate $\hat{\eta}$

The empirical Bayes approach can be applied to find a point estimate  $\hat{\eta}$  of  $\eta$ , by maximizing the marginal likelihood<sup>3</sup>

$$p(y^d | \eta) = \mathcal{N}\left(y^d; \mu_\eta(x^d), K_\eta^{dd}\right), \quad (4.9)$$

<sup>3</sup>The convention to name (4.9) marginal likelihood is because of the following: In a parametric model, the likelihood is  $p(y | \theta)$ , whereas  $p(y | \eta)$  is its marginal with respect to  $\theta$ :  $p(y | \eta) = \int p(y | \theta)p(\theta | \eta)d\theta$ .

where we have added the subscript  $\eta$  to stress the dependence on the hyperparameters. Note that (4.9) is nothing but a multivariate Gaussian distribution. Due to the way  $\eta$  enters into the problem, this is typically a highly non-convex problem, calling for numerical optimization tools. The major benefit with the point estimate is indeed that a single numerical value  $\widehat{\eta}$  is obtained, which is easy to use in, e.g., prediction

$$p(y^\star | y^d, \widehat{\eta}) = \mathcal{N}\left(y^\star; \mu_{\widehat{\eta}}(x^\star) + K_{\widehat{\eta}}^{\star d} \left(K_{\widehat{\eta}}^{dd}\right)^{-1} \left(y^d - \mu_{\widehat{\eta}}(x^d)\right), K_{\widehat{\eta}}^{\star\star} - K_{\widehat{\eta}}^{\star d} \left(K_{\widehat{\eta}}^{dd}\right)^{-1} K_{\widehat{\eta}}^{d\star}\right), \quad (4.10)$$

a lengthy but computationally tractable expression. That a single numerical value for the hyperparameters is chosen is however also a major drawback of the approach. In many cases the ‘landscape’ of (4.9) is widespread and multimodal, which makes the optimization very hard, and the global optimum sensitive to small changes in data  $y^d$  as well as initialization of the optimization procedure. A further discussion with examples is found in Paper VII.

### 4.3.2 Hyperpriors: Marginalizing out $\eta$

The alternative approach to a point estimate is the Bayesian approach with hyperpriors. This approach amounts to inferring the posterior distribution  $p(\eta | y^d) \propto p(y^d | \eta)p(\eta)$ , and use this posterior distribution rather than a point estimate in subsequent tasks, such as prediction

$$p(y^\star | y^d) = \int \mathcal{N}\left(y^\star; \mu_\eta(x^\star) + K_\eta^{\star d} \left(K_\eta^{dd}\right)^{-1} \left(y^d - \mu_\eta(x^d)\right), K_\eta^{\star\star} - K_\eta^{\star d} \left(K_\eta^{dd}\right)^{-1} K_\eta^{d\star}\right) p(\eta | y^d) d\eta. \quad (4.11)$$

Because of the integral over  $\eta$  in (4.11), we will also refer to this approach as *marginalization*: the integrand of (4.11) is  $p(y^\star, \eta | y^d)$ , and the integral computes the marginal distribution  $p(y^\star | y^d)$ . However, the marginalization (4.11) is in general not analytically tractable, in contrast to the prediction with a point estimate (4.10). Typically not even the posterior distribution  $p(\eta | y^d)$  itself is tractable, which perhaps is the major drawback of this approach.

A numerical solution for this is to draw Monte Carlo samples from  $p(\eta | y^d)$ , and then approximate the integral in (4.11) with a sum over these samples. One method for acquiring such samples is presented in Paper VII. Other alternatives include variational inference, see, e.g., Titsias and Lázaro-Gredilla (2014).

## 4.4 Computational aspects

The computational load of (4.8), the main workhorse of the GP model, is dominated by the inversion of the matrix  $K^{dd}$ , an operation essentially of complexity  $\mathcal{O}(m^3)$ . Thus, the computational complexity of the GP grows with data in a rather unfavorable way, which may prohibit its use in many applications. A rich literature on approximations is therefore available, e.g., Rasmussen and Williams (2006, Chapter 8); Snelson (2007) and Chalupka et al. (2013). In particular, we will make use of an approximation proposed by Solin and Särkkä (2014) in Paper I.

Essentially, most approximative methods amount to creating a lower-dimensional representation of the data. The lower-dimensional representation resembles a parameter  $\theta$ . A naïve but illustrative such approximation method is the ‘subset of data’ method, where only a subset of all data  $y$  (chosen either randomly or in a more systematic way) is considered. If the subset is of size  $p < m$ , the computational load of the GP reduces from  $\mathcal{O}(m^3)$  to  $\mathcal{O}(p^3)$ .

## 4.5 Two remarks

The GP provides a widely used and perhaps intuitively appealing model for nonlinear functions. In this section, we will make two remarks that are important to keep in mind when working with GPs for modeling.

### 4.5.1 A posterior variance independent of observed function values?

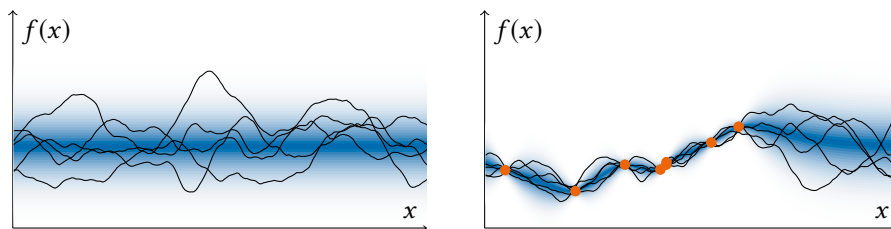
The Gaussian process is a flexible model, as seen in, e.g., Figure 4.5. However, in its use with fixed hyperparameters  $\eta_0$ , it has the peculiarity that its variance is independent of the actually observed function values  $f(x^d)$ , for instance in the predictive variance<sup>4</sup> in (4.8). This is nothing but a direct consequence of the prior assumption that the data was generated by a GP with the specified mean and covariance function with hyperparameters  $\eta_0$ . However, if the hyperparameters are not considered fixed, but inferred from data (either using empirical Bayes or by assuming hyperpriors and marginalizing them), the predictive variance depends on the observed function values, somewhat indirectly via the hyperparameter learning procedure.

### 4.5.2 What is a typical sample of a GP?

The mean of the GP can be used to characterize the distribution. It is, however, important to remember that the mean is a rather atypical sample of the GP, just as 0 is a very particular sample of a  $\mathcal{N}(0, 1)$  distribution. Furthermore, the GP also encodes a smoothness assumption, which is not very clear in the plotting style of Figure 4.5: in Figure 4.6, 5 samples are drawn from these distributions, where it is clear that also the correlation (along the  $x$ -axis) contains important information in the GP distribution as well. This is essentially the same point as we discussed when we considered the 2-dimensional Gaussian distribution in Figure 4.1, and only plotted its marginal distributions in 4.2 even though a full joint distribution is specified by the model.

---

<sup>4</sup>This point has its counterpart in the Kalman filter, Section 3.4.2, where also the predictive covariance is independent of the observed measurements. It is essentially the very same phenomenon, since the Kalman filter can be interpreted as a Gaussian process (Solin and Särkkä 2014).



**Figure 4.6:** Five samples of the GP from Figure 4.5 (with a Matérn  $\nu = 3/2$  covariance function). Note, in particular, that the samples are more wiggly than the mean function: a reminder that the blue shades do not contain all information, but is only the marginal distribution for each  $x$  (cf. Figure 4.1 and 4.2).

## 4.6 Gaussian-process state-space models

A combination of the state-space model and the GP is the relatively recent GP state-space model,

$$p(x_{t+1} | x_t) = \mathcal{N}(x_{t+1}; f(x_t), Q), \quad f \sim \mathcal{GP}(\mu_f(\cdot), \kappa_f(\cdot, \cdot)), \quad (4.12a)$$

$$p(y_t | x_t) = \mathcal{N}(y_t; g(x_t), R), \quad g \sim \mathcal{GP}(\mu_g(\cdot), \kappa_g(\cdot, \cdot)). \quad (4.12b)$$

The somewhat cumbersome notation should simply be read as ‘ $x_{t+1}$  equals a GP of  $x_t$  plus Gaussian noise’, and similar for  $y_t$ . The promising feature of the model is that it combines the nonparametric flexibility of a GP with the dynamical nature of the state-space model, allowing for complex and highly nonlinear dynamical phenomena to be described. Currently the best overview of the GP state-space model is probably found in the thesis by Frigola-Alcade (2015).

Due to the somewhat entangled use of the GP in (4.12a), where the output of the GP,  $x_{t+1}$ , is the input at the next time step, the inference problem becomes relatively hard. Frigola, Lindsten, et al. (2013) proposed a conceptually interesting but computationally brutal solution, and the subsequent Frigola, Y. Chen, et al. (2014) and Paper I (and in particular, its predecessor Svensson, Solin, et al. 2016) present further developments in different directions; For a numerical example which took Frigola, Lindsten, et al. (2013) about 10 hours of computational time, Svensson, Solin, et al. 2016 and Paper Frigola, Y. Chen, et al. (2014) only requires a few minutes.

## 4.7 Summary of the chapter

This chapter has introduced the GP as a generalization of the multivariate Gaussian distribution. A crucial aspect is that some important expressions are available in closed form, such as (4.10). The use of the GP in machine learning is as a model for (nonlinear) functions  $f$ , of which we only have observed the values in a few points (cf. Figure 4.5). It can also be combined with the state-space model into the GP state-space model (4.12).



*“Expose yourself to as much randomness as possible.”*

Ben Casnocha

# 5

## Monte Carlo methods for machine learning

**M**ONTE Carlo methods are a class of numerical methods named after the casino in the capital of Monaco (Figure 5.1). They originated in physics research with disputable purposes during the first half of the 20th century. An accessible introduction from that era, still well worth reading, is ‘The Monte Carlo method’ by Metropolis and Ulam (1949). Today, Monte Carlo methods are established tools within many different scientific fields, in particular in machine learning and some related areas.

Monte Carlo methods are useful when the mathematical computations are not analytically tractable, meaning, e.g., that an integral lacks a closed-form solution. There are also other alternatives, such as the variational approach (see Blei et al. 2016 for an overview). The idea in the variational approach is to impose additional assumptions until the modified problem becomes tractable. This thesis, however, focus on the Monte Carlo approach.

We give in this chapter an overview and introduction to sequential Monte Carlo (SMC) and Markov chain Monte Carlo (MCMC) in general, as well as their application for learning in state-space models.

### 5.1 The Monte Carlo idea

Consider a probability density  $\pi(\cdot)$  over the space of a parameter  $\theta$ , that is defined in such a way that the analysis of interest (e.g., computing the variance of  $\theta$ ) is not analytically tractable. The Monte Carlo idea is to approximately represent  $\pi$  by random samples (an empirical measure). Those random samples should be generated such that their properties resemble the properties of the distribution  $\pi$ . The samples are nothing but numerical values stored in a computer, and it is (hopefully) easier to analyze those samples than analyzing  $\pi$  directly.



**Figure 5.1:** Casino de Monte-Carlo in Monaco. A place of gambling and broken dreams, and moreover the source of the name ‘Monte Carlo method’. Photo: Andreas Svensson.

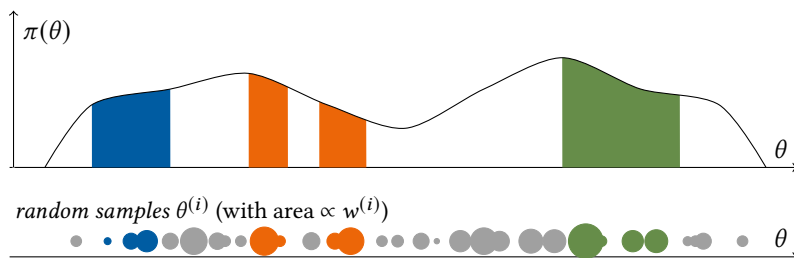
Formally, we introduce the notation of  $N$  weighted<sup>1</sup> samples  $\{\theta^{(i)}, w^{(i)}\}_{i=1}^N$ . This collection of weighted samples is a *Monte Carlo* (or *particle*) *approximation* of the density  $\pi$  if it holds that the empirical measure is ‘close’ to  $\pi$ , by which we mean

$$\frac{1}{\sum_{j=1}^N w^{(j)}} \sum_{i=1}^N w^{(i)} \mathbb{I}_A(\theta_i) \approx \int_A \pi(\theta) d\theta \tag{5.1}$$

for every measurable set  $A$ , with equality almost surely in the limit as  $N \rightarrow \infty$ . This is illustrated in Figure 5.2. If it is possible to draw samples from  $\pi$  directly, one may simply draw  $N$  such samples and set all weights to 1. If samples cannot be drawn from  $\pi$  directly, there are alternatives, of which we will review some.

For some methods, (5.1) does not only hold in the limit as  $N \rightarrow \infty$ , but also when taking the expectation over different realization of the Monte Carlo method itself as  $\mathbb{E} \left[ \frac{1}{\sum_{j=1}^N w^{(j)}} \sum_{i=1}^N w^{(i)} \mathbb{I}_A(\theta_i) \right] = \int_A \pi(\theta) d\theta$  for a fixed  $N$ . That is a stronger property, which holds for, e.g., rejection sampling but not  $p(x_t | y_{1:t}, \vartheta)$  in a particle filter.

<sup>1</sup>Note that we use non-normalized weights throughout this chapter.



**Figure 5.2:** The Monte Carlo idea: A probability density  $\pi(\theta)$  at the top, and weighted random samples of that distribution below (the area of each sample is proportional to its weight). Each color is a choice of  $A$  in (5.1), so we expect each colored area in the upper part of the figure (i.e.,  $\int_A \pi(\theta) d\theta$ ) to be roughly proportional to the area of its corresponding samples (i.e.,  $\sum_{i=1}^N w^{(i)} \mathbb{I}_A(\theta_i)$ ).

**Algorithm 1:** Bootstrap particle filter

---

**Input:** State space model  $f(\cdot | \cdot)$ ,  $g(\cdot | \cdot)$ ,  $p(x_0)$ , and data  $y_{1:T}$ .

**Output:** Weighted samples  $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_x}$  from  $p(x_t | y_{1:t}, \vartheta)$  for  $t = 1, \dots, T$ .

