Thesis for the degree of Doctor of Philosophy

Probabilistic Sequence Models with Speech and Language Applications

Gustav Eje Henter

KTH Electrical Engineering

Communication Theory
School of Electrical Engineering
KTH – Royal Institute of Technology

Stockholm 2013

Henter, Gustav Eje
Probabilistic Sequence Models with Speech and Language Applications

ISBN 978-91-7501-932-1
TRITA-EE 2013:042
ISSN 1653-5146

Communication Theory
School of Electrical Engineering
KTH – Royal Institute of Technology
SE-100 44 Stockholm, Sweden

Abstract

Series data, sequences of measured values, are ubiquitous. Whenever observations are made along a path in space or time, a data sequence results. To comprehend nature and shape it to our will, or to make informed decisions based on what we know, we need methods to make sense of such data. Of particular interest are probabilistic descriptions, which enable us to represent uncertainty and random variation inherent to the world around us.

This thesis presents and expands upon some tools for creating probabilistic models of sequences, with an eye towards applications involving speech and language. Modelling speech and language is not only of use for creating listening, reading, talking, and writing machines—for instance allowing human-friendly interfaces to future computational intelligences and smart devices of today—but probabilistic models may also ultimately tell us something about ourselves and the world we occupy.

The central theme of the thesis is the creation of new or improved models more appropriate for our intended applications, by weakening limiting and questionable assumptions made by standard modelling techniques. One contribution of this thesis examines causal-state splitting reconstruction (CSSR), an algorithm for learning discrete-valued sequence models whose states are minimal sufficient statistics for prediction. Unlike many traditional techniques, CSSR does not require the number of process states to be specified a priori, but builds a pattern vocabulary from data alone, making it applicable for language acquisition and the identification of stochastic grammars. A paper in the thesis shows that CSSR handles noise and errors expected in natural data poorly, but that the learner can be extended in a simple manner to yield more robust and stable results also in the presence of corruptions.

Even when the complexities of language are put aside, challenges remain. The seemingly simple task of accurately describing human speech signals, so that natural synthetic speech can be generated, has proved difficult, as humans are highly attuned to what speech should sound like. A pair of papers in the thesis therefore study nonparametric techniques suitable for improved acoustic modelling of speech for synthesis applications. Each of the two papers targets a known-incorrect assumption of established

methods, based on the hypothesis that nonparametric techniques can better represent and recreate essential characteristics of natural speech.

In the first paper of the pair, Gaussian process dynamical models (GPDMs), nonlinear, continuous state-space dynamical models based on Gaussian processes, are shown to better replicate voiced speech, without traditional dynamical features or assumptions that cepstral parameters follow linear autoregressive processes. Additional dimensions of the state-space are able to represent other salient signal aspects such as prosodic variation. The second paper, meanwhile, introduces KDE-HMMs, asymptotically-consistent Markov models for continuous-valued data based on kernel density estimation, that additionally have been extended with a fixed-cardinality discrete hidden state. This construction is shown to provide improved probabilistic descriptions of nonlinear time series, compared to reference models from different paradigms. The hidden state can be used to control process output, making KDE-HMMs compelling as a probabilistic alternative to hybrid speech-synthesis approaches.

A final paper of the thesis discusses how models can be improved even when one is restricted to a fundamentally imperfect model class. Minimum entropy rate simplification (MERS), an information-theoretic scheme for postprocessing models for generative applications involving both speech and text, is introduced. MERS reduces the entropy rate of a model while remaining as close as possible to the starting model. This is shown to produce simplified models that concentrate on the most common and characteristic behaviours, and provides a continuum of simplifications between the original model and zero-entropy, completely predictable output. As the tails of fitted distributions may be inflated by noise or empirical variability that a model has failed to capture, MERS's ability to concentrate on high-probability output is also demonstrated to be useful for denoising models trained on disturbed data.

Keywords: Time series, acoustic modelling, speech synthesis, stochastic processes, causal-state splitting reconstruction, robust causal states, pattern discovery, Markov models, HMMs, nonparametric models, Gaussian processes, Gaussian process dynamical models, nonlinear Kalman filters, information theory, minimum entropy rate simplification, kernel density estimation, time-series bootstrap.

List of Papers

The thesis is based on the following papers:

[A] G. E. Henter, M. R. Frean, and W. B. Kleijn, "Gaussian Process Dynamical Models for Nonparametric Speech Representation and Synthesis," in *Proc. ICASSP*, 2012, pp. 4505–4508.

[B] G. E. Henter and W. B. Kleijn, "Picking Up the Pieces: Causal States in Noisy Data, and How to Recover Them," *Pattern Recogn. Lett.*, vol. 34, no. 5, pp. 587–594, 2013.

[C] G. E. Henter and W. B. Kleijn, "Minimum Entropy Rate Simplification of Stochastic Processes," *IEEE T. Pattern Anal.*, submitted.

[D] G. E. Henter, A. Leijon, and W. B. Kleijn, "Kernel Density Estimation-Based Markov Models with Hidden State," manuscript in preparation.

In addition to papers A–D, the following papers have also been produced in part by the author of the thesis:

[E] G. E. Henter and W. B. Kleijn, "Simplified Probability Models for Generative Tasks: A Rate-Distortion Approach," in *Proc. EUSIPCO*, vol. 18, 2010, pp. 1159–1163.

[F] G. E. Henter and W. B. Kleijn, "Intermediate-State HMMs to Capture Continuously-Changing Signal Features," in *Proc. Interspeech*, vol. 12, 2011, pp. 1817–1820.

[G] P. N. Petkov, W. B. Kleijn, and G. E. Henter, "Enhancing Subjective Speech Intelligibility Using a Statistical Model of Speech," in *Proc. Interspeech*, vol. 13, 2012, pp. 166–169.

[H] P. N. Petkov, G. E. Henter, and W. B. Kleijn, "Maximizing Phoneme Recognition Accuracy for Enhanced Speech Intelligibility in Noise," *IEEE T. Audio Speech*, vol. 21, no. 5, pp. 1035–1045, 2013.

Acknowledgements

While this thesis may have just one name printed on the front page, it owes its existence to the input and support of many people, and I am grateful for each and every one. First and foremost, I must give thanks to my two scientific guiding stars at SIP who offered me the opportunity to pursue a PhD, and first among them my principal supervisor, supermind Professor Bastiaan Kleijn: Your scientific creativity is immense and inspiring, your considerable experience is dispensed in lucid and compact pearls of wisdom, your formidable skill in debate has helped hone mine, and your editing capabilities and sense for greatness in scientific presentation are second to none, with our papers being much stronger as a result. Thank you for your immeasurable teachings, and for developing me from a student into a researcher. I hope you can see hints of yourself reflected back in this thesis.

Next, I am deeply grateful to my second supervisor, Professor Arne Leijon: Your abilities as a scientist and engineer, your didactic skills, your approachability, and your dependability are as great as your humility. Thank you for turning me from a student into a teacher. It has been a delight to work with you on the Pattern Recognition course, and I value your opinion immensely, whether in science or outside of it.

Beside my two supervisors, I am also indebted to many other seniors in the department, including Professor Markus Flierl, Doctor Saikat Chatterje, and Professor Mikael Skoglund—not only directly, but also for their work in keeping electrical engineering at KTH a thriving research environment. On that note, I also must thank Dora Söderberg for her happy assistance with administrative matters. Additionally, I am grateful to Professor Ragnar Thobaben for helpfully performing quality review of the thesis.

Next I must thank my coworkers at SIP, and lately Communication Theory, who have made my years as a PhD student rewarding both scientifically and socially. Aside from Ermin, Konrad, Pravin, Zhanyu, and Sebastian, who I've had the pleasure of sharing an office with, special thanks go to all who have worked on their sound and imaging PhDs alongside me, including, alphabetically, Anders, Chris, David Z, Du, Guoqiang, Haopeng, Jalil, Jan, Janusz, Minyue, Nasser, Obada, Petko (also a scientific collaborator), and Svante. Among the post-docs, Cees, Hannes, and Timo also come to mind. Our many interesting discussions at lunch, around the fika table, at

seminars, and on many other occasions will remain a fond memory of mine.

Speaking of interesting discussions, I would be remiss if I did not mention my MSc student Andreas and the many inquisitive, ambitious, and creative students that I've interacted with in the Pattern Recognition course. What I learned through you in my teaching efforts has sharpened my scientific understanding and contributed tangibly to my research.

Outside the confines of KTH, I would like to express my gratitude to those who helped make my visit to Wellington enjoyable, including Bastiaan, officemate Yusuke, scientific collaborator Marcus, as well as Kristi and Daniel D. Thanks also go to the scientists in other countries whom I've have the pleasure of working with on the ACORNS and LISTA projects, with a particular mention going to Simon and Cassie at the University of Edinburgh, who generously have decided to provide me with employment for the coming year. I also thank Matt in Cambridge for thoughtful e-mail conversations.

Leaving the scientific domain, I wish to extend many warm thanks to my friends and peers, both in Sweden and across the globe, who have helped keep my life fun and interesting during the PhD. This includes Leigh, Leon, and Michael in the Pacific Northwest; Daniel P and Richard in California; Natalie in Quebec; Brian in Oklahoma; the eminent Douglas and the crew on the East Coast; Danny in France; Philip in Germany; Donna in Denmark; and Eivind in Norway.

In my home base of Sweden I am grateful to a multitude of friends hailing from different circles and eras. This includes peers from my education, with a particular nod to the class at Höglandsskolan, including Gabriel, Mattias, Erik, Rikard, Gustav N, Ulf, Tobias, Björn, Johanna, Jonas, and Clara, studymates Fritjof, Eric, and Torbjörn from KTH, as well as more recent good friends such as Joel, Patrik, Agata, Antonio, and David B, to name just a few. I hope I can deepen these friendships and continue to add new ones in years to come.

I would not be who I am today if not for my educators in school and at university. Suffice it to say that, among many, I am particularly grateful to Agneta, Katarina, and Leena for their positive and enduring influence on my life. I also have had the fortune of growing up in the context of a large and stable extended family, with grandparents Ave, Maud, Berit, and Edgar who always have had a keen interest in what I've been doing.

Having peeled away many layers of close friends and acquaintances, we get to the core of being. While for the scientific content of this dissertation I am most indebted to my supervisors, the existence of the thesis is at least as attributable to four people who I can always count on both in times of joy and in times need. These are Gabriel, my friend for more than half my life, my father Jan-Inge and my mother Maria—givers of life and constant sources of wisdom, love, and support from childhood until the present day—and my brother and best friend Viking. Thank you for believing in me even

when I did not. Without your encouragement, this thesis would not have come to pass.

Gustav Eje Henter
Edinburgh, October 2013

Contents

Acronyms and Abbreviations

ACORNS	Acquisition of Communication and Recognition Skills
AD-converter	Analog-to-Digital converter
AIC	Akaike Information Criterion
AMISE	Asymptotic Mean Integrated Squared Error
AR	Autoregressive
AR-HMM	Autoregressive Hidden Markov Model
ARMA	Autoregressive Moving Average
ASR	Automatic Speech Recognition
BIC	Bayesian Information Criterion
CCSA	Clustered Causal-State Algorithm
CELP	Code-Excited Linear Prediction
CMN	Cepstral Mean Normalization
CSSR	Causal-State Splitting Reconstruction
CTW	Context Tree Weighting
DAG	Directed Acyclic Graph
DNA	Deoxyribonucleic Acid
DNN	Deep Neural Network
EBW	Extended Baum-Welch
ECG	Electrocardiogram

EM	Expectation Maximization
FET	Future and Emerging Technologies
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GMM	Gaussian Mixture Model
GP	Gaussian Process
GPDM	Gaussian Process Dynamical Model
GP-LVM	Gaussian Process Latent variable Model
HMM	Hidden Markov Model
HSMM	Hidden Semi-Markov Model
HTS	HMM-based Speech Synthesis system
IID, i.i.d.	Independent and Identically Distributed
KCDE	Kernel Conditional Density Estimation
KDE	Kernel Density Estimation
KDE-HMM	KDE Hidden Markov Model
KDE/HMM	KDE Hidden Markov Model (special case of KDE-HMM)
KDE-MM	KDE Markov Model
KL-divergence	Kullback-Leibler divergence
LISTA	The Listening Talker
LSTM	Long Short-Term Memory
MA	Moving Average
MAP	Maximum A-Posteriori
MCE	Minimum Classification Error
MCMC	Markov Chain Monte Carlo
MERS	Minimum Entropy Rate Simplification
MFCC	Mel-Frequency Cepstral Coefficient
MFD	Markov Forecast Density
MGE	Minimum Generation Error
ML	Maximum Likelihood

MLP	Multilayer Perceptron
MLPG	Maximum Likelihood Parameter Generation
MM	Minorize-Maximization, Markov Model
MMI	Maximum Mutual Information
MRF	Markov Random Field
MSE	Mean Squared Error
MUSHRA	Multiple Stimuli with Hidden Reference and Anchor
NLP	Natural Language Processing
NP-hard	Non-deterministic Polynomial-time hard
PAC learning	Probably Approximately Correct learning
PCA	Principal Component Analysis
pdf	Probability Density Function
PDFA	Probabilistic Deterministic Finite Automaton
pmf	Probability Mass Function
POMDP	Partially Observable Markov Decision Process
POS	Part-of-Speech
PPCA	Probabilistic Principal Component Analysis
PPM	Prediction by Partial Matching
RBF	Radial Basis Function
RBM	Restricted Boltzmann Machine
RCS	Robust Causal States
RMS	Root Mean Square
RNN	Recurrent Neural Network
RVM	Relevance Vector Machine
RV	Random Variable
SETAR	Self-Exciting Threshold Autoregressive
SNR	Signal-to-Noise Ration
STRAIGHT	Speech Transformation and Representation using Adaptive Interpolation of weighted spectrum

SVM	Support Vector Machine
TDNN	Time-Delay Neural Network
VLMM	Variable-Length Markov Model
VOM	Variable-Order Markov model
VTLN	Vocal Tract Length Normalization
WSS	Wide-Sense Stationary

Part I

Introduction

Introduction

1 The Importance of Sequence Modelling

In the last few centuries, human society has evolved at a pace unprecedented in the historical record. This development has propelled humanity from a pre-industrial civilization to a technologically advanced information society, leaving nary any aspect of life untouched.

A driving force in the recent societal evolution has been broad and systematic application of the scientific method of iteratively forming hypotheses and validating these against natural data, rather than relying on intuition and subjective beliefs alone. Identifying more accurate and appropriate models of observed patterns and phenomena has allowed scientists and engineers to better predict and shape the world. Such breakthroughs, in turn, have enabled data collection of higher accuracy and greater scope than before, creating a feedback loop. Along the way, scientists and engineers have amassed an ever-growing toolchest of methods for analyzing and describing natural data. The purpose of this thesis is to present and expand on some of these tools relevant for probabilistic sequence modelling.

1.1 Probability Theory

Since the beginning of the 20th century, probability theory has seen rapid development and risen to prominence as an invaluable modelling tool in applied sciences. Important early milestones of probability theory include the presentation of Hilbert's so-called sixth problem—a call [1], in 1900, to put physics (including probabilities) on a firm mathematical basis—followed by Kolmogorov's subsequent axiomatization of probability theory [2] in the 1930s. Today, probability theory and randomness sit at the heart of our gold-standard physical theories of the universe, as a key component of quantum mechanical models. The famous Schrödinger equation, for instance, is of fundamentally stochastic nature.

Probability theory has also seen widespread application far beyond the microcosmic realm. An important early promoter of applications of

probability theory outside physics was the biologist and statistician Sir R. A. Fisher. At any scale, the real world is fraught with uncertainty, as is the data we gather from it. Probability theory, which provides a principled method to represent, quantify, and perform computation under uncertainty, is therefore of interest.[1] Some, such as Jaynes [4], have taken this reasoning further, and interpret probability theory as an extension of logic to the case where we do not have sufficient information. Cox's theorem [5, 6] is an attempt to derive this correspondence from first principles.

At present, probability theory permeates much of the scientific theory and methodology in many fields of science—natural and social—including speech and language as considered in this thesis.

1.2 Overview of Sequence Data Applications

Time-dependent data series are everywhere. They are in the stars, and in the Earth. They appear in our society, and within ourselves. In astronomy, we encounter series showing periodic patterns of sunspots [7] or the transient intensity profiles of gamma ray bursts and many other sources [8]. In geology and meteorology, time dependence is ubiquitous in, for instance, temperature series and rainfall data [9, 10]. In biology and ecology, time series map the rise and fall of predator and prey populations [11], and appear as sampled acoustic recordings of marine and terrestrial life [12]. In military as well as civilian life, sonar and radar data take the shape of time series. In economy and finance, time chronicles the wealth of nations as well as the fluctuating prices of stocks and other financial instruments [13, 14, 15]. In social sciences, political approval ratings and crime statistics, for instance, change over time [16]. In physiology and medicine, electrocardiograms and electroencephalograms are some well-known time-dependent signals, as are epidemiological data. In culture and entertainment, series data manifestations include music (scores and performances), motion-capture data for computer graphics, as well as sports results.

Not all series data are necessarily time-dependent, either. The base-pair sequences in DNA and RNA are important examples of dataseries indexed by space rather than time. Other non-temporal examples include quantities measured along spatial paths, for instance elevation profiles of roads.

1.3 Applications in Speech and Language

A particularly rich trove of sequence data concerns human communication in different forms, most prominently speech and text. Our interest in collecting and analyzing such data is not surprising: we use speech and language to express and communicate abstract thought. In essence, these processes

[1] Other techniques for quantifying uncertainty also exist, e.g., *fuzzy logic*, which allows non-binary (partial) set membership [3].

map out the spaces where our thoughts reside. There is even evidence that language shapes our cognition [17], and language development has (through the "social brain hypothesis") been proposed as an intimate correlate of brain size in human evolution [18].

Much research effort in signal processing, natural language processing, linguistics, machine learning, and artificial intelligence has been devoted to devising models of speech and text. Such models are also of interest in fields such as neurobiology and psychoacoustics, for instance for speech perception research [19].

In some cases, breakthroughs in speech and text modelling have ushered in widespread social and societal changes. The speech modelling and signal processing (source coding) necessary for compressing and transmitting speech across cellular phone networks (e.g., CELP [20]), in particular, has had substantial impact in shaping modern society. In other areas, the promise of automatic speech and language processing technology has not been realized in full, and is subject to ongoing research. This includes many speech and language understanding tasks such as automatic speech recognition, or text translation and summarization, where machines still lag behind human performance considerably, except, perhaps, in the most narrowly-defined tasks [21].[2] Another interesting example is speech synthesis, where artificial speech can be at least as intelligible as human speech, while the naturalness of synthetic speech remains noticeably inferior to human speech [23, 24].

Interestingly, models and tools developed for speech and language data have frequently been useful in other applications as well. In fact, speech and language applications have been a driving force in the development of several widely used sequence-modelling paradigms, such as Hidden Markov models and the recent efforts to use deep neural networks for time series modelling [25] (both of which will be discussed in later sections).

1.4 Thesis Disposition

As seen in section 1.2, series data comes from many sources, and our study of such data can have many goals. Given the myriad applications, there exists a vast body of different techniques for modelling such data. Not every method is suited to every application, however, and it is therefore important to select an approach that is appropriate for the task at hand. Discrete-valued data, for example, is typically handled using different methods than continuous-valued series. More subtle differences between applications exist as well: for instance, it turns out that optimal parameter estimates in resolution-constrained source coding are different from entropy-constrained

[2] The curious fact that computers excel at many tasks that cause humans great difficulty, but have problems with tasks we humans find easy, is known as *Moravec's paradox* [22].

coding [26]. One aim of this thesis is to also consider in which situations different techniques are preferable, for instance in classification versus synthesis applications.

The remainder of this thesis introduction provides a more in-depth overview of sequence models with applications to speech and language. Specifically, section 2 defines stochastic processes, and describes some properties that set sequence data apart from other datasets. Section 3 then outlines characteristics that define various problem classes of interest, and how we may use models to solve the problems in the face of uncertainty. A taxonomy of different types of models is presented in section 4, followed by a closer study of common sequence modelling paradigms and some associated assumptions in section 5. Section 6 then discusses the surrounding procedure and considerations involved in applying the models in practice. Section 7, finally, sets the stage for the manuscripts that constitute the body of the thesis and concludes the introduction by discussing the results and relations between the papers.

2 The Data

The goal of this section is to introduce stochastic processes as a general framework for representing sources of random sequence data. This entails defining stochastic processes (sections 2.1 and 2.2), discussing finite and infinite duration data sources (section 2.3), and outlining the one-dimensional between-sample dependence properties that distinguish discrete-time processes from many other data sources (sections 2.4 and 2.5).

2.1 Preliminary Notation

We begin by introducing some standard notation. In the following, capital letters, e.g., $\boldsymbol{X}$, denote random variables (RVs), while lower-case letters identify specific, nonrandom realizations $\boldsymbol{x}$ of the random variables, for instance an observed sample value. Boldface indicates (possibly) vector or matrix-valued variables, scalars are non-bold, while curly fonts generally identify sets. In particular, $\mathcal{X}$ will represent the *state space* of $\boldsymbol{X}$, the set of values the random variable can take, i.e., $\boldsymbol{x} \in \mathcal{X}$.

The function f will represent both probability density functions (pdfs) of continuous-valued random variables and probability density functions (pmfs) of discrete-valued RVs, depending on context. To distinguish different distributions, we will write, e.g., $f_{\boldsymbol{X}}(\boldsymbol{x})$ to represent the pdf or pmf of $\boldsymbol{X}$ evaluated at $\boldsymbol{x}$. $f_{\boldsymbol{X}}(\boldsymbol{x})$, $f_{\boldsymbol{X}}(\cdot)$, or the shorthand $f_{\boldsymbol{X}}$ may sometimes be used to represent the entire distribution, that is, the function evaluated at all $\boldsymbol{x} \in \mathcal{X}$. Conditional distributions, for example of $\boldsymbol{X}$ given $\boldsymbol{\Theta} = \boldsymbol{\theta}$, are written as $f_{\boldsymbol{X}|\boldsymbol{\Theta}}(\boldsymbol{x} \mid \boldsymbol{\theta})$, while $f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta})$ with a semicolon represents a de-

pendence on a non-stochastic, but possibly unknown, quantity or parameter $\boldsymbol{\theta}$; these two examples also illustrate how Bayesian and frequentist interpretations are differentiated. The set of $\boldsymbol{x}$ for which $f_{\boldsymbol{X}}(\boldsymbol{x})$ is nonzero is the *support* of $\boldsymbol{X}$: it is a (not necessarily strict) subset of $\mathcal{X}$. Occasionally, the somewhat loose notation $\{\boldsymbol{X}\}$ may be used in pdfs or pmfs to indicate a set of random variables, each taking values on the same space $\mathcal{X}$.

2.2 Stochastic Processes

Simply put, a *stochastic process* is a collection of possibly dependent random variables $\boldsymbol{X}_t$ that share the same *observation space* or *state space*, here $\mathcal{X}$. The variables in the set are indexed by elements t of some *index set* $\mathcal{T}$, so we can think of the stochastic process as the set $\{\boldsymbol{X}_t : t \in \mathcal{T}\}$. [3]

A rigorous definition of a stochastic process requires a measurable space $(\mathcal{X}, \Sigma_{\mathcal{X}})$ and a probability space $(\Omega, \Sigma_{\Omega}, f)$, where Ω is a sample space, $\Sigma_{\mathcal{S}} \subseteq 2^{\mathcal{S}}$ signifies a sigma algebra over a set $\mathcal{S}$, and $f : \Sigma_{\Omega} \to [0, 1]$ is a measurable function assigning consistent probabilities to the events (elements) in Σ_{Ω}. A stochastic process is then a set of random variables $\{\boldsymbol{X}_t : t \in \mathcal{T}\}$ taking values on $\mathcal{X}$. For more on this formal treatment, see, e.g., [28].

Throughout this thesis, we will apply a probabilistic perspective, and consider the data under examination to be generated by a stochastic process of some sort. (This does not necessarily imply that one needs to use probabilistic methods to solve sequence-related tasks later on.) Discrete-time sequence data, specifically, is characterized by stochastic processes where the index set is scalar, fully ordered, and discrete. We will assume that this set is integer valued: either the positive integers for processes with a specific starting point, or all of $\mathbb{Z}$ for bi-infinite sequences. (Other index sets for sequences can generally be mapped onto these in an order-preserving fashion.) We will term the resulting stochastic processes *time series models*, in contrast to *time series*, which are observed, nonrandom data sequences.[4]

Unlike the index set $\mathcal{T}$, few constraints will be imposed on the set of possible values $\mathcal{X}$. Nevertheless, as with regular random variables, the nature of the state space has a substantial impact on the nature of the process, as well as what modelling techniques that are preferable. Generally

[3] An alternative interpretation is that a stochastic process is a function-valued random variable, that is, a distribution over the space of functions $\boldsymbol{x}(t)$, where the index set $\mathcal{T}$ defines the domain of the function while the state space $\mathcal{X}$ defines its range. This "function-space view" can be particularly helpful for processes on continuous index sets, such as Gaussian processes as discussed in [27].

