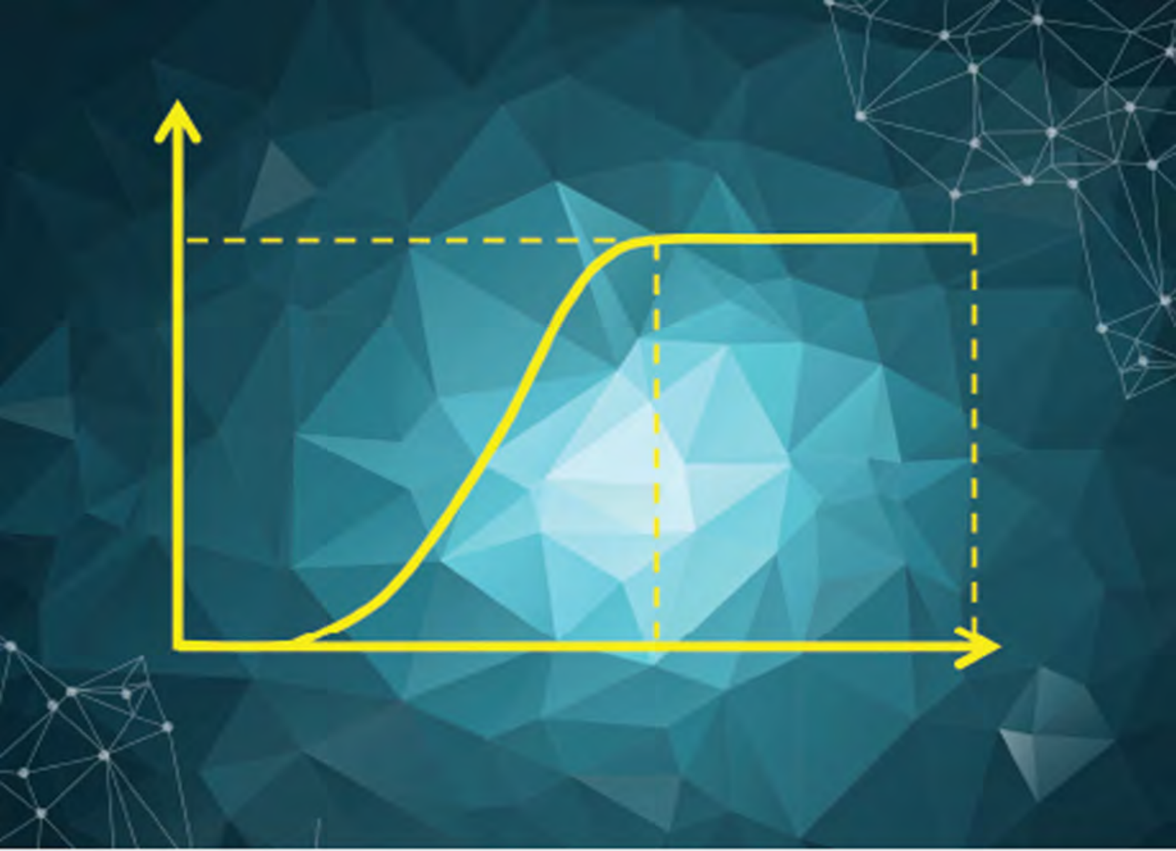

Data Science for Wind Energy

Yu Ding

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

Data Science for Wind Energy

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Data Science for Wind Energy

Yu Ding

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Printed on acid-free paper
Version Date: 20190508

International Standard Book Number-13: 978-1-138-59052-6 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

To my parents.

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

Contents

Foreword

Wind power is a rapidly growing source of renewable energy in many parts of the globe. Building wind farms and maintaining turbine assets also provide numerous job opportunities. As a result, the wind energy sector plays an increasingly important role in the new economy. While being scaled up, efficiency and reliability become the key to making wind energy competitive. With the arrival of the data science and machine learning era, a lot of discussions are being made in the related research community and wind industry, contemplating strategies to take full advantage of the potentials and opportunities unleashed by the large amount of data to address the efficiency and reliability challenges.