- 1 Draw  $x_0^{(i)} \sim p(x_0)$  and set  $w_0^{(i)} = 1$
- 2 **for**  $t = 1$  **to**  $T$  **do**
- 3     Draw  $a_t^{(i)}$  with  $\mathbb{P}(a_t^{(i)} = j) \propto w_{t-1}^{(j)}$      *resampling,  $\{x_{t-1}^{a_t^{(i)}}, 1\} \approx p(x_{t-1} | y_{1:t-1}, \vartheta)$*
- 4     Draw  $x_t^{(i)}$  from  $f(x_t | x_{t-1}^{a_t^{(i)}})$      *propagation,  $\{x_t^{(i)}, 1\} \approx p(x_t | y_{1:t-1}, \vartheta)$*
- 5     Set  $w_t^{(i)} = g(y_t | x_t^{(i)})$      *weighting,  $\{x_t^{(i)}, w_t^{(i)}\} \approx p(x_t | y_{1:t}, \vartheta)$*
- 6 **end**

All statements with  $(i)$  are for  $i = 1, \dots, N_x$ . The notation  $\approx$  means that the weighted samples on the left hand side are approximately (in the meaning of (5.1)) the density on the right hand side.

---

## 5.2 The bootstrap particle filter

As a popular example of a non-trivial Monte Carlo algorithm, we start by introducing the particle filter. The origin of the particle filter is to be found in Gordon et al. (1993) and Stewart and McCarty (1992). It is a Monte Carlo implementation of the Bayesian filtering recursion (3.6) solving the *state inference* problem, i.e., computing the filtering distributions  $p(x_1 | y_1, \vartheta), \dots, p(x_T | y_{1:T}, \vartheta)$  (cf. the generic  $\pi$  in the previous section) in the state-space model, when the model parameters  $\vartheta$  are known. An animated beginner's introduction to the particle filter is found in Svensson (2013), and there is a myriad of written introductions, e.g. Arulampalam et al. (2002), Gustafsson et al. (2002), Haykin and Freitas (2004), and Särkkä (2013). A good overview (but perhaps not a first introduction) is provided by Doucet and Johansen (2011).

The key idea of the particle filter is to propagate a set of  $N_x$  weighted particles  $\{x_t^{(i)}, w_t^{(i)}\}_{i=1}^{N_x}$  (samples of the state) along the time dimension  $t$ , by propagating them from time  $t - 1$  to the next time step  $t$  by drawing samples from  $f(\cdot | x_{t-1}^{(i)})$  (3.2a), and adapt them to the measurements according to  $g(y_t | x_t^{(i)})$  (3.2b). An important step in the implementation is also the resampling step, where (loosely speaking) particles with small weights are discarded and particles with large weights are duplicated. This is summarized in Algorithm 1, the so-called bootstrap particle filter.<sup>2</sup>

### 5.2.1 Resampling

The resampling step ensures that computational resources are spent in the most interesting parts of the state-space, and that a situation where all but one particle eventually have zero weights is avoided. This can be seen as deciding a genealogy of the particles, i.e., how many descendants a certain particle will have, and which of

---

<sup>2</sup>The connection between Algorithm 1 and the straps aimed for helping when putting on a pair of leather boots may seem rather weak. The history involves the saying 'pull oneself up by one's bootstraps' (often, but probably falsely, attributed to the fictional character Baron Munchausen by Raspe 1786), which is the background for the naming of the statistical idea 'bootstrap' (Efron 1979), which has a close connection to the resampling.

the particle branches that will become extinct. (The genealogy analogue can be particularly helpful when considering the inference problem of the entire sequence  $x_{1:T}$ ; clearly,  $x_t$  is correlated with  $x_{t-1}$ ). To obtain a consistent algorithm, the resampling scheme has to be constructed such that

$$\mathbb{E} \left[ \# \text{ of descendants to } x_{t-1}^{(i)} \right] = \sum_{j=1}^{N_x} \mathbb{P} \left( a_t^{(j)} = i \right) \propto w_{t-1}^{(i)}. \quad (5.2)$$

There are alternatives when it comes to designing a resampling algorithm that fulfills (5.2), see, e.g., Douc and Cappé (2005) and Murray, Lee, et al. (2015) for overviews. It is also possible to design resampling schemes where the duplicated particles are not assigned unit weights (as implicitly done in Algorithm 1) see, Paige et al. (2014) for an example. This is also the underlying key observation for the novel method proposed in Paper VI.

In all non-trivial cases the resampling step is a stochastic procedure, which unfortunately also adds to the variance of the final estimates obtained from the particle filter. It is therefore common to perform the resampling only when needed, which is usually determined by monitoring the so-called effective sample size (ESS, Kong et al. 1994)  $\left( \sum_{i=1}^{N_x} (w^{(i)} / \sum_{j=1}^{N_x} w^{(j)})^2 \right)^{-1}$ , taking values between 1 and  $N_x$ , and perform resampling only when the ESS falls below a certain threshold, e.g.,  $N_x/2$ . If an adaptive resampling scheme is used, a slight modification of the weight update in Algorithm 1 is needed.

### 5.2.2 Positive and unbiased estimates of $p(y_{1:T} | \vartheta)$

The particle filter was first used as a tool for solving the filtering problem in nonlinear state-space models, but it can also be used to estimate the likelihood  $p(y_{1:T} | \vartheta)$  (3.7). The estimate is created from the weights  $w_t^{(i)}$  in Algorithm 1 as

$$\widehat{p}_{N_x}(y_{1:T} | \vartheta) = \prod_{t=1}^T \left( \frac{1}{N_x} \sum_{i=1}^{N_x} w_t^{(i)} \right), \quad (5.3)$$

where we emphasize in the notation that it is a Monte Carlo-based estimate based on  $N_x$  particles. It can be shown (see, e.g., Appendix A) that (5.3) is an unbiased estimate of the likelihood, i.e.,

$$\mathbb{E} \left[ \widehat{p}_{N_x}(y_{1:T} | \vartheta) \right] = p(y_{1:T} | \vartheta), \quad (5.4)$$

This claim is not asymptotic in  $N_x$ , but holds for any finite number  $N_x \geq 1$  of particles. The expectation in (5.4) is over realizations of Algorithm 1 itself, i.e., the randomness involved in the propagation and resampling step. It further holds (as can be seen by inspection of (5.3)) that  $\widehat{p}(y_{1:T} | \vartheta) \geq 0$ . This can, as we will see, be used in algorithms for learning the model parameters  $\vartheta$ . We will also mention a few more theoretical properties about Algorithm 1 later in Section 5.4.4.

---

**Algorithm 2:** Markov chain Monte Carlo sampler

---

**Input:** A transition kernel  $\mathcal{K}$  with stationary distribution  $\pi$ .**Output:** Unweighted samples  $\{\theta^{(k)}\}_{k=0}^K$  from (in the limit  $K \rightarrow \infty$ )  $\pi$ .

- 1 Draw  $\theta^{(0)}$  arbitrarily
  - 2 **for**  $k = 1$  **to**  $K$  **do**
  - 3 | Draw  $\theta^{(k)}$  from  $\mathcal{K}(\theta | \theta^{(k-1)})$
  - 4 **end**
- 

## 5.3 The Markov chain Monte Carlo sampler

Let us now leave the particle filter and the state-space model aside, and return to the general problem we formulated in Section 5.1. That is, we are interested in drawing conclusions about some analytically intractable distribution  $\pi(\theta)$ , typically a posterior  $p(\theta | y)$ . If we can not draw samples from  $\pi$  directly, but instead evaluate  $\pi$  point wise (i.e., query the value of  $\pi(\theta)$  for any  $\theta$ , at least up to proportionality), we can use the Markov chain Monte Carlo (MCMC) methodology to generate samples from  $\pi$ . The MCMC sampler is an algorithm that stochastically explores the  $\theta$ -space, and thereby defines a stochastic process (a Markov chain) in that space. We denote the realization of the stochastic process, i.e., the outcome of one run of the algorithm, as  $\{\theta^{(0)}, \theta^{(1)}, \dots, \theta^{(K)}\}$ . An MCMC sampler is designed such that  $\{\theta^{(0)}, \theta^{(1)}, \dots, \theta^{(K)}\}$  becomes an (unweighted) particle approximation of  $\pi$  in the limit<sup>3</sup> as  $K \rightarrow \infty$ .

We briefly review the essential ideas of how to construct an MCMC sampler. A more complete treatment of the topic is found in, e.g., Tierney (1994), Andrieu, Freitas, et al. (2003), Robert and Casella (2004, Chapter 6) and Liang et al. (2010). The key ingredient in an MCMC algorithm is a transition kernel  $\mathcal{K}(\cdot | \cdot)$  with a certain stationary distribution. A transition kernel is any function  $\mathcal{K}(\cdot | \cdot)$  (where both arguments live in  $\theta$ -space) such that  $\mathcal{K}(\cdot | \theta')$  is a probability density for every  $\theta'$ . A stationary distribution  $\pi$  of  $\mathcal{K}$  is such that  $\mathcal{K}(\cdot | \pi) = \pi(\cdot)$ , where we use the shorthand notation  $\mathcal{K}(\cdot | \pi) \triangleq \int \mathcal{K}(\cdot | \theta') \pi(\theta') d\theta'$ . If  $\mathcal{K}$  fulfills certain technical conditions, it can be applied in Algorithm 2 to produce samples from  $\pi$  in the limit as  $K \rightarrow \infty$ . The conditions are essentially that  $\mathcal{K}$  should not admit periodic cycles and that for any  $\theta$  and  $\theta'$ , there should exist an  $n$  such that  $\mathcal{K}^n(\theta | \theta') > 0$  (where  $\mathcal{K}^n$  denotes an  $n$ -fold iterative application of  $\mathcal{K}$ ),

The transition kernel  $\mathcal{K}$  in MCMC is often defined by an algorithm itself, rather than a closed form expression. Many different methods for designing MCMC kernels exist, such as slice sampling (Neal 2003), Hamiltonian Monte Carlo (Duane et al. 1987; Neal 2011), the bouncy particle sampler (Bouchard-Côté et al. 2017; Peters and With 2012), etc. We will now introduce the perhaps two most basic and standard algorithms for designing  $\mathcal{K}$ , Metropolis-Hastings and Gibbs sampling.

---

<sup>3</sup>The asymptotic behavior as  $K \rightarrow \infty$  is (if the sampler fulfills certain conditions) independent of the initialization  $\theta^{(0)}$ , but in practice a so-called burn-in period of some length  $K_b$ , typically has to be considered, and the corresponding first  $K_b$  samples are discarded. For the performance in practice, it can be crucial to consider and analyze this transient behavior of the MCMC sampler. We will, however, not reflect any more on this, but refer to, e.g., Chapter 12 of Robert and Casella (2004).

**Algorithm 3:** Metropolis-Hastings transition kernel  $\mathcal{K}$ **Input:**  $\theta^{(k-1)}$ **Output:**  $\theta^{(k)}$ 

- 1 Draw  $\theta'$  from  $q(\theta' | \theta^{(k-1)})$  *A candidate for  $\theta^{(k)}$*
- 2 Compute  $\alpha = \min\left(\frac{\gamma(\theta')}{\gamma(\theta^{(k-1)})} \frac{q(\theta^{(k-1)} | \theta')}{q(\theta' | \theta^{(k-1)})}\right)$  *The acceptance probability*
- 3 Set  $\theta^{(k)} = \begin{cases} \theta' & \text{with probability } \alpha \\ \theta^{(k-1)} & \text{with probability } 1 - \alpha \end{cases}$  *Decide if candidate is accepted or not*

**Algorithm 4:** Gibbs transition kernel  $\mathcal{K}$ **Input:**  $\theta^{(k-1)}$ **Output:**  $\theta^{(k)}$ 

- 1 Draw  $\theta_1^{(k)}$  from  $p(\theta_1 | \theta_2^{(k-1)})$
- 2 Draw  $\theta_2^{(k)}$  from  $p(\theta_2 | \theta_1^{(k)})$

## 5.3.1 The Metropolis-Hastings kernel

The Metropolis-Hastings algorithm (named<sup>4</sup> after Nicholas Metropolis, Rosenbluth, et al. 1953 and Wilfred K. Hastings 1970) is a popular plug-in kernel, only requiring that  $\pi$  can be evaluated point wise up to proportionality as  $\pi(\theta) = \gamma(\theta)/Z$ . A proposal density  $q(\cdot | \theta^{(k-1)})$  is also needed, from which samples of  $\theta$  can be drawn, and is either symmetric ( $q(\theta | \theta') = q(\theta' | \theta)$ ) or can be evaluated point wise. The Metropolis-Hastings algorithm is outlined by Algorithm 3. The idea is to sample a candidate  $\theta'$  from the proposal, and always (with an adjustment to account for bias caused by the proposal) accept the candidate as  $\theta^{(k)}$  if  $\pi(\theta') \geq \pi(\theta^{(k-1)})$ . However, also if  $\pi(\theta') < \pi(\theta^{(k-1)})$ , the candidate may be accepted with a certain acceptance probability, designed in a way to create the correct stationary distribution. If the support of the proposal  $q(\cdot | \theta^{(k-1)})$  covers the support of  $\pi$ , it can be proved (e.g., Robert and Casella 2004, Theorem 7.2) that  $\pi$  is the stationary distribution of Algorithm 3, and it can be used in the MCMC sampler (Algorithm 2) to generate samples from  $\pi$ .

## 5.3.2 The Gibbs kernel

The Metropolis-Hastings algorithm has an element of rejection sampling, effectively a trial and error approach where a large fraction of the computational resources may be spent on computing  $\gamma(\theta')$  for proposals that are never accepted. The Gibbs algorithm (named after Josiah Willard Gibbs, coined by S. Geman and D. Geman 1984) is an alternative kernel that does not suffer from this drawback, but produces samples that are always accepted (but may on the other hand suffer from a high autocorrelation). The Gibbs kernel requires that  $\theta$  can be partitioned as  $\theta = \{\theta_1, \theta_2, \dots, \theta_M\}$  (preferably with low cross-dependence between the partitions) so that it is possible to draw samples from  $p(\theta_m | \theta \setminus \theta_m) = \frac{\pi(\theta)}{\int \pi(\theta) d\theta_m}$  for every partition  $m$ . Then, this sampling is

<sup>4</sup>It should, however, be remembered that the original article has 5 authors, and Metropolis happened to be the first one in the alphabetical ordering.

iterated over all  $m$ , as summarized by Algorithm 4 for the case  $M = 2$ . The analysis for the Gibbs sampler is, however, rather intricate (see, e.g., Robert and Casella 2004, Chapter 9 and 10 and references therein), but the resulting Markov chain can under certain conditions be proven to fulfill the necessary conditions for producing samples of  $\pi$  when used in the MCMC sampler (Algorithm 2) as  $K \rightarrow \infty$ .