[4] Note that continuous-time processes ($\mathcal{T} = \mathbb{R}$, say) can be sampled by picking out a discrete subset of the random variables contained in the process. If limitations are imposed on the set of possible realizations of the continuous process, the samples may sometimes be sufficient for reconstructing the full, continuous time function—an example is the case of equidistant sampling from a band-limited function, which can be reconstructed perfectly from samples with a sufficiently high sampling frequency, as per the sampling theorem [29].

speaking, discrete-valued processes over finite alphabets are the simplest to model, but countably infinite alphabets, e.g., $\mathcal{X} = \mathbb{Z}$, are also possible. For continuous-valued processes, where $\mathcal{X} \subseteq \mathbb{R}^D$ in general, it is common to make assumptions on the marginal pdfs $f_{\boldsymbol{X}_t}(\boldsymbol{x}_t)$ such as continuity or a specific parametric shape.

As a point of notation, we will from now on write $\underline{\boldsymbol{X}}$ to denote a discrete-time sequence of random variables—typically distributed according to some time series model—and $\underline{\boldsymbol{x}}$ to signify a realized string of observations, e.g., a sampled series. In particular, we may write $\underline{\boldsymbol{X}}$ to represent the stochastic process itself. In situations where we want to refer to particular random variables or observations from the process, we shall write $\underline{\boldsymbol{X}}_\tau$, where $\tau \subseteq \mathcal{T}$ is a set of time indices. The special notation

$$\underline{\boldsymbol{X}}_{t_1}^{t_2} = (\boldsymbol{X}_{t_1}, \boldsymbol{X}_{t_1+1}, \ldots, \boldsymbol{X}_{t_2}) \tag{1}$$

identifies a contiguous sequence or sample from t_1 up to and including t_2, i.e., $\underline{\boldsymbol{X}}_{t_1}^{t_2} = \underline{\boldsymbol{X}}_\tau$ with

$$\tau = \{t_1, t_1 + 1, \ldots, t_2\}. \tag{2}$$

The capital letter T is frequently used to denote the end time of a sequence; thus $\underline{\boldsymbol{X}}_1^T$ is a set of T random variables, not a vector transpose, which would be written $\boldsymbol{X}^\intercal$. In rare cases, we write $\underline{\boldsymbol{x}}^{(n)}$ to refer to a specific sequence (the nth sequence in some set), in contrast to $\boldsymbol{x}_n$, which identifies a single sampled value. The notation $\underline{\mathcal{X}}$, finally, identifies the set of all possible sequences and sub-sequences with elements taking values on $\mathcal{X}$; essentially $\underline{\mathcal{X}} = \mathcal{X}^{2^\mathcal{T}}$ (or a subset if only contiguous sub-sequences are considered), $2^\mathcal{T}$ being the powerset of $\mathcal{T}$, representing all possible index sequences.

2.3 Finite and Infinite Duration

Some practical data sources only generate observations for a short time, and, as a consequence, the length of observed sequences $\underline{\boldsymbol{x}}$ may be limited by the source itself. Other processes keep going forever, at least in theory. We distinguish these as *finite-duration* and *infinite-duration* stochastic processes, respectively. We can define finite-duration processes based on infinite duration processes with the positive integers as index set, where we also introduce a special output (observation) symbol $\emptyset$, representing "no output." Finite-duration processes are then, in principle, the processes supported on

$$\underline{\mathcal{X}}_{\text{fin}} = \left\{ \underline{\boldsymbol{x}} \in \{\mathcal{X} \cup \emptyset\}^\mathcal{T} : t, t' \in \mathcal{T} \wedge \boldsymbol{x}_t = \emptyset \wedge t' \geq t \Rightarrow \boldsymbol{x}_{t'} = \emptyset \right\}, \tag{3}$$

the set of infinite sequences where all values beyond the first $\emptyset$ also are $\emptyset$. Since $\emptyset$ represents no output and the sequence ending, one generally works

with finite sequences $\underline{\boldsymbol{x}}_1^T$ in practice; whether the considered sequence actually ended at T, or potentially continues beyond that point, is sometimes ambiguous and has to be inferred from the context.

Finite-duration sequences can be created by chopping up infinite-duration sequences at specific points, or otherwise extracting finite segments from an infinite series, ignoring any dependences between the segments. Sentences, for instance, are often seen as realizations of a finite-duration process, even if they are part of a continuous stream of text or speech.

2.4 Sample Interdependence

While many elementary probability-theory problems assume that data samples are mutually independent, stochastic processes can be used as a framework for representing dependence between a set of comparable observations. Since it is common in practice for data samples to exhibit interdependence, stochastic processes are very useful modelling tools.[5]

Dependence between variables, as for instance described by stochastic processes, is a double-edged sword. If $\boldsymbol{X}$ and $\boldsymbol{Y}$ are two independent random variables, then the joint probability density function factors as

$$f_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x},\boldsymbol{y}) \equiv f_{\boldsymbol{X}}(\boldsymbol{x})\, f_{\boldsymbol{Y}}(\boldsymbol{y})\,. \tag{4}$$

For dependent variables, this is no longer true. This complicates learning, as a dependent pdf essentially has to be learned for all pairs $(\boldsymbol{x}, \boldsymbol{y})$, which is generally more difficult—compare the pmf $f_X(x)\, f_Y(y)$ of two independent discrete random variables on an N-symbol alphabet, which has $2(N-1)$ degrees of freedom, against the general joint pmf $f_{X,Y}(x, y)$ which has $N^2 - 1$.

Another effect when samples covary is that each new sample contains less new information because of the redundancy. Consider the extreme case when samples are known to be totally dependent, such that $X_t = X_1\ \forall t \in \mathcal{T}$: Here, all samples will have the same value, and no sample beyond the first observation contributes any additional information about f_{X_t}. Hence we cannot accurately infer, e.g., the mean $\mu = \mathbb{E}(X_t)$ even from an infinitely long sample sequence. More generally, it can be shown that when nearby samples are somewhat dependent, but not too strongly so, it is possible to converge on the true mean at the same asymptotic rate as when samples are IID, but with an error that is greater by a constant factor for any large but fixed sample size. In other words, a larger amount of samples (by a certain factor) is required to obtain a desired precision, compared to the

[5] From a philosophical point of view, it can even be argued that *no* events in the world are truly independent, since the quantum-mechanical wave function of a particle generally has infinite support except when confined to infinitely deep potential wells. This makes it theoretically possible, though prohibitively unlikely, for particles to influence each other at arbitrarily large distances.

independent case. A more formal discussion of this property, complete with an example, is provided in section 5.2.

On the other hand, non-independence between random variables is an essential property in many applications, e.g., in machine learning, since dependence allows analysts to infer the properties (distribution) of unobserved quantities from observed ones. For instance, such dependence between observed feature data and class labels is a necessity for informed classification. In the case of time series specifically, dependence between nearby samples can allow more accurate short-term time-series forecasting (prediction of future values from past values), compared to the case where no nearby samples are available. For example, it is much easier to predict tomorrow's weather if one knows what the weather was like today.

2.5 The Time Dimension

As mentioned earlier, we consider stochastic processes where the index set is the integers or a (typically infinite) contiguous subset thereof. This introduces a natural ordering of the random variables $\boldsymbol{X}_t$, where variables are assigned to regularly-spaced points on the real line. For other kinds of stochastic processes, the index set can be grids or lattices, and it is also possible to assign random variables to nodes in a general graph (cf. the framework in section 5.3).

A key property of natural processes is that spatial or temporal ordering reflects a natural tendency of nearby variables to covary more, while variables at great separation are virtually independent. In other words, dependences are localized. For discrete-time sequence data, where variables are arranged along a line, this means that dependences may be expected to have a one-dimensional structure of some kind. (Note that "one-dimensional" in this case only refers to the manner in which variables influence each other, so each individual variable $\boldsymbol{X}_t$ may be a high-dimensional vector, e.g., pixels in a frame of video.)

Some specific localized dependence structures are introduced in sections 5.4 through 5.6. For now, we point out that these one-dimensional dependences have a common property in that they simplify inference: Given sufficient information about the current (and recent) values of the process, the process variables are partitioned into a past set and a future set that are conditionally independent. As a consequence, fast inference algorithms such as the forward recursion in section 5.5 can be developed. The existence of such algorithms is highly significant for the practice of time-series modelling, in that many models and methods that are computationally infeasible in the general case can, and are, used effectively to solve problems involving sequence data.

In theory, grids and many other graphical network structures may also be partitioned into conditionally independent sets if the values of a sufficient

number of observables are given, but this number usually grows with graph size, and generally does not lend itself to developing efficient algorithms.

3 Solving Problems Using Data

Our interest in sequence data stems from the fact that it carries information about the process that generated it. Real-world data therefore can be used to solve practical problems. While the details may vary between applications, this section provides an overview of different tasks, what sets them apart, and how, in principle, we can use data to solve them.

We begin by characterizing different types of tasks (section 3.1), and then consider methods for quantifying task performance (section 3.2) and making optimal decisions under uncertainty (section 3.3). A discussion of different constraints which may impose restrictions on how problems are solved is given in section 3.4. Section 3.5 discusses the necessity of making assumptions, such that decisions can be made based on empirical data (section 3.6).

3.1 Task Archetypes

The central task of applied problem-solving can generally be phrased as making a decision of some kind, based on the available data—after all, if we already have decided what to do, and cannot be influenced by data on the problem, there is rarely any interest in further analysis.

A function that accepts data as input and returns a decision is known as a *decision function* or *decision rule*. In this context, a decision means choosing an element from a set of different possibilities. If the decision function is not completely manually specified, but also has a direct dependence on other data from related decision situations, it may also be called a *learner*. The ability to not decide solely on the prior beliefs of the analyst alone, but to also provide an explicit entry point for information gleaned from the real world to guide the decisions, has proved to be an extraordinarily powerful and versatile approach to successful decision making. A more extensive discussion of statistical decision theory can be found in [30].

Some common task archetypes, which differ in the nature of the inputs and the outputs of the decision function, are classification, synthesis, regression, prediction, and estimation. These are outlined below in turn.

3.1.1 Classification

In a classification problem, the training data $\mathcal{D}$ takes the form of labelled examples, that is, pairs of observed features $\boldsymbol{x}$ and an associated label variable

c:

$$\mathcal{D} = \{(\boldsymbol{x}_n, c_n)\}_{n=1}^{N}, \tag{5}$$

where N forthwith represents the sample size. Since we are working with sequence data, we are particularly interested in cases where the each observation actually is a sequence,

$$\underline{\boldsymbol{x}}^{(n)} = \left(\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_2^{(n)}, \ldots, \boldsymbol{x}_{T_n}^{(n)}\right). \tag{6}$$

These sequences may have different lengths (T_n for sequence n). The classification task is to predict the unknown labels of new examples where only the feature sequences can be observed. In other words, one should devise a *classifier*, a function $\widehat{c}(\underline{\boldsymbol{x}})$ that, given a sequence, returns an estimate of the class label. The set of labels $\mathcal{C}$ (the possible decision choices) is generally discrete and finite; at the very least, we expect $\mathcal{C}$ to have lower cardinality than the set of possible feature sequences $\underline{\mathcal{X}}$.

Two subtypes of classification problems can be identified: in the simplest kind, each feature sequence is associated with exactly one label. Without loss of generality, $c \in \mathcal{C} \subseteq \mathbb{Z}$. Some examples of this setting are isolated word recognition for speech [31, 32], or sentiment classification [33] or spam filtering [34] for text data. In a more general setting, the "label" may actually be a sequence $\underline{c}$ of discrete labels itself: $\underline{c} \in \underline{\mathcal{C}}$. One example of this is automatic speech recognition (ASR), where the task is to convert speech sequence input to appropriate sequences of words or phones [35]. Another case is part-of-speech (POS) tagging in NLP, e.g., [36], where each word in the input text should be labelled with an appropriate part-of-speech tag. State estimation in automatic control is a third example.

The umbrella of classification is quite broad. Detection problems [37] can be seen as classification tasks, where one is classifying sequences or sequence frames into categories "signal present" and "signal not present." Certain collaborative filtering tasks can also be formulated as predicting the user-rating label based on other labelled examples [38].

3.1.2 Synthesis

Synthesis problems are essentially classification problems in reverse. The training set $\mathcal{D}$ takes the same feature-label pair form as in equation (5) from before, but the task is different. Specifically, one is given a label c or label sequence $\underline{c}$, and asked to reconstruct an appropriate observation sequence $\widehat{\underline{\boldsymbol{x}}}(\cdot)$ to go with it. The set of possible choices in the decision problem is then defined by the state space, i.e., $\widehat{\underline{\boldsymbol{x}}} \in \underline{\mathcal{X}}$, the conceivable realizations of the stochastic process. The set $\underline{\mathcal{X}}$ generally has greater cardinality than the label set, which is the opposite situation from classification. Also, the synthesis output is generally considered observable, whereas the class labels

returned by a classifier may not be possible to observe directly. In case the samples $\boldsymbol{X}_t$ are continuous-valued, this is essentially a kind of regression problem.

A prominent synthesis application is speech synthesis, where the task is to create new, synthetic speech signals corresponding to given words or phone sequences (labels) that are to be spoken [39]. The synthetic speech features $\widehat{\boldsymbol{x}}_t$, e.g., [40], are generally continuous-valued. Tasks such as machine translation [41] where the output is a text also have an element of synthesis, as the output lives in the same space as the observations. In contrast to speech synthesis, however, text output is discrete, and typically consists of strings of characters or words, $\widehat{x}_t \in \mathcal{X}$, $\mathcal{X}$ being a finite alphabet or dictionary.

3.1.3 Regression and Prediction

Regression is another type of supervised learning problem, generally characterized by finding a functional relationship (which may be stochastic) that relates one continuous-valued random variable to another. The training data takes the form

$$\mathcal{D} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n)\}_{n=1}^{N}, \tag{7}$$

with the task being is to reconstruct the *dependent variable* $\boldsymbol{y}_n$ from the *independent variables* $\boldsymbol{x}_n$, i.e., create a predictor $\widehat{\boldsymbol{y}}(\boldsymbol{x})$. (The dependent/independent nomenclature here, although standard, can be confusing, as it does not imply statistical independence among the elements of $\boldsymbol{X}$.) Unlike previous scenarios, $\boldsymbol{X}$ and $\boldsymbol{Y}$ typically take values on spaces of similar cardinality.

We are interested in cases where the datapoints are sequences. A prominent application area involving sequence pair data $\left(\underline{\boldsymbol{x}}^{(n)}, \underline{\boldsymbol{y}}^{(n)}\right)$ is *control theory*, where one wants to use a series of *inputs* $\underline{\boldsymbol{x}}$ to control the current and future outputs $\underline{\boldsymbol{Y}}$ of some process in order to maintain specific values or satisfy certain constraints [42]. The specific task of identifying the relationship between the input and output series is known as *system identification* [43].

A notable special case of the above arises when a sequence of observations from some process is available, with the variables of interest for prediction being future values of the same process. In other words, the task is to create a function $\widehat{\underline{\boldsymbol{x}}}_{\tau'}(\underline{\boldsymbol{x}}_{\tau})$, where $\tau \subset \mathcal{T}$ and $\tau' \subset \mathcal{T}$ here denote disjoint, nonempty sets of time indices. (There is thus no distinction between $\boldsymbol{X}_t$ and $\boldsymbol{Y}_t$-variables anymore, just between different times.) When the times in τ' exceed those in τ, this prediction situation is frequently known as *forecasting*. Since one essentially has a regression problem mapping values of a process to other values of the process itself, another term is *autoregression*

(see, in particular, section 5.4). Other variants involve extrapolating data series backwards in time, or filling in missing values (*imputation*).

Like in synthesis problems, the decision variable in these prediction problems lives in the same space as the observations, and is considered explicitly observable. However, the training data only consists of single dataseries without explicit labels, similar to the unsupervised data for estimation tasks below.

Forecasting is very common in certain application areas such as meteorology [44], climatology [45], and economics [46], but is only rarely a goal in itself when working with speech or language data. Nevertheless, predictive models have important applications in these areas as well: good short-term speech-signal predictors are of importance for predictive speech coding [20], while predictive models of text can be used as a component of automatic speech recognition systems, to correct misheard or unintelligible words based on context [47].

3.1.4 Estimation and Model Selection

Classification, synthesis, and regression are examples of so-called *supervised learning*, as the training data both contains observations $\boldsymbol{x}$ and other values, e.g., labels c, that interpret them or provide a reference or answer of some kind. In *unsupervised learning*, only observed data samples $\boldsymbol{x}$ are available, with no interpretation. The most prevalent example of unsupervised learning is *model selection* of some sort, where, for a given training dataset of independent samples

$$\mathcal{D}_{(\mathrm{us})} = \{\boldsymbol{x}_n\}_{n=1}^{N}\,, \tag{8}$$

the task is to select the most appropriate description m of the data from a set of possible stochastic models $\mathcal{M}$, each a distribution $f_m(\boldsymbol{x})$ over $\mathcal{X}$. (The letters "us" here stand for unsupervised.) Especially relevant to this thesis is the case where the training dataset contains sequences $\underline{\boldsymbol{x}}^{(n)}$. Each sequence is assumed to be independent, but samples within the same sequence may show dependence.

Typically, the set of models to choose from forms a manifold or a set of manifolds in the space of all possible distributions. If we introduce a coordinate system over the manifold, the problem of model selection then reduces to what is known as *parameter estimation*, where the parameter $\boldsymbol{\theta} \in \mathcal{P}$ corresponds to the coordinates of a unique model on the manifold. In other words, there is a bijection between $\mathcal{M}$ and $\mathcal{P}$, and we have

$$f_m(\boldsymbol{x}) \equiv f(\boldsymbol{x};\, \boldsymbol{\theta}) \tag{9}$$

for some $m(\boldsymbol{\theta})$. An *estimator* is thus a function $\widehat{\boldsymbol{\theta}}(\{\boldsymbol{X}\})$ of random observations from a distribution, which we apply to the available (training) data

$\mathcal{D}_{(\text{us})}$.[6]

Parameter estimation is most frequently encountered as a sub-task in standard approaches to classification or synthesis: first, one uses the training data subsets

$$\mathcal{D}_c = \{\boldsymbol{x}_n \in \mathcal{D} : c_n = c\} \; \forall c \in \mathcal{C} \tag{10}$$

to create a separate model m_c for each label c; the resulting set of models is then used to generate appropriate classifier outputs for given inputs. The procedure will be discussed further in sections 3.6 and 6.1, while more comprehensive coverage specifically of estimation is provided in [48]. Most often, the set of parameters $\mathcal{P}$ (possible decision choices) is continuous, as they are coordinates on a manifold. Discrete parameters and estimation problems also exist, for instance order selection [49] (estimating the order parameter of a model, e.g., the degree of a polynomial in a regression model). These typically correspond to selecting one of several possible manifolds in distribution space; such discrete problems are often referred to as model selection rather than parameter estimation.

There are also situations where the estimated value is seen as a goal in itself, such as determining the signal-to-noise ratio of a wireless system [50], estimating the mean value of a population, tracking an object on radar, or identifying the value of fundamental constants in physics. These examples illustrate that estimation problems also can arise in non-probabilistic settings, such as classical physics, and that there need not be a one-to-one mapping between the estimated parameter and the model that generated the data (e.g., many different distributions can have the same mean). In this case, the one-to-one relationship from equation (9) might not apply, and we may instead see the quantity $\boldsymbol{\theta}$ to be estimated as a many-to-one function $\boldsymbol{\theta}(f_{\boldsymbol{X}})$ of the actual distribution $f_{\boldsymbol{X}}$ of the observations. However, it is still suffices to estimate $f_{\boldsymbol{X}}$ to obtain an estimate of $\boldsymbol{\theta}$.

In addition to the situations discussed here, there are many other kinds of decision problems, such as ranking [51] (choosing an ordering of a set of elements) or the related problem of selecting a subset of items. Most of these, however, are less common in sequence data applications, and will therefore not be considered further in this thesis.

3.2 Loss Functions

For a given decision task, we frequently want to select a course of action that is as good as possible under the circumstances. This suggests expressing the

[6] This presentation tends towards the *frequentist* perspective, where the parameter $\boldsymbol{\theta}$ is treated as an unknown number without any probabilistic interpretation. A deeper look at the alternative *Bayesian* view, where the parameters are treated as random variables $\boldsymbol{\Theta}$ which covary with other observables, is provided in section 4.4, among others.

task as a mathematical optimization problem. To make this more concrete, we must quantify what constitutes good and bad performance.

A highly general method for evaluating different options in a decision problem is to define a *loss function* or *cost function* $L\left(\widehat{\boldsymbol{y}}, \boldsymbol{y}\right)$ which associates a cost with choosing the element $\widehat{\boldsymbol{y}}$ when the optimal choice would have been $\boldsymbol{y}$. (The symbol here $\boldsymbol{y}$ is a stand-in that could denote a class label, observation sequence, or a distribution parameter—we are by no means referring to regression exclusively.) Typically the loss $L\left(\boldsymbol{y}, \boldsymbol{y}\right)$ of making the optimal choice is zero, with other alternatives having equal or greater loss. The loss-function concept can also be generalized to situations where the decision and the world state occupy different spaces, though we will not do so explicitly here.

The loss function encodes our priorities between different choices, so the details are highly application dependent. Nevertheless, some common alternatives exist. For classification, it is common to take

$$L_{01}\left(\widehat{\boldsymbol{y}}, \boldsymbol{y}\right) = I\left(\widehat{\boldsymbol{y}} \neq \boldsymbol{y}\right), \tag{11}$$

where $I\left(\cdot\right)$ is an indicator function. This is called *0-1 loss* [52]. 0-1 loss considers each deviation from optimality "equally bad," and simply counts the number of misclassifications a decision rule makes when applied to a dataset. On continuous sets, for instance in regression problems, the *quadratic loss function*

$$L_2\left(\widehat{\boldsymbol{y}}, \boldsymbol{y}\right) = \left\|\widehat{\boldsymbol{y}} - \boldsymbol{y}\right\|_2^2 \tag{12}$$

$$= \sum_i \left(\widehat{y}_i - y_i\right)^2 \tag{13}$$

is often used. As this is a continuous function, the derivative of which is linear in $\widehat{\boldsymbol{y}}$, it frequently leads to linear equations, making it easier to optimize than 0-1 loss. However, the quadratic loss is not invariant under non-affine transformations of the input variables. For example,

$$L_2\left(0, 1\right) = 1 = L_2\left(1, 2\right) \tag{14}$$

$$L_2\left(\ln 0, \ln 1\right) = \infty \neq L_2\left(\ln 1, \ln 2\right). \tag{15}$$

It is therefore important to choose an appropriate data representation when using the quadratic loss.

The quadratic loss function is also applicable for comparing densities of random variables. If $f_{\widehat{\boldsymbol{Y}}}\left(\boldsymbol{y}\right)$ is a selected density, while the true, reference distribution (optimal choice) is $f_{\boldsymbol{Y}}\left(\boldsymbol{y}\right)$, we can simply replace the sum in (12) by an integral, and obtain

$$L_2\left(f_{\widehat{\boldsymbol{Y}}}\left(\boldsymbol{y}\right), f_{\boldsymbol{Y}}\left(\boldsymbol{y}\right)\right) = \int \left(f_{\widehat{\boldsymbol{Y}}}\left(\boldsymbol{y}\right) - f_{\boldsymbol{Y}}\left(\boldsymbol{y}\right)\right)^2 \mathrm{d}\boldsymbol{y}. \tag{16}$$

Another important loss function for comparing densities the *Kullback-Leibler divergence* [53], or *KL-divergence* for short, which is defined by

$$L_{\mathrm{KL}}\left(f_{\widehat{\boldsymbol{Y}}}\left(\boldsymbol{y}\right), f_{\boldsymbol{Y}}\left(\boldsymbol{y}\right)\right) = \int f_{\boldsymbol{Y}}\left(\boldsymbol{y}\right) \ln \frac{f_{\boldsymbol{Y}}\left(\boldsymbol{y}\right)}{f_{\widehat{\boldsymbol{Y}}}\left(\boldsymbol{y}\right)} \mathrm{d}\boldsymbol{y} \tag{17}$$

$$= h\left(\boldsymbol{Y}\right) - \mathbb{E}\left(\ln f_{\widehat{\boldsymbol{Y}}}\left(\boldsymbol{Y}\right)\right), \tag{18}$$

where h denotes the differential entropy [54]. Unlike the other loss functions presented, the KL-divergence is asymmetric in the arguments. Specifically, it tends to strongly penalize situations where $f_{\widehat{\boldsymbol{Y}}}$ has smaller support than the reference $f_{\boldsymbol{Y}}$ [55]. The KL-divergence has connections to information theory, and may, for instance, be interpreted as the number of excess bits sent in source coding when using a coding scheme optimal for the distribution $f_{\widehat{\boldsymbol{Y}}}$ when the true distribution is $f_{\boldsymbol{Y}}$.

If it is possible to assign reasonable numerical costs to the various possible errors in an application, that is usually preferable to using a standard loss function like those above. Assessing the costs of different scenarios may be relatively straightforward in financial mathematics, but can be challenging in other areas, where values different wins and losses may not be monetary, and thus not necessarily directly comparable.