Data Science for Wind Energy arrives at the right time, becoming one of the first dedicated volumes to bridge the gap, and provides expositions of relevant data science methods and abundant case studies, tailored to address research and practical challenges in wind energy applications.

This book of eleven technical chapters is divided into three parts, unified by a general data science formulation presented in Chapter 1. The overarching formulation entails the modeling and solution of a set of probability density functions, conditional or otherwise, not only to account for the mean estimation or prediction, but also to allow for uncertainty quantification. The first part of the book embodies the modeling of a spatio-temporal random wind field and uses that as a springboard for better forecasting. Chapter 2 recaps the existing methods for modeling data in a univariate time series, and Chapters 3 and 4 bring to the readers many new data science concepts and methods. The asymmetry quantification and asymmetric spatio-temporal modeling introduced in Chapter 3 and the regime-switching methods discussed in Chapter 4 are particularly interesting. The second part of the book concentrates on the system-level, power production-oriented turbine performance assessment. This part starts off with a power curve analysis (Chapter 5), followed by adding physically informed constraints to power curve modeling for devising productive efficiency metrics (Chapter 6). Chapters 7 and 8 further discuss, respectively, the circumstances when a turbine's performance can be enhanced by a purposeful action or diminished due to the wake effect. The third part of the book focuses on reliability management and load analysis for wind turbines, nested within an integrative framework combining models, simulations and data (Chapter 9). The load analysis for reliability assessment involves heavily statistical sampling techniques, as detailed in Chapters 10 and

11, and those methods are useful to general reliability engineering purposes—my own research on electrical power system reliability has been benefited by these data science methodologies. I am pleased to see the anomaly detection and fault diagnosis methods presented in Chapter 12, borrowing experiences and successes from other industries for the benefit of wind energy practice.

One of the reasons I am fond of this book is the author's diligence and generosity in collecting, arranging, and releasing ten important wind farm datasets, more than 150 megabytes in volume, plus another 440 megabytes of simulated data used in reliability verification. On top of that, the author provides computer codes for all eleven technical chapters, most of them in `R` while some are in `MATLAB`®, either for reproducing figures and tables in the book or implementing some major algorithm. I am sure that those data and codes will immensely help both academic researchers and practitioners.

To appreciate a book, it is helpful to understand the author. I had the good fortune to get to know Dr. Yu Ding shortly after he joined Texas A&M faculty in 2001. There was a university-wide event celebrating the 125th anniversary of Texas A&M University. Yu and I happened to sit next to each other at the same table, and at that moment, I had been with the university for 23 years, while Yu had been for about 25 days. In the ensuing years, Yu's path and mine have crossed often. We served on the committees of each other's students, co-authored papers and co-directed research projects, and because of these, I am reasonably familiar with most of the materials presented in this book. I have witnessed Yu's quick ascent to a leading and authoritative researcher on the intersection of data science and wind energy. Yu's unique multidisciplinary training and penetrating insights allow him and his research team to produce many influential works, contributing to methodology development and benefiting practices. Yu's work on turbine performance assessment in particular leads to large-scale fleet-wide implementations, rendering multi-million-dollar extra revenues. Not surprisingly, Yu was recognized with a Research Impact Award by Texas A&M College of Engineering in May 2018 "*for innovations in data and quality science impacting the wind energy industry.*"

It is thus a great pleasure for me to introduce this unique and timely book and a dear colleague to the academia and practitioners who want to know more about data science for wind energy.

Chanan Singh
Regents Professor and Irma Runyon Chair Professor
Electrical & Computer Engineering Department
Texas A&M University, College Station, Texas

June 2019

Preface

All models are wrong but some are useful.

— George E. P. Box