It is also possible to construct combinations of the Metropolis-Hastings and Gibbs algorithm (Liang et al. 2010, Section 3.4; Müller 1991 and Robert and Casella 2004, Section 10.3), although care must be taken in order not to change the stationary distribution (Dyk and Jiao 2014).

### 5.3.3 Convergence

The convergence of Algorithm 2 in the asymptotic case  $K \rightarrow \infty$  follows, under some additional assumptions on  $\mathcal{K}$ , a central limit theorem. For a measurable test function  $h(\theta)$ , the difference between the true (and after-sought) expectation  $\mathbb{E}[h(\theta)]$  and the sample-based estimate of it  $h_K(\{\theta^{(k)}\}_{k=1}^K) = \frac{1}{K} \sum_{k=1}^K h(\theta^{(k)})$  is

$$\sqrt{K} \left( h_K(\{\theta^{(k)}\}_{k=1}^K) - \mathbb{E}[h(\theta)] \right) \xrightarrow{d} \mathcal{N}(0, \sigma_{\text{MCMC}}^2(h, \pi)) \quad (5.5)$$

where  $\sigma_{\text{MCMC}}^2(h)$  is a bounded function of  $h$  and  $\pi$  (Tierney 1994, Theorem 4 and 5; Robert and Casella 2004, Theorem 6.65 and 6.67).

## 5.4 The Sequential Monte Carlo sampler

As discussed in Section 3.4.1, the state inference in a state-space model is a particular learning problem. Similarly, the particle filter can be seen as a particular instance of the more general sequential Monte Carlo (SMC) method. SMC can also be formulated for other types of models, such as graphical models (Naesseth, Lindsten, and Schön 2014).

The most generic formulation of SMC can be found in the Feynman-Kac formalism (Del Moral 2004; Del Moral and Doucet 2014). Yet another instance of SMC is the SMC sampler (Del Moral, Doucet, and Jasra 2006), here presented as Algorithm 5. The SMC sampler is formulated for the same problem as the MCMC sampler, namely to sample from a static density  $\pi$  which only can be evaluated point wise up to proportionality.

The particle filter targets the filtering distributions (3.6) sequentially<sup>5</sup>. For the SMC sampler, only a static distribution  $\pi$  is typically of user interest, but a sequence of probability distributions  $\{\pi_0, \pi_1, \dots, \pi_p\}$  is introduced as an intermediate tool, and the particles are then propagated along this sequence. It is assumed that all  $\pi_p$  can be evaluated up to proportionality, i.e.,  $\pi_p(\theta) = \gamma_p(\theta)/Z_p$ , where  $\gamma_p(\theta)$  can be computed for any  $\theta$ .

<sup>5</sup>Hence the name *sequential* Monte Carlo.

---

**Algorithm 5:** Sequential Monte Carlo sampler
 

---

**Input:** Sequence of densities  $\{\pi_0, \pi_1, \dots, \pi_P\}$  on the form  $\pi_p(\theta) = \gamma_p(\theta)/Z_p$ , with  $\gamma_p(\theta)$  possible to evaluate point wise.

**Output:** Weighted samples  $\{\theta_p^{(i)}, w_p^{(i)}\}_{i=1}^{N_\theta}$  from  $\pi_p(\theta)$ , for each  $p = 0, \dots, P$ .

- 1 Draw  $\theta_0^{(i)} \sim \gamma_0(\theta_0)$  and set  $w_0^{(i)} = 1$
- 2 **for**  $p = 1$  **to**  $P$  **do**
- 3     Draw  $a_p^{(i)}$  with  $\mathbb{P}(a_p^{(i)} = j) \propto w_{p-1}^{(j)}$  *resampling*,  $\{\theta_{p-1}^{(i)}, 1\} \approx \pi_{p-1}$
- 4     Draw  $\theta_p^{(i)}$  from  $\mathcal{K}_p(\theta_p | \theta_{p-1}^{(i)})$  *propagation*,  $\{\theta_p^{(i)}, 1\} \approx \mathcal{K}_p(\cdot | \pi_{p-1})$
- 5     Set  $w_p^{(i)} = W_p(\theta_p^{(i)}, \theta_{p-1}^{(i)})$  *weighting*,  $\{\theta_p^{(i)}, w_p^{(i)}\} \approx \pi_p$
- 6 **end**

All statements with (i) are for  $i = 1, \dots, N_\theta$ , and  $\mathcal{K}_p$  can be taken as Algorithm 3.

---

### 5.4.1 Connection to particle filters

We can retrieve the bootstrap particle filter (Algorithm 1) from the SMC sampler (Algorithm 5) by letting  $\theta = x$ ,  $P = T$ ,  $\pi_p(\theta_p) = p(x_t | y_{1:t})$ ,  $W_p(\theta_p, \theta_{p-1}) = g(y_t | x_t)$  and  $K_p(\theta_p, \theta_{p-1}) = f(x_t | x_{t-1})$ . More advanced versions of the particle filter are also possible to formulate, where  $f(x_t | x_{t-1})$  is replaced by a more general proposal density, and the weighting is adjusted accordingly (see, e.g., Doucet and Johansen 2011 for an overview). The aim of such a construction is typically to decrease the variance of the particle weights and the final estimates.

### 5.4.2 Constructing a sequence $\{\pi_p\}_{p=0}^P$

The particle filter sequentially targets the densities  $p(x_t | y_{1:t}, \vartheta)$ . The SMC sampler, on the other hand, targets a static density  $\pi$ . Therefore, we have to construct an artificial sequence of distributions  $\{\pi_p\}_{p=0}^P$  (with  $\pi_0$  easy to sample from and  $\pi_P = \pi$ ) along which the particles can be propagated. Preferably, the distance between any consecutive  $\pi_{p-1}$  and  $\pi_p$  should be ‘small’ in order to guide the particles well towards  $\pi_P = \pi$ . This idea resembles simulated annealing (also introduced by Metropolis, Rosenbluth, et al. 1953) and continuation methods (Richter and DeCarlo 1983).

If  $\pi(\theta)$  is a posterior, i.e.,  $\propto p(\theta)p(y | \theta)$ , one option is to construct  $\{\pi_p\}_{p=0}^P$  as the likelihood-tempered sequence

$$\pi_p \propto p(\theta)p(y | \theta)^{p/P}. \quad (5.6)$$

Another alternative is the data-tempered sequence

$$\pi_p \propto p(\theta | y_{B_{0:p}}), \quad (5.7)$$

where  $\{B_p\}_{p=0}^P$  is a sequence with batches of the data  $y$ , such that  $B_0$  is empty and  $B_{0:P}$  contains all data  $y$ . A third option is proposed in Paper V.

### 5.4.3 Propagating the particles

For the SMC sampler, there is no underlying state-space model as for the particle filter that can be used to propagate or weight the particles. Therefore,  $W_p$  and  $\mathcal{K}_p$  has to be chosen by the user. Different alternatives are possible (Del Moral, Doucet, and Jasra 2006, Section 3.3), but one choice is

$$\mathcal{K}_p(\cdot | \cdot) \text{ Metropolis-Hastings kernel with stationary distribution } \pi_{p-1}, \quad (5.8a)$$

$$W_p(\theta_p, \theta_{p-1}) = \frac{\pi_p(\theta_{p-1})}{\pi_{p-1}(\theta_{p-1})}, \quad (5.8b)$$

which can be shown to yield a consistent algorithm. The SMC sampler with the choices (5.6-5.8) is a rather general scheme, which can be applied to a broad range of problems. One example is found in Paper VII and another in Del Moral, Doucet, and Jasra (2012a). We will later also review how it can be applied to the parameters  $\vartheta$  in the state-space model, resulting in the SMC<sup>2</sup> algorithm (Chopin, Jacob, et al. 2013).

### 5.4.4 Convergence

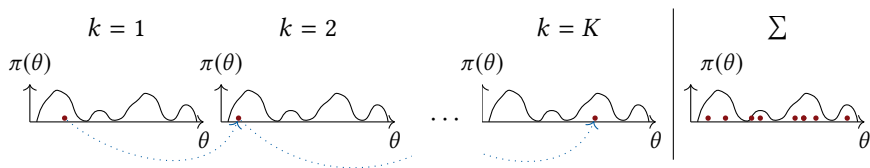
We have already discussed an important property of the particle filter (Algorithm 1), namely that  $\widehat{p}_{N_x}(y_{1:T} | \vartheta)$  is unbiased for any finite  $N_x \geq 1$ . In a similar manner, it is possible to construct an unbiased estimator also for the normalizing constants  $Z_p$  in the SMC sampler. Results concerning the long-term stability of SMC, and in particular particle filters, also exist (Douc, Moulines, et al. 2014; Whiteley 2013).

Akin to the MCMC case, results are available also for the asymptotic case  $N_\theta \rightarrow \infty$ . As for MCMC (Section 5.4.4), we can for every measurable test function  $h(\theta)$  establish (under some technical assumptions) the central limit theorem for Algorithm 5

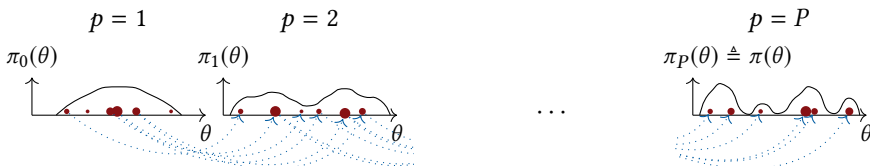
$$\sqrt{N_\theta} \left( h_{N_\theta}(\{\theta_p^{(i)}, w_p^{(i)}\}_{i=1}^{N_\theta}) - \mathbb{E}[h(\theta)] \right) \xrightarrow{d} \mathcal{N}(0, \sigma_{\text{SMC}}^2(h, \pi)), \quad (5.9)$$

when  $N_\theta \rightarrow \infty$ , where  $\sigma_{\text{SMC}}^2(h)$  is a bounded function of  $h$  and  $\pi$  (Del Moral, Doucet, and Jasra 2006, Proposition 2). This result is applicable to any SMC algorithm (Chopin 2004; Del Moral 2004), and in particular also for the particle filter in Algorithm 1. The case when resampling is performed only adaptively (as discussed in Section 5.2.1; also applicable to Algorithm 5) is more intricate to analyze, but similar results have been presented by Del Moral, Doucet, and Jasra (2012b).

To summarize, the bottom line is that the SMC sampler has a central limit theorem on the same form as MCMC.



(a) The MCMC idea: propagate a single sample (red dot) through the landscape of  $\pi$ , such that its random trace (summarized in the rightmost plot) eventually becomes samples of the distribution of interest  $\pi$ . That is, the chain has to ‘visit’ areas where  $\pi$  is large more often than areas where  $\pi$  is small. It will most likely have visited every mode of  $\pi$  as  $K \rightarrow \infty$ , but not necessarily within a reasonable finite time (i.e., before the user’s computational budget is consumed).



(b) The SMC idea: propagate a set of  $N_x$  ( $N_x = 6$  in this illustration) particles (samples, red dots) through a sequence of  $P$  distributions  $\pi_0, \dots, \pi_P$ , to eventually end up with samples from the distribution of interest  $\pi(\cdot) \triangleq \pi_P(\cdot)$ . By making a ‘smooth’ transition from the easy-to-sample distribution  $\pi_0$  to the distribution of interest  $\pi$  the hope is that the samples represent  $\pi$  more efficiently than in the MCMC setting (by exploring different modes in parallel, etc.).

**Figure 5.3:** The key concept of the MCMC (a) and SMC samplers (b). The idea of MCMC is to make a (more or less informed) stochastic walk with a single particle in  $\theta$ -space such that the walk will be proportional to the density  $\pi$ . The SMC idea is to propagate a whole bunch of particles through an evolving landscape (cf. how the particle filter solves the state inference problem), which after a pre-defined number of iterations  $P$  ends up in  $\pi$ .

## 5.5 Markov Chain or Sequential Monte Carlo?

MCMC has been around since the 1950’s, whereas SMC is younger than the author of this thesis<sup>6</sup>. With that perspective, it is perhaps not surprising that the MCMC sampler can essentially be seen as the special case of the SMC sampler with  $\pi_p = \pi$  and  $N_\theta = 1$ . For this reason, we may also expect (as confirmed by Svensson and Schön 2016 for a particular case) that the SMC sampler requires more user effort, in terms of implementation time. It is also worth highlighting that the number of iterations  $K$  in the MCMC sampler (Algorithm 2) does not have to be specified beforehand, but the algorithm can be run until the computational budget is consumed, a so-called anytime algorithm. For the SMC sampler (Algorithm 5), both  $N_\theta$  and  $P$  have to be specified before beforehand, and is thereby not an anytime algorithm.

The different underlying ideas on how the samples are drawn are illustrated in Figure 5.3. Any attempt to claim superiority of one approach in general is probably fruitless. However, a rudimentary knowledge about both alternatives can probably help in making wise choices: the historical timeline might have given MCMC an advantage.

<sup>6</sup>Who would like to claim that he is rather young.

## 5.6 Monte Carlo for state-space model parameters $\vartheta$

The particle filter (Section 5.2) with its various extensions and generalizations provides an often unbeaten Monte Carlo solution for inferring the states  $x_t$  in the state-space model (3.2). (MCMC may, however, be beneficial for some particular problem settings, such as the case in Svensson, Schön, et al. 2015.) For the problem of finding model parameters  $\vartheta$ , on the other hand, the particle filter cannot provide a solution on its own<sup>7</sup>. However, the particle filter can be a very useful building block of an MCMC or SMC sampler to construct well-performing and theoretically consistent algorithms for inferring the posterior  $p(\vartheta | y_{1:T})$ , as well as the maximum likelihood estimate  $\hat{\vartheta}$ .