3.3 Optimal Decisions under Uncertainty

A practical problem with loss functions as presented thus far is that one must know the true optimal decision in order to be able to compute the loss. Obviously, that information is not accessible in a real decision situation, where the optimal course of action is unknown. Instead, one has to act based on other information, available in the form of random variables that have been observed. Such information is seldom sufficient to determine the true optimal course of action with certainty, but induces a conditional probability distribution over the different possible actions, indicating how likely it is that each one of them is the reference option (the one with the lowest cost once the answer is revealed). Based on this distribution, one can form a decision in many different ways:

3.3.1 Supervised Problems

If, in a supervised learning problem, the joint distribution $f_{\boldsymbol{X},C}\left(\boldsymbol{x}, c\right)$ of features and labels is available, it is possible to minimize the *expected loss,* also known as the *risk*, given the input data. In fact, risk minimization only requires knowledge about the conditional distribution of the unknown quantity given the knowns, e.g., the conditional distribution $f_{C|\boldsymbol{X}}\left(c \mid \boldsymbol{x}\right)$ of labels given the features in classification; the joint distribution is not

necessary. The associated minimization problem can be written

$$\widehat{c}\left(\boldsymbol{x}\right) = \operatorname*{argmin}_{c \in \mathcal{C}} \mathbb{E}\left(L\left(c,\, C\right) \mid \boldsymbol{X} = \boldsymbol{x}\right) \tag{19}$$

$$= \operatorname*{argmin}_{c \in \mathcal{C}} \sum_{c' \in \mathcal{C}} L\left(c,\, c'\right) f_{C|\boldsymbol{X}}\left(c' \mid \boldsymbol{x}\right). \tag{20}$$

(Classification will be used as a running example in much of this section. The corresponding problem for other supervised situations is completely analogous.)

In supervised learning problems, minimizing the expected 0-1 loss L_{01} leads to a decision rule of the form

$$\widehat{c}_{\mathrm{MAP}}\left(\boldsymbol{x}\right) = \operatorname*{argmax}_{c \in \mathcal{C}} f_{C|\boldsymbol{X}}\left(c \mid \boldsymbol{x}\right) \tag{21}$$

$$= \operatorname*{argmax}_{c \in \mathcal{C}} f_{\boldsymbol{X}|C}\left(\boldsymbol{x} \mid c\right) f_{C}\left(c\right). \tag{22}$$

This is known as the *maximum a-posteriori* (MAP) decision rule, as it selects the most probable alternative given the input data $\boldsymbol{x}$. The term $f_C\left(c\right)$, the distribution over the class labels in the absence of any further information about the instance (i.e., before observing $\boldsymbol{x}$), is termed the *prior distribution*. In the special case when all class labels are equally likely a priori, so that f_C is a uniform distribution, this reduces to

$$\widehat{c}_{\mathrm{ML}}\left(\boldsymbol{x}\right) = \operatorname*{argmax}_{c \in \mathcal{C}} f_{\boldsymbol{X}|C}\left(\boldsymbol{x} \mid c\right). \tag{23}$$

This is known as the *maximum likelihood* (ML) rule as it maximizes the *likelihood function* $f_{\boldsymbol{X}|C}\left(\boldsymbol{x} \mid c\right)$, which is the probability of the observed data as a function of an unobserved quantity or parameter, here c. In speech synthesis, the MAP rule of maximizing the probability of the output given the input is widely used, though it is confusingly labelled as "*maximum likelihood parameter generation*" (MLPG) [56],

$$\widehat{\underline{\boldsymbol{x}}}_{\mathrm{MLPG}}\left(\underline{c}\right) = \operatorname*{argmax}_{\boldsymbol{x} \in \mathcal{X}} f_{\underline{\boldsymbol{X}}|\underline{C}}\left(\underline{\boldsymbol{x}} \mid \underline{c}\right). \tag{24}$$

For regression-type tasks, the expected quadratic loss, also known as *mean square error* (MSE), is minimized by the conditional expected value of the missing variable, e.g.,

$$\widehat{\boldsymbol{x}}_{\mathrm{MSE}}\left(c\right) = \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} \mathbb{E}\left(\left\|\boldsymbol{x} - \boldsymbol{X}\right\|_2^2 \mid C = c\right) \tag{25}$$

$$= \mathbb{E}\left(\boldsymbol{X} \mid C = c\right) \tag{26}$$

$$= \int \boldsymbol{x} f_{\boldsymbol{X}|C}\left(\boldsymbol{x} \mid c\right) \mathrm{d}\boldsymbol{x}. \tag{27}$$

(This tacitly assumes that the conditional mean is an element of $\mathcal{X}$, so that it can be selected by the optimization. This is for instance satisfied if $\mathcal{X}$ is convex. If $\mathbb{E}(\boldsymbol{X} \mid C = c) \notin \mathcal{X}$, one may have to project the mean onto $\mathcal{X}$.) For Gaussian distributions, the mean is the same as the mode, and this gives the same result as MAP. There are also other loss functions that are minimized by the same mean value, in particular the so-called *Bregman divergences* [57, 58], of which the quadratic loss function is a special case.

A more risk-averse alternative to the expected loss is to minimize the worst-case-scenario loss. This can be written

$$\widehat{c}_{\text{minimax}}(\boldsymbol{x}) = \underset{c \in \mathcal{C}}{\operatorname{argmin}} \sup_{c' \in \mathcal{C}(\boldsymbol{x})} L(c, c'), \tag{28}$$

where $\mathcal{C}(\boldsymbol{x})$ denotes the set of possible c-values given that the input features are $\boldsymbol{x}$. This is known as the *minimax criterion*, as it minimizes the maximum loss. Notably, the decision does not depend on the exact probability distribution of the output variable given the input, but only on the possible losses that can be incurred for a specific decision (the support of the pdf). The principle is therefore also applicable in cases where there is no explicit probability distribution over the output variable.

3.3.2 Unsupervised Problems

In unsupervised problems, where we want to identify the distribution or other structure of an unlabelled dataset, we can use the quadratic or Kullback-Leibler loss, respectively, to define the two optimization problems

$$\widehat{f}_2(\cdot) = \underset{g(\cdot) \in \mathcal{M}}{\operatorname{argmin}} \int \left(g(\boldsymbol{x}) - f_{\boldsymbol{X}}(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x} \tag{29}$$

$$\widehat{f}_{\text{KL}}(\cdot) = \underset{g(\cdot) \in \mathcal{M}}{\operatorname{argmin}} \int f_{\boldsymbol{X}}(\boldsymbol{x}) \ln \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{g(\boldsymbol{x})} \mathrm{d}\boldsymbol{x}, \tag{30}$$

where $\mathcal{M}$ is a set of models (probability densities). Like the supervised case above, this requires that the data distribution $f_{\boldsymbol{X}}$ is known. Note that the problems are nontrivial if $f_{\boldsymbol{X}}(\cdot) \notin \mathcal{M}$. The set of models $\mathcal{M}$ may, for instance, be a parametric family of distributions $f(\boldsymbol{x}; \boldsymbol{\theta})$ indexed by the parameter $\boldsymbol{\theta}$, so that

$$\mathcal{M} = \{f(\cdot; \boldsymbol{\theta})\}_{\boldsymbol{\theta} \in \mathcal{P}}. \tag{31}$$

The previous optimization problems are then equivalent to the parameter estimation problems

$$\widehat{\boldsymbol{\theta}}_2 = \underset{\boldsymbol{\theta} \in \mathcal{P}}{\operatorname{argmin}} \int \left(f(\boldsymbol{x}; \boldsymbol{\theta}) - f_{\boldsymbol{X}}(\boldsymbol{x})\right)^2 \mathrm{d}\boldsymbol{x} \tag{32}$$

$$\widehat{\boldsymbol{\theta}}_{\text{KL}} = \underset{\boldsymbol{\theta} \in \mathcal{P}}{\operatorname{argmin}} \int f_{\boldsymbol{X}}(\boldsymbol{x}) \ln \frac{f_{\boldsymbol{X}}(\boldsymbol{x})}{f(\boldsymbol{x}; \boldsymbol{\theta})} \mathrm{d}\boldsymbol{x}. \tag{33}$$

Among the two loss functions, the KL-divergence is significantly more common in parameter estimation, whereas L_2 minimization is the norm in non-parametric density estimation, possibly because it simplifies calculations [59]. Generally speaking, the KL-divergence is more sensitive to capturing the tails of the distribution than is the L_2 loss, so a distribution obtained through KL-minimization is often more spread out, whereas L_2-minimizers are comparatively more accurate near peaks of $f_{\boldsymbol{X}}$.

Interestingly, KL-divergence minimization can, similarly to the supervised learning problems earlier, be formulated as minimizing an expected value,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{KL}} = \underset{\boldsymbol{\theta}\in\mathcal{P}}{\operatorname{argmin}} \mathbb{E}\left(\ln f_{\boldsymbol{X}}\left(\boldsymbol{X}\right) - \ln f\left(\boldsymbol{X};\, \boldsymbol{\theta}\right)\right) \tag{34}$$

$$= \underset{\boldsymbol{\theta}\in\mathcal{P}}{\operatorname{argmax}} \mathbb{E}\left(\ln f\left(\boldsymbol{X};\, \boldsymbol{\theta}\right)\right). \tag{35}$$

Notice that this amounts to maximizing the (expected) likelihood of the data, and that the derivation and criterion do not require the parameter $\boldsymbol{\theta}$ to be a realization of a random variable, which is appropriate for frequentist methods. The ML-KL connection is explored further in section 3.6.

3.4 Tasks Constraints

Practical problems are not necessarily always as straightforward as outlined above. Other side constraints on the data and how it is acquired or presented may also affect the solution to the task. Some of these will be discussed in this section.

Sometimes, there are restrictions on the information that is available. In a supervised learning application, label data c may sometimes be expensive to acquire, whereas feature data $\boldsymbol{x}$ comes cheap. Both speech and texts, for instance, may be readily obtained in bulk from the Internet, but accurate annotation (e.g., speech transcription or POS-tagging) requires costly human intervention. In this case, a middle ground between supervised and unsupervised learning, known as *semi-supervised learning* [60], exists. In semi-supervised learning, only some of the training examples are labelled. Often, it is possible to improve performance over pure supervised learning on the labelled examples only, as the distribution of unlabelled points may carry information about where (for instance) different clusters in the data may be located, which may then be identified as belonging to different classes using the few labelled examples.

Another information restriction is *censored data*, where, for some datapoints, only a subset of the elements in $\boldsymbol{x}$ are available. And even if all elements are available, they may not be accurate. There could be saturation effects in the features, or noise bursts that make elements unreliable. Particularly challenging are the situations where entire datapoints are misleading,

due to labels being incorrect, as situation known as *label noise* [61].

Another class of constraints surrounds the manner in which data are presented. The classic learning scenario—where a large set of training data is provided, a classifier, synthesizer or other similar-problem solver is created using the data, and the resulting decision function is applied to the task at hand—is known as *batch learning*. Sometimes, however, data arrives instance by instance in a stream. This is called *online learning*. In this scenario, only part of the information in a new instance is provided at first, and the learner is required to make a decision based on the incomplete datum (e.g., predict the class from given features); after this, the true answer is revealed, and there is an opportunity to refine the decision function based on the new information.

The online learning situation is more demanding that batch learning, because new information is constantly coming in, and procedures that cannot be updated or generate output efficiently may be unable to keep up. Generally speaking, an online algorithm can always be applied in a batch setting, by presenting the training examples one by one until they have been exhausted, and then applying the resulting problem-solver to the actual task at hand. However, some results on converting batch learning algorithms to an online setting do exist, e.g., [62].

In some online learning settings, it may be possible for the problem-solving algorithm to influence the examples that arrive, for instance by presenting different feature sets and being told the appropriate label. This is known as *active learning* [63]. As this procedure has similarities with how human infants and children develop, active learning may, among other things, be of interest in artificial intelligence applications.

Though the above restrictions and paradigms are important, the remainder of this thesis will focus on offline, purely supervised or unsupervised learning—specifically how to create and identify useful models for sequence data of different kinds.

3.5 The Necessity of Assumptions

So far, we have seen how a variety of decision problems involving sequence data can be formulated, and how they can be solved if the behaviour of the data (meaning the distribution) is provided. In practice, the distribution is unknown, and one must make some sort of assumptions to solve the problem. This is a symptom of a general property in science and philosophy that some things have to be taken for granted—even the fundamental building blocks of mathematics are expressed as axioms [64], whose internal consistency cannot be established beyond all doubt [65].

Without making assumptions that connect future decisions with past decisions in the problems we are considering, every new situation is a blank slate, and it is not possible to use past data to aid future decisions. There

are theoretic arguments (see, e.g., chapter 2 in [66]) that assumption-free learning cannot generalize to new instances. In machine learning, assumptions are often referred to as *inductive bias*, since they bias a learner towards certain conclusions, particularly when data is scarce.

The question then becomes what assumptions to make. As usual, the answer depends on the situation. In general, one typically makes assumptions such that the problem is in some way solvable. Preferably, the assumptions should also match the characteristics of the practical situation under consideration.

Beyond this, there is a myriad of different possible situations. A useful view is to place different assumptions on a scale ranging from strong to weak. Strong assumptions are typically necessary when only little data is available. At the extreme end, if we propose a single, specific distribution $f_{\boldsymbol{X}}$ or $f_{\boldsymbol{X},C}$, there is no need for training data at all—optimal decisions under the assumed distribution can be computed directly from previously presented principles. The hazards of making such strong assumptions are that they could be wildly wrong, and there is no way for data to steer the decision making in the right direction.

If the training dataset is large, one may consider making weaker assumptions, leaving a bigger hole for the data to fill. The most conservative assumption that can be made is arguably that the data distribution follows the so called empirical distribution (in equation (36) below), which is supported exclusively on the observations in the given dataset. This does not assume the existence of any possible observations beyond the values seen in the training data. Sadly, this is too conservative and cannot generalize at all to new situations; see equation (45). Nevertheless, the empirical distribution, as a probability distribution which (together with the sample size N) uniquely represents the dataset, can be used as a tool for developing techniques for making decisions based on a mixture of assumptions and empirical data, as outlined below.

As a middle ground between no assumptions, and assumptions only, the classic approach to problem solving with data follows the example of estimation and model selection: one assumes that the data was generated by a distribution in a specific set $\mathcal{M}$, and then uses the data to pick a fitting model (or several—cf. 4.4 and 4.6) from the set. If the dimensionality of the set is fixed, such that that it can be parameterized by a coordinate $\boldsymbol{\theta} \in \mathbb{R}^p$, $\mathcal{M}$ is known as a *parametric family*, and the models that it contains are called *parametric models*. If, on the other hand, the dimensionality of the set of possible models grows without bound as the size N of the training dataset increases, the setup is typically considered *nonparametric*.

3.6 Data-Based Decision Making

We shall now look at techniques for making informed decisions and solving the tasks from section 3.1, using a mixture of data and assumptions. The hope is that, if the assumptions are reasonably close to the true situation, a practically useful solution will result.

3.6.1 Unsupervised Problems

The decision criteria presented in section 3.3 are unrealistic, as they assume that the relevant distribution $f_{\boldsymbol{X},C}(\boldsymbol{x}, c)$ or $f_{\boldsymbol{X}}(\boldsymbol{x})$ is known, whereas in a practical application only training data $\mathcal{D}$ or $\mathcal{D}_{\text{un}}$ is available. To get around this, one can define the *empirical distribution*

$$\dot{f}(\boldsymbol{x}) = \dot{f}\left(\boldsymbol{x} \mid \mathcal{D}_{(\text{us})}\right) = \frac{1}{N}\sum_{n=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_n) \tag{36}$$

consisting of Dirac spikes at each datapoint. This distribution essentially turns expected values into sums over the available datapoints. Inserting this approximation as the reference distribution into equation (35), one obtains

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}_{\text{ML}}\left(\mathcal{D}_{(\text{us})}\right) &= \operatorname*{argmax}_{\boldsymbol{\theta}\in\mathcal{P}} \frac{1}{N}\sum_{n=1}^{N} \ln f(\boldsymbol{x}_n; \boldsymbol{\theta}) && (37)\\
&= \operatorname*{argmax}_{\boldsymbol{\theta}\in\mathcal{P}} \ln f\left(\mathcal{D}_{(\text{us})}; \boldsymbol{\theta}\right) && (38)\\
&= \operatorname*{argmax}_{\boldsymbol{\theta}\in\mathcal{P}} f\left(\mathcal{D}_{(\text{us})}; \boldsymbol{\theta}\right) && (39)
\end{aligned}$$

In other words, minimizing the KL-divergence between the parametric approximation and the empirical distribution is the same as choosing the parameter according to the maximum likelihood rule in equation (23) (cf. [67]); $\widehat{\boldsymbol{\theta}}_{\text{ML}}$ is the parameter choice which maximizes the probability of the given data.[7]

Maximum likelihood is the classic frequentist parameter estimation procedure, as it has numerous desirable properties. Most importantly, it is [70, 71]:

[7] For the record, minimizing the quadratic loss (L_2-norm) between the empirical distribution and a parametric family yields the estimator

$$\widehat{\boldsymbol{\theta}}_2\left(\mathcal{D}_{(\text{us})}\right) = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathcal{P}} \left(\frac{1}{2}\int \left(f(\boldsymbol{x}; \boldsymbol{\theta})\right)^2 \mathrm{d}\boldsymbol{x} - \frac{1}{N}\sum_{n=1}^{N} f(\boldsymbol{x}_n; \boldsymbol{\theta})\right). \tag{40}$$

A cross-validation version of this estimator is commonly used with nonparametric methods, specifically in bandwidth selection for kernel density estimation (KDE) [68, 69], but the formula has received relatively little attention in the traditional parameter estimation literature, compared to the maximum likelihood approach.

1. Consistent: if the true $f_{\boldsymbol{X}}(\boldsymbol{x}) \equiv f(\boldsymbol{x}; \boldsymbol{\theta})$ for some $\boldsymbol{\theta}$, the estimator (37) will converge in probability on this value as $N \to \infty$.

2. Asymptotically efficient: as N grows large, $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ will be as close as possible to the true value $\boldsymbol{\theta}$ for the given dataset size (again assuming the true model is in $\mathcal{M}$).

3. Parameterization invariant: if $\boldsymbol{\theta}' = \boldsymbol{g}(\boldsymbol{\theta})$ is an alternative parameterization of the parametric family $\mathcal{M}$, and $\widehat{\boldsymbol{\theta}}'_{\mathrm{ML}}\left(\mathcal{D}_{(\mathrm{us})}\right)$ is the maximum likelihood estimate of this alternative parameter for a given dataset, then $\widehat{\boldsymbol{\theta}}'_{\mathrm{ML}}\left(\mathcal{D}_{(\mathrm{us})}\right) \equiv \boldsymbol{g}\left(\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}\left(\mathcal{D}_{(\mathrm{us})}\right)\right)$. As parameters are not considered directly observable, there may be more than one compelling parameterization of a family $\mathcal{M}$; this result assures us that how we represent the family of distributions will not affect the result.

This is a frequentist treatment, in that the parameter $\boldsymbol{\theta}$ is considered unknown but not stochastic, and we do not use a probability distribution to quantify the uncertainty in $\boldsymbol{\theta}$. The Bayesian alternative is to instead introduce a probability distribution over the models in $\mathcal{M}$. This is equivalent to letting the true parameter be a random variable $\boldsymbol{\Theta}$ with a prior distribution $f_{\boldsymbol{\Theta}}(\boldsymbol{\theta})$. The joint distribution can be written

$$f_{\boldsymbol{X},\boldsymbol{\Theta}}(\boldsymbol{x}, \boldsymbol{\theta}) \equiv f_{\boldsymbol{X}|\boldsymbol{\Theta}}(\boldsymbol{x} \mid \boldsymbol{\theta})\, f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}), \tag{41}$$

where $f_{\boldsymbol{X}|\boldsymbol{\Theta}}(\boldsymbol{x} \mid \boldsymbol{\theta}) \equiv f_{\boldsymbol{X}}(\boldsymbol{x}; \boldsymbol{\theta})$ are functionally identical, but carry different interpretations. Minimizing the expected 0-1 loss for $\boldsymbol{\theta}$ leads to the maximum a-posteriori parameter estimate

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}\left(\mathcal{D}_{(\mathrm{us})}\right) = \operatorname*{argmax}_{\boldsymbol{\theta} \in \mathcal{P}} f_{\boldsymbol{\Theta}}(\boldsymbol{\theta}) \prod_{n=1}^{N} f_{\boldsymbol{X}|\boldsymbol{\Theta}}(\boldsymbol{x}_n \mid \boldsymbol{\theta}) \tag{42}$$

$$= \operatorname*{argmax}_{\boldsymbol{\theta} \in \mathcal{P}} f_{\boldsymbol{\Theta}|\{\boldsymbol{X}\}}\left(\boldsymbol{\theta} \mid \mathcal{D}_{(\mathrm{us})}\right), \tag{43}$$

completely analogous to the MAP rule in equation (21). A more in-depth treatment of the Bayesian perspective will be reserved for section 4.4.

3.6.2 Supervised Problems

For supervised problems, the situation is usually a little more complex. The empirical distribution in (36) has very narrow support, so the induced conditional distribution

$$\dot{f}_{C|\boldsymbol{X}}(c \mid \boldsymbol{x}) = \frac{\dot{f}_{\boldsymbol{X},C}(\boldsymbol{x}, c)}{\dot{f}_{\boldsymbol{X}}(\boldsymbol{x})} \tag{44}$$

$$= \frac{\dot{f}_{\boldsymbol{X},C}(\boldsymbol{x}, c)}{\sum_{c' \in \mathcal{C}} \dot{f}_{\boldsymbol{X}}(\boldsymbol{x}, c')} \tag{45}$$

is undefined unless $\boldsymbol{x} \in \{\boldsymbol{x}_n\}_{n=1}^N$. The risk minimization rule (20) then cannot be applied directly. In classification, the classic way to get around this is to:

1. propose a parametric family

$$\mathcal{M} = \{f_{\boldsymbol{X},C}(\cdot, \cdot; \boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}} \tag{46}$$

   for the joint distribution $f_{\boldsymbol{X},C}$,

2. estimate the parameters $\boldsymbol{\theta}$ of the joint distribution using unsupervised learning (typically maximum likelihood) on the data $\mathcal{D}$, and

3. insert the selected pdf $f_{\boldsymbol{X},C}\left(\boldsymbol{x}, c; \widehat{\boldsymbol{\theta}}\right)$ into a decision rule, commonly MAP (21) or maximum likelihood (23), to obtain a classifier.

Assuming data samples are independent, the maximum likelihood objective function for the parameter estimation factors as

$$f_{\{\boldsymbol{X},C\}}(\mathcal{D}; \boldsymbol{\theta}) = \prod_{n=1}^{N} f_{\boldsymbol{X},C}(\boldsymbol{x}_n, c_n; \boldsymbol{\theta}) \tag{47}$$

$$= \prod_{c\in\mathcal{C}} \prod_{n=1}^{N} \left(f_{\boldsymbol{X}|C}(\boldsymbol{x}_n \mid c; \boldsymbol{\theta}) f_C(c; \boldsymbol{\theta})\right)^{I(c_n=c)} \tag{48}$$

$$= \prod_{c\in\mathcal{C}} f_C(c; \boldsymbol{\theta})^{|\mathcal{D}_c|} f_{\{\boldsymbol{X}\}|C}(\mathcal{D}_c \mid c; \boldsymbol{\theta}), \tag{49}$$

where $\mathcal{D}_c$ from equation (10) is the subset of datapoints corresponding to label c, and $|\mathcal{D}_c|$ denotes its cardinality. Typically, the class-conditional distributions $f_{\{\boldsymbol{X}\}|C}(\mathcal{D}_c \mid c; \boldsymbol{\theta})$ depend on disjoint subsets $\boldsymbol{\theta}_c$ of the parameters for each class. As the objective function factors, these parameters can then be estimated independently, class by class, simplifying the computations.

3.6.3 Discriminative Procedures

The proposed maximum likelihood classification approach based on unsupervised parameter estimation typically yields reasonable results, but may underperform in cases where the assumptions are incorrect and the true distribution of the data is not part of $\mathcal{M}$. There are several techniques, known as *discriminative learning*, which attempt to optimize objectives that are closer to the supervised task at hand. The general idea is to worry less about describing the distributions accurately, and instead concentrate on the *decision boundaries* (the inputs $\boldsymbol{x}$ at which $\widehat{c}$ changes value), and try to position these as appropriately as possible.

One discriminative strategy, known as *maximum mutual information* (MMI) training [72], is to select the parameters to maximize the conditional probability

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MMI}}\left(\mathcal{D}\right) = \operatorname*{argmax}_{\boldsymbol{\theta}\in\mathcal{P}} \prod_{n=1}^{N} f_{C|\boldsymbol{X}}\left(c_n \mid \boldsymbol{x}_n; \boldsymbol{\theta}\right) \tag{50}$$

of the correct class. This maximizes the information-theoretic mutual information between labels and observations, evaluated on the training data. Compared to MAP parameter estimation for unsupervised data, this procedure optimizes the conditional probabilities used in a MAP decision (21) for the supervised task. This is much closer to the supervised task performance we ultimately are interested in, and hence (50) tends to improve results in practice, at the expense of being more difficult to optimize than traditional approaches such as (37).

Another possibility is to consider the decision function, e.g., (21) or (23), as an abstract function of both the input data and the parameters $\boldsymbol{\theta}$. Minimizing the expected loss as a function of the parameters then yields the formulation

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathcal{P}} \mathbb{E}\left(L\left(\widehat{c}\left(\boldsymbol{X}; \boldsymbol{\theta}\right), C\right)\right), \tag{51}$$

in the case of classification. Note that this is a valid problem also in cases where the classifier $\widehat{c}$ is just an arbitrary function with no probabilistic interpretation at all. We can treat the performance of the decision function on the training data (i.e., the empirical distribution) as a proxy for performance on future examples. This suggests selecting parameters according to

$$\widehat{\boldsymbol{\theta}}\left(\mathcal{D}\right) = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathcal{P}} \frac{1}{N} \sum_{n=1}^{N} L\left(\widehat{c}\left(\boldsymbol{x}_n; \boldsymbol{\theta}\right), c_n\right). \tag{52}$$

For the special case of 0-1 loss, this yields an estimate

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MCE}}\left(\mathcal{D}\right) = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathcal{P}} \sum_{n=1}^{N} I\left(\widehat{c}\left(\boldsymbol{x}_n; \boldsymbol{\theta}\right) \neq c_n\right). \tag{53}$$

This is known as *minimum classification error* (MCE) [73], as it chooses a parameter set that minimizes the number of misclassifications that $\widehat{c}$ makes on the training data. While MCE often achieves good practical performance on the task under consideration, the objective function is not straightforward to optimize, as it only takes on N different, discrete values.

Equation (52) and the reasoning behind it also applies to synthesis tasks. For speech, assuming a quadratic loss function (more appropriate than 0-1

loss now that $\mathcal{X}$ is continuous and equipped with a metric) and a given data-generation procedure $\widehat{\boldsymbol{x}}(\cdot;\boldsymbol{\theta})$ such as MLPG, one obtains the parameter estimation problem

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MGE}}(\mathcal{D}) = \operatorname*{argmin}_{\boldsymbol{\theta}\in\mathcal{P}} \sum_{n=1}^{N} \left(\widehat{\boldsymbol{x}}(c_n;\boldsymbol{\theta}) - \boldsymbol{x}_n\right)^2. \tag{54}$$

This is known as *minimum generation error training* (MGE), and tends to improve subjective results over synthesis from ML-estimated models [74]. Again, this is achieved by considering an optimization problem more closely related to the practical problem of interest.

3.6.4 Computational Considerations

As a final remark, it is not uncommon that the eventual optimization problem one obtains cannot be solved analytically, especially for discriminative methods. Often, however, optimization techniques such as gradient descent can be applied to solve the problem iteratively. Typically, these methods take a proposed solution and improve it, such that the value of the objective function increases. Applying the procedure multiple times eventually converges on a local stationary point of the original objective [75]. Aside from gradient descent another example of this procedure is the EM-algorithm [76], which can be formulated as interleaving two optimizations of two different sets of variables [77]. In other cases, the problem can simplified or approximated to a form which can be solved more readily (e.g., by relaxing the optimization problem [78]), but the resulting solution may then not be identical to that of the original formulation.

4 Common Model Paradigms

The assumptions made when solving practical problems with the methods from 3.6 are to a large part embodied by the model set $\mathcal{M}$. Consequently, the different modelling techniques that have been proposed are almost as varied as the decision problems we may face.

This section establishes some of the major paradigms in general statistical modelling and problem solving, complete with example techniques from the different classes; the companion section 5 discusses assumptions and models specific to sequence data. Topics covered here include the distinctions between parametric and nonparametric models (section 4.1), between discriminative and generative models (section 4.2), and between probabilistic and geometric approaches (section 4.3). Methods that combine multiple models, either from a Bayesian perspective (section 4.4), using mixtures (section 4.5), or through other means (section 4.6), are also considered.

4.1 Parametric vs. Nonparametric

The distinction between parametric and nonparametric methods, introduced in section 3.5, is quite significant in practice. The ever-growing set of distributions $\mathcal{M}$ that nonparametric methods can describe as the number of samples N increases enables asymptotic convergence as $N \to \infty$ even under very weak assumptions on the generating distribution (see, for instance, [79]). On the other hand, the weak assumptions mean that larger datasets generally are necessary to attain a certain performance; [80] gives some upper bounds for performance in density estimation. In addition, computational complexity often grows superlinearly with dataset size N, whereas common parametric approaches typically are linear in N, though they may be superlinear in p, the dimensionality of the parameter space $\mathcal{P}$. Approximations can frequently be employed in nonparametric approaches to make the computational load linear in N, but at the expense of the asymptotic convergence properties; in a sense, such approximations make nonparametric methods parametric.

Parametric distributions are exceedingly common in applications. Most well-known is probably the normal or Gaussian distribution, which is a member of the important *exponential family*, consisting of distributions that can be written as

$$f_{\boldsymbol{X}}(\boldsymbol{x};\, \boldsymbol{\theta}) = h(\boldsymbol{x}) \exp\left(\boldsymbol{\eta}^{\mathsf{T}}(\boldsymbol{\theta})\, \boldsymbol{T}(\boldsymbol{x}) - K(\boldsymbol{\theta})\right). \tag{55}$$

Here, $\boldsymbol{T}$ is a vector of *natural sufficient statistics*,[8] $\boldsymbol{\eta}$ is a corresponding set of *natural parameters* (the distribution is written in *natural form* if $\boldsymbol{\eta}(\boldsymbol{\theta}) \equiv \boldsymbol{\theta}$), while $\ln K(\boldsymbol{\theta})$ acts as a normalization. This family also includes several other important continuous and discrete distributions such as the beta and gamma distributions, the von Mises-Fisher distribution, as well as the geometric and Poisson distributions. Mixture distributions, see section 4.5, are generally not in the exponential family. Student's t-distribution, in particular, can be written as an infinite mixture and is not in the exponential family.

To be in the exponential family it is additionally required that the support of the distribution is independent of $\boldsymbol{\theta}$. The set of uniform distributions on, e.g., $[0,\, \theta]$, is therefore not an exponential family. Being in the exponential family has certain advantages for parameter estimation, for instance the existence of a sufficient statistic (the Pitman-Koopman-Darmois theorem) whose dimension remains bounded as the sample size $N \to \infty$, but we shall not elaborate further on these here.

[8] Here, a *statistic* is any function of the available data, and a *sufficient statistic* is a function computable over the data which contains all information relevant for optimal parameter estimation. This means that the mutual information between the parameter and the sufficient statistic is the same as the mutual information between the parameter and the data, regardless of how the parameter is distributed [54].

The most prominent nonparametric density estimation method is arguably kernel density estimation (KDE) [81, 82], shadowed by its closely related "dual" k-nearest neighbour density estimation, although the latter does not lead to normalizable densities [83]. Other techniques include series expansions and maximum penalized likelihood methods (closely related to splines). Some discussion of different nonparametric density estimation methods is available in [59, 80].

4.2 Generative vs. Discriminative

Equation (20) shows that minimum-risk decisions only require knowledge about the conditional distribution of the decision quantity given the input. Approximate minimum-risk decisions can thus be made based on approximations of this distribution, e.g., $f_{C|\boldsymbol{X}}\left(c \mid \boldsymbol{x}; \widehat{\boldsymbol{\theta}}\right)$. A model that describes this conditional probability directly is said to be *discriminative*, since it is designed for class discrimination. The alternative is to instead approximate the full joint feature-label distribution $f_{\boldsymbol{X},C}(\boldsymbol{x}, c)$—such a model can subsequently be used for discrimination by rearranging the relations

$$f_{\boldsymbol{X},C}(\boldsymbol{x}, c) \equiv f_{\boldsymbol{X}}(\boldsymbol{x})\, f_{C|\boldsymbol{X}}(c \mid \boldsymbol{x}) \tag{56}$$

$$\equiv f_C(c)\, f_{\boldsymbol{X}|C}(\boldsymbol{x} \mid c)\,, \tag{57}$$

but is in addition capable of sampling new feature-label pairs, or new features given labels, which is useful in synthesis tasks. Such models are known as *generative* models. We note that (56) suggests that $f_{C|\boldsymbol{X}}$ may have fewer degrees of freedom than $f_{\boldsymbol{X},C}$, and thus may be easier to estimate from finite datasets, giving discriminative methods a statistical advantage.

The density models discussed in the previous section can all easily be adapted as class-conditional feature distributions $f_{\boldsymbol{X}|C}$, which, together with a categorical distribution f_C for the classes, form a full, generative model—cf. (57). The spline-based method for log-likelihood ratios in [84], in contrast, exclusively describes the conditional class distribution $f_{C|\boldsymbol{X}}$, without suggesting a joint distribution $f_{\boldsymbol{X},C}$. A curious intermediate case is k-nearest neighbour classification, which gives unnormalizable pdfs in unsupervised density estimation, but does lead to well-formed probabilities in classification [83].

The difference between using a discriminative and a generative model for a discriminative task is not substantial. When parameters in generative models are chosen using MAP or ML, it becomes possible to use the generative properties of the approach to investigate what the model has learned about the typical example, for instance by sampling. This interpretability can often be advantageous in understanding how well a certain statistical model works for a problem. If, however, the model is misspecified (the true

pdf is not in $\mathcal{M}$) a parameter estimation approach grounded in the discriminative perspective, such as MMI or MCE training, often yields better ultimate performance on the task. Aside from experimental evidence, this can be seen theoretically following [85], where it is shown that MMI is equivalent to ordinary maximum-likelihood estimation in an expanded model; the greater flexibility (increased number of parameters) of this model makes it possible to adjust to a larger variety of situations than before.

One downside of discriminative models is that they only are trained to describe the conditional distribution $f_{C|\boldsymbol{X}}$ accurately, so any joint distribution $f_{\boldsymbol{X},C}$ formed by slotting in discriminatively trained parameter values such as $\widehat{\boldsymbol{\theta}}_{\mathrm{MMI}}$ or $\widehat{\boldsymbol{\theta}}_{\mathrm{MGE}}$ into a generative family may not be meaningful. It can also happen that a generative view may increase end performance in discrimination: one example is unsupervised pre-training for deep neural networks (DNN), which is an explicit data-generation perspective that has shown recent success in initializing DNNs for discriminative tasks [86].

4.3 Probabilistic vs. Geometric

In equation (53), we saw one example that solutions to classification problems need not be derived from probabilistic principles in order for empirical risk minimization to be possible. In general, as long as the task is not explicitly to estimate a density or density parameter, there also exist nonprobabilistic solution techniques. Instead of being based on maximizing probabilities, these are generally founded on minimizing distances, and are therefore termed *geometric methods*. Because they do not describe any probabilities or distributions, geometric models do not allow sampling and can only be used discriminatively.

A simple geometric approach is exemplar-based classification, where instances are classified based on which of two class examples they are closest to in feature space, according to some distance metric. Under the Euclidean distance metric, the decision regions are then separated by a hyperplane. A more general example of this geometric notion is maximum-margin methods such as support vector machines (SVM) [87], which separate classes using a hyperplane in a high-dimensional feature space [88]. SVMs have turned out to be very strong performers on many binary classification tasks. Another geometric method, used, e.g., in prediction and forecasting, is linear regression by least squares, the "squares" being square Euclidean distances [83].

It has turned out that many geometric methods are special cases of a corresponding probabilistic model. For instance, the two-exemplar based approach above yields identical decision surfaces to classification based on a model where both classes are Gaussian distributed with identical covariance matrices. Furthermore, while the arithmetic mean is the point which minimizes the sum of square distances to all datapoints in a set, the mean is also

the maximum likelihood estimate of the location parameter and mode of a Gaussian distribution for the same dataset. The correspondence extends to finding sets of reference points that minimize the square distances to the closest reference point (through the k-means algorithm [89]), and training mixtures of Gaussian components with identical standard deviations. Probabilities of different points for a Gaussian distribution with arbitrary covariance matrix $\boldsymbol{C}$ can be mapped one-to-one to Mahalanobis distances [90]

$$d(\boldsymbol{x}, \boldsymbol{\mu}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^\intercal \boldsymbol{C}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})} \tag{58}$$

from the mean vector $\boldsymbol{\mu}$. In general, every Bregman divergence, interpretable as a kind of asymmetric distance measure, can be connected to a corresponding distribution in the exponential family [58].

Another example that highlights the connection between geometric and probabilistic methods is nearest-neighbour classification, where the estimated class is chosen as the class of the example in the training data that is closest to the input features according to some distance metric. This is obviously a geometric decision—yet the same technique can also be interpreted probabilistically, as a special case of k-nearest neighbour from before. A notable example where, on the other hand, identifying an equivalent probabilistic model has been difficult is the case of SVMs; [91] represents a more successful interpretation attempt. There exist, however, methods closely related to SVMs that are probabilistic from the ground up, e.g., relevance vector machines (RVMs) [92].

Geometric intuition can often aid the solution of machine learning problems, as well as add to the understanding of statistical methods, e.g., by considering ML-estimation as a minimal KL-divergence projection in information geometry—this is how equation (37) was derived. Furthermore, many discrimination and prediction rules, also those derived from probabilistic principles, can be formulated in terms of distances.

Probabilistic approaches, however, bring more to the table than their geometric counterparts, since they in addition explicitly model the uncertainty in the decision situation. This makes them naturally capable of estimating stochastic quantities such as the overall expected risk of a classifier, or how certain a learner is of its decision for a given input. (One caveat is that the selected models are biased towards overestimating their own performance on the training material, since they were selected among all other models for having the greatest apparent accuracy there [93, 94].) It may also be possible for probabilistic methods to flag unusual input data that falls in feature regions where the model has not seen many examples and its decisions may be inaccurate.

4.4 Fully Bayesian Approaches

Section 3.6 introduced the Bayesian notion of assigning initial (prior) probabilities to the different models in the set $\mathcal{M}$. These probabilities, together with the observations, determine the final model selected, as seen in equation (42). By taking the logarithm of the objective function,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}\left(\mathcal{D}\right) = \operatorname*{argmax}_{c \in \mathcal{C}} \left(\ln f_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}\right) + \ln f_{\{\boldsymbol{X},\, C\}|\boldsymbol{\Theta}}\left(\mathcal{D} \mid \boldsymbol{\theta}\right) \right) \tag{59}$$

$$= \operatorname*{argmax}_{c \in \mathcal{C}} \left(\ln f_{\boldsymbol{\Theta}}\left(\boldsymbol{\theta}\right) + \sum_{n=1}^{N} \ln f_{\boldsymbol{X},\, C|\boldsymbol{\Theta}}\left(\boldsymbol{x}_n,\, c_n \mid \boldsymbol{\theta}\right) \right), \tag{60}$$

we see that the log prior $\ln f_{\boldsymbol{\Theta}}$ can be seen as an additive regularizer of the data log-likelihood $\ln f_{\{\boldsymbol{X},\, C\}|\boldsymbol{\Theta}}$, biasing the estimate towards certain regions of parameter space, with the additional property that $f_{\boldsymbol{\Theta}}$ also has a probabilistic interpretation. The Bayesian framework has more implications than this, however. Since the parameter $\boldsymbol{\Theta}$ in the Bayesian framework is a random variable, one can marginalize it out in many decision situations: both in supervised settings such as classification, where we can form

$$\widehat{c}\left(\boldsymbol{x} \mid \mathcal{D}\right) = \operatorname*{argmin}_{c \in \mathcal{C}} \mathbb{E}\left(L\left(c,\, C\right) \mid \boldsymbol{X} = \boldsymbol{x},\, \mathcal{D}\right) \tag{61}$$

$$= \operatorname*{argmin}_{c \in \mathcal{C}} \sum_{c' \in \mathcal{C}} L\left(c,\, c'\right) f_{C|\boldsymbol{X},\, \{\boldsymbol{X},\, C\}}\left(c' \mid \boldsymbol{x},\, \mathcal{D}\right), \tag{62}$$

using the law of total probability to compute

$$\widehat{f}_{C|\boldsymbol{X},\, \{\boldsymbol{X},\, C\}}\left(c' \mid \boldsymbol{x},\, \mathcal{D}\right) = \int f_{C|\boldsymbol{X},\, \boldsymbol{\Theta}}\left(c' \mid \boldsymbol{x},\, \boldsymbol{\theta}\right) f_{\boldsymbol{\Theta}|\{\boldsymbol{X},\, C\}}\left(\boldsymbol{\theta} \mid \mathcal{D}\right) \mathrm{d}\boldsymbol{\theta}, \tag{63}$$

as well as in unsupervised density estimation, where one obtains the *predictive distribution*

$$\widehat{f}_{\boldsymbol{X}|\{\boldsymbol{X}\}}\left(\boldsymbol{x} \mid \mathcal{D}_{(\mathrm{us})}\right) = \int f_{\boldsymbol{X}|\boldsymbol{\Theta}}\left(\boldsymbol{x} \mid \boldsymbol{\theta}\right) f_{\boldsymbol{\Theta}|\{\boldsymbol{X}\}}\left(\boldsymbol{\theta} \mid \mathcal{D}_{(\mathrm{us})}\right) \mathrm{d}\boldsymbol{\theta}. \tag{64}$$

(We assume samples are conditionally independent given $\boldsymbol{\Theta}$.) Methods like the above, which take into account the entire posterior distribution $f_{\boldsymbol{\Theta}|\{\boldsymbol{X},\, C\}}$ or $f_{\boldsymbol{\Theta}|\{\boldsymbol{X}\}}$ for the parameter, are known as *fully Bayesian*, in contrast to MAP or maximum likelihood, which are based on *point estimates* (single values) $\widehat{\boldsymbol{\theta}}_{\mathrm{MAP}}$ or $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$.

The fully Bayesian strategy of incorporating all possibilities into decisions lends some protection against situations where there is substantial uncertainty about what parameter value that is appropriate. In such cases, picking a single value through point estimation can be highly misleading, as will be discussed in section 6.3. Weighting models together as in (64)

also enables the approach to represent certain distributions not contained in $\mathcal{M}$, but within the convex hull spanned by $\mathcal{M}$ [95]. On the other hand, marginalizing over all parameter values complicates the mathematics of fully Bayesian approaches, and often necessitates approximations, e.g., so-called variational inference [96], or computationally intensive simulation methods such as Markov chain Monte Carlo (MCMC) [97, 98]. Sometimes the distributions formed by summing over models in $\mathcal{M}$ can be physically unreasonable, for instance leading to speech models having more formants (vocal tract resonances) than natural human speech can contain.

Despite the conceptual appeal, Bayesian methods have a somewhat controversial history in statistics. This is partially because the choice of prior distribution often cannot be motivated on theoretical grounds, and partially because of the subjectivist philosophical interpretations that are sometimes attached to Bayesian methods.

In certain applications a prior distribution can be inferred from previously available information. When creating a speaker-specific speech recognizer, for instance, we can use the distribution of voice parameters seen in a previous training material of many speakers as our prior. Sometimes, however, there is a desire to select a "*non-informative prior*," which represents having no knowledge about the distribution. This condition of perfect ignorance has proved difficult to describe mathematically. An early proposal, attributed to Bayes and Laplace, is the *principle of indifference*: assume a uniform prior over all parameter values. Unlike ML-estimation, which only is a point estimate, the distribution given by this rule depends on the parameterization for continuous variables. This is unappealing, since it is hard to argue which representation should be preferred for a fundamentally unobservable quantity such as a distribution parameter.

Other proposals for objective prior selection exist, notably the so-called *Jeffreys prior* [99]

$$f_{\text{Jeff}}(\boldsymbol{\theta}) \propto \sqrt{\det \mathbb{E}\left(\left(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{X}|\boldsymbol{\Theta}}(\boldsymbol{X} \mid \boldsymbol{\theta})\right)\left(\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{X}|\boldsymbol{\Theta}}(\boldsymbol{X} \mid \boldsymbol{\theta})\right)^{\mathsf{T}} \mid \boldsymbol{\Theta} = \boldsymbol{\theta}\right)}, \quad (65)$$

which is independent of representation and only depends on the model $f_{\boldsymbol{X}|\boldsymbol{\Theta}}$. While attractive, the Jeffreys prior is not a panacea. Frequently, it is not a normalizable probability distribution (it has infinite integral over the parameter space), in which case it is said to be *improper*. This may be acceptable as long as the posterior distribution used for making decisions based on data is a proper distribution, which is often the case. However, there are situations (notably mixture models [100]) where the posterior is always improper. There are also arguments that the Jeffreys prior is unsatisfactory for multidimensional parameters [101].[9]

[9] An alternative approach to non-informative priors, investigated, e.g., in [102], is to formulate *weakly informative priors*: prior distributions designed with the intention

In recent years, the prior distribution has come to be seen as more of an asset and less of a liability, and Bayesian methods have experienced increased use in applications. For one thing, it can be argued that non-Bayesian methods also make assumptions, but that these simply are not stated as explicitly. Having the assumptions up front, and formulated probabilistically as a Bayesian prior distribution, provides a clearer picture of what is being taken for granted. The prior also provides a very explicit entry-point for other available information about the expected behaviour of the variable(s) in question. This can be used to (gently or strongly) bias results in the right direction, and possibly increase performance compared to methods that make weaker assumptions.

In contrast to the use of Bayesian methods as an applied tool in specific situations, the philosophical interpretation of Bayesian probabilities as subjective probabilities (as in, e.g., [103]) remains controversial. For some discussion see [104] and the accompanying commentary.

4.5 Mixture Models

Fully Bayesian methods are part of a broad category of approaches where more than one model is employed to solve a single problem. A general term for such techniques is *ensemble methods*. From a high-level perspective, the idea of combining different approaches and opinions is compelling, as it is frequently used in real-life decision making, e.g., panels of experts, juries, and scientific research groups—cf. [105].

Perhaps the simplest class of ensemble models is *mixture models*, where the pdf is a weighted sum of different models. Most well-known are *Gaussian mixture models* (GMMs), which have the form

$$f_{\boldsymbol{X}}\left(\boldsymbol{x};\,\boldsymbol{\theta}\right)=\sum_{k=1}^{K}p_k\mathcal{N}\left(\boldsymbol{x};\,\boldsymbol{\mu}_k,\,\boldsymbol{C}_k\right), \tag{66}$$

where $\mathcal{N}\left(\boldsymbol{x};\,\boldsymbol{\mu},\,\boldsymbol{C}\right)$ denotes a normal-distribution pdf with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{C}$ evaluated at $\boldsymbol{x}$, while $\boldsymbol{\mu}_k$ and $\boldsymbol{C}_k$ are parameters for component k, with nonnegative component probabilities p_k satisfying $\sum_{k=1}^{K}p_k=1$. K, the number of components, is typically fixed (nonrandom). Many component shapes other than Gaussians are possible, for instance mixtures of beta distributions for data confined to an interval [106].

Mixture-model samples are most easily understood as generated through a two stage process: For every datapoint $\boldsymbol{X}_n$, a discrete random variable Q_n is first drawn from the distribution $f_Q\left(k\right)=p_k$. The realization $k=q_n$ then tells us which component distribution $f_{\boldsymbol{X}|Q}\left(\boldsymbol{x}_n\mid q_n\right)$ to use when generating the observed $\boldsymbol{x}$-value. Q_n here is thus an unobservable (hidden)

to exert little influence over the conclusions of an experiment while being sufficiently informative to avoid the pathologies of standard non-informative priors.

random variable that helps generate the nth observation. Such unobservable variables, which are sampled anew for each datapoint, are known as *latent variables*, in contrast to Bayesian parameters $\boldsymbol{\Theta}$, which have the same, though unobservable, value for the entire dataset. This "hidden selector" structure also suggests that mixture models may be particularly appropriate in cases where the studied distribution is composed of many different subpopulations with potentially different properties.

Mixture models are superficially similar to fully Bayesian approaches in that the predictive data distributions (64) and (66) in both cases are a weighted combination of different pdfs from simpler models, often from the exponential family. However, the distributions resulting from the two methods often can show quite different asymptotic learning behaviour (that is, when we let the training set size N grow). For a Bayesian model, the likelihood term in (60) will eventually swamp the prior (wherever the prior is nonzero) as the amount of data becomes large. If, for example, $\mathcal{M}$ is a set of distinct categorical distributions, the posterior probability will eventually come to strongly favour the one among these that is closest to the true data distribution. The resulting predictive distribution (64) will be virtually indistinguishable to one of the distributions in $\mathcal{M}$. A mixture model, in contrast, may retain all components and keep them distinct even as $N \to \infty$, and can therefore more easily converge on a variety of shapes that are different and more complex than any of the components. In cases where the true generating distribution is not in $\mathcal{M}$, the standard Bayesian approach with conjugate priors often converges on a compromise model which best explains the entire dataset (but see also [95]), while mixture models, roughly speaking, partition the state space $\mathcal{X}$ into K pieces, and use a separate model for each part. However, the two ideas can be combined to form *Bayesian mixture models*, e.g., [83, 106], in which the component probabilities and parameters are random variables.