### 5.6.1 MCMC for nonlinear state-space models: PMCMC

For inferring  $\vartheta$  in linear state-space models (3.3), MCMC can be used essentially out of the box. The use of a Metropolis-Hastings sampler is shown in Ninness and Henriksen (2010) (although formulated for transfer functions; a state-space model formulation is found in Schön, Lindsten, et al. 2015, Example 4), and the Gibbs sampler in Wills et al. 2012. In both cases the Kalman filter (and some extensions of it) provides the required expressions for  $p(\vartheta | y_{1:T})$  (for the Metropolis-Hastings solution) and  $p(x_{1:T} | \vartheta, y_{1:T})$  (for the Gibbs solution). In the Gibbs solution we also need an expression for  $p(\vartheta | x_{1:T}, y_{1:T})$ , which is (if the conjugate prior is used) provided by the matrix normal inverse Wishart distribution (Appendix B).

The two cases in the previous paragraph are special cases in that the required expressions are available analytically. For the general nonlinear state-space model (3.2), neither the likelihood  $p(y_{1:T} | \vartheta)$  nor the conditional distribution  $p(x_{1:T} | \vartheta, y_{1:T})$  are available in closed form, nor can they be computed exactly. It turns out that the particle filter provides a good approach for approximating these distributions, in the combined particle-filter-within-MCMC framework<sup>8</sup>, PMCMC (Andrieu, Doucet, et al. 2010).

#### Pseudo-marginal Metropolis-Hastings

What happens to the Metropolis-Hastings sampler (Algorithm 3) if  $\pi(\theta)$  cannot be evaluated exactly, but only stochastically estimated  $\hat{\pi}(\theta)$ ? A naïve approach would perhaps be to pretend that  $\hat{\pi}(\theta)$  is exact (i.e., contains no stochastic element) and apply Algorithm 3. (Another attempt could be to average over a few realizations of  $\hat{\pi}(\theta)$  for every  $\theta$ , and use that average when computing the acceptance probability  $\alpha$ .) It turns out (Andrieu and Roberts 2009), somewhat surprisingly, that if  $\hat{\pi}(\theta)$  is positive and unbiased, i.e.,  $\mathbb{E}[\hat{\pi}(\theta)] = \pi(\theta)$  and  $\hat{\pi}(\theta) > 0$ , using  $\hat{\pi}(\theta)$  as if it were exact (the approach suggested above) creates a consistent algorithm, in the sense that the stationary distribution of Algorithm 3 remains unchanged!

<sup>7</sup>If the unknown parameters have a low dimensionality, they can possibly be considered as part of the state  $x_t$  (and modeled to be slowly time-varying), and the problem is thereby transferred to a vanilla state inference setting. This solution highlights the (mild) arbitrariness of splitting unknown parameters  $\theta$  in the state-space model into model parameters  $\vartheta$  and states  $x_t$ .

<sup>8</sup>The original meaning of PMCMC is simply ‘particle Markov chain Monte Carlo’, but ‘particle-filter-within-MCMC’ is a more explanatory interpretation.

This quite remarkable fact can be proven by handling the randomness of  $\widehat{\pi}(\theta)$  explicitly by introducing another random variable  $v$ , and considering  $\widehat{\pi}(\theta)$  to be deterministic when conditioned on  $v$ . Then, it is possible to show that the Metropolis-Hastings sampler targets an extended distribution  $p(\theta, v)$ , and that  $\pi(\theta)$  can be obtained by integrating  $v$  out. Thus the name of the approach, pseudo-marginal.

### Particle marginal Metropolis-Hastings

Following the pseudo-marginal Metropolis-Hastings approach with  $\widehat{\pi}(\theta) \propto \widehat{p}_{N_x}(y_{1:T} | \vartheta)p(\vartheta)$  from (5.3), the particle marginal Metropolis-Hastings approach is obtained (Andrieu, Doucet, et al. 2010, Section 2.4.2). Although not affecting the asymptotical properties, the choice of the number of particles  $N_x \geq 1$  and the proposal density  $q(\cdot | \cdot)$  are crucial for its practical performance. Some discussion on how to choose  $N_x$  can be found in Andrieu, Doucet, et al. (2010), and some design methods for  $q$  can be found in Dahlin, Lindsten, et al. (2015). Two beginner's introduction to particle Metropolis-Hastings are provided by Dahlin and Schön (2016) and Schön, Svensson, et al. (2018).

### Particle Gibbs

It is also possible to construct a Gibbs sampler, Algorithm 4, for state-space model parameters  $\vartheta$ . Such a construction is possible by taking  $\theta$  in Algorithm 4 as  $\{x_{1:T}, \vartheta\}$ , i.e., iteratively sample  $x_{1:T}^{(k)}$  conditional on the model parameters  $\vartheta^{(k-1)}$ , and the model parameters  $\vartheta^{(k)}$  conditional on the state  $x_{1:T}^{(k)}$ . Thus, we need to draw samples from  $p(x_{1:T} | \vartheta^{(k)})$  as well as  $p(\vartheta | x_{1:T}^{(k)})$ .

For certain state-space model structures (e.g., the linear model in Wills et al. 2012, the models in Section 7 in Lindsten, Jordan, et al. 2014 and the model in Paper I),  $p(\vartheta | x_{1:T}^{(k)})$  is available in closed form and possible to sample from. If that is not the case, other sampling strategies can be used, see, e.g., Example 8 of Schön, Lindsten, et al. (2015).

To sample approximately from  $p(x_{1:T} | \vartheta^{(k)})$ , a particle filter can be used: the approximation is due to the finite number of particles  $N_x$  in the particle filter. However, with a slightly more involved Gibbs sampling scheme it is possible to draw MCMC samples of  $x_{1:T}$  with a kernel (constructed using the so-called conditional particle filter) with exactly  $p(x_{1:T} | \vartheta^{(k)})$  as its stationary distribution. A particularly well-performing conditional particle filter construction has proven to be the one introduced by Lindsten, Jordan, et al. (2014), the conditional particle filter with ancestor sampling. We will not detail this construction any further here, but we refer to Andrieu, Doucet, et al. (2010) and Lindsten, Jordan, et al. (2014) for all technical details on this so-called particle Gibbs construction.

## 5.6.2 Particle Gibbs for maximum likelihood estimation

If the maximum likelihood estimate  $\widehat{\vartheta}$  (rather than the posterior  $p(\vartheta | y_{1:T})$ ) is of interest, Lindsten (2013) and Paper III presents a combination of particle Gibbs and a stochastic approximation (Robbins and Monro 1951) version of the expectation maximization (EM) algorithm (Dempster et al. 1977). The construction makes use of particle

Gibbs only for the state inference problem, and uses the stochastic approximation EM framework (Delyon et al. 1999; Kuhn and Lavielle 2004) for the maximum likelihood estimation of  $\vartheta$ . The use of EM for maximum likelihood estimation of  $\vartheta$  in nonlinear state-space models has been around since at least Ghahramani and Roweis (1998), and the combination of SMC and EM for this purpose has been proposed by Cappé et al. (2005), Olsson et al. (2008), and Schön, Wills, et al. (2011). The combination of particle Gibbs and stochastic approximation EM, as proposed by Lindsten (2013), improves the convergence properties and reduces the computational load compared to previous algorithms. A more detailed introduction is given in Paper III, and Papers I and IV both use the method for two particular model structures.

### 5.6.3 SMC for state-space model parameters: SMC<sup>2</sup>

In the same spirit as the MCMC methodology can be used for sampling the posterior  $p(\vartheta | y_{1:T})$  of the state-space model parameters, so can the SMC sampler. The SMC sampler can be applied directly to a linear state-space model, akin to the MCMC sampler case, since  $p(y_{1:T} | \vartheta)$  is explicitly available from the Kalman filter. A natural way to construct a sequence of densities is the data-tempered alternative  $P = T$ ,  $\pi_0(\vartheta) = p(\vartheta)$ ,  $\pi_1(\vartheta) = p(\vartheta | y_1)$ ,  $\dots$ ,  $\pi_T(\vartheta) = p(\vartheta | y_{1:T})$ . An alternative construction, for the special case when the state-space model has very little measurement noise, is proposed in Paper V. For the general case with a nonlinear state-space model, the particle filter is required to approximately evaluate  $p(y_{1:t} | \vartheta)$  as  $\widehat{p}_{N_x}(y_{1:t} | \vartheta)$ , yielding the SMC<sup>2</sup> algorithm<sup>9</sup> (Chopin, Jacob, et al. 2013; Fulop and Li 2013). For propagating the particles in step 4 in Algorithm 5, the particle Metropolis-Hastings kernel (Algorithm 3) can be used. Once again the unbiasedness  $\mathbb{E}[\widehat{p}_{N_x}(y_{1:t} | \vartheta)] = p(y_{1:t} | \vartheta)$  is key to obtaining a consistent algorithm; the details are found in Section 3.1 in Chopin, Jacob, et al. (2013).

This somewhat involved construction leaves the user with several design choices, for instance the trade-off between the number of particles  $N_x$  in the particle filters and the number of particles  $N_\vartheta$  at the SMC sampler level. Chopin, Ridgway, et al. (2015) have suggested how to automatically adapt these numbers.

SMC<sup>2</sup> is not to be confused with nested SMC (Naesseth, Lindsten, and Schön 2015), which is a general framework for using SMC to construct proposal densities within an SMC algorithm.

## 5.7 Summary of the chapter

This chapter has introduced some Monte Carlo ideas useful for machine learning, and we have in particular considered the particle filter (for state inference in the state-space model) as well as the MCMC and SMC samplers (for general problems). We have also introduced the combinations PMCMC and SMC<sup>2</sup>, both primarily aimed for learning model parameters  $\vartheta$  in state-space models.

---

<sup>9</sup>The naming should be read as ‘SMC square’, i.e., SMC to the power of two; a particle filter (an SMC algorithm) is used within an SMC sampler (another SMC algorithm).



*“Wait, what if these quote marks are inside out, so everything in the rest of the document is the quotation and this part isn’t? Duuuuude.”*

Randall Munroe

# 6

## Conclusions and future work

THIS chapter contains some overall conclusions of the research presented in Paper I–VII. In addition to what is written in each paper, this chapter also has an outlook into further possible research directions.

### 6.1 Conclusions

The contributions of this thesis include applications of state-of-the-art learning methods to non-trivial models, new versions of learning methods themselves, as well as a contribution to model validation methodology. At the heart of all methods lies the use of Monte Carlo approximations to handle otherwise intractable integrals, and often the sequential Monte Carlo method in particular. The results are often promising, but the existence of this thesis suggests that:

- The work is most likely not done yet, but there are probably more problems where (sequential) Monte Carlo can make a difference than the ones included in this thesis. With increasing access to computational power, together with a raised interest for the Bayesian approach, there are probably plenty of opportunities.
- Applications and tweaking of Monte Carlo, and sequential Monte Carlo in particular, is apparently complicated enough to be topic for a doctoral thesis. There is probably work to be done in packaging and providing sequential Monte Carlo as an off-the-shelf method to practitioners not having a PhD degree on the topic.

## 6.2 Future work

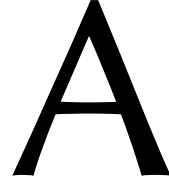
As suggested above, sequential Monte Carlo is potentially useful for a wider range of situations than those where it is used today. Its use is, however, probably limited by the relative complexity of implementing it. A natural research direction would therefore be to provide a ‘user interface’ to sequential Monte Carlo, accessible for an application domain expert not knowledgeable within Monte Carlo methods. Indeed, such initiatives already exist (I have partly been funded by such a project: ASSEMBLE, Murray and Schön 2018), but the work is nowhere close to be finished.

Parts of this thesis are focused around Bayesian learning in somewhat complicated state-space models. However, to the best of my knowledge, little research has been done on how to interpret and efficiently transform the posterior  $p(\vartheta | y)$  into a ‘posterior’ for the dynamical behavior (such as the input-output relationship). Such results would be of particular interest when the parameters  $\vartheta$  themselves do not bear a physical meaning (such as in Paper I).

A common argument for the Bayesian approach is the quantification of the present uncertainty provided by the posterior distribution. It is, however, important to bear in mind that the posterior distribution, and thereby also the uncertainty, is *conditional* on the choice of model. (This conditioning is not only a formality, but critical for the results obtained.) This reflection was part of the inspiration behind Paper II, but several (fundamental) questions still remain: Are there more aspects of a model that should (and can) be validated than the one proposed in Paper II? Is it possible to judge the severity of model misspecifications from a posterior distribution? As a tough and applied question, consider decision making in self-driving cars based on posterior uncertainties from a ‘Bayesian neural network’ (whatever that would mean): what aspects of the data/reality have to be present in the model (the neural network), in order to guarantee that the posterior uncertainty represents something meaningful?

*“Policy should always be rooted  
in unbiased science.”*

Christine Todd Whitman



## The unbiased estimator $\widehat{p}_{N_x}(y_{1:T})$

This chapter contains a proof of the fact that (5.3),

$$\widehat{p}_{N_x}(y_{1:T} | \vartheta) = \prod_{t=1}^T \left( \frac{1}{N_x} \sum_{i=1}^{N_x} w_t^{(i)} \right), \quad (\text{A.1})$$

with  $w_t^{(i)}$  generated by the bootstrap particle filter Algorithm 1 in Chapter 5, is an unbiased estimator of the likelihood  $p(y_{1:T} | \vartheta)$  (3.7) of a state-space model with model parameters  $\vartheta$ , for any finite  $N_x \geq 1$ . With unbiasedness, we mean  $\mathbb{E}[\widehat{p}_{N_x}(y_{1:T} | \vartheta)] = p(y_{1:T} | \vartheta)$ , where the expectation is over the randomness in the particle filter algorithm itself. This result was first presented by Del Moral (2004, Section 7.4.2) and is important to many parameter learning strategies, such as particle marginal Metropolis-Hastings (Section 5.6.1) and Paper VI. The proof here follows closely that of Pitt et al. (2012), which is written for the more general case of the auxiliary particle filter.