Theoretical results show that mixture models can approximate any distribution well, given a sufficient number of components [83]. Models with a large number of components may require substantial amounts of data to be accurate, however. Somewhat surprising, then, is that Bayesian mixture models with an effectively *infinite* number of components can be formulated and made practical [107]; these are of particular interest when it is difficult to propose an adequate number of components K a priori.

4.6 Ensemble Methods

While Bayesian approaches and mixture models have a long history in statistics, other kinds of ensemble models have recently surged in popularity due to their consistently strong showings in prediction contests such as the widely publicized Netflix Prize. Three important approaches—bagging, boosting, and stacking—are discussed below:

Bagging [108] stands for *bootstrap aggregating*, and is the simplest of the three techniques. In this method, the bootstrap [109] (resampling from $\mathcal{D}$ with replacement) is applied to create a large number of different datasets based on the original data. Individual problem solvers with the same structure are then trained on the resampled datasets, leading to a potentially different parameter estimate for each. To form a decision, the outputs of the trained problem solvers are combined using, e.g., averaging (minimizing the squared error on continuous spaces) or simple majority voting (discrete spaces).

Compared to bagging, *boosting*, e.g., [110, 111], is a more directed method specifically for classification. Boosting trains a sequence of classifiers on different weightings of the available data. After each model is trained, the data is reweighted so that incorrectly classified examples are given more weight when the next classifier is created. This creates a sequence of models that in some sense complement each other. To obtain the final decision, the different learners are typically weighted together depending on their classification accuracy. There are some impressive theoretical results showing that boosting asymptotically can approach optimal classification accuracy, even if the component classifiers are only marginally different from chance [112].

Stacking, finally, is the idea to create a pool of learners, and then use another, higher-level learner to combine their outputs [113]. Notably, the two best-performing teams from the Netflix Prize competition both used large, stacked ensembles for their predictors [114]. Artificial neural networks can be seen as an extreme example of stacking, where each layer is an ensemble of simple classifiers (perceptrons) with linear decision surfaces.

Generally, ensemble methods appear to perform better the more different the component models are, as this allows them to complement each other more effectively. Bayesian methods are therefore sometimes considered relatively weak ensembles [114], as all component models share the same structure and the parameter distribution tends to concentrate on a small subset of the possible models once sufficient amounts of data becomes available. Nevertheless, the ability to average over multiple models makes it possible for Bayesian approaches to converge on distributions not present in $\mathcal{M}$ [95]. This is appealing, as many traditional convergence and optimality results, e.g., for maximum likelihood, only consider the situation where the true distribution is in $\mathcal{M}$.

5 Modelling Sequence Data

The model paradigms from section 4 all apply to sequence models as well, but sequence models have the additional complication that samples also may depend on each other. Handling this requires assumptions on the na-

ture of how samples interact, and models that express and quantify the dependences. This section introduces stationarity (section 5.1) and ergodicity (section 5.2) as two central assumptions in sequence data models, and outlines Bayesian networks as a framework for visualizing dependence structures (section 5.3). Thereafter, Markov models (section 5.4), hidden Markov models (section 5.5), and various combinations (section 5.6) are described, along with example techniques from each class. Finally, we sketch how the models can be applied to situations that do not satisfy stationarity or ergodicity (section 5.7), and touch upon some alternative paradigms which do not fit well in standard state-based sequence modelling frameworks (section 5.8), for instance detector-based techniques.

5.1 Stationarity

For sequence data, as considered in this thesis, there are two commonly-seen standard assumptions: that data series are stationary, and that they are ergodic. Stationarity can be seen as a way to make the dependences of the process uniform and structured, while ergodicity is a concept for localizing the influence of the dependences. In this section we explore stationarity in greater detail, with a discussion of ergodicity reserved for section 5.2.

Stationarity is the notion that the probabilistic properties of samples from different times are comparable. Just as it is commonly assumed that the laws of nature are invariant over time, a stationary process is governed by the same distribution regardless of what the time-variable t is. More specifically, a sequence is *strictly stationary* if, for any sequence of observations $\underline{\boldsymbol{x}}$, the probability (or pdf for continuous-valued processes) of that sequence appearing in a pattern is identical at all times—that is, the process satisfies

$$f_{\underline{\boldsymbol{X}}_\tau}\left(\underline{\boldsymbol{x}}\right) \equiv f_{\underline{\boldsymbol{X}}_{\tau+\ell}}\left(\underline{\boldsymbol{x}}\right) \tag{67}$$

for all ℓ, as long as the time indices

$$\tau = \{t_1, \ldots, t_T\} \tag{68}$$

$$\tau + \ell = \{t_1 + \ell, \ldots, t_T + \ell\} \tag{69}$$

for the pattern both are subsets of the index set $\mathcal{T}$. In other words, any substring $\underline{\boldsymbol{x}}$ has the same probability of appearing at any time.

Stationarity also comes in other forms than strict stationarity. A process is *weakly stationary* or *wide-sense stationary* (WSS) if its second-order properties do not depend on time,

$$\mathbb{E}\left(\boldsymbol{X}_1\right) = \mathbb{E}\left(\boldsymbol{X}_t\right) \tag{70}$$

$$\mathbb{E}\left(\boldsymbol{X}_1 \boldsymbol{X}_\ell^\intercal\right) = \mathbb{E}\left(\boldsymbol{X}_t \boldsymbol{X}_{t+\ell}^\intercal\right) \tag{71}$$

for all valid t and lags ℓ; in other words, means and covariances are time-independent. Note that this does not require the distributions at different times to be the same.

The advantage of stationarity is that is simplifies learning. Under strong stationarity, there essentially is a single, translated distribution to learn, rather than a new distribution for each time t. In the case of weak stationarity, the distributions at individual times may differ, but there are still invariant means and variances that can be identified.

If the properties of the process (its distribution) change slowly with time or otherwise are confined to a small set of similar distributions, a model that assumes stationarity can still yield good results on a dataset. Speech signals, for instance, change relatively slowly on the millisecond scale, so stationary models are often sufficiently accurate over intervals of 20 ms or so.

In other cases, nonstationarity may have to be incorporated into the model itself. Seasonal dependences, which are common in time series of geological, biological, or societal origin, can be expressed through *cyclostationary processes*. These can be thought of as a set of T_p alternating (and possibly dependent) stationary processes, T_p being the period, where sample $\boldsymbol{X}_t$ is generated from process number $t \bmod T_p$. Cyclostationary models can be created by letting process parameters (e.g., the mean) vary according to a given pattern [115]. It may then be possible to remove the nonstationary aspects of the data and obtain a stationary residual process. There are also nonparametric approaches to seasonality, for instance the climate data example from [27]. Common cycle periods are days and years; tides also have a monthly component, while weekly patterns often appear in data affected by human activity. Similar techniques can also be applied to describe ongoing trends in the data, for instance by adding terms that model linear growth.

Many deviations from stationarity are, unlike cyclostationarity, difficult to describe in a structured manner. One example is one-shot, transient processes, i.e., finite-duration phenomena. To accurately estimate the behaviour of such processes, it is typically necessary to have several independent realizations of the process. A deeper look at methods for working with nonstationary time series can be found in [116].

5.2 Ergodicity

Ergodicity is often presented as a short-memory property. Essentially, it is the idea that long samples from a process will be representative of all the possible behaviours of the process. More formally, if τ and τ' are two sets of time indices and $(\underline{\boldsymbol{X}}_\tau = \underline{\boldsymbol{x}})$ and $(\underline{\boldsymbol{X}}_{\tau'} = \underline{\boldsymbol{x}}')$ are two positive-measure events, meaning that the two output patterns can be generated by the process, then the process is *ergodic* if the event $\left(\underline{\boldsymbol{X}}_\tau = \underline{\boldsymbol{x}} \cap \underline{\boldsymbol{X}}_{\tau'+\ell} = \underline{\boldsymbol{x}}'\right)$ also has positive measure for some ℓ. In other words, all possible patterns can also co-occur

in the same realization, at least if they are sufficiently far apart. (This definition is a slight simplification of that given in [117].)

Though, technically, ergodicity can be defined though a measure-preserving transform without requiring stationarity, stationarity is commonly assumed before ergodicity is considered. If a process is both stationary and ergodic, which is common to assume in applications, this means that the behaviour is time-invariant, and that time average of any integrable function $g(\boldsymbol{x})$ converges to the ensemble average (the standard expected value of the function),

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} g(\boldsymbol{X}_t) = \int g(\boldsymbol{x})\, f_{\boldsymbol{X}_1}(\boldsymbol{x})\, d\boldsymbol{x} \tag{72}$$

almost everywhere [118]. For such a process, a single sample sequence, if long enough, suffices to learn the behaviour of the process to an arbitrary (given) accuracy. Ergodicity thus implies that the data satisfies a kind of law of large numbers for dependent samples.

Of course, some ergodic systems converge to the ensemble average faster than others, and between independence and simple ergodicity exist several different kinds of so-called *mixing* (weak and strong). This is reminiscent of the various strengthened versions of the law of large numbers, including the central limit theorem. Exactly how quickly an ergodic system forgets can be quantified by *mixing coefficients* or by the *autocovariance time* T_{Cov}. If the process $\underline{X}$ is weakly stationary with mean value μ and variance σ^2, the autocovariance time is defined through

$$\sum_{\ell=1}^{\infty} |\mathrm{Cov}(X_t,\, X_{t+\ell})| = \sigma^2 T_{\mathrm{Cov}}, \tag{73}$$

assuming the sum is finite. This number affects how quickly the sample mean

$$\widehat{\mu}_N(\underline{x}) = \frac{1}{N} \sum_{t=1}^{N} x_t \tag{74}$$

converges to the true mean $\mu = \mathbb{E}(X_t)$—specifically one can show [119] that

$$\mathbb{E}\left(L_Q\left(\widehat{\mu}_N(\underline{X}),\, \mu\right)\right) = \mathbb{E}\left(\left(\widehat{\mu}_N(\underline{X}) - \mu\right)^2\right) \tag{75}$$

$$\leq \frac{\sigma^2}{N}\left(1 + 2T_{\mathrm{Cov}}\right). \tag{76}$$

This result can be used to extend the law of large numbers, and also quantifies how learning from dependent data requires more samples than learning

from independent samples due to the redundancy, as was mentioned in section 2.4. More precisely, an accuracy that is achievable with N datapoints for independent data (which has $T_{\text{Cov}} = 0$) now requires $N\left(1 + 2T_{\text{Cov}}\right)$ samples instead, an increase by a constant factor. It is as if one is working with uncorrelated datapoints spaced $2T_{\text{Cov}}$ samples apart; this explains the interpretation of T_{Cov} as an amount of time.

It is a theorem that non-ergodic stationary processes can be decomposed into a set of component processes, each of which is ergodic [120]. Essentially, a process is either free to show all possible behaviours in a single realization, or restricted to pick a subset of them; within such a subset, all behaviours associated with the subset can be observed in a single realization. As a consequence, ergodicity is not always necessary for prediction: if we are asked to predict the likely future observations of a long sample, the sample will be relevant to the current ergodic component, and that is the only ergodic component that matters for the prediction task at hand.

5.3 Bayesian Networks

We have seen that, in order to be able to efficiently learn a process from its output data, variable dependences must be localized (ergodicity). Learning also becomes much easier if the dependences have some form of invariant structure (stationarity). Fortunately, many observed processes satisfy these criteria to a large degree.

In the following subsections, we introduce the two dominant paradigms for modelling stationary and ergodic sequence data. The methods can be extended to also handle certain types of nonstationarity (particularly trends and seasonality) and to model finite-duration sources.

When classifying sequence models, dependences among variables will be illustrated in the language of Bayesian networks [121, 122], which are a type of graphical model. More specifically, a Bayesian network is a directed acyclic graph (DAG) which expresses the causal dependences among a set of random variables.[10] To be concrete, let $\mathcal{V} = \{1, \ldots, N\}$ be a topologically ordered indexing of the nodes in a DAG, so that the set of vertices $\mathcal{E}$ satisfies $\mathcal{E} \subseteq \{(i, i') \in \mathcal{V} \times \mathcal{V} : i' > i\}$. Let further $\mathcal{J}(i) \subseteq \mathcal{E}$ be the edges pointing to i. Also let $\boldsymbol{X}_{\mathcal{V}} = \{\boldsymbol{X}_i : i \in \mathcal{V}\}$ be a set of possibly dependent random variables. The graph $(\mathcal{V}, \mathcal{E})$ is then a *Bayesian network with respect to* $\boldsymbol{X}_{\mathcal{V}}$

[10] Markov random fields (MRFs), one alternative to Bayesian networks, utilize undirected graphs, and can therefore be symmetric in the relations between variables. This is useful, e.g., in models of atomic interaction such as the Ising model and spin glasses [123]. Reduced Boltzmann machines, common building blocks in deep neural networks, are also expressed as MRFs [124, 125]. Hybrid models which combine both paradigms exist as well [126, 25].

if the joint probability distribution of $\boldsymbol{X}_{\mathcal{V}}$ satisfies

$$f_{\boldsymbol{X}_{\mathcal{V}}}\left(\boldsymbol{x}_{\mathcal{V}}\right) \equiv \prod_{i \in \mathcal{V}} f_{\boldsymbol{X}_i | \boldsymbol{X}_{\mathcal{J}(i)}}\left(\boldsymbol{x}_i \mid \boldsymbol{x}_{\mathcal{J}(i)}\right). \tag{77}$$

(Note that this is different from the factorization

$$f_{\boldsymbol{X}_{\mathcal{V}}}\left(\boldsymbol{x}_{\mathcal{V}}\right) \equiv \prod_{i \in \mathcal{V}} f_{\boldsymbol{X}_i | \boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i-1}}\left(\boldsymbol{x}_i \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{i-1}\right) \tag{78}$$

which applies to all sets of random variables.)

We see that the absence of an edge between i and $i' > i$ in a Bayesian network means that the corresponding variables are *conditionally independent*, given the variables in $\mathcal{J}\left(i'\right)$—that is, the joint pdf factors as

$$\begin{aligned} &f_{\boldsymbol{X}_i, \boldsymbol{X}_{i'} | \boldsymbol{X}_{\mathcal{J}(i')}}\left(\boldsymbol{x}_i, \boldsymbol{x}_{i'} \mid \boldsymbol{x}_{\mathcal{J}(i')}\right) \\ &\qquad \equiv f_{\boldsymbol{X}_i | \boldsymbol{X}_{\mathcal{J}(i')}}\left(\boldsymbol{x}_i \mid \boldsymbol{x}_{\mathcal{J}(i')}\right) f_{\boldsymbol{X}_{i'} | \boldsymbol{X}_{\mathcal{J}(i')}}\left(\boldsymbol{x}_{i'} \mid \boldsymbol{x}_{\mathcal{J}(i')}\right) \end{aligned} \tag{79}$$

when $i \notin \mathcal{J}\left(i'\right), i < i'$. Though they may not look like much, such independences can substantially simplify computations.[11]

We shall see that a useful notion of localized dependences is models that have Bayesian networks on $\mathcal{V} \supseteq \mathcal{T}$ whose causal dependences, i.e., the sets $\mathcal{J}\left(i\right)$, do not grow indefinitely with $i \in \mathcal{V}$. For probability distributions that represent stationary processes, the sets $\mathcal{J}\left(i\right)$ generally have some recurring structure to them, such that the causal dependences at any point simply are translated versions of the dependences at any other point. Such a Bayesian network is known as a *dynamical Bayesian network*.

Although general inference in a Bayesian network is NP-hard to even approximate [128], the factorization in (77) suggests a sequential computation such that sampling or probability computation for sets of contiguous variables starting at $i = 1$ can be performed efficiently. This property is important for efficient computations with the models presented here.

5.4 Markov Processes

Possibly the simplest assumption that can be made when creating a sequence-data model is that the generating process has limited-range memory, so that only the most recent data samples are informative for predicting future behaviour, assuming the underlying model is known. This notion is formalized by the well-known Markov property

$$f_{\boldsymbol{X}_t | \underline{\boldsymbol{X}}_{-\infty}^{t-1}}\left(\boldsymbol{x}_t \mid \underline{\boldsymbol{x}}_{-\infty}^{t-1}\right) \equiv f_{\boldsymbol{X}_t | \underline{\boldsymbol{X}}_{t-p}^{t-1}}\left(\boldsymbol{x}_t \mid \underline{\boldsymbol{x}}_{t-p}^{t-1}\right); \tag{80}$$

[11] An important caveat is that dependences in the network change depending on whether or not the value of a variable is known, though simple algorithms exist to take this into account, e.g., [127].

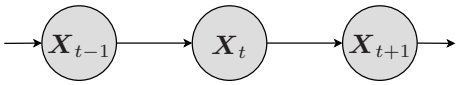

Figure 1: Bayesian network representing first-order Markov model.

any process that satisfies this for all t is said to be a *Markov process of order* p.

Relation (80) implies that future evolution is conditionally independent of the past, given the current *context* or *state* $\underline{\boldsymbol{x}}_{t-p}^{t-1}$.[12] Given $\underline{\boldsymbol{x}}_{t-p}^{t-1}$, we can recursively apply (80) to find the joint probability distribution of any future samples. The state is thus what is known as a sufficient statistic for prediction. In Bayesian network terminology, we have $\mathcal{J}(t) = \{t-p, \ldots, t-1\}$. The associated between-variable dependences (and conditional independences) are represented graphically in figure 1 for $p = 1$. As a special case, one can assume that a process has no memory at all and $p = 0$, in which case the data is IID.

5.4.1 Markov Chains

Discrete-valued Markov processes are often known as *Markov chains*. (Some authors use this term to refer to continuous-valued processes, but this text will not.) Stationary Markov chains on finite alphabets (meaning $|\mathcal{X}| < \infty$) are extremely common as language models in speech and text modelling. A highly appealing property is that the general *next-step* or *transition distribution* $f_{X_p|\underline{X}_1^{p-1}} : \mathcal{X}^{p+1} \to [0, 1]$ of these models can be specified using only $|\mathcal{X}|^{p+1}$ numbers (i.e., parameters), which is polynomial in the alphabet size. The values of $f_{X_p|\underline{X}_1^{p-1}}$ are called *transition probabilities*.

Any finite-order Markov chain on $\mathcal{X}$ can be converted to a first-order Markov chain over the alphabet $\mathcal{X}^p$. For first-order Markov chains on finite alphabets with time-independent next-step distribution, the transition probabilities can be arranged into a transition matrix $\boldsymbol{A} \in [0, 1]^{|\mathcal{X}|\times|\mathcal{X}|}$ with elements

$$(\boldsymbol{A})_{ii'} = a_{ii'} = f_{X_2|X_1}(i' \mid i). \tag{81}$$

Despite the capital font, $\boldsymbol{A}$ is typically not a random quantity; the capital here merely highlights that it is a matrix. Note that some texts define transition matrices as $\boldsymbol{A}^\intercal$, the transpose of our $\boldsymbol{A}$.

[12] In this view, the *state* is essentially a random variable which, if known, decouples the future evolution of a process from its past. As will become apparent in section 5.5, such a state need not take values on $\mathcal{X}$. The term "state space" will therefore sometimes be used to refer to a space other than $\mathcal{X}$, depending on what we mean by the "state" of the process under consideration.

The simple idea of collecting transition probabilities in a matrix is surprisingly powerful, in that many meaningful properties or relevant probabilities of Markov chains can be expressed in linear algebraic terms using $\boldsymbol{A}$. Since the elements on each row of $\boldsymbol{A}$ form a probability distribution, one for example has that

$$\boldsymbol{A}\mathbf{1} = \mathbf{1}, \tag{82}$$

where $\mathbf{1}$ is a column vector of all ones.[13] $\mathbf{1}$ is thus a right eigenvector with eigenvalue 1, which necessarily is the largest eigenvalue of $\boldsymbol{A}$. Many important properties of stationary and ergodic models can then be derived from the Perron-Frobenius theorem [129, 130]. If $f_{X_t}(i)$ is a stationary distribution of the Markov chain and the vector $\boldsymbol{\pi}$ has elements $\pi_i = f_{X_t}(i)$, we have

$$\boldsymbol{A}^\intercal \boldsymbol{\pi} = \boldsymbol{\pi}, \tag{83}$$

so $\boldsymbol{\pi}$ is a left eigenvector of $\boldsymbol{A}$. If this eigenvector is unique (the eigenvalue 1 has multiplicity 1) and $\boldsymbol{\pi} > \mathbf{0}$, the Markov chain is ergodic. The *spectral gap* of $\boldsymbol{A}$ (the difference between the moduli of the two largest eigenvalues) bounds the asymptotic decay rate towards the stationary distribution $\boldsymbol{\pi}$ for a Markov chain with an arbitrary starting state.

5.4.2 Continuous-Valued Markov Models

Continuous-valued data with stationary and finite Markovian dependences can be modelled using numerous different techniques. Most prominent are perhaps standard *linear autoregressive models* (AR models), where new values are generated as a linear combination of recent observations plus some weakly stationary, independent zero-mean driving noise $\underline{\boldsymbol{\mathcal{E}}}$,[14] i.e.,

$$\boldsymbol{X}_t = \sum_{\ell=1}^{p} \boldsymbol{A}_\ell \boldsymbol{X}_{t-\ell} + \boldsymbol{\mathcal{E}}_t \tag{84}$$

$$f_{\boldsymbol{X}_{p+1}|\underline{\boldsymbol{X}}_1^p}\left(\boldsymbol{x}_{p+1} \mid \underline{\boldsymbol{x}}_1^p\right) = f_{\boldsymbol{\mathcal{E}}}\left(\boldsymbol{x}_{p+1} - \sum_{\ell=1}^{p} \boldsymbol{A}_\ell \boldsymbol{x}_{p+1-\ell}\right), \tag{85}$$

where $\boldsymbol{A}_\ell$ are (typically nonrandom) matrices of auto-regressive parameters. These are models for which linear prediction based on past values is optimal. Certain conditions apply on $\{\boldsymbol{A}_\ell\}_{\ell=1}^p$ and the variance of $\boldsymbol{\mathcal{E}}_t$ for the process to be stable and stationary. Extending the model to a nonzero mean is straightforward.

[13] Nonnegative real matrices satisfying (82) are said to be *right stochastic*, which is an unfortunate term as the matrix itself need not be a stochastic quantity. Matrix-valued random variables are instead known as *random matrices*.

[14] Think of this notation as "capital epsilon," with $\boldsymbol{\varepsilon}$ representing a specific realization.

Autoregressive models are sometimes introduced from a geometric perspective, as methods for minimizing the mean square prediction error. Predicting the conditional mean of the next datapoint then becomes a linear regression problem. These geometric models have straightforward probabilistic analogues where samples are perturbed (driven) by, for instance, IID zero-mean Gaussian noise $\underline{\boldsymbol{\mathcal{E}}}$ around the geometric predictions. Such a model leads to one additional parameter to estimate, namely the noise standard deviation, but in turn provides a generative, probabilistic model of the data. As with the Gaussian distribution examples in section 4.3, the mean of the probabilistic and the geometric predictions coincide, so estimates of the remaining parameters are the same in the two interpretations.

Autoregressive moving-average models (ARMA) [131] are also very common in applications. These are a minor generalization of AR models where the noise is a moving-average process, as in

$$\boldsymbol{X}_t = \boldsymbol{\mu} + \sum_{\ell=1}^{p} \boldsymbol{A}_\ell \left(\boldsymbol{X}_{t-\ell} - \boldsymbol{\mu}\right) + \sum_{\ell=0}^{o} \boldsymbol{B}_\ell \boldsymbol{\mathcal{E}}_{t-\ell}, \tag{86}$$

o being the order of the moving average, $\boldsymbol{\mu}$ being the process mean parameter, and $\{\boldsymbol{B}_\ell\}_{\ell=0}^{o}$ being parameters of the moving average. It is generally assumed that $\boldsymbol{B}_0 = \boldsymbol{I}$, the identity matrix.

AR and ARMA models are often among the first techniques a practitioner will apply to a new problem. The Wold decomposition [132] shows that any weakly stationary and ergodic process can be decomposed into a (possibly infinite-order) linear ARMA process driven by a white noise process, meaning a process which is weakly stationary and has samples which are mutually independent but not necessarily IID. This decomposition can be seen as a motivation for the widespread use of truncated (i.e., finite-order) linear ARMA models in applications.

For more complex data, nonlinear approaches are generally of interest. A simple trick is to transform the data, in the hope that the process becomes more linear, for instance by taking the logarithm of dataseries which experience multiplicative updates. If this is not sufficient, a plethora of nonlinear models exists for the analyst to choose from. One may for example consider piecewise-linear, regime-switching approaches such as self-exciting threshold autoregressive (SETAR) models [133], when driven by a function of the context variable, or geometric methods such as (non-recurrent) time-delay neural networks for recognition [134] or kernel-based AR models for prediction [135, 136], to name a few different alternatives.

Common to all these models is the fact that they are parametric, and thus only can represent a restricted family of continuous-valued Markov processes of a given order. Somewhat surprisingly, perhaps, there exist procedures which are capable of learning much wider classes of continuous-valued Markov models, e.g., [137]. The idea, also used as a component

of thesis paper [138], is to use KDE for nonparametric estimation of the Markov transition density $f_{\boldsymbol{X}_t|\underline{\boldsymbol{X}}_{t-p}^{t-1}}$.

5.4.3 Variable-Order Models

In the above treatment, the order p is treated as given and fixed. In practice, it is not known what p is, or even if it has a finite value. As models with low p nearly always constitute special cases of models with higher p, one may be tempted to choose large-order models, but these have many parameters and can be difficult to learn, cf. section 6.3. A more refined approach is to treat p as a parameter and attempt to estimate it, i.e., perform order selection. A review of some order-selection criteria is provided in [49]; cross-validation techniques are a popular choice in applications. See also section 6.3.

For Markov chains, there also exist methods that let the order of the model depend adaptively on the latest observations (the context): sometimes many symbols into the past have to be evaluated before the state can be uniquely determined, whereas at other times only a few of the most recent observations may suffice. This is essentially the same as grouping certain strings in $\mathcal{X}^p$ whose latest symbols are similar, and assuming that their statistics are the same. These descriptions are known as *variable-order Markov models* (VOMs) or *variable-length Markov models* (VLMMs). The order required by each context can be adaptively learned from data in various ways, for instance following [139].

Variable-order techniques are also referred to as *context-tree methods*, since contexts can be arranged in a tree, with observed symbols on the branches and where each leaf corresponds to a specific predicted distribution for the next sample. A more sophisticated embodiment of the same idea is the causal state learning algorithm in [140, 141], which compactly can represent certain strictly sofic processes [142], where some contexts may have essentially infinite order. Specifically, an unbounded amount of lookback may be required for a strictly sofic process in order to uniquely identify the state, though, once a sufficiently long history is provided, the number of possible distributions for future samples is finite.

Instead of simply selecting a single order, or a single order per context, it is also possible to combine models of different orders. This results in ensemble models of various sorts, with the potential to provide greater predictive accuracy; see section 4.6. The (binary) *context-tree weighting* (CTW) method in [143] is an example of a weighted ensemble of Markov chains of varying order.

5.5 Long-Memory Models

For processes with long-range dependences, a Markov model may require high order, many parameters, and lots of data to provide a reasonable de-

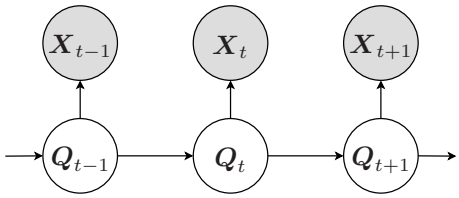

Figure 2: Bayesian network representing a first-order hidden-state model. Shaded nodes represent observables while other nodes are unobserved latent variables.

scription. A more common approach to create models with long memory is to let the state of the process be a latent variable $\boldsymbol{Q}_t$. This latent variable generates the data we see at time t, but is itself not directly observable. We call this a *hidden-state model*. Like before, we let the state follow a finite-order Markovian process, i.e., satisfy (80), so that the entire model may be specified by the two distributions $f_{\boldsymbol{X}_t|\boldsymbol{Q}_t}$ and $f_{\boldsymbol{Q}_t|\underline{\boldsymbol{Q}}_{t-p}^{t-1}}$. This leads to a Bayesian network over both state and observation variables, with a dependence structure as illustrated in figure 2 for a first-order hidden-state process.

5.5.1 Hidden-Markov Models and Kalman Filters

Naturally, the nature of the space $\mathcal{Q}$ on which the $\boldsymbol{q}_t$-variables take values a substantial effect on the processes that result. Discrete state-spaces, in particular, lead to *hidden Markov models* (HMMs) [35], of which innumerable variations exist. Continuous state-spaces, on the other hand, produce *Kalman filters* [144] and nonlinear extensions thereof, e.g., [145, 146].

HMMs are a very natural fit for connecting speech and language (text) data, as they can represent continuous-valued observations, as in a sound signal, which are generated based on an underlying discrete grammar or language over a set of phones or words. In speech modelling for recognition and synthesis, one thus takes $\underline{\boldsymbol{x}}$ as a representation of the speech signal, and identifies the corresponding label sequence $\underline{c}$ with a sequence of HMM states $\underline{q}$. This enables the use of the forward-backward or Viterbi algorithm to perform decoding for automatic speech recognition [35] or provide specific state sequences as input to control parametric speech synthesis [39].

Even though the underlying hidden state has limited-range memory in HMMs and Kalman filters, the $\underline{\boldsymbol{X}}$ process observations need *not* satisfy the Markov property. Among other things, this is evident in how the forward algorithm

$$f_{Q_t|\underline{\boldsymbol{X}}_{t_0}^t}\left(q \mid \underline{\boldsymbol{x}}_{t_0}^t\right)$$

$$= f_{\boldsymbol{X}_t|Q_t}\left(\boldsymbol{x}_t \mid q\right) \sum_{q' \in \mathcal{Q}} f_{Q_t|Q_{t-1}}\left(q \mid q'\right) f_{Q_{t-1}|\underline{\boldsymbol{X}}_{t_0}^{t-1}}\left(q' \mid \underline{\boldsymbol{x}}_{t_0}^{t-1}\right) \quad (87)$$

for state-inference in HMMs [35] recursively depends on all previous observations. Hidden-state models can therefore integrate predictive information over arbitrary lengths of time. This is crucial in many applications, as it makes it simple to model long-range signal structure such as, for instance, the order of sounds in speech signals or various chart patterns in financial technical analysis, using only a limited state space (possible values for the hidden-state variable). Patterns with variable durations are especially easy to represent.

All Markov chains can be formulated as HMMs where $f_{X_t|Q_t}\left(x_t \mid q_t\right) = I\left(x_t = q_t\right)$, so HMMs are strictly more general than finite-order Markov chains. At the same time, it is sufficient that the underlying Markov chain is ergodic for the entire HMM to be ergodic as well [147]. Furthermore, unlike many other possible models with long-range memory, fast inference is possible for HMMs using the forward-backward algorithm [35]. (The same holds for linear Kalman filters where all variables are Gaussian [148, 149].) This combination of expressive power and computational efficiency has been central to the success of HMMs in applications, since it enables the models to be efficiently trained and refined on very large databases. Efficient algorithms are not as easily derived for multidimensional stochastic processes, e.g., on grids or lattices; more general results on graphs which admit fast inference are provided by [150, 151].

Another advantage of the discrete state in a hidden-Markov model is that it, like the components in a mixture model, effectively partitions the data into different subsets, each of which may be explained by different distributions. This makes it possible to capture detailed features of the observation distribution. Unlike regular mixture models, however, HMMs also describe how different components correlate across time, creating a class of mixture models suitable for describing time series.

On the other hand, the fact that the state is not directly visible complicates learning. While the EM-algorithm [76] can be used for iterative parameter-estimation in HMMs, and is guaranteed to converge on a local optimum in some special cases [152, 153], the method assumes that the number of states is known a priori. This is not necessarily true in practice, and data-driven criteria for deciding on a suitable number of states are therefore of interest. The causal-state learning algorithm in [140, 141] can be seen as an example of such a method, capable of learning more general representations than finite-order Markov chains, and yielding a number of states that converges on the true number of so-called *causal states* (a certain information-theoretic representation of stochastic processes [154]). Thesis paper [155] extends these ideas to be more robust to noise.

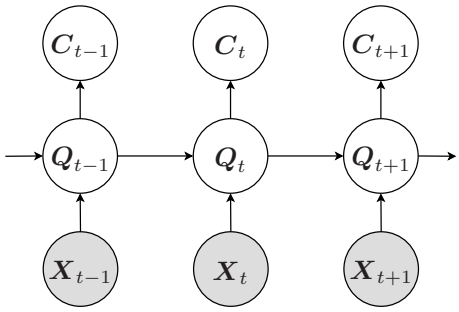

Figure 3: Illustration of dependences and observables in a recurrent neural network for classification. The model is discriminative rather than generative, so the graph cannot directly be interpreted as a Bayesian network.

5.5.2 Models Incorporating Neural Networks

Hidden state-variables can also be introduced to add long memory to non-probabilistic discriminatory models, for instance by creating artificial neural networks where the activations of certain units, instead of being expressed directly in the output, are fed into the inputs of other units at the next time step. Such constructions are known as *recurrent neural networks* (RNN). Many variants exist, with one interesting example being *long short-term memory* (LSTM) networks [156], where internal network activations do not decay with time and gating units allow information to flow in or out of memory.

Probabilistic extensions of LSTMs have been shown to do well on certain time series tasks such as recognizing handwritten text [157]. Unlike naïve HMM implementations, these models do not assume a one-to-one mapping between states and recognition labels, and instead use their memory state $\boldsymbol{Q}_t$ as a middle layer between $\boldsymbol{X}_t$ and C_t, as in figure 3. This is similar to the traditional input-state-output setup of Kalman filters in control theory, e.g., [148, 149].

Other approaches use neural networks for state-to-observation mappings, but employ HMMs for the state-dynamics back end. An example is the use of MLP features for speech recognition, e.g., [158, 159], which are features based on phone activations for each frame, derived from multilayer perceptrons (a type of artificial neural network) trained to recognize phones. These features can be used in isolation or in tandem with more traditional feature representations.

In recent years, deep belief networks based on reduced Boltzmann machines (RBMs) have produced strong results in ASR when paired with an

HMM back-end [25]. This work has garnered substantial interest from the speech community. Unlike feedforward MLPs, these constructions have probabilistic interpretations and are often initialized from a probabilistic perspective. They are thus compelling to use in a generative setting, and applications of deep neural networks to speech synthesis have just begun to spring up, e.g., [160, 161, 162].

5.5.3 Hidden Semi-Markov Models

A practical shortcoming of standard HMMs is that they have very poor ability to model different duration distributions. Since the HMM only remembers its current state, and essentially flips a coin at each t to decide whether it should stay in the state or transition elsewhere, the time spent in each state follows a discrete memoryless (i.e., geometric) distribution

$$f_D(d; a) = a^{d-1}(1-a), \tag{88}$$

where the parameter a is the probability of remaining in the state and d is a positive integer duration value. This distribution has a relatively large standard deviation that is coupled to the average duration, and attains its mode at the shortest possible duration $d = 1$ for all valid a. Natural processes such as phone durations in speech, in contrast, often show distributions peaked around a value larger than the minimal duration, and with a narrower dispersion around this value than a fitted geometric distribution would provide; see also figure 2 in [163] for an example plot of ground-truth durations from TV programming data.

One possibility to better model non-geometric state-duration distributions is to let each state also contain a model of the duration distribution of the state. Whenever a new state is entered, a duration is generated from this distribution, and the process remains in the state for exactly as many steps as the generated duration shows. Following this, the process moves to another state according to a Markov chain with time-independent transition probabilities where self-transitions are forbidden, and the procedure repeats. This creates a so-called *hidden semi-Markov model* (HSMM), which can describe arbitrary state-duration distributions.

The name *semi-Markov* comes from the fact that the process is Markovian (in the sense that past and future are conditionally independent) given knowledge about the current state Q_t *and* a counter D_t specifying how long the process has remained in the current state. The state space is therefore effectively two-dimensional, which creates issues with applying standard algorithms efficiently. If, however, the duration distribution is log-concave (which implies unimodality), fast dynamic-programming analogues of the Viterbi algorithm exist [163]; these can be used as part of approximate EM training.

Alternatively, certain non-exponential duration distributions, e.g., the negative binomial distribution, can be represented as ordinary HMMs by creating a group of states tied to have the same output distribution, so that the H(S)MM has identical output distributions as long as the state remains in the tied group. This enables one to apply standard EM training algorithms. The duration distributions that can be represented by combining geometric distributions in this manner are known as *phase-type distributions*. Continuous-valued phase-type distributions are dense on $[0, \infty)$, and can therefore approximate any duration distribution well, when given a sufficient number of sub-states [164].

Hidden semi-Markov models have been used in automatic speech recognition [165], but the gains they produce there are perceived as relatively minor. For parametric speech synthesis, in contrast, the improvements in duration modelling delivered by HSMM systems are more noticeable [166], and HSMMs are used in state-of-the-art text-to speech systems, e.g., HTS [167]. An application to segmentation of TV recordings substantially lowered frame classification error [163]. Other HSMM uses involve modelling durations of human motion patterns, for example [168] and [169].

An alternative to using hidden-semi Markov models to obtain natural duration distributions is to introduce a continuous state space (that is, use a possibly nonlinear Kalman filter), which can represent gradual, intermediate progress between states and behaviours even while remaining strictly Markovian. This approach was explored in papers [170] and [171] by the thesis author and collaborators.

5.6 Combining Models and Paradigms

Often, real-world data series exhibit both short-range correlations and long-range structure and patterns. Theoretically, all this can be represented by a discrete-state HMM, given a sufficient number of states, but in practice that number may be prohibitively large. In a nutshell, HMMs trade the limited memory *range* of Markov models for limited memory *resolution*. Instead, it is often preferable to combine the Markovian observation dependences and hidden-state approaches discussed above within a single model. The result is a network structure as illustrated in figure 4, where between-sample dependences follow a Markov process with properties determined by the hidden state.[15] Importantly, fast inference remains possible.

Typically, the Markovian part in a model of the above type captures simple and predictable local aspects of the data such as continuity, allowing the hidden state to concentrate on more complex, long-range interactions. In

[15] More general dynamic Bayesian networks for speech recognition have been considered in [172]. There are also algorithms to learn the structure of dynamic networks from data [173].

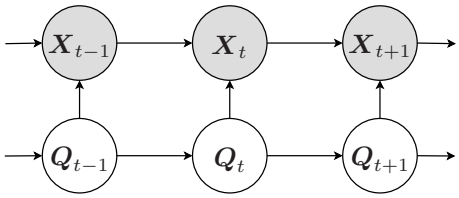

Figure 4: Bayesian network representing a model with Markovian and hidden-state dependences. Shaded nodes represent observables while other nodes are unobserved latent variables.

speech signal modelling, for example, autoregressive models describe correlations between analysis frames due to physical constraints on the motion of speech articulators, while the hidden states typically are based on a language model (e.g., n-grams), thus accounting for grammar and other large-scale structures of speech and language.

Joint Markovian and hidden-state structure is seen in, e.g., GARCH models [14] from econometrics, or in SETAR models driven by an unobservable Markov chain, so-called Markov switching models [174, 133]. The practice of adding dynamic features such as velocity and acceleration—used in acoustic models for speech recognition and synthesis [175], and in signature verification [176], for example—can be seen as a trick for introducing implicit between-frame correlations. However, this is mathematically inconsistent, in that it assigns probability mass to impossible feature sequences and tends to underestimate data variability [177]. More principled approaches to modelling temporal correlations such as autoregressive HMMs (AR-HMMs) [178] or trajectory models [126] yield improved model accuracy in the case of speech [177]. Thesis paper [138] can be seen as a combination of continuous-valued nonparametric Markov models similar to [137] with the discrete hidden state of HMMs. Such a hidden state provides a method to control the Markov process output, making it useful for language modelling in speech signals.

If may not be strictly necessary to explicitly introduce Markovian dependences in order to enforce output with a high degree of continuity. Papers [170] and [171] investigate ideas based on continuous-valued state spaces, where the dynamics and output mapping pdfs change continuously with state-space position.

In the context of long-memory modelling of text and other discrete data, there are methods which, rather than add a hidden state, instead create Markov-chain models without any explicit bound on the memory length, where the model order then grows with every observed symbol. Examples include context trees extended to unbounded depth [179], unbounded-length prediction by partial matching (PPM*) [180], and the sequence memoizer

[181, 182]. All of these can be viewed as ensemble methods, combining predictions based on Markov models of different orders. In contrast to the other proposals, the sequence memoizer is Bayesian and based on Pitman-Yor processes [183], which are capable of replicating the power-law like behaviour exhibited by many text sources. (We say power-law like, as it turns out that many sources claimed to be governed by power-laws in the literature show significant differences from true power-law distributions [184].) A shortcoming of creating and combining Markov-chain models of unbounded order is that the computational complexity of training on a given sequence often becomes quadratic in sequence length.

5.7 Finite Sequences

For processes that are not bi-infinite, a distribution over the starting state ($\boldsymbol{X}_1$ or $\boldsymbol{Q}_1$) must be specified. To ensure stationarity, this $f_{\boldsymbol{Q}_1}$ must satisfy

$$f_{\boldsymbol{Q}_2}(\boldsymbol{q}) = \int f_{\boldsymbol{Q}_2|\boldsymbol{Q}_1}(\boldsymbol{q} \mid \boldsymbol{q}')\, f_{\boldsymbol{Q}_1}(\boldsymbol{q}')\, \mathrm{d}\boldsymbol{q}' \tag{89}$$

$$= f_{\boldsymbol{Q}_1}(\boldsymbol{q}) \tag{90}$$

for all $\boldsymbol{q}$. If the process is ergodic, $f_{\boldsymbol{Q}_1}$ is uniquely is determined by the conditional next-step distribution $f_{\boldsymbol{Q}_2|\boldsymbol{Q}_1}$. For Markov chains, this is the stationary distribution vector $\boldsymbol{\pi}$ from section 5.4.

Finite-duration Markov chains and HMMs can be formed by adding an attracting state (a state that only transitions to itself) associated with the "no output" symbol $\emptyset$. By starting in a different state than the attracting one, transient behavior results. Even if the resulting model is not stationary in the ordinary sense of the word—for one thing, $f_{Q_1}(i) \neq \pi_i$ for several i—it can still be governed by an unchanging (time-independent) next-step distribution

$$f_{\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}}(\boldsymbol{q} \mid \boldsymbol{q}') \equiv f_{\boldsymbol{Q}_2|\boldsymbol{Q}_1}(\boldsymbol{q} \mid \boldsymbol{q}')\,. \tag{91}$$

This introduces similar advantages for learning as conventional stationarity does in infinite-duration models. Finite-duration models of this type are commonly used to represent single speech utterances.

5.8 Other Approaches

Not all techniques for working with sequence data fit neatly into the above state-based framework. Early efforts in speech recognition were based on templates or *exemplars*; incoming speech was time stretched or compressed through dynamic time warping [185, 31] such that a maximal geometric similarity to each reference example could be computed. Hidden Markov

models, popularized in part by a timely 1989 tutorial [35] by Rabiner, provided a more principled method for assigning alignment and similarity costs, and have largely superseded time-warping methods.[16] Under the surface, however, both paradigms use very similar dynamic programming techniques to compute how well duration-variable speech matches a given hypothesis.

In speech synthesis, HMM-based methods show substantial flexibility, and can adapt their voice to recreate different speakers and affective states. Higher naturalness, however, is generally provided by *concatenative* or *unit selection* synthesis approaches (cf. [187]), where fast search algorithms are used to find fitting signal segments from a speech database, segments which then are concatenated to express a desired target phrase. These techniques can be seen as geometric methods, as they are based on non-probabilistic target and join costs. (The target costs measure how well a segment well it expresses the desired phone or phone sequence in context, and the join cost how the segment matches neighbouring segment candidates). Unit selection approaches have some connections with nonparametric methods, in that the different possibilities that the method can express is explicitly linked to the training database and its size. (They are not ensembles, however, since only a single model—a single realization even—is used at any given point in time.)

There are also problem-solving techniques that do not model or treat the datastream on a sample-by-sample or frame-by-frame basis. One such approach to classification, and specifically speech recognition, is to use *detectors*—small functions, each of which is designed to be sensitive to the occurrence of a certain events in a datastream. In ASR, relevant events may be the occurrence of speech building blocks such as phones or simple words.

A set of detectors continuously scan the incoming data, and if the probability is sufficiently large for a specific event to have occurred at some t, a detection is flagged for that time for the detector in question. Subsequent processing tries to make sense of the set of detections, which is not a sampled time series in any typical sense, nor is it modelled as such. An example of a detector framework of this kind is [188]. An alternative view grounded in (continuous-time) point processes can be found in [189].

Detectors can also be utilized in a more traditional time-series approach: If the detection probabilities for the set of detectors for each frame are arranged in a vector, a sequence of detector-based feature vectors is obtained. This data can be described with traditional time-series models. The use of MLP features in speech recognition [159], already mentioned in section 5.5, is an example of this approach.

[16] This is not to say that geometric methods are out of the picture; [186] provides an innovative recent SVM-based discriminative model for ASR.

6 Solving Practical Problems

So far, we have introduced the most common tasks involving sequence data, as well as the mathematical models and techniques used for solving them. In this section, we summarize how everything can be put together and discuss some important considerations for successful applied problem solving; [114] provides another view of several key issues in practical machine learning.

We begin by outlining a typical protocol followed when approaching a new problem (section 6.1). Thereafter (in section 6.2), feature engineering, the step that adapts the data to a form suitable for modelling, is considered. The phenomenon of overfitting, perhaps the most famous nemesis of data analysts and statisticians, is explored subsequently (section 6.3). Another notable practical problem is mismatch between training and application, where we discuss (in section 6.4) some methods, falling partially or wholly outside traditional parameter estimation, to enhance models for specific applications.

6.1 Solution Procedure

In solving a practical problem with the assistance of sequence data, there are a few standard steps:

1. **Define the problem** by deciding on a specific quantitative performance measure to optimize.

2. **Gather data.** This can, e.g., entail recording speech data or obtaining a suitable text corpus, but also transcribing speech recordings or annotating corpora (i.e., adding labels to the data).

3. **Create a feature extractor.** This is a very important step, especially for supervised learning, in which the data is transformed to a form that is more suitable for modelling. An overview of the considerations involved is given in the next section.

4. **Specify a model** for the feature data. Many of the distinguishing characteristics of different model paradigms have been covered previously, and will not be repeated here.

5. **Estimate the model parameters** based on training data. Even nonparametric methods often have tuning parameters that need to be adjusted. Sometimes the procedure is not as simple as just applying the standard parameter estimation techniques from section 3.6, but may also involve additional processing as discussed in section 6.4 to improve performance.

6. **Estimate performance**. Since the apparent performance on the training data can be significantly misleading, this is usually done on held-out data not used during training, a form of cross-validation, but other methods also exist.

7. **Tweak** the feature extractor, the model, and possibly other processing steps until system performance is satisfactory.

8. **Apply** the resulting solution model to the original problem.

A few of these steps have been discussed in previous sections, particularly the parts that relate to model building. We now turn to discuss some of the surrounding aspects, such as feature creation and various types of preprocessing and postprocessing to adapt to different situations. While crucial to successful problem solving, these tend to be highly application dependent and are frequently difficult to put on an objective basis. Although a substantial amount of work has been done to put several of these considerations on a firmer theoretical footing, the fact remains that successful problem solving requires a combination of art and science.

6.2 Feature Engineering

Especially in supervised learning problems, it is quite uncommon to model the raw input data directly. Often, the feature data that the model describes is a processed version of the original input. The process that transforms the raw input to suitable features is known as *feature extraction*, and the process of designing such extractors is known as *feature engineering*. This is generally the most application-dependent aspect of any machine-learning application, as once the data is in a suitable form, one can just apply one of the many available mathematical models to describe the situation and obtain a solution to the problem.

In classification, feature extraction is often considered a one-way transformation, but to perform synthesis it is important that the feature data can be inverted to recover a signal in the original input domain. It is not necessary for this inverse to be unique, just that some reasonable "pseudoinverse" in the original input space can be selected.

6.2.1 Information Removal

Feature extraction generally has two goals:

1. To remove information in the data that is redundant or of no or little value to the task at hand. This yields a lower-dimensional description that requires fewer bits and is faster to process.

2. To transform the remaining information into a form that exposes the structure of the problem and allows it to be modelled efficiently.

To see why the first objective, information removal, can be beneficial, one can think of CD-quality speech signals. These have a very high bitrate, but many frequencies may be fundamentally inaudible or masked by other sounds at different times—this is why lossy compression of audio can be so efficient [190]. Obviously, the human-audible signal aspects alone suffice for high-accuracy speech recognition, as evidenced by human recognition performance on lossily compressed speech data. Furthermore, many perceptible vocal cues such as pitch may convey emphasis, speaker emotional state, and facilitate turn-taking, but are not of central importance for recognizing the words themselves (though pitch can be of importance in tonal languages such as Mandarin).

In situations where the experimenter does not know a-priori what information to keep, feature extractors can be made to incorporate unsupervised dimensionality-reduction techniques such as principal component analysis (PCA) or factor analysis [191] to discard information while retaining most of the empirical variability. The use of autoencoders in deep neural networks [192] can be seen as an approach for performing nonlinear dimensionality reduction. There are also dimensionality-reduction techniques utilizing labelled data, e.g., [193], which may be better at selecting information aspects relevant for specific supervised tasks. Though PCA and the concept of dimensionality reduction often are motivated on geometric grounds, there exist probabilistic extensions as well, for instance probabilistic PCA (PPCA) [194].