In the following,  $\vartheta$  will be suppressed in the notation, since all expressions are conditioned on  $\vartheta$ . We start by introducing the estimator<sup>1</sup>

$$\widehat{p}_{N_x}(y_t | y_{1:t-1}) = \frac{1}{N_x} \sum_{i=1}^{N_x} w_t^{(i)}, \quad (\text{A.2})$$

which has the natural property that  $\prod_{t=1}^T \widehat{p}_{N_x}(y_t | y_{1:t-1}) = \widehat{p}_{N_x}(y_{1:T})$ . We also define  $\widehat{p}_{N_x}(y_{t-h:t} | y_{1:t-h-1})$  naturally as  $\prod_{t'=t-h}^t \widehat{p}_{N_x}(y_{t'} | y_{1:t'-1})$  for  $h \geq 0$ .

The structure of the proof is as follows: First, in Lemma 1, it will be proved that

$$\mathbb{E} \left[ \widehat{p}_{N_x}(y_t | y_{1:t-1}) | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] = \sum_{i=1}^{N_x} \frac{w_{t-1}^{(i)}}{\sum_{j=1}^{N_x} w_{t-1}^{(j)}} p(y_t | x_{t-1}^{(i)}), \quad (\text{A.3})$$

<sup>1</sup>Note the somewhat subtle notation:  $p$  denotes probability densities, whereas  $\widehat{p}_{N_x}$  denotes deterministic functions (which we distinguish by their different arguments) of quantities stochastically generated by the particle filter. The point with the proof is to show that the  $\widehat{p}_{N_x}$ -function (A.1) is an unbiased estimator of the corresponding  $p$ .

i.e.,  $\widehat{p}_{N_x}(y_t | y_{1:t-1})$  (the contribution to (A.1) from iteration  $t$  of the particle filter) is unbiased, if conditioned on a realization of particles from the previous iteration at time  $t - 1$ . Then, in Lemma 2, we prove that it also holds for  $h \geq 1$  sequential iterations of the particle filter, once again conditioned on a realization of particles at time  $t - h - 1$ . Finally, by letting  $h = T$ , we conclude in Theorem 1 that if  $\{x_0^{(i)}\}_{i=1}^{N_x}$  are unbiased samples from  $p(x_0)$ , then must  $\widehat{p}_{N_x}(y_{1:T})$  (A.1) also be unbiased.

**Lemma 1.** *With the definition of  $\widehat{p}_{N_x}(y_t | y_{1:t-1})$  in (A.2), it holds that*

$$\mathbb{E} \left[ \widehat{p}_{N_x}(y_t | y_{1:t-1}) | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] = \sum_{i=1}^{N_x} \frac{w_{t-1}^{(i)}}{\sum_{j=1}^{N_x} w_{t-1}^{(j)}} p(y_t | x_{t-1}^{(i)}). \quad (\text{A.4})$$

*Proof.*

$$\begin{aligned} \mathbb{E} \left[ w_t^{(j)} | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] &= \\ &= \mathbb{E} \left[ \mathbb{E} \left[ w_t^{(j)} | a_t^{(j)}, \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] = \\ &= \mathbb{E}_{a_t^{(j)}} \left[ \mathbb{E}_{x_t^{(j)} \sim f(x_t^{(j)} | x_{t-1}^{(j)})} \left[ g(y_t | x_t^{(j)}) | a_t^{(j)}, \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] = \\ &= \mathbb{E}_{a_t^{(j)}} \left[ p(y_t | x_{t-1}^{(j)}) | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] = \sum_{k=1}^{N_x} p(a_t^{(j)} = k | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x}) p(y_t | x_{t-1}^{(k)}). \end{aligned} \quad (\text{A.5})$$

Then,

$$\begin{aligned} \mathbb{E} \left[ \sum_{j=1}^{N_x} w_t^{(j)} | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] &= \sum_{j=1}^{N_x} \mathbb{E} \left[ w_t^{(j)} | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x} \right] = /(\text{A.5})/ = \\ &= \sum_{k=1}^{N_x} \left( \sum_{j=1}^{N_x} p(a_t^{(j)} = k | \{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x}) \right) p(y_t | x_{t-1}^{(k)}) = \\ &= /(\text{5.2})/ = N_x \sum_{k=1}^{N_x} \frac{w_{t-1}^{(k)}}{\sum_{i=1}^{N_x} w_{t-1}^{(i)}} p(y_t | x_{t-1}^{(k)}), \end{aligned} \quad (\text{A.6})$$

and the lemma follows.  $\square$

We have now proved that given a realization of weighted particles  $\{x_{t-1}^{(i)}, w_{t-1}^{(i)}\}_{i=1}^{N_x}$  representing  $p(x_{t-1} | y_{t-1})$ , the estimator  $\widehat{p}_{N_x}(y_t | y_{1:t-1})$  (A.2), i.e., the contribution to (A.1) from one single iteration of the particle filter for the following time  $t$ , is unbiased. We now present the next lemma, concerning the corresponding unbiasedness of  $\widehat{p}_{N_x}(y_{t-h:t} | y_{1:t-h-1})$ .

**Lemma 2.** *With the definitions of  $\widehat{p}_{N_x}(y_t | y_{1:t-1})$  and  $\widehat{p}_{N_x}(y_{t-h:t} | y_{1:t-h-1})$  from above, it holds that*

$$\mathbb{E} \left[ \widehat{p}_{N_x}(y_{t-h:t} | y_{1:t-h-1}) | \{x_{t-h-1}^{(i)}, w_{t-h-1}^{(i)}\}_{i=1}^{N_x} \right] = \sum_{k=1}^{N_x} \frac{w_{t-h-1}^{(k)}}{\sum_{i=1}^{N_x} w_{t-h-1}^{(i)}} p(y_{t-h:t} | x_{t-h-1}^{(k)}). \quad (\text{A.7})$$

*Proof.* The proof is by induction. For  $h = 0$ , (A.7) is true by Lemma 1. We now assume that (A.7) holds also for an arbitrary  $h$ , and show that it implies that (A.7) also holds for  $h + 1$ . For  $h + 1$ , the left hand side of (A.7) is

$$\begin{aligned}
& \mathbb{E} \left[ \widehat{p}_{N_x}(y_{t-h-1:t} | y_{1:t-h-2}) | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x} \right] = \\
&= \mathbb{E} \left[ \widehat{p}_{N_x}(y_{t-h:t} | y_{1:t-h-1}) \widehat{p}_{N_x}(y_{t-h-1} | y_{1:t-h-2}) | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x} \right] = \\
&= \mathbb{E} \left[ \mathbb{E} \left[ \widehat{p}_{N_x}(y_{t-h:t} | y_{1:t-h-1}) | \{x_{t-h-1}^{(i)}, w_{t-h-1}^{(i)}\}_{i=1}^{N_x} \right] \times \right. \\
&\quad \left. \widehat{p}_{N_x}(y_{t-h-1} | y_{1:t-h-2}) | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x} \right] = \\
&= / \text{Induction assumption and (A.2)} / = \\
&= \mathbb{E} \left[ \sum_{j=1}^{N_x} \frac{w_{t-h-1}^{(j)}}{\sum_{i=1}^{N_x} w_{t-h-1}^{(i)}} p(y_{t-h:t} | x_{t-h-1}^{(j)}) \frac{1}{N_x} \sum_{i=1}^{N_x} w_{t-h-1}^{(i)} | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x} \right] = \\
&= \mathbb{E} \left[ \sum_{j=1}^{N_x} w_{t-h-1}^{(j)} p(y_{t-h:t} | x_{t-h-1}^{(j)}) \frac{1}{N_x} | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x} \right] = \\
&= / \text{akin to (A.5)} : \mathbb{E} \left[ w_{t-h-1}^{(j)} p(y_{t-h:t} | x_{t-h-1}^{(j)}) | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x} \right] = \\
&= \sum_{k=1}^{N_x} p(a_{t-h-1}^{(j)} = k | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x}) p(y_{t-h-1:t} | x_{t-h-2}^{(k)}) / = \\
&= \frac{1}{N_x} \sum_{k=1}^{N_x} \left( \sum_{j=1}^{N_x} p(a_{t-h-1}^{(j)} = k | \{x_{t-h-2}^{(i)}, w_{t-h-2}^{(i)}\}_{i=1}^{N_x}) \right) p(y_{t-h-1:t} | x_{t-h-2}^{(k)}) = \\
&= \sum_{k=1}^{N_x} \frac{w_{t-1}^{(k)}}{\sum_{i=1}^{N_x} w_{t-1}^{(i)}} p(y_{t-h-1:t} | x_{t-h-2}^{(k)}), \quad (\text{A.8})
\end{aligned}$$

and the lemma follows.  $\square$

We have proved that the result from Lemma 1 also holds for  $h \geq 1$  iterations of the particle filter. From Lemma 2, we now have that (with  $t = T$  and  $h = t - 1$ )

$$\mathbb{E} \left[ \widehat{p}_{N_x}(y_{1:T}) | \{x_0^{(i)}, w_0^{(i)}\}_{i=1}^{N_x} \right] = \sum_{k=1}^{N_x} \frac{w_0^{(k)}}{\sum_{i=1}^{N_x} w_0^{(i)}} p(y_{1:T} | x_0^{(k)}) = \sum_{k=1}^{N_x} p(y_{1:T} | x_0^{(k)}) \frac{1}{N_x}. \quad (\text{A.9})$$

If  $x_0^{(k)} \sim p(x_0)$ , we can conclude that

$$\mathbb{E} \left[ \frac{1}{N_x} \sum_{k=1}^{N_x} p(y_{1:T} | x_0^{(k)}) \right] = \int p(y_{1:T} | x_0) p(x_0) dx_0 = p(y_{1:T}). \quad (\text{A.10})$$

We can now formulate the following theorem (where we have re-introduced  $\vartheta$  to the notation).

**Theorem 1.** *For the estimator  $\widehat{p}_{N_x}(y_{1:T} | \vartheta)$ , as defined by (A.1) and Algorithm 1 in Chapter 5, it holds that*

$$\mathbb{E} \left[ \widehat{p}_{N_x}(y_{1:T} | \vartheta) \right] = p(y_{1:T} | \vartheta), \quad (\text{A.11})$$

for any finite  $N_x \geq 1$ .



*“The most important questions of life (...) are indeed for the most part only problems of probability.”*

Pierre-Simon Laplace

# B

## The matrix normal inverse Wishart distribution in linear regression

This appendix gives an introduction to the matrix normal inverse Wishart distribution (and its scalar case normal inverse gamma). The normal inverse gamma and some of its generalizations is often in the literature highlighted as the conjugate prior for a data likelihood model on the form  $p(y | \mu, \sigma^2) = \mathcal{N}(y; \mu, \sigma^2)$ , where both  $\mu$  and  $\sigma^2$  are unknown. In this appendix, we will derive the expressions for the slightly more involved case of a linear regression model, i.e.,  $p(y | a, \sigma^2) = \mathcal{N}(y; ax, \sigma^2)$ , with  $x$  known and  $a$  and  $\sigma^2$  unknown, and also its multivariable extension. Similar expressions can also be found in Quintana (1987).

### B.1 The matrix normal and inverse Wishart distributions

In this section, we introduce the matrix normal inverse Wishart distribution, by first considering the scalar case, and thereafter its multivariable generalization. Introductions can also be found in Dawid (1981) and Press (1982). We will assume a basic familiarity with the Gaussian and the gamma distributions.

B.1.1 The scalar case:  $NI\mathcal{G}$ 

The Gaussian distribution,

$$\mathcal{N}(y; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y - \mu)^2}{2\sigma^2}\right), \quad (\text{B.1})$$

is a probability with support on the entire real line, however with a clear preference for values around its mean  $\mu \pm$  a few standard deviations  $\sigma$ . Because of these easy-to-grasp properties, in combination with its frequent appearance as a limiting distribution (cf. the central limit theorem) and its analytically tractable form, it is ubiquitous in statistical modeling.

A simple problem is that of inferring  $\theta = \mu$  when we observe data  $y_{1:T}$  as exchangeable observations  $p(y_t | \theta) = \mathcal{N}(y_t; \mu, \sigma^2)$ . If we decide to follow the Bayesian way of reasoning, we formulate a prior  $p(\theta)$ . A natural choice for the prior might be  $p(\mu) = \mathcal{N}(\mu; m, \zeta^2)$ , and the posterior then becomes (after some algebra)  $p(\theta | y_{1:T}) = \mathcal{N}\left(\mu; \left(\frac{m}{\zeta^2} + \frac{\sum_t y_t}{\sigma^2}\right) \left(\frac{1}{\zeta^2} + \frac{T}{\sigma^2}\right)^{-1}, \left(\frac{1}{\zeta^2} + \frac{T}{\sigma^2}\right)^{-1}\right)$ , i.e., another Gaussian distribution. Thus, the Gaussian distribution is the conjugate prior for a Gaussian likelihood model with unknown mean.

The above example is, however, somewhat unrealistic, since the mean is unknown whereas the variance is assumed to be known! A less artificial situation would be the problem of inferring  $\theta = \{\mu, \sigma^2\}$  jointly. However, the Gaussian distribution is clearly not a good prior for  $\sigma^2$ , since the Gaussian distribution has support on the entire real line, whereas a negative variance bears no meaning in our model. A way of constructing a distribution with support only on the positive real line, is Proceedings of 26th to consider the square of a standard Gaussian random variable  $z$ , or more generally, the sum of  $\ell$  such squared Gaussian random variables  $z_j$ ,

$$q = \sum_{j=1}^{\ell} z_j^2, \quad p(z_j) \sim \mathcal{N}(z_j; 0, 1). \quad (\text{B.2})$$

The density for  $q$  can be written as

$$p(q) = \frac{1}{2^{\ell/2} \Gamma\left(\frac{\ell}{2}\right)} (q)^{\ell/2-1} \exp\left(-\frac{q}{2}\right) \triangleq \mathcal{G}(q; 1, \ell), \quad (\text{B.3})$$

where we use  $\mathcal{G}$  to be the notation for the so-called *gamma* distribution. By its construction (B.2), we may realize that the mean of  $\mathcal{G}(q; 1, \ell)$  is  $\ell$ , and its variance increases with  $\ell$ . The gamma distribution can be generalized to non-integer  $\ell > 1$ , and also a scale parameter  $\lambda > 0$  can be introduced, as

$$\mathcal{G}(q; \lambda, \ell) = \frac{\lambda^{\ell/2}}{2^{\ell/2} \Gamma\left(\frac{\ell}{2}\right)} (q)^{\ell/2-1} \exp\left(-\frac{q\lambda}{2}\right). \quad (\text{B.4})$$

Now, this distribution could be used as a prior for  $\sigma^2$ . However, to retain conjugacy properties, we have to work with the inverse of  $q$ : if  $q$  is gamma distributed, then is

its inverse  $\sigma^2 \triangleq 1/q$ , distributed as

$$\mathcal{IG}(\sigma^2; \lambda, \ell) = \frac{\lambda^{\ell/2}}{2^{\ell/2} \Gamma\left(\frac{\ell}{2}\right)} (\sigma^2)^{-\ell/2-1} \exp\left(-\frac{\lambda}{2\sigma^2}\right), \quad (\text{B.5})$$

the so-called *inverse gamma* ( $\mathcal{IG}$ ) distribution<sup>1</sup>, with support on  $(0, \infty)$ , mean  $\frac{2\lambda}{\ell-1}$  and variance increasing with  $\lambda$  and decreasing with  $\ell$ .

The inverse gamma distribution can now be combined with the Gaussian distribution into the normal inverse gamma distribution ( $\mathcal{NIG}$ ) in the following way:

$$\begin{aligned} \mathcal{NIG}(\mu, \sigma^2; m, v, \lambda, \ell) &\triangleq \mathcal{N}(\mu; m, v\sigma^2) \mathcal{IG}(\sigma^2; \lambda, \ell) \propto \\ &\propto (\sigma^2)^{-\ell/2-3/2} \exp\left(-\frac{\frac{1}{v}(\mu-m)^2 + \lambda}{2\sigma^2}\right) \end{aligned} \quad (\text{B.6})$$

Note that this is a hierarchical construction on the form  $p(\mu, \sigma^2) = p(\mu | \sigma^2)p(\sigma^2)$ , and *not* the independent form  $p(\mu, \sigma^2) = p(\mu)p(\sigma^2)$ . If we again assume the observations  $y_{1:T}$  are exchangeable and observed as  $p(y_t | \theta) = \mathcal{N}(y_t; \mu, \sigma^2)$ , now with both mean and variance unknown, the posterior becomes  $p(\theta | y) = \mathcal{NIG}\left(\mu, \sigma^2; \frac{m/v + \sum_t y_t}{1/v+T}, \frac{1}{1/v+T}, \lambda + \sum_t y_t^2 + m^2/v - \frac{(m/v + \sum_t y_t)^2}{1/v+T}, \ell + T\right)$ . That is, the posterior is just another normal inverse gamma distribution, which indeed is the conjugate prior to  $\mathcal{N}(y_t; \mu, \sigma^2)$  with unknown mean and variance.

### B.1.2 Generalizing to the matrix case: $\mathcal{MN}\mathcal{IW}$

The generalization of the univariate Gaussian distribution to the multivariate Gaussian distribution is well established. The generalization to the matrix case is, however, perhaps less so. Following Dawid (1981), we introduce the matrix normal ( $\mathcal{MN}$ ) distribution as follows: If the random  $k \times p$  matrix  $Z$  has independent standard Gaussian entries, we write  $p(Z) = \mathcal{MN}(Z; 0, I_k, I_p)$ . If, more generally, the rows of  $Z$  are independent, and each column has a multivariate Gaussian  $\mathcal{N}(0, V)$  distribution ( $V$  is  $p \times p$ ), we write  $p(Z) = \mathcal{MN}(Z; 0, I_k, V)$ . Similarly, we write  $p(Z) = \mathcal{MN}(Z; 0, U, I_p)$  if each column of  $Z$  is independent, and each row has a multivariate Gaussian  $\mathcal{N}(0, U)$  distribution ( $U$  is  $k \times k$ ).

In the most general form, we may say that if all elements  $z_{i,j}$  of the  $k \times p$  random matrix  $Z$  have a jointly Gaussian distribution, element  $z_{i,j}$  has the marginal distribution  $p(z_{i,j}) = \mathcal{N}(z_{i,j}; m_{i,j}, u_{i,i} \cdot v_{j,j})$ , and the covariance between  $z_{i,j}$  and  $z_{m,T}$  is  $\text{cov}[z_{i,j}, z_{m,T}] = u_{i,m} \cdot v_{j,T}$ , then the distribution of  $Z$  is  $p(Z) = \mathcal{MN}(Z; M, U, V)$ . We may write its density as

$$\mathcal{MN}(Z; M, U, V) = (2\pi v)^{-kp/2} |U|^{-p/2} |V|^{-k/2} \exp\left(-\frac{1}{2} \text{tr}\left((A-M)^\top U^{-1} (A-M) V^{-1}\right)\right). \quad (\text{B.7})$$

<sup>1</sup>Note that this is not the most common parameterization of the inverse gamma distribution.

Analogously to the gamma  $\mathcal{G}(1, \ell)$  distribution, we can construct the Wishart distribution  $\mathcal{W}(I_k, \ell)$  (named after John Wishart 1928) as follows: Let  $Z$  be distributed as  $p(Z) = \mathcal{MN}(Z; 0, I_k, I_\ell)$ . Then  $ZZ^\top$  is distributed as  $\mathcal{W}(I_k, \ell)$ . As in the scalar case, we can generalize to non-integer  $\ell$ , introduce a scale parameter (in the matrix case, a  $k \times k$  symmetric positive definite matrix  $\Lambda$ ) and consider the inverse  $(ZZ^\top)^{-1}$  (which exists with probability 1 if  $\ell > k - 1$ ), yielding the inverse Wishart distribution with density

$$\mathcal{IW}(\Sigma; \Lambda, \ell) = \frac{|\Lambda|^{\ell/2}}{2^{\ell/2} \Gamma_k\left(\frac{\ell}{2}\right)} |\Sigma|^{-\frac{\ell+k+1}{2}} \exp\left(-\frac{1}{2} \text{tr}(\Lambda \Sigma^{-1})\right) \quad (\text{B.8})$$

if  $\Sigma$  is symmetric positive definite, and  $\Gamma_k(\cdot)$  is the multivariate gamma function.  $\mathcal{IW}(\Sigma; \Lambda, \ell)$  has a mean  $\Lambda/(\ell - k - 1)$  (for  $\ell > k - 1$ ) and a variance increasing (element-wise) with  $\Lambda$  and decreasing with  $\ell$  (e.g., Rosen 1988). The diagonal elements of  $\Sigma$  are distributed as inverse gamma (e.g., Theorem 5.2.1 in Press 1982).

Following the scalar case, we construct the  $\mathcal{MN}\mathcal{IW}$  distribution as

$$\begin{aligned} \mathcal{MN}\mathcal{IW}(A; \Sigma; M, V, \Lambda, \ell) &\triangleq \mathcal{MN}(A; M, \Sigma, V) \mathcal{IW}(\Sigma; \Lambda, \ell) \propto \\ &\propto |\Sigma|^{-(\ell+p)/2-1} \exp\left(-\frac{1}{2} \text{tr}\left(\Sigma^{-1} \left((A - M)V^{-1}(A - M)^\top + \Lambda\right)\right)\right). \end{aligned} \quad (\text{B.9})$$

The special case  $p = 1$ , when  $\mathcal{MN}(M, \Sigma, 1) = \mathcal{N}(M, \Sigma)$  is often referred to as the normal inverse Wishart distribution, the conjugate prior<sup>2</sup> for the case when observing vector-valued data  $p(y_t | \theta) = \mathcal{N}(y_t; \mu, \Sigma)$  (e.g., Gelman et al. 2014, Section 3.6).

---

<sup>2</sup>The inverse Wishart is indeed the conjugate prior, but whether it is a sensible choice of prior is subject to debate, e.g. Alvarez et al. (2014) and Yang and J. O. Berger (1994) and references therein.

## B.2 Scalar linear regression: $y_t = ax_t + e_t$

We now consider the problem of scalar linear regression with  $T$  exchangeable observations; i.e.,  $y_t = ax_t + e_t$ ,  $e_t \sim \mathcal{N}(0, \sigma^2)$  and  $x_t$  is known. That is, we have the model  $p(y_{1:T} | a, \sigma^2) = \prod_{t=1}^T \mathcal{N}(y_t; ax_t, \sigma^2)$ . We want to infer  $a \in \mathbb{R}$  and  $\sigma^2 \in \mathbb{R}^+$  with the Bayesian approach, and assume a normal inverse gamma (B.6) prior  $\mathcal{NIG}(a, \sigma^2; m, v, \lambda, \ell)$ . This yields the posterior

$$\begin{aligned}
 p(a, \sigma^2) &\propto \mathcal{NIG}(a, \sigma^2; m, v, \lambda, \ell) \cdot \prod_{t=1}^T \mathcal{N}(y_t; ax_t, \sigma^2) \propto \\
 &\propto (\sigma^2)^{-1/2-3/2-T/2} \exp\left(-\frac{\frac{1}{v}(a-m)^2 + \lambda + \sum_{t=1}^T (y_t - ax_t)^2}{2\sigma^2}\right) = \\
 &\left/ \frac{1}{v}(a-m)^2 + \lambda + \sum_t (y_t - ax_t)^2 = \frac{1}{v}(a^2 - 2am + m^2) + \lambda + \sum_t y_t^2 - 2a \sum_t y_t x_t + a^2 \sum_t x_t^2 = \right. \\
 &\left. \left( \frac{1}{v} + \sum_t x_t^2 \right) \left( a - \frac{m/v + \sum_t y_t x_t}{1/v + \sum_t x_t^2} \right)^2 + \lambda + \sum_t y_t^2 + \frac{m^2}{v} - \frac{(m/v + \sum_t x_t y_t)^2}{\sum_t x_t^2 + 1/v} \right) \\
 &= (\sigma^2)^{-(\ell+T)/2-3/2} \exp\left(-\frac{\left( \frac{1}{v} + \sum_t x_t^2 \right) \left( a - \frac{m/v + \sum_t y_t x_t}{1/v + \sum_t x_t^2} \right)^2 + \lambda + \sum_t y_t^2 + \frac{m^2}{v} - \frac{(m/v + \sum_t x_t y_t)^2}{\sum_t x_t^2 + 1/v}}{2\sigma^2}\right) \propto \\
 &\propto / \text{cf. (B.6)} / \propto \mathcal{NIG}\left(a, \sigma^2; \bar{m}, \bar{v}, \bar{\lambda}, \bar{\ell}\right) \quad (\text{B.10})
 \end{aligned}$$

with

$$\bar{m} = \frac{m/v + \sum_t y_t x_t}{1/v + \sum_t x_t^2}, \quad (\text{B.11a})$$

$$\frac{1}{\bar{v}} = \frac{1}{v} + \sum_t x_t^2, \quad (\text{B.11b})$$

$$\bar{\lambda} = \lambda + \sum_t y_t^2 + \frac{m^2}{v} - \frac{(m/v + \sum_t x_t y_t)^2}{1/v + \sum_t x_t^2}, \quad (\text{B.11c})$$

$$\bar{\ell} = \ell + T. \quad (\text{B.11d})$$

### B.3 Multivariable linear regression: $y_t = Ax_t + e_t$

We now consider the matrix case, where we observe  $T$  exchangeable observations

$$\underbrace{\begin{bmatrix} y_t \\ \vdots \\ y_t \end{bmatrix}}_{k \times 1} = \underbrace{\begin{bmatrix} & & \\ & A & \\ & & \end{bmatrix}}_{k \times p} \underbrace{\begin{bmatrix} x_t \\ \vdots \\ x_t \end{bmatrix}}_{p \times 1} + \underbrace{\begin{bmatrix} e_t \\ \vdots \\ e_t \end{bmatrix}}_{k \times 1}, \quad e_t \sim \mathcal{N}(0, \Sigma), \quad (\text{B.12})$$

with known  $x_t$ . The data likelihood is given by

$$\begin{aligned} p(y_{1:T} | A, \Sigma) &= \prod_{t=1}^T \mathcal{N}(y_t; Ax_t, \Sigma) = \\ &= \prod_{t=1}^T |\Sigma|^{-k/2} \exp\left(-\frac{1}{2}(y_t - Ax_t)^\top \Sigma^{-1} (y_t - Ax_t)\right) \end{aligned} \quad (\text{B.13})$$

We want to infer  $A \in \mathbb{R}^{k \times p}$  and the  $k \times k$  a covariance matrix  $\Sigma$ , in a Bayesian fashion. As a prior, we assume  $\mathcal{MNITW}(A, \Sigma; M, V, \Lambda, \ell)$  (B.9). This gives the posterior

$$\begin{aligned} p(A, \Sigma | y_{1:T}) &\propto \mathcal{MNITW}(A, \Sigma; M, V, \Lambda, \ell) \cdot \prod_{t=1}^T \mathcal{N}(y_t; Ax_t, \Sigma) \propto \\ &\propto |\Sigma|^{-(\ell+p+kn)/2-1} \exp\left(-\frac{1}{2} \text{tr}\left(\Sigma^{-1} \left((A-M)V^{-1}(A-M)^\top + \Lambda + \sum_{t=1}^T (y_t - Ax_t)(y_t - Ax_t)^\top\right)\right)\right) = \\ &= \int \underbrace{\left( (A-M)V^{-1}(A-M)^\top + \Lambda + \sum_{t=1}^T (y_t - Ax_t)(y_t - Ax_t)^\top \right)}_{(\star)} = \\ &= \underbrace{\left[ A \cdot \left( MV^{-1} + \sum_t y_t x_t^\top \right) \left( V^{-1} + \sum_t x_t x_t^\top \right)^{-1} \right] \left( V^{-1} + \sum_t x_t x_t^\top \right) \left[ A \cdot \left( MV^{-1} + \sum_t y_t x_t^\top \right) \left( V^{-1} + \sum_t x_t x_t^\top \right)^{-1} \right]^\top}_{(\star\star)} + \\ &\quad + \Lambda + \sum_t y_t y_t^\top + MV^{-1}M^\top - \underbrace{\left( MV^{-1} + \sum_t y_t x_t^\top \right) \left( V^{-1} + \sum_t x_t x_t^\top \right)^{-1} \left( MV^{-1} + \sum_t y_t x_t^\top \right)^\top}_{(\star\star)} \Bigg| = \\ &= |\Sigma|^{-(\ell+p+kn)/2-1} \exp\left(-\frac{1}{2} \text{tr}((\star) + (\star\star))\right) \propto / \text{cf. (B.9)} / \propto \\ &\propto \mathcal{NIW}\left(a, \Sigma; \bar{M}, \bar{V}, \bar{\Lambda}, \bar{\ell}\right) \end{aligned} \quad (\text{B.14})$$

with

$$\bar{M} = \left( MV^{-1} + \sum_t y_t x_t^\top \right) \left( V^{-1} + \sum_t x_t x_t^\top \right)^{-1}, \quad (\text{B.15a})$$

$$\bar{V}^{-1} = V^{-1} + \sum_t x_t x_t^\top, \quad (\text{B.15b})$$

$$\bar{\Lambda} = (\star\star), \quad (\text{B.15c})$$

$$\bar{\ell} = \ell + kT. \quad (\text{B.15d})$$

“We could, of course, use any notation we want.”  
Richard Feynman

# Notation list

The notation used in the introductory chapters is summarized below. The notation used in the papers is introduced separately in each paper.