6.2.2 Exposing Problem Structure

The second objective of feature extraction is to expose the structure of a problem in a way that can be described and learned efficiently. (This assumes that there is *some* kind of structure to the problem, otherwise learning is impossible, as discussed in section 3.5). One way to think about this is to consider where the features fall in the high-dimensional space of the raw input—typically, certain values go together, so that the data lies on a low-dimensional manifold in the high-dimensional space. The feature extraction is then essentially the task of extracting meaningful, or at least useful, coordinates on this manifold. It is easy to see that coordinates based on the structure of the problem can be expected to be helpful, e.g., for geometric methods. In the case of discrete-valued data such as text, the situation is a little less straightforward to visualize, but elements of this reasoning still remain.

A subtask of exposing the structure of the data is to separate independent or uncorrelated aspects of the material. This can make a significant difference in the number of parameters that need to be estimated: compare the diagonal covariance matrices appropriate for Gaussian-distributed vectors with independent elements against the general covariance matrix,

which has D^2 degrees of freedom. (D here denotes the feature-vector dimensionality; naturally, dimensionality reduction will reduce the parameter set as well.)

6.2.3 Finding Fitting Features

Solving a learning task efficiently requires a synergy between features and models: the distribution and behaviour of the features should match the assumptions made by the model. If, for instance, the features as designed are naturally Gaussian distributed, then a Gaussian model may be expected to be appropriate. If feature-vector components are independent, Gaussian distributions with diagonal covariances or other models that assume independence may be used, otherwise correlated Gaussians or GMMs may be required. Sometimes, it can be necessary to apply mathematical transformations to adjust the range of the data and express the data in a natural and well-behaved way.

Often, nature can be used as a source of inspiration in creating effective feature extractors. Take speech signals as an example: *Mel-frequency cepstrum coefficients*, MFCCs, common in speech recognition [195] both incorporate a short-term Fourier transform (similar to the effect of the human cochlea) as a step to decorrelate the feature data and remove inaudible aspects based on a psychoacoustic frequency-discrimination scale. The perceptual theme can be used to reduce feature vector dimensionality even further based on models of human hearing [196]. Features related to MFCCs are also used in speech synthesis, though somewhat different and augmented by pitch tracking and other feature data, e.g., [40], in order not to sacrifice quality by discarding too much information. (See also the ASR versus synthesis comparison in [175].)

The auditory frequency scales used in MFCCs and similar representations, originally introduced based on the human auditory system and psychoacoustics, have recently been related to mathematical invariance and stability properties under frequency transposition and other perturbations in so-called *wavelet scatterings* [197]. This interesting line of work also connects with modulation features, another biologically-linked speech feature [198], and with cascades of nonlinear processing units seen in convolutional deep belief networks [199].

6.3 Bias, Variance, and Overfitting

Apart from selecting features and models that go well together, an important practical concern is choosing a model family of adequate complexity. Theoretical results show that methods such as maximum likelihood are asymptotically consistent if the data generating distribution is an element in $\mathcal{M}$. This seems to suggest that model families should be chosen as in-

clusive as possible, to ensure that they contain the "true generating model," or a close approximation thereof. This intuition is however very wrong in finite-sample situations. In fact, even *if* the data were generated by a model in $\mathcal{M}$ (which is so unlikely in an application as to be practically impossible), a simplified model might—surprisingly—give better predictions both in supervised and unsupervised settings. The task of the analyst is thus not to determine the theoretically correct parametric family, but to propose a set of models $\mathcal{M}$ which yields a good solution to the task, given the available data. This was succinctly articulated by George E. P. Box in his famous quotation that "all models are wrong, but some models are useful" [200].

The problem of choosing an appropriate model family is compounded by the fact that large, complex model families $\mathcal{M}$ often are able to fit the training data well, even when their actual explanatory value is very poor. This phenomenon is known as *overfitting*, and relates to the fact that a model with many parameters may be capable of fitting all the random peculiarities of the data, whereas a simpler model is forced to choose a shape which captures the broad tendencies of the material, in order to fit the data as well as possible. Simpler models therefore often generalize better to unseen instances, and thus provide more reliable solutions to the problem at hand.

In section 4.3 we mentioned how the apparent training data performance is biased towards overly optimistic values, simply because one chooses the model that maximizes this value. This bias becomes worse the more different models we choose from (i.e., the larger $\mathcal{M}$ is) [93], which essentially is why overfitted models are so easily selected by the unwary analyst. There exist analytical methods to compensate for the inherent bias in the training-data performance of large models, e.g., Akaike's information criterion (AIC) [201] or the Bayesian information criterion (BIC) [202],[17] but cross-validation as touched upon in section 5.4 is possibly the most common approach in practice.

Fully Bayesian methods as in section 4.4 lend some protection against overfitting, since the distribution is computed as the average over all plausible models, rather than picking the single model that looks most appealing. If the Bayesian approach is uncertain about which model that is most appropriate, the set of models considered probable will be broad, and one may expect the predicted distribution to be smoothed out and be less sensitive to stochastic variation than a distribution based on a point estimate. Nonparametric modelling can be seen as another method to combat overfitting,

[17] As an aside, it can be proved that AIC is an inconsistent estimator of the true order [203, 204], tending to overestimate the order of Markov chains, where BIC, in contrast, is consistent. This is often a bit of a red herring, however, since models of fixed, overestimated order generally learn to let higher-order terms go to zero, and thus also converge on the correct predictor in the limit—hence, consistent order estimation is not necessary for consistent distribution estimation.

by gradually growing the model set as more data becomes available, letting model complexity adapt to dataset size.

For any technique we use, there are limits on how much data that is sufficient or necessary to have a high probability of learning a well-performing model. More specifically, the excess risk in classification (i.e., the difference between the average loss of a learner and the lowest achievable average loss) can be bounded using results from *computational learning theory* [66]. One important tool in such efforts is Hoeffding's inequality, which puts bounds on the difference between the empirical and the true probability of an event, and thus can be used to relate the expected generalization performance to the observed error rate.

Mathematically, the task of selecting a proper model (family) complexity can be expressed as a so-called *bias-variance trade-off*. Simple model families have a large asymptotic *bias* (systematic error), in that their best model may never be close to the generating model. The optimal parameters, being few, can, however, be estimated accurately from modest amounts of data. Complex $\mathcal{M}$, on the other hand, typically have the potential to come closer to the true distribution, but as there are many models to choose from, more data will be needed to get there. The selected models therefore have a strong dependence on random properties of the sample data, and show a lot of variation (random error) around the most appropriate model.

To make the bias-variance trade-off more concrete, consider a case where the estimate $\widehat{\theta}$ of an unknown parameter θ satisfies $\widehat{\theta} \sim \mathcal{N}\left(\mu, \sigma^2\right)$. The mean square error of the estimate is then

$$\mathbb{E}\left(\left(\widehat{\theta} - \theta\right)^2\right) = \mathbb{E}\left(\left(\widehat{\theta} - \mu + \mu - \theta\right)^2\right) \tag{92}$$

$$= (\theta - \mu)^2 + \sigma^2, \tag{93}$$

which elegantly decomposes into a term relating to the systematic bias, and a term relating to the random uncertainty. Both bias and variance have to be small for the total expected error to be small. An example where the above trade-off can be used explicitly in model selection is bandwidth selection for KDE, where the asymptotic mean integrated square error (AMISE) takes the form of a sum of a bias and a variance term [59]. Bias-variance-type trade-offs also exist for other loss functions, e.g., 0-1 loss [52].

6.4 Preprocessing and Postprocessing

In applications of machine learning, speech, or language technology to real life, there is often a mismatch between how a model is created and trained, and how it later is applied. An example is the profusion of different noise environments, speaker accents, and other signal degradation sources that a general speech recognizer must face without having seen them all in the lab.

The mismatch between training and application is challenging to predict through theory, and often has to be compensated for by the analyst, using methods external to the standard estimation and decision procedures. Frequently, as we shall see in this section, this compensation occurs either as a data preprocessing (typically as part of the feature extraction stage) or as a kind of postprocessing after model selection or parameter estimation has been performed.

6.4.1 Domain Adaptation

A canonical example of how training data and application can differ is that models often are trained on data from a subset of individuals or cases, while the input data in subsequent applications typically comes from other individuals or cases, having a different probability distribution. As a matter of notation, the data is said to come from the *source domain*, with the *target domain* being the intended application; methods to mitigate problems due to the mismatch are collectively referred to as *domain adaptation* or *transfer learning* [205].

Feature extraction, as covered in section 6.2, can be seen as a form of preprocessing. Certain types of feature normalization can, for instance, make a speech recognizer more robust to variations in the speaker voice and the ambient acoustic environment. An example is *cepstral mean normalization* (CMN), e.g., [206], where each utterance feature sequence is translated to have mean zero. This moves both training and test data to a common region in feature space, and constitutes an example of *feature-based domain adaptation*. A simple extension is to normalize both mean and variance. A preprocessing to correct spelling errors could be seen as a method for reducing uneven performance when working with texts written by different individuals. Another notable feature-based domain adaptation scheme from NLP which takes the form of a preprocessing step is described in [207].

If the mismatch is simply due to uneven sampling, where cases that are commonly encountered in practice are less common in the training database, this can be overcome by selecting or weighting datapoints prior to training [208], a type of *instance-based domain adaptation*. This requires retraining (picking a new model) every time the domain changes, however, and cannot compensate for cases where application-relevant training examples are not merely scarce, but absent altogether.

A third option is *model-based domain adaptation*, where the trained model is adjusted to fit the application. This can, for instance, be done by re-estimating certain parameters, e.g., *vocal tract length normalization* (VTLN) in speech recognition [209, 210], or by transforming means and variance parameters in HMMs [211]. This is thus a kind of model postprocessing.

6.4.2 Smoothing

Even if we do not know any specifics about the target domain, models can be estimated and processed in ways that typically increase performance in applications. A basic, but important, example of this is model *smoothing*, which is of importance in language modelling for speech recognition, as well as in other NLP tasks [212].

Smoothing addresses a problem with maximum likelihood estimation for categorical distributions, for instance in language models, which assigns zero probability to any event not observed in the training data. Such an event could be a unique word or word combination (n-gram). Paradoxically, extremely uncommon events can occur quite often in practice: for large text corpora, for instance, it is typical that about half the words in the corpus only occur once, so-called *hapax legomena*. Even if a previously unseen word combination occurs only once in an entire sequence of test data, the probability of the entire sequence will still be estimated to be zero. The model therefor rejects many perfectly valid texts, which is known as the *zero-probability* or *zero-frequency problem* [213].

The idea of smoothing is to broaden the support of the estimated distribution, so that events previously considered impossible now have some probability of occurring. This can be seen as a recognition of the fact that practical diversity is greater than the variability found in the training data. A simple scheme for categorical data is to add an initial count of one, or some other number, to every bin (possible observations), e.g., [214]; these *pseudocounts*, together with the empirical frequencies from the data, are used to determine the estimated distribution. The pseudocounts can be interpreted as a Bayesian prior, where the experimenter a priori creates "fake observations" to indicate that all elements in $\mathcal{X}$ have a real possibility of occurring. Jeffreys prior, in particular, amounts to adding a count of $1/2$ to every bin.

Smoothing to broaden the support of estimated distributions can also be performed after model estimation, as a postprocessing step. For uncommon word combinations, another approach is to interpolate between a high-order Markov (n-gram) model and a lower order model [215], a kind of model combination.

The opposite situation of the zero-probability problem, when a selected model assigns much probability mass to observations that are very unlikely to be generated by the true process, can be an issue in generative tasks. Speech sampled from state-of-the-art parametric speech synthesis models, for instance, sounds warbly and bubbly in a manner human speech does not [177]. Evidently, most of the probability mass is assigned to models which are radically unlike natural speech. Texts generated from Markov chains are another example: while the words and local word combinations may be standard, the text as a whole is generally incoherent in a manner atypical

for texts of human origin.

The above failings can be seen as fundamental shortcomings of the models, being unable to capture and represent important sources of variation and instead interpreting these as random noise. For want of better models, the issues can be lessened by reducing the support of the output distributions after training to concentrate on the most probable and representative observations. Speech synthesis using MLPG [56], for instance, takes this to the extreme, by only generating the single most probable output element. An information-theoretic scheme for continuously selectable levels of this type of "sharpening" (as an opposite of smoothing) is the topic of thesis paper [55], some ideas of which were first published in [216].

7 The Thesis Research

We now move our focus towards the specific research contributions contained in this thesis. To start, some context is provided (section 7.1), explaining why research on speech and language models remains relevant. Thereafter an overview of the different research papers and their objectives is presented (section 7.2), followed by a list of main contributions of the individual papers and some brief words on future topics to explore (in section 7.3).

7.1 Thesis Context

A common theme of machine learning and artificial intelligence has been that machines can do things that no human can, but many things that humans find easy are very difficult for a computer. Speech and language is a good example of this divide. While computers can record, edit, store, transmit, and reproduce text and sound in ways that could hardly have been imagined just fifty years ago, we still cannot interface with machines through speech and text nearly as naturally as we do with human beings.

That speech and natural language technology falls short in important areas is not for lack of trying. Massive research efforts have been directed towards automatic speech recognition, text-to-speech systems, machine translation, and similar applications, and have produced numerous insights and breakthroughs big and small. Today, tools and technologies exploiting these efforts are widely deployed, but remain held back from realizing their full potential by lingering limitations in the underlying science. The hunt is thus still on for better models which capture the true essence of speech and language, or at least improve our modelling capacity to deliver solutions that improve on the state of the art.

Paper	A	B	C	D
Core technique	GPDMs	CSSR/RCS	MERS	KDE-HMMs
Type	Generative	Generative	Generative	Generative
Observations	Continuous	Discrete	Any	Continuous
Application	Synthesis	Language acquisition	Synthesis, denoising	Synthesis
Has Markov dependence?		✓	✓	✓
Long-memory mechanism	Continuous state	Unbounded order	Unbounded order/none	Discrete state
Nonparametric	✓	✓		Yes and no
What's new?	Application	Criterion, algorithm	Algorithm, application	Model, algorithm

Table 1: Overview of the ideas presented in the thesis papers. Tick marks mean "yes" while empty cells signify a "no." Further explanation and discussion is provided in the text.

7.2 Overview of Work

We need better speech and language models, and the research contained in this thesis represents efforts to identify and examine promising candidates. In one way or another, each paper in the thesis demonstrates a method for overcoming limitations of entrenched modelling paradigms in particular application scenarios. Often, but not always, this is accomplished by investigating model classes with greater descriptive power than the standard methods, e.g., through nonparametric techniques, or Markov chains with potentially unbounded memory.

The majority of the thesis papers focus on synthesis applications, but paper B considers a method primarily suitable for NLP-related recognition tasks (e.g., named-entity recognition) or pattern discovery for language acquisition. Papers A and D, meanwhile, both look to overcome shortcomings in traditional HMM-based acoustic models used in parametric speech synthesis, either through applying a more powerful and appropriate state-space representation (replacing the oversimplification of a discrete, one-dimensional phonetic state), or by developing novel, controllable and consistent nonparametric Markov models for continuous-valued processes (which may capture speech dynamics better than the linear Markov models currently in use). Paper C, finally, considers how we best can make do with the imperfect models we have, with results that are applicable to continuous as well as discrete-valued sources, i.e., both speech and text.

To illustrate how the research efforts included in thesis connect, and

where they differ, table 1 provides a structured comparison of the four thesis papers. While the table is largely self-explanatory, a few comments are in order: The "Markov dependence" row refers to whether or not causal dependences between observations exist if all latent variables (if any) are given. GPDMs show that a hidden, continuous-valued state suffices to generate smoothly changing output, without additional between-observation Markovian dependences, though paper A conjectures that introducing such dependences (which is straightforward using dynamic features) may further improve synthesis quality. "Unbounded order" long memory in the table means that the most straightforward way to think about the technique in question is as a Markov process of possibly unbounded or infinite order, at least for certain contexts. MERS can be solved explicitly for (continuous-valued) Gaussian processes with arbitrary power spectral densities—meaning infinite order—but in the case of discrete-valued processes only finite-order Markov chains are considered in the paper. KDE-HMMs have a set of nonparametric Markov processes that generate the observations, but the models switch between the different processes based on a parametric hidden Markov chain; these models therefore contain both parametric and nonparametric elements.

7.3 Summary of Contributions and Outlook

To provide some more detail on the individual research efforts, the main achievements of the thesis papers can be summarized as follows:

- Paper A: The first (to our knowledge) application of *Gaussian process dynamical models* (GPDMs) [217, 218] to speech generation. GPDMs are a class of nonparametric dynamic models where autoregressive state dynamics $f_{\boldsymbol{Q}_t|\boldsymbol{Q}_{t-1}}$ and state-conditional output distributions $f_{\boldsymbol{X}_t|\boldsymbol{Q}_t}$ are given by Gaussian processes [27]. It is shown that GPDMs produce more natural output for voiced speech than do comparable hidden Markov models, both under sampling and MLPG. This is attributed to the ability of the continuous state-space to represent gradual transitions and recreate natural phone durations. Additional state-space dimensions are shown to enable the representation of prosodic variation.

- Paper B: An algebraic criterion for deciding when the causal-state representation of a process can be learned by the *causal-state splitting reconstruction* (CSSR) algorithm from [140, 141]. The criterion is applied to show that CSSR cannot learn a simple two-state process when disturbed by non-zero probability substitution noise. A specification of how to apply the criterion algorithmically is also provided.

- Paper B: A modified CSSR algorithm, dubbed *robust causal states*

(RCS). It is proved that, unlike CSSR, the algorithm is capable of recovering the underlying causal states of a process also in the presence of noise (substitutions, deletions, and insertions). The conclusions are supported by simulation data.

- Paper C: A general information-theoretic scheme, *minimum entropy rate simplification* (MERS), for simplifying stochastic processes by concentrating on the most representative observations. Inspired by rate-distortion theory in source coding [54], MERS is formulated as an optimization problem over a set of stochastic processes, selecting the process with the least entropy rate, subject to a constraint on the permitted dissimilarity from the original reference process. A multitude of different process classes and dissimilarity measures can be used.

- Paper C: Explicit solutions to the MERS problem for Gaussian processes under different dissimilarity measures and constraints on the process under consideration. The solutions formulas also apply to entropy maximization under similar constraints.

- Paper C: An explicit solution formula to the MERS problem for first-order Markov chains under a reverse KL-divergence dissimilarity. The solution also applies to entropy maximization under similar constraints. Experiments demonstrate the results of MERS on Markov chains, including text generation and an application to denoising a model trained on disturbed data, where MERS outperforms an alternative scheme based on thresholding.

- Paper D: KDE-HMMs, a new signal model extending asymptotically consistent continuous-valued Markov processes based on kernel density estimation (a construction likely first considered in [137]) with a discrete hidden Markov state. This is promising for modelling the trajectories of acoustic parameters in speech (which have proved difficult to describe well-enough to generate natural-sounding speech [177]), with the added latent state enabling synthesis output to be controlled.

- Paper D: Guaranteed-ascent generalized EM-based parameter update formulas for KDE-HMMs, as well as relaxed update formulas which allow faster convergence in practice. Experiments show that KDE-HMMs can be trained using the relaxed formulas to achieve superior prediction performance on nonlinear time series, compared to several reference models.

In-depth information can be found in the papers themselves, which are included in part II below.

Of course, many threads remain to be explored. To name a few items, the acoustic models of paper A need to be extended to handle discontinuities and to allow the synthesis of arbitrary speech utterances. The learning techniques in paper B are not yet practical in large-scale scenarios. Minimum entropy rate simplification from paper C may investigate new applications and additional dissimilarity measures. And the models of paper D, despite their many appealing properties, have yet to be adapted to perform non-parametric speech synthesis. These are but a few of many possible research tasks still ahead of us in speech and language modelling.

References

[1] D. Hilbert, "Mathematische probleme," *Nachrichten von der Königl. Gesellschaft der Wiss. zu Göttingen*, pp. 253–297, 1900.

[2] A. N. Kolmogorov, *Grundbegriffe der Wahrscheinlichkeitsrechnung*. Berlin: Julius Springer, 1933.

[3] T. J. Ross, *Fuzzy Logic with Engineering Applications*, 3rd ed. Chichester, UK: John Wiley & Sons, 2010.

[4] E. T. Jaynes, *Probability Theory: The Logic of Science*, G. L. Bretthorst, Ed. Cambridge, UK: Cambridge University Press, 2003.

[5] R. T. Cox, "Probability, frequency, and reasonable expectation," *Am. Jour. Phys.*, vol. 14, pp. 1–13, 1946.

[6] ——, *The Algebra of Probable Inference*. Baltimore, MD: Johns Hopkins Press, 1961.

[7] G. U. Yule, "On a method of investigating periodicities in disturbed series, with special reference to Wolfer's sunspot numbers," *Phil. Trans. R. Soc. A*, vol. 226, pp. 267–298, 1927.

[8] S. Vaughan, "Random time series in astronomy," *Phil. Trans. R. Soc. A*, vol. 371, no. 1984, 2013.

[9] M. E. Mann, R. S. Bradley, and M. K. Hughes, "Northern hemisphere temperatures during the past millennium: inferences, uncertainties, and limitations," *Geophys. Res. Lett.*, vol. 26, no. 6, pp. 759–762, 1999.

[10] C. Duchon and R. Hale, *Time Series Analysis in Meteorology and Climatology: An Introduction*. Wiley-Blackwell, 2012.

[11] P. Turchin, *Complex Population Dynamics: A Theoretical/Empirical Synthesis*. Princeton, NJ: Princeton University Press, 2003.

[12] D. Stern, Ed., *Remote Sensing of Animals*, ser. Acou. Today, vol. 8, no. 3, 2012.

[13] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *J. Polit. Econ.*, vol. 81, no. 3, pp. 637–654, 1973.

[14] T. P. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econom.*, vol. 31, no. 3, pp. 307–327, 1986.

[15] W. Enders, *Applied Econometric Time Series*, 2nd ed. Hoboken, NJ: John Wiley & Sons, 2004.

[16] R. McCleary and R. A. Hay, Jr., *Applied Time Series Analysis for the Social Sciences*. Beverly Hills, CA: SAGE Publications, 1980.

[17] A. L. Gilbert, T. Regier, P. Kay, and R. B. Ivry, "Whorf hypothesis is supported in the right visual field but not the left," *P. Natl. Acad. Sci. USA*, vol. 103, no. 2, pp. 489–494, 2006.

[18] R. I. M. Dunbar, "Coevolution of neocortical size, group size and language in humans," *Behav. Brain Sci.*, vol. 16, no. 4, pp. 681–735, 1993.

[19] S. Stadler, A. Leijon, and B. Hagerman, "An information theoretic approach to estimate speech intelligibility for normal and impaired hearing," in *Proc. Interspeech*, Antwerpen, BE, Aug. 2007, pp. 398–401.

[20] M. R. Schroeder and B. S. Atal, "Code-excited linear prediction (celp): High-quality speech at very low bit rates," in *Proc. ICASSP*, vol. 10, 1985, pp. 937–940.

[21] J. R. Hershey, S. J. Rennie, P. A. Olsen, and T. T. Kristjansson, "Super-human multi-talker speech recognition: A graphical modeling approach," *Comput. Speech Lang.*, vol. 24, no. 1, pp. 45–66, 2010.

[22] H. Moravec, *Mind Children*. Harvard University Press, 1988.

[23] S. King and V. Karaiskos, "The blizzard challenge 2010," in *Proc. Blizzard Challenge 2010 workshop*, 2010.

[24] C. Valentini-Botinhao, E. Godoy, Y. Stylianou, B. Sauert, S. King, and J. Yamagishi, "Improving intelligibility in noise of HMM-generated speech via noise-dependent and -independent methods," in *Proc. ICASSP*, May 2013.

[25] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal. Proc. Mag.*, vol. 29, no. 6, pp. 82–97, 2012.

[26] A. Ozerov and W. B. Kleijn, "Asymptotically optimal model estimation for quantization," *IEEE Trans. Comm.*, vol. 59, no. 4, pp. 1031–1042, 2011.

[27] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

[28] J. L. Doob, *Stochastic Processes*. New York, NY: John Wiley & Sons, 1953.

[29] C. E. Shannon, "Communication in the presence of noise," *Proc. IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[30] J. O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd ed. New York, NY: Springer-Verlag, 1985.

[31] C. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE T. Acoust. Speech*, vol. 28, no. 6, pp. 623–635, 1980.

[32] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural Networks*, vol. 3, no. 1, pp. 23–43, 1990.

[33] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? sentiment classification using machine learning techniques," in *Proc. EMNLP*, vol. 10, 2002, pp. 79–86.

[34] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos, "An evaluation of naive Bayesian anti-spam filtering," in *Proc. ECML Workshop Mach. Learn. New Inf. Age*, G. Potamias, V. Moustakis, and M. van Someren, Eds., 2000, pp. 9–17.

[35] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[36] A. Ratnaparkhi, "A maximum entropy model for part-of-speech tagging," in *Proc. EMNLP*, vol. 1, 1996, pp. 133–142.

[37] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Upper Saddle River, NJ: Prentice Hall, 1998.

[38] K. Miyahara and M. J. Pazzani, "Collaborative filtering with the simple Bayesian classifier," in *Proc. PRICAI*, 2000, pp. 679–689.

[39] H. Zen, K. Tokuda, and A. W. Black, "Statistical parametric speech synthesis," *Speech Commun.*, vol. 51, no. 11, pp. 1039–1064, 2009.