Symbol	Meaning
<i>General</i>	
$a$	A scalar or vector
$A$	A matrix or a set
$\mathbb{I}_A(\theta)$	Indicator function: 1 if $\theta \in A$ , 0 otherwise
$\setminus$	Relative complement
$\mathbb{R}$	The set of real numbers
$\  \cdot \ $	The Euclidean distance
$\Gamma(\cdot)$	Gamma function
$K_\nu(\cdot)$	Modified Bessel function (Rasmussen and Williams 2006, p. 84)
$p$	Probability density or mass
$\mathbb{P}$	Probability
$\mathbb{E}[\cdot]$	The expected value of the argument
$\mathcal{N}(\cdot; \mu, \sigma^2)$	The density for a univariate Gaussian distribution with mean $\mu$ and variance $\sigma^2$ .
$\mathcal{N}(\cdot; \mu, \Sigma)$	The density for a multivariate Gaussian distribution with mean $\mu$ and covariance matrix $\Sigma$ .
$\xrightarrow{d}$	Convergence in distribution
<i>Data, models and learning</i>	
$y$	Data
$y_t$	The data sample with index $t$
$T$	The number of data samples
$y_{1:T}$	$\{y_t\}_{t=1}^T$
$n_y$	The dimension of one data sample
$\theta$	Parameters in a model
$\eta$	Hyperparameters in a model
$p(\theta)$	Prior distribution for $\theta$
$p(\theta   y)$	Posterior distribution for $\theta$
$p(y   \theta)$	Density for $y$ given $\theta$
$\mathcal{L}(\theta)$	Likelihood function for $\theta$ (2.2)
$\hat{\theta}$	Point estimate of $\theta$

## NOTATION LIST

### *State-space models*

$x_t$	The state (at time $t$ ) in a state-space model
$n_x$	The dimension of the state in a state-space model
$n_u$	The dimension of the input to a state-space model
$f(\cdot   \cdot)$	The state transition function in a state-space model
$g(\cdot   \cdot)$	The observation function in a state-space model
$\vartheta$	The parameters in a state-space model

### *Gaussian processes*

$x^\star$	The points where the value of the Gaussian process is predicted
$x^d$	The points where the Gaussian process has been observed
$\mu(\cdot)$	The mean function
$\kappa(\cdot, \cdot)$	The covariance function
$\varepsilon$	Observation noise
$K^{\star\star}, K^{\star d}, K^{d\star}, K^{dd}$	Shorthand notation for $\kappa$ evaluated in certain points; see definitions on page 30

### *Monte Carlo*

$N$	The number of particles in a general particle approximation
$w^{(i)}$	The weight of particle $i$ in a weighted particle approximation
$N_x$	The number of particles in the particle filter, Algorithm 1 in Chapter 5
$K$	The number of iterations of the MCMC sampler, Algorithm 2 in Chapter 5
$\mathcal{K}(\cdot   \cdot)$	The transition kernel in the MCMC sampler, Algorithm 2 in Chapter 5
$N_\theta$	The number of particles in the SMC sampler, Algorithm 5 in Chapter 5
$P$	The number of iterations of the SMC sampler, Algorithm 5 in Chapter 5

“Citing an author whose ideas or information you used is paying a debt.”

Umberto Eco

## References

- Hirotsugu Akaike (1974). “A new look at the statistical model identification”. In: *IEEE Transactions on Automatic Control* 19.6, pp. 716–723.
- Ignacio Alvarez, Jarad Niemi, and Matt Simpson (2014). “Bayesian inference for a covariance matrix”. In: *Proceedings of the 26<sup>th</sup> Annual Conference on Applied Statistics in Agriculture*. Manhattan, KS, USA, pp. 71–82.
- Christophe Andrieu, Arnaud Doucet, and Roman Holenstein (2010). “Particle Markov chain Monte Carlo methods”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 72.3, pp. 269–342.
- Christophe Andrieu, Nando de Freitas, Arnaud Doucet, and Michael I. Jordan (2003). “An introduction to MCMC for machine learning”. In: *Machine Learning* 50.1, pp. 5–43.
- Christophe Andrieu and Gareth O. Roberts (2009). “The pseudo-marginal approach for efficient Monte Carlo computations”. In: *Annals of Statistics* 37.2, pp. 967–725.
- M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp (2002). “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking”. In: *IEEE Transactions on Signal Processing* 50.2, pp. 174–188.
- Thomas Bayes (1763). “An essay towards solving a problem in the doctrine of chances”. In: *Philosophical Transactions (1683-1775)* 53, pp. 370–418.
- James O. Berger (1985). *Statistical decision theory and Bayesian analysis*. 2nd ed. New York, NY, USA: Springer.
- James O. Berger (2006). “The case for objective Bayesian analysis”. In: *Bayesian Analysis* 1.3, pp. 385–402.
- Christopher M. Bishop (2006). *Pattern recognition and machine learning*. New York, NY, USA: Springer.
- David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe (2016). “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112 (518), pp. 859–877.
- Alexandre Bouchard-Côté, Sebastian J. Vollmer, and Arnaud Doucet (2017). “The Bouncy Particle Sampler: A non-reversible rejection-free Markov chain Monte Carlo method”. In: *arXiv:1510.02451*.
- Olivier Cappé, Éric Moulines, and Tobias Rydén (2005). *Inference in hidden Markov models*. Springer Series in Statistics. New York, NY, USA: Springer.
- George Casella and Roger L. Berger (2002). *Statistical inference*. 2nd ed. Pacific Grove, CA, USA: Duxbury.
- Kathryn Chaloner and Isabella Verdinelli (1995). “Bayesian experimental design: a review”. In: *Statistical Science* 10.3, pp. 273–304.
- Krzysztof Chalupka, Christopher K. I. Williams, and Iain Murray (2013). “A framework for evaluating approximation methods for Gaussian process regression”. In: *The Journal of Machine Learning Research (JMLR)* 14.2, pp. 333–350.

## REFERENCES

- Tianshi Chen, Henrik Ohlsson, and Lennart Ljung (2012). “On the estimation of transfer functions, regularizations and Gaussian processes—Revisited”. In: *Automatica* 48.8, pp. 1525–1535.
- Nicolas Chopin (2004). “Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference”. In: *Annals of Statistics* 36.6, pp. 2385–2411.
- Nicolas Chopin, Pierre E. Jacob, and Omiros Papaspiliopoulos (2013). “SMC<sup>2</sup>: an efficient algorithm for sequential analysis of state space models”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75.3, pp. 397–426.
- Nicolas Chopin, James Ridgway, Mathieu Gerber, and Omiros Papaspiliopoulos (2015). “Towards automatic calibration of the number of state particles within the SMC<sup>2</sup> algorithm”. In: *arXiv:1506.00570*.
- Johan Dahlin, Fredrik Lindsten, and Thomas B. Schön (2015). “Particle Metropolis-Hastings using gradient and Hessian information”. In: *Statistics and Computing* 25.1, pp. 81–92.
- Johan Dahlin and Thomas B. Schön (2016). “Getting started with particle Metropolis-Hastings for inference in nonlinear models”. In: *arXiv:1511.01707*.
- A. Philip Dawid (1981). “Some matrix-variate distribution theory: notational considerations and a Bayesian application”. In: *Biometrika* 68.1, pp. 265–274.
- Bruno de Finetti (1992). “Foresight: its logical laws, its subjective sources”. In: *Breakthroughs in Statistics: Foundations and Basic Theory*. Ed. by Samuel Kotz and L. Norman Johnson. Trans. by Henry E. Kyberg. Vol. 1. (Originally published in 1937 as “La prévision: ses lois logiques, ses sources subjectives” in *Annales de l’Institut Henri Poincaré* 7, pp. 1–68.) New York, NY, USA: Springer, pp. 134–174.
- Pierre Del Moral (2004). *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. New York, NY, US: Springer.
- Pierre Del Moral and Arnaud Doucet (2014). “Particle methods: an introduction with applications”. In: *ESAIM: Proceedings* 44.1, pp. 1–46.
- Pierre Del Moral, Arnaud Doucet, and Ajay Jasra (2006). “Sequential Monte Carlo samplers”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.3, pp. 411–436.
- Pierre Del Moral, Arnaud Doucet, and Ajay Jasra (2012a). “An adaptive sequential Monte Carlo method for approximate Bayesian computation”. In: *Statistics and Computing* 22.5, pp. 1009–1020.
- Pierre Del Moral, Arnaud Doucet, and Ajay Jasra (2012b). “On adaptive resampling strategies for sequential Monte Carlo methods”. In: *Bernoulli* 18.1, pp. 252–278.
- Bernard Delyon, Marc Lavielle, and Éric Moulines (1999). “Convergence of a stochastic approximation version of the EM algorithm”. In: *Annals of Statistics* 27.1, pp. 94–128.
- Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin (1977). “Maximum likelihood from incomplete data via the EM algorithm”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38.
- Randal Douc and Olivier Cappé (2005). “Comparison of resampling schemes for particle filtering”. In: *Proceedings of the 4<sup>th</sup> International Symposium on Image and Signal Processing and Analysis (ISPA)*. Zagreb, Croatia, pp. 64–69.

- Randal Douc, Éric Moulines, and Jimmy Olsson (2014). “Long-term stability of sequential Monte Carlo methods under verifiable conditions”. In: *Annals of Applied Probability* 24.5, pp. 1767–1802.
- Arnaud Doucet and Adam M. Johansen (2011). “A tutorial on particle filtering and smoothing: fifteen years later”. In: *Nonlinear Filtering Handbook*. Ed. by D. Crisan and B. Rozovsky. Oxford, UK: Oxford University Press, pp. 656–704.
- Simon Duane, A. D. Kennedy, Brian J. Pendleton, and Duncan Roweth (1987). “Hybrid Monte Carlo”. In: *Physics Letter B* 195.2, pp. 216–222.
- David Duvenaud, James Lloyd, Roger Grosse, Joshua Tenenbaum, and Zoubin Ghahramani (2013). “Structure discovery in nonparametric regression through compositional kernel search”. In: *Proceedings of the 30<sup>th</sup> International Conference on Machine Learning (ICML)*. Atlanta, GA, USA, pp. 1166–1174.
- David Duvenaud, Dougal Maclaurin, and Ryan Adams (2016). “Early stopping as nonparametric variational inference”. In: *Proceedings of the 19<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS)*. Cádiz, Spain, pp. 1070–1077.
- David A. van Dyk and Xiyun Jiao (2014). “Metropolis-Hastings within partially collapsed Gibbs samplers”. In: *Journal of Computational and Graphical Statistics* 24.2, pp. 301–327.
- Bradley Efron (1979). “Bootstrap methods: another look at the jackknife”. In: *Annals of Statistics* 7.1, pp. 1–26.
- Bradley Efron (1986). “Why isn’t everyone a Bayesian?” In: *The American Statistician* 40.1. Including discussion by H. Chernoff, D. V. Lindley, C.N. Morris, S. J. Press and A. F. M. Smith, pp. 1–5.
- Bradley Efron (2013). “A 250-year argument: belief, behavior, and the bootstrap”. In: *Bulletin of the American Mathematical Society* 50.1, pp. 129–146.
- Bradley Efron and Trevor Hastie (2016). *Computer age statistical inference*. Cambridge, UK: Cambridge University Press.
- Yonina C. Eldar and Gitta Kutyniok, eds. (2012). *Compressed sensing: theory and applications*. Cambridge, UK: Cambridge University Press.
- Roger Frigola-Alcade (2015). “Bayesian time series learning with Gaussian processes”. PhD thesis. UK: University of Cambridge.
- Roger Frigola, Yutian Chen, and Carl Rasmussen (2014). “Variational Gaussian process state-space models”. In: *Advances in Neural Information Processing Systems 27 (NIPS)*. Montréal, QC, Canada, pp. 3680–3688.
- Roger Frigola, Fredrik Lindsten, Thomas B. Schön, and Carl Rasmussen (2013). “Bayesian inference and learning in Gaussian process state-space models with particle MCMC”. In: *Advances in Neural Information Processing Systems 26 (NIPS)*. Lake Tahoe, NV, USA, pp. 3156–3164.
- Andras Fulop and Junye Li (2013). “Efficient learning via simulation: a marginalized resample-move approach”. In: *Journal of Econometrics* 176.2, pp. 146–161.
- Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin (2014). *Bayesian data analysis*. 3rd ed. Boca Raton, FL, USA: Chapman & Hall/ CRC Press.
- Stuart Geman and Donald Geman (1984). “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6.6, pp. 721–741.

## REFERENCES

- Zoubin Ghahramani and Sam T. Roweis (1998). "Learning nonlinear dynamical systems using an EM algorithm". In: *Advances in Neural Information Processing Systems (NIPS) 11*. Denver, CO, USA, pp. 431–437.
- Neil J. Gordon, David J. Salmond, and Adrian F.M. Smith (1993). "Novel approach to nonlinear/non-Gaussian Bayesian state estimation". In: *IEE Proceedings F - Radar and Signal Processing*, pp. 107–113.
- Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund (2002). "Particle filters for positioning, navigation, and tracking". In: *IEEE Transactions on Signal Processing* 50.2, pp. 425–437.
- Trevor Hastie, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. 2nd ed. New York, NY, USA: Springer.
- Wilfred K. Hastings (1970). "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1.
- "Sequential state estimation: From Kalman filters to particle filters" (2004). In: *Proceedings of the IEEE* 92.3. Ed. by Simon Haykin and Nando de Freitas. Special issue.
- Markus Heinonen, Henrik Mannerström, Juho Rousu, Samuel Kaski, and Harri Lähdesmäki (2016). "Non-stationary Gaussian process regression with Hamiltonian Monte Carlo". In: *Proceedings of the 19<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS)*. Cádiz, Spain, pp. 737–740.
- Håkan Hjalmarsson (2009). "System identification of complex and structured systems". In: *European Journal of Control* 15.3–4, pp. 275–310.
- Arthur E. Hoerl and Robert W. Kennard (1970). "Ridge regression: biased estimation for nonorthogonal problems". In: *Technometrics* 42.1, pp. 80–86.
- Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B. Schön (2018). "Linearly constrained Gaussian processes". In: *Advances in Neural Information Processing Systems 30 (NIPS)*. Long Beach, CA, USA, pp. 217–224.
- Thomas Kailath (1980). *Linear systems*. Englewood Cliffs, NJ, USA: Prentice Hall.
- Rudolf E. Kálmán (1960). "A new approach to linear filtering and prediction problems". In: *Journal of Basic Engineering* 82.1, pp. 35–45.
- Augustine Kong, Jun S. Liu, and Wing Hung Wong (1994). "Sequential imputations and Bayesian missing data problems". In: *Journal of the American Statistical Association* 89.425, pp. 278–288.
- Estelle Kuhn and Marc Lavielle (2004). "Coupling a stochastic approximation version of EM with an MCMC procedure". In: *ESAIM: Probability and Statistics* 8, pp. 115–131.
- Pierre Simon de Laplace (1820). *Théorie analytique des probabilités*. 3rd ed. Paris, France: Mme Ve Courcier, imprimeur-libraire pour les mathématiques.
- Faming Liang, Chuanhai Liu, and Raymond Carroll (2010). *Advanced Markov chain Monte Carlo methods: learning from past samples*. West Sussex, United Kingdom: John Wiley & Sons.
- Dennis V. Lindley (1990). "The 1988 Wald memorial lectures: the present position in Bayesian statistics". In: *Statistical Science* 6.1, pp. 44–65.
- Fredrik Lindsten (2013). "An efficient stochastic approximation EM algorithm using conditional particle filters". In: *Proceedings of the 38<sup>th</sup> International Conference on*

- Acoustics, Speech, and Signal Processing (ICASSP)*. Vancouver, BC, Canada, pp. 6274–6278.
- Fredrik Lindsten, Michael I. Jordan, and Thomas B. Schön (2014). “Particle Gibbs with ancestor sampling”. In: *The Journal of Machine Learning Research (JMLR)* 15.1, pp. 2145–2184.
- Fredrik Lindsten and Thomas B. Schön (2013). “Backward simulation methods for Monte Carlo statistical inference”. In: *Foundations and Trends in Machine Learning* 6.1, pp. 1–143.
- Lennart Ljung (1999). *System identification: theory for the user*. 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall.
- Lennart Ljung and Torkel Glad (2004). *Modellbygge och simulering*. 2nd ed. Lund, Sweden: Studentlitteratur.
- Michael Lustig, David Donoho, and John M. Pauly (2007). “Sparse MRI: the application of compressed sensing for rapid MR imaging”. In: *Magnetic resonance in medicine* 58.6, pp. 1182–1195.
- David J. C. MacKay (1998). “Introduction to Gaussian processes”. In: *Neural Networks and Machine Learning*. Ed. by C. M. Bishop. Vol. 168. NATO ASI Series F: Computational and Systems Sciences. Berlin, Germany: Springer-Verlag, pp. 133–165.
- Bertil Matérn (1960). “Spatial Variation”. PhD thesis. Sweden: Statens skogsforskningsinstitut.
- Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller (1953). “Equation of state calculations by fast computing machines”. In: *Journal of Chemical Physics* 21.6, pp. 1087–1092.
- Nicholas Metropolis and Stanisław Ulam (1949). “The Monte Carlo method”. In: *Journal of the American Statistical Association* 44.247, pp. 335–341.
- Peter Müller (1991). *A generic approach to posterior intergration and Gibbs sampling*. Tech. rep. West Lafayette, IN, USA: Department of Statistics, Purdue University.
- Lawrence M. Murray, Anthony Lee, and Pierre E. Jacob (2015). “Parallel resampling in the particle filter”. In: *Journal of Computational and Graphical Statistics* 25.3, pp. 789–805.
- Lawrence M. Murray and Thomas B. Schön (2018). “Automated learning with a probabilistic programming language: Birch”. To be submitted.
- Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön (2014). “Sequential Monte Carlo for graphical models”. In: *Advances in Neural Information Processing Systems 27 (NIPS)*. Montréal, QC, Canada, pp. 1862–1870.
- Christian A. Naesseth, Fredrik Lindsten, and Thomas B. Schön (2015). “Nested sequential Monte Carlo methods”. In: *Proceedings of the 32<sup>nd</sup> International Conference on Machine Learning (ICML)*. Lille, France, pp. 1292–1301.
- Radford M. Neal (2003). “Slice sampling”. In: *Annals of Statistics* 31.3, pp. 705–767.
- Radford M. Neal (2011). “MCMC using Hamiltonian dynamics”. In: *Handbook of Markov Chain Monte Carlo*. Ed. by Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. Chapman & Hall/CRC Press.
- Brett Ninness and Soren Henriksen (2010). “Bayesian system identification via Markov chain Monte Carlo techniques”. In: *Automatica* 46.1, pp. 40–51.
- Jimmy Olsson, Olivier Cappé, Randal Douc, and Éric Moulines (2008). “Sequential Monte Carlo smoothing with application to parameter estimation in nonlinear state-space models”. In: *Bernoulli* 14.1, pp. 155–179.

- Brooks Paige, Frank Wood, Arnaud Doucet, and Yee Whye Teh (2014). “Asynchronous anytime sequential Monte Carlo”. In: *Advances in Neural Information Processing Systems 27 (NIPS)*. Montréal, QC, Canada, pp. 1–9.
- Václav Peterka (1981). “Bayesian system identification”. In: *Automatica* 17.1, pp. 41–53.
- E. A. J. Frank Peters and Gijsbertus de With (2012). “Rejection-free Monte Carlo sampling for general potentials”. In: *Physical Review E* 85.2, pp. 1–5.
- David L. Phillips (1962). “A technique for the numerical solution of certain integral equations of the first kind”. In: *Journal of the ACM* 9.1, pp. 84–97.
- Michael K. Pitt, Ralph dos Santos Silva, Paolo Giordani, and Robert Kohn (2012). “On some properties of Markov chain Monte Carlo simulation methods based on the particle filter”. In: *Journal of Econometrics* 171.2, pp. 134–151.
- S. James Press (1982). *Applied multivariate analysis: using Bayesian and frequentist methods of inference*. Malabar, FL, USA: Robert E. Krieger Publishing Company.
- Friedrich Pukelsheim (1993). *Optimal design of experiments*. New York, NY, USA: Wiley.
- José Mario Quintana (1987). “Multivariate Bayesian forecasting models”. PhD thesis. UK: University of Warwick.
- Carl E. Rasmussen and Christopher K. I. Williams (2006). *Gaussian processes for machine learning*. Cambridge, MA, USA: MIT Press.
- Rudolf Erich Raspe (1786). *Baron Munchausen’s narrative of his marvellous travels and campaigns in Russia*. Oxford, UK: Smith.
- Stephen L. Richter and Raymond A. DeCarlo (1983). “Continuation methods: theory and applications”. In: *IEEE Transactions on Circuits and Systems* 30.6, pp. 347–352.
- Herbert Robbins and Sutton Monro (1951). “A stochastic approximation method”. In: *The Annals of Mathematical Statistics* 22.3, pp. 400–407.
- Christian P. Robert and George Casella (2004). *Monte Carlo statistical methods*. 2nd ed. New York, NY, USA: Springer.
- Dietrich von Rosen (1988). “Moments for the inverted Wishart distribution”. In: *Scandinavian Journal of Statistics* 15.2, pp. 91–109.
- Wilson J. Rugh (1993). *Linear system theory*. Englewood Cliffs, NJ, USA: Prentice Hall.
- Simo Särkkä (2013). *Bayesian filtering and smoothing*. Cambridge, UK: Cambridge University Press.
- Mark J. Schervish (1995). *Theory of statistics*. New York, NY, USA: Springer.
- Thomas B. Schön and Fredrik Lindsten (2011). *Manipulating the multivariate Gaussian density*. Tech. rep. Linköping, Sweden: Division of Automatic Control, Linköping University.
- Thomas B. Schön, Andreas Svensson, Lawrence M. Murray, and Fredrik Lindsten (2018). “Probabilistic learning of nonlinear dynamical systems using sequential Monte Carlo”. In: *Mechanical Systems and Signal Processing* 104, pp. 866–883.
- Thomas B. Schön, Adrian Wills, and Brett Ninness (2011). “System identification of nonlinear state-space models”. In: *Automatica* 47.1, pp. 39–49.
- Gideon Schwarz (1978). “Estimating the dimension of a model”. In: *Annals of Statistics* 6.2, pp. 461–464.
- Amar Shah, Andrew Gordon Wilson, and Zoubin Ghahramani (2014). “Student- $t$  processes as alternatives to Gaussian processes”. In: *Proceedings of the 17<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS)*. Reykjavik, Iceland, pp. 877–885.

- Jonas Sjöberg and Lennart Ljung (1995). “Overtraining, regularization and searching for a minimum, with application to neural networks”. In: *International Journal of Control* 62.6, pp. 1391–1407.
- Edward Snelson (2007). “Flexible and efficient Gaussian process models for machine learning”. PhD thesis. UK: University College London.
- Torsten Söderström and Petre Stoica (1989). *System identification*. Hemel Hempstead, UK: Prentice-Hall, Inc.
- Arno Solin, Manon Kok, Niklas Wahlström, Thomas B. Schön, and Simo Särkkä (2018). “Modeling and interpolation of the ambient magnetic field by Gaussian processes”. In: *IEEE Transactions on Robotics*. Accepted for publication.
- Arno Solin and Simo Särkkä (2014). “Explicit link between periodic covariance functions and state space models”. In: *Proceedings of the 17<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS)*. Reykjavik, Iceland, pp. 904–912.
- Arno Solin and Simo Särkkä (2014). “Hilbert space methods for reduced-rank Gaussian process regression”. In: *arXiv:1401.5508*.
- Leland Stewart and Perry Jr. McCarty (1992). “Use of Bayesian belief networks to fuse continuous and discrete information for target recognition, tracking, and situation assessment”. In: *Proceedings of SPIE 1699, Signal Processing, Sensor Fusion, and Target Recognition*. Orlando, FL, USA, pp. 177–185.
- Stephen M. Stigler (1986). “Laplace’s 1774 memoir on inverse probability”. In: *Statistical Science* 1.3, pp. 359–378.
- Andreas Svensson (2013). *Particle Filter Explained without Equations*. URL: <https://www.youtube.com/watch?v=aUkBa1zMKv4>.
- Andreas Svensson and Thomas B. Schön (2016). *Comparing two recent particle filter implementations of Bayesian system identification*. Tech. rep. 2016-008. (Presented at Reglermöte 2016, Gothenburg, Sweden). Department of Information Technology, Uppsala University.
- Andreas Svensson, Thomas B. Schön, and Manon Kok (2015). “Nonlinear state space smoothing using the conditional particle filter”. In: *Proceedings of the 17<sup>th</sup> IFAC Symposium on System Identification (SYSID)*. Beijing, China, pp. 975–980.
- Andreas Svensson, Arno Solin, Simo Särkkä, and Thomas B. Schön (2016). “Computationally efficient Bayesian learning of Gaussian process state space models”. In: *Proceedings of the 19<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS)*. Cádiz, Spain, pp. 213–221.
- Robert Tibshirani (1996). “Regression shrinkage and selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 58.1, pp. 267–288.
- Luke Tierney (1994). “Markov chains for exploring posterior distributions”. In: *Annals of Statistics* 22.4, pp. 1701–1728.
- Michail K. Titsias and Lázaro-Gredilla (2014). “Doubly Stochastic Variational Bayes for non-Conjugate Inference”. In: *Proceedings of the 31<sup>st</sup> International Conference on Machine Learning (ICML)*. Beijing, China, pp. 1971–1979.
- Aad W. van der Vaart (1998). *Asymptotic Statistics*. Cambridge, UK: Cambridge University Press.
- Nick Whiteley (2013). “Stability properties of some particle filters”. In: *The Annals of Applied Probability* 23.6, pp. 2500–2537.

- Adrian Wills, Thomas B. Schön, Fredrik Lindsten, and Brett Ninness (2012). "Estimation of linear systems using a Gibbs sampler". In: *Proceedings of the 16<sup>th</sup> IFAC Symposium on System Identification (SYSID)*. Brussels, Belgium, pp. 203–208.
- John Wishart (1928). "The generalised product moment distribution in samples from a normal multivariate population". In: *Biometrika* 20A.1/2, pp. 32–52.
- John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma (2009). "Robust face recognition via sparse representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.2, pp. 210–227.
- Ruoyong Yang and James O. Berger (1994). "Estimation of a covariance matrix using the reference prior". In: *Annals of Statistics* 22.3, pp. 1195–1211.
- Hui Zou and Trevor Hastie (2005). "Regularization and variable selection via the elastic net". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 67.2, pp. 301–320.



# Acta Universitatis Upsaliensis

*Digital Comprehensive Summaries of Uppsala Dissertations  
from the Faculty of Science and Technology 1709*

Editor: The Dean of the Faculty of Science and Technology

A doctoral dissertation from the Faculty of Science and Technology, Uppsala University, is usually a summary of a number of papers. A few copies of the complete dissertation are kept at major Swedish research libraries, while the summary alone is distributed internationally through the series Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology. (Prior to January, 2005, the series was published under the title “Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology”.)

Distribution: [publications.uu.se](http://publications.uu.se)  
urn:nbn:se:uu:diva-357611



ACTA  
UNIVERSITATIS  
UPSALIENSIS  
UPPSALA  
2018