[40] H. Kawahara, "STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds," *Acoust Sci & Tech*, vol. 27, no. 6, pp. 349–353, 2006.

[41] P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, J. D. Lafferty, and R. L. Mercer, "Analysis, statistical transfer, and synthesis in machine translation," in *Proc. TMI*, 1992, pp. 83–100.

[42] T. Glad and L. Ljung, *Control Theory: Multivariable and Nonlinear Methods*. London, UK: Taylor & Francis, 2000.

[43] L. Ljung, *System Identification – Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[44] A. C. Lorenc, "Analysis methods for numerical weather prediction," *Q. J. Roy. Meteor. Soc.*, vol. 112, no. 474, pp. 1177–1194, 1986.

[45] M. Collins, "Ensembles and probabilities: a new era in the prediction of climate change," *Phil. Trans. R. Soc. A*, vol. 365, no. 1857, pp. 1957–1970, 2007.

[46] M. P. Clements and D. E. Hendry, *Forecasting Economic Time Series*. Cambridge, UK: Cambridge University Press, 1998.

[47] C. Chelba and F. Jelinek, "Structured language modeling," *Comput. Speech Lang.*, vol. 14, no. 4, pp. 283–332, 2000.

[48] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Upper Saddle River, NJ: Prentice Hall, 1993.

[49] P. Stoica and Y. Selén, "Model-order selection: a review of information criterion rules," *IEEE Signal. Proc. Mag.*, vol. 21, no. 4, pp. 36–47, 2004.

[50] D. R. Pauluzzi and N. C. Beaulieu, "A comparison of SNR estimation techniques for the AWGN channel," *IEEE T. Commun.*, vol. 48, no. 10, pp. 1681–1691, 2000.

[51] O. Chapelle and Y. Zhang, "A dynamic Bayesian network click model for web search ranking," in *Proc. WWW*, 2009, pp. 1–10.

[52] R. Kohavi and D. H. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Proc. ICML*, 1996, pp. 275–283.

[53] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Stat.*, vol. 22, no. 1, pp. 79–86, 1951.

[54] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY: John Wiley & Sons, 1991.

[55] G. E. Henter and W. B. Kleijn, "Minimum entropy rate simplification of stochastic processes," *IEEE T. Pattern Anal.*, submitted.

[56] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *Proc. ICASSP*, vol. 3, 2000, pp. 1315–1318.

[57] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR Comp. Math. Math+*, vol. 7, no. 3, pp. 200–217, 1967.

[58] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, 2005.

[59] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, UK, 1986.

[60] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

[61] C. E. Brodley and M. A. Friedl, "Identifying and eliminating mislabeled training instances," in *Proc. Natl. Conf. Artif. Intell.*, vol. 13, 1996, pp. 799–805.

[62] S. M. Kakade and A. T. Kalai, "From batch to transductive online learning," in *Proc. NIPS*, 2005.

[63] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proc. ACM-SIGIR*, vol. 17, 1994, pp. 3–12.

[64] K. Kunen, *Set Theory*. Amsterdam, NL: Elsevier, 1980.

[65] K. Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," *Monatsh. Math.*, vol. 38, no. 1, pp. 173–198, 1931.

[66] T. M. Mitchell, *Machine Learning*. New York, NY: McGraw-Hill, 1997.

[67] H. Akaike, "Information theory and an extension of the maximum likelihood principle," in *Proc. Second Int. Symp. Inf. Theory*, 1973, pp. 267–281.

[68] M. Rudemo, "Empirical choice of histograms and kernel density estimators," *Scand. J. Statist.*, vol. 9, pp. 65–78, 1982.

[69] B. A. Turlach, "Bandwidth selection in kernel density estimation: A review," Institut de Statistique, Université catholique de Louvain, Voie du Roman Pays 34, B-1348 Louvain-la-Neuve, Belgium, Tech. Rep. Discussion Paper 9317, 1993.

[70] T. S. Ferguson, *A Course in Large Sample Theory.* London, UK: Chapman & Hall, 1996.

[71] U. Grenander and M. I. Miller, *Pattern Theory: From Representation to Inference.* Oxford, UK: Oxford University Press, 2007.

[72] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. ICASSP*, vol. 11, 1986, pp. 49–52.

[73] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE T. Signal Proces.*, vol. 40, no. 12, pp. 3043–3054, 1992.

[74] Y.-J. Wu and R.-H. Wang, "Minimum generation error training for HMM-based speech synthesis," in *Proc. ICASSP*, vol. 1, 2006, pp. I–89–I–92.

[75] A. Gunawardana and W. Byrne, "Discriminative speaker adaptation with conditional maximum likelihood linear regression," in *Proc. Eurospeech*, 2001, pp. 1203–1206.

[76] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[77] I. Csiszár and G. Tusnády, "Information geometry and alternating minimization procedures," in *Stat. Decis., Suppl. Issue Number 1*, 1984, pp. 205–237.

[78] S. P. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge, UK: Cambridge University Press, 2004.

[79] L. Devroye and L. Györfi, *Nonparametric Density Estimation: The L1 View.* John Wiley & Sons, 1985.

[80] G. Wahba, "Optimal convergence properties of variable knot, kernel, and orthogonal series methods for density estimation," *Ann. Stat.*, vol. 3, no. 1, pp. 15–29, 1975.

[81] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *Ann. Math. Stat.*, vol. 27, no. 3, pp. 832–837, 1956.

[82] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Stat.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[83] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, New York, NY, 2006.

[84] B. W. Silverman, "Density ratios, empirical likelihood and cot death," *Appl. Stat. – J. Roy. St. C*, vol. 27, no. 1, pp. 26–33, 1978.

[85] T. Minka, "Discriminative models, not discriminative training," Microsoft Research Cambridge, Tech. Rep. MSR-TR-2005-144, 2005. [Online]. Available: http://research.microsoft.com/pubs/70229/tr-2005-144.pdf

[86] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *J. Mach. Learn. Res.*, vol. 11, pp. 625–660, 2010.

[87] C. Cortes and V. N. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[88] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, 1st ed. Cambridge, MA: MIT Press, 2001.

[89] S. Lloyd, "Least squares quantization in PCM," *IEEE T. Inform. Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[90] P. C. Mahalanobis, "On the generalised distance in statistics," *Proc. Natl. Inst. Sci. India*, vol. 2, no. 1, pp. 49–55, 1936.

[91] V. Franc, A. Zien, and B. Schölkopf, "Support vector machines as probabilistic models," in *Proc. ICML*, 2011, pp. 665–672.

[92] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[93] B. Efron, "How biased is the apparent error rate of a prediction rule?" *J. Am. Stat. Assoc.*, vol. 81, no. 394, pp. 461–470, 1986.

[94] D. L. Verbyla, "Potential prediction bias in regression and discriminant analysis," *Can. J. Forest Res.*, vol. 16, no. 6, pp. 1255–1257, 1986.

[95] C. R. Shalizi, "Dynamics of Bayesian updating with dependent data and misspecified models," *Electron. J. Statist.*, vol. 3, pp. 1039–1074, 2009.

[96] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, University College London, 2003.

[97] R. M. Neal, "Probabilistic inference using markov chain monte carlo methods," Dept. of Computer Science, University of Toronto, Tech. Rep. CRG-TR-93-1, 1993.

[98] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.

[99] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *P. Roy. Soc. A. – Math. Phy.*, vol. 186, no. 1007, pp. 453–461, 1946.

[100] L. Wasserman, "Asymptotic inference for mixture models by using data-dependent priors," *J. Roy. Stat. Soc. B*, vol. 62, no. 1, pp. 159–180, 2000.

[101] A. R. Syversveen, "Noninformative bayesian priors: Interpretation and problems with construction and applications," Norges Teknisk-Naturvitenskapelige Universitet, Tech. Rep. 3, 1998.

[102] A. Gelman, A. Jakulin, M. G. Pittau, and Y.-S. Su, "A weakly informative default prior distribution for logistic and other regression models," *Ann. Appl. Stat.*, vol. 2, no. 4, pp. 1360–1383, 2008.

[103] L. J. Savage, *The Foundations of Statistics*, 2nd ed. New York, NY: Dover Publications, 1972.

[104] A. Gelman, "Objections to Bayesian statistics," *Bayesian Analysis*, vol. 3, no. 3, pp. 445–450, 2008.

[105] J. Surowiecki, *The Wisdom of Crowds*. New York, NY: Doubleday, 2005.

[106] Z. Ma and A. Leijon, "Bayesian estimation of beta mixture models with variational inference," *IEEE T. Pattern Anal.*, vol. 33, no. 11, pp. 2160–2173, 2011.

[107] C. E. Rasmussen, "The infinite Gaussian mixture model," in *Proc. NIPS*, vol. 12, 2000, pp. 554–560.

[108] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[109] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Stat.*, vol. 7, pp. 1–26, 1979.

[110] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[111] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

[112] R. E. Schapire, "The strength of weak learnability," *Mach. Learn.*, vol. 5, no. 2, pp. 197–227, 1990.

[113] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[114] P. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[115] P. J. Brockwell and R. A. Davis, *Introduction to Time Series and Forecasting*, 2nd ed. New York, NY: Springer, 2002.

[116] M. B. Priestley, *Non-Linear and Non-Stationary Time Series Analysis*. Waltham, MA: Academic Press, 1988.

[117] M. Brin and S. Garrett, *Introduction to Dynamical Systems*. Cambridge University Press, 2002.

[118] G. D. Birkhoff, "Proof of the ergodic theorem," *Proc. Natl. Acad. Sci. USA*, vol. 17, no. 12, pp. 656–660, 1931.

[119] C. R. Shalizi. (2010) The world's simplest ergodic theorem. [Online]. Available: http://bactra.org/weblog/668.html

[120] C. R. Shalizi and A. Kontorovich. (2010) Almost None of the Theory of Stochastic Processes. Version 0.1.1. http://www.stat.cmu.edu/~cshalizi/almost-none/.

[121] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, ser. Representation and Reasoning. San Mateo CA: Morgan Kaufmann, 1988.

[122] T. Koski and J. M. Noble, *Bayesian Networks: An Introduction*. Chichester, UK: John Wiley & Sons, 2009.

[123] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*, ser. International Series of Monographs on Physics. Oxford, UK: Oxford University Press, 2001, vol. 111.

[124] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," in *Parallel Distributed Computing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations*. MIT Press, 1986, ch. 6.

[125] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using 2-layer networks," in *Proc. NIPS*, 1992, pp. 912–919.

[126] H. Zen, K. Tokuda, and T. Kitamura, "Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences," *Comput. Speech Lang.*, vol. 21, no. 1, pp. 153–173, 2007.

[127] R. D. Shachter, "Bayes-ball: Rational pastime (for determining irrelevance and requisite information in belief networks and influence diagrams)," in *Proc. UAI*, 1998, pp. 480–487.

[128] P. Dagum and M. Luby, "Approximating probabilistic inference in Bayesian belief networks is NP-hard," *Artif. Intell.*, vol. 60, no. 1, pp. 141–153, 1993.

[129] O. Perron, "Zur Theorie der Matrices," *Math. Ann.*, vol. 64, no. 2, pp. 248–263, 1907.

[130] F. G. Frobenius, "Über Matrizen aus nicht negativen Elementen," *Sitzungsber. Königl. Preuss. Akad. Wiss.*, pp. 456–477, 1912.

[131] P. Whittle, *Hypothesis Testing in Time Series Analysis*. Stockholm, Sweden: Almqvist & Wiksell, 1951.

[132] H. O. A. Wold, *A Study in the Analysis of Stationary Time Series*, 2nd ed. Stockholm, Sweden: Almqvist & Wiksell, 1954.

[133] H. Tong and K. S. Lim, "Threshold autoregression, limit cycles and cyclical data," *J. Roy. Stat. Soc. B*, vol. 42, no. 3, pp. 245–292, 1980.

[134] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE T. Acoust. Speech*, vol. 37, no. 3, pp. 328–339, 1989.

[135] M. Kallas, P. Honeine, C. Richard, C. Francis, and H. Amoud, "Kernel-based autoregressive modeling with a pre-image technique," in *Proc. SSP*, 2011, pp. 281–284.

[136] ——, "Prediction of time series using Yule-Walker equations with kernels," in *Proc. ICASSP*, 2012, pp. 2185–2188.

[137] M. B. Rajarshi, "Bootstrap in Markov-sequences based on estimates of transition density," *Ann. I. Stat. Math.*, vol. 42, no. 2, pp. 253–268, 1990.

[138] G. E. Henter, A. Leijon, and W. B. Kleijn, "Kernel density estimation-based Markov models with hidden state," manuscript in preparation.

[139] P. Bühlmann and A. J. Wyner, "Variable length Markov chains," *Ann. Stat.*, vol. 27, no. 2, pp. 480–513, 1999.

[140] C. Shalizi and K. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in *Proc. UAI*, vol. 20, 2004, pp. 504–511.

[141] C. Shalizi, K. Shalizi, and J. Crutchfield, "An algorithm for pattern discovery in time series," Santa Fe Institute, Tech. Rep. 02-10-060, 2002, http://arxiv.org/abs/cs.LG/0210025.

[142] B. Weiss, "Subshifts of finite type and sofic systems," *Monatsh. Math.*, vol. 77, no. 5, pp. 462–474, Oct. 1973.

[143] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE T. Inform. Theory*, vol. 41, no. 3, pp. 653–664, 1995.

[144] G. Welch and G. Bishop, "An introduction to the Kalman filter," SIGGRAPH 2001 Course 8, 2001.

[145] E. A. Wan and A. T. Nelson, "Dual Kalman filtering methods for nonlinear prediction, smoothing, and estimation," in *Proc. NIPS 1996*, vol. 9, 1997.

[146] E. A. Wan, R. van der Merwe, and A. T. Nelson, "Dual estimation and the unscented transformation," in *Proc. NIPS 1999*, vol. 11, 2000, pp. 666–672.

[147] A. Schönhuth and H. Jaeger, "Characterization of ergodic hidden Markov sources," *IEEE T. Inform. Theory*, vol. 55, no. 5, pp. 2107–2118, 2009.

[148] R. E. Kálmán, "A new approach to linear filtering and prediction problems," *J. Basic Eng. – T. ASME*, vol. 82, no. 1, pp. 35–45, 1960.

[149] R. E. Kálmán and R. S. Bucy, "New results in linear filtering and prediction theory," *J. Basic Eng. – T. ASME*, vol. 83, no. 3, pp. 95–108, 1961.

[150] T. Jebara, "MAP estimation, message passing, and perfect graphs," in *Proc. UAI*. AUAI Press, 2009, pp. 258–267.

[151] J. R. Foulds, N. Navaroli, P. Smyth, and A. T. Ihler, "Revisiting MAP estimation, message passing and perfect graphs," in *Proc. AISTATS*, 2011, pp. 278–286.

[152] P. S. Gopalakrishnan, D. Kanevsky, A. Nádas, and D. Nahamoo, "An inequality for rational functions with applications to some statistical estimation problems," *IEEE T. Inform. Theory*, vol. 37, no. 1, pp. 107–113, 1991.

[153] D. Kanevsky, "Extended Baum transformations for general functions," in *Proc. ICASSP*, vol. 1, 2004, pp. I – 821–824.

[154] C. Shalizi and J. Crutchfield, "Computational mechanics: Pattern and prediction, structure and simplicity," *J. Stat. Phys.*, vol. 104, no. 3–4, pp. 817–879, 2001.

[155] G. E. Henter and W. B. Kleijn, "Picking up the pieces: Causal states in noisy data, and how to recover them," *Pattern Recogn. Lett.*, vol. 34, no. 5, pp. 587–594, 2013.

[156] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[157] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE T. Pattern Anal.*, vol. 31, no. 5, pp. 855–868, 2009.

[158] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, vol. 3, 2000, pp. 1635–1638.

[159] Q. Zhu, B. Y. Chen, N. Morgan, and A. Stolcke, "On using MLP features in LVCSR," in *Proc. Interspeech*, 2004.

[160] H. Zen, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *Proc. ICASSP*, 2013, pp. 7962–7966.

[161] Z.-H. Ling, L. Deng, and D. Yu, "Modeling spectral envelopes using restricted Boltzmann machines and deep belief networks for statistical parametric speech synthesis," *IEEE T. Speech Audi. P.*, vol. 21, no. 10, pp. 2129–2139, 2013.

[162] S. Kang, X. Qian, and H. Meng, "Multi-distribution deep belief network for speech synthesis," in *Proc. ICASSP*, 2013, pp. 8012–8016.

[163] D. Tweed, R. Fisher, J. Bins, and T. List, "Efficient hidden semi-Markov model inference for structured video sequences," in *Proc. VSPETS*, 2005, pp. 247–254.

[164] S. Asmussen, O. Nerman, and M. Olsson, "Fitting phase-type distributions via the EM algorithm," *Scand. J. Statist.*, vol. 23, no. 4, pp. 419–441, 1996.

[165] M. J. Russell and R. K. Moore, "Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition," in *Proc. ICASSP*, vol. 10, 1985, pp. 5–8.

[166] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Hidden semi-Markov model based speech synthesis," in *Proc. ICSLP*, 2004, pp. 1393–1396.

[167] J. Yamagishi, H. Zen, Y.-J. Wu, T. Toda, and K. Tokuda, "The HTS-2008 system: Yet another evaluation of the speaker-adaptive HMM-based speech synthesis system in the 2008 Blizzard Challenge," in *Blizzard Challenge 2008 Workshop*, 2008.

[168] N. Niwase, J. Yamagishi, and T. Kobayashi, "Human walking motion synthesis with desired pace and stride length based on HSMM," *IEICE T. Inf. Syst.*, vol. 88, no. 11, pp. 2492–2499, 2005.

[169] E. Bozkurt, S. Asta, S. Özkul, Y. Yemez, and E. Erzin, "Multimodal analysis of speech prosody and upper body gestures using hidden semi-Markov models," in *Proc. ICASSP*, 2013, pp. 3562–3656.

[170] G. E. Henter and W. B. Kleijn, "Intermediate-state HMMs to capture continuously-changing signal features," in *Proc. Interspeech*, vol. 12, 2011, pp. 1817–1820.

[171] G. E. Henter, M. R. Frean, and W. B. Kleijn, "Gaussian process dynamical models for nonparametric speech representation and synthesis," in *Proc. ICASSP*, 2012, pp. 4505–4508.

[172] J. A. Bilmes and C. Bartels, "Graphical model architectures for speech recognition," *IEEE Signal Proc. Mag.*, vol. 22, no. 5, pp. 89–100, 2005.

[173] N. Friedman, K. Murphy, and S. Russell, "Learning the structure of dynamic probabilistic networks," in *Proc. UAI*, vol. 14, 1998, pp. 139–147.

[174] J. D. Hamilton, "A new approach to the economic analysis of non-stationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, 1989.

[175] J. Dines, J. Yamagishi, and S. King, "Measuring the gap between HMM-based ASR and TTS," in *Proc. Interspeech*, 2009, pp. 1391–1394.

[176] G. Rigoll and A. Kosmala, "A systematic comparison between on-line and off-line methods for signature verification with hidden Markov models," in *Proc. ICPR*, vol. 2, 1998, pp. 1755–1757.

[177] M. Shannon, H. Zen, and W. Byrne, "The effect of using normalized models in statistical speech synthesis," in *Proc. Interspeech*, vol. 12, 2011.

[178] M. Shannon and W. Byrne, "Autoregressive HMMs for speech synthesis," in *Proc. Interspeech*, vol. 10, 2009, pp. 400–403.

[179] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE T. Inform. Theory*, vol. 44, no. 2, pp. 792–798, 1998.

[180] J. G. Cleary and W. J. Teahan, "Unbounded length contexts for PPM," *Comput. J.*, vol. 40, no. 2 and 3, pp. 67–75, 1997.

[181] F. Wood, J. Gasthaus, C. Archambeau, L. James, and Y. W. Teh, "The sequence memoizer," *Commun. ACM*, vol. 54, no. 2, pp. 91–98, 2011.

[182] J. Gasthaus, F. Wood, and Y. W. Teh, "Lossless compression based on the sequence memoizer," in *Proc. DCC*, 2010, pp. 337–345.

[183] J. Pitman and M. Yor, "The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator," *Ann. Probab.*, vol. 25, no. 2, pp. 855–900, 1997.

[184] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.

[185] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE T. Acoust. Speech*, vol. 26, no. 1, pp. 43–49, 1978.

[186] S.-X. Zhang and M. J. F. Gales, "Structured support vector machines for noise robust continuous speech recognition," in *Proc. Interspeech*, 2011, pp. 989–990.

[187] A. J. Hunt and A. W. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *Proc. ICASSP*, 1996, pp. 373–376.

[188] C.-H. Lee, M. A. Clements, S. Dusan, E. Fosler-Lussier, K. Johnson, B.-H. Juang, and L. R. Rabiner, "An overview on automatic speech attribute transcription (ASAT)," in *Proc. Interspeech*, 2007, pp. 1825–1828.

[189] A. Jansen and P. Niyogi, "Point process models for event-based speech recognition," *Speech Commun.*, vol. 51, no. 12, pp. 1155–1168, 2009.

[190] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. New York, NY: Springer, 2008.

[191] C. J. C. Burges, "Geometric methods for feature extraction and dimensional reduction," in *Data Mining and Knowledge Discovery Handbook*. New York, NY: Springer, 2005, pp. 59–91.

[192] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[193] D. Zhang, Z.-H. Zhou, and S. Chen, "Semi-supervised dimensionality reduction," in *Proc. SDM*, vol. 7, 2007, pp. 629–634.

[194] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *J. Roy. Stat. Soc. B*, vol. 61, no. 3, pp. 611–622, 1999.

[195] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Upper Saddle River, NJ: Prentice Hall, 1993.

[196] C. Koniaris, S. Chatterjee, and W. B. Kleijn, "Selecting static and dynamic features using an advanced auditory model for speech recognition," in *Proc. ICASSP*, 2010, pp. 4342–4345.

[197] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE T. Signal Proces.*, submitted. [Online]. Available: http://arxiv.org/abs/1304.6763

[198] J. H. McDermott and E. P. Simoncelli, "Sound texture perception via statistics of the auditory periphery: Evidence from sound synthesis," *Neuron*, vol. 71, no. 5, pp. 926–940, 2011.

[199] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. ICML*, 2009, pp. 609–616.

[200] G. E. P. Box and N. R. Draper, *Empirical Model-Building and Response Surfaces*. New York, NY: John Wiley & Sons, 1987.

[201] H. Akaike, "A new look at the statistical model identification," *IEEE T. Automat. Contr.*, vol. 19, no. 6, pp. 716–723, 1974.

[202] G. Schwarz, "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, no. 2, pp. 461–464, 1978.

[203] R. L. Kashyap, "Inconsistency of the AIC rule for estimating the order of autoregressive models," *IEEE T. Automat. Contr.*, vol. 25, no. 5, pp. 996–998, 1980.

[204] R. W. Katz, "On some criteria for estimating the order of a Markov chain," *Technometrics*, vol. 23, no. 3, pp. 243–249, 1981.

[205] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE T. Knowl. Data En.*, vol. 22, no. 10, pp. 1345–1359, 2010.

[206] S. Molau, F. Hilger, and H. Ney, "Feature space normalization in adverse acoustic conditions," in *Proc. ICASSP*, 2003, pp. I–656–I–659.

[207] H. Daumé III, "Frustratingly easy domain adaptation," in *Proc. ACL*, vol. 45, 2007, pp. 256–263.

[208] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Stat. Plan. Infer.*, vol. 90, no. 2, pp. 227–244, 2000.

[209] E. Eide and H. Gish, "A parametric approach to vocal tract length normalization," in *Proc. ICASSP*, 1996, pp. 346–348.

[210] T. Claes, I. Dologlou, L. ten Bosch, and D. Van Compernolle, "A novel feature transformation for vocal tract length normalization in automatic speech recognition," *IEEE T. Speech Audi. P.*, vol. 6, no. 6, pp. 549–557, 1998.

[211] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Comput. Speech Lang.*, vol. 12, no. 2, pp. 75–98, 1998.

[212] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. ACL*, vol. 34, 1996, pp. 310–318.

[213] I. H. Witten and T. C. Bell, "The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression," *IEEE T. Inform. Theory*, vol. 37, no. 4, pp. 1085–1094, 1991.

[214] G. J. Lidstone, "Note on the general case of the Bayes-Laplace formula for inductive or a posteriori probabilities," *T. Fac. Actuar.*, vol. 8, pp. 182–192, 1920.

[215] F. Jelinek and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," in *Proc. Workshop Pattern Recognit. Pract.*, 1980.

[216] G. E. Henter and W. B. Kleijn, "Simplified probability models for generative tasks: a rate-distortion approach," in *Proc. EUSIPCO*, vol. 18, 2010, pp. 1159–1163.

[217] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models," in *Proc NIPS 2005*, vol. 18, 2006, pp. 1441–1448.

[218] ——, "Gaussian process dynamical models for human motion," *IEEE T. Pattern Anal.*, vol. 30, no. 2, pp. 283–298, Feb. 2008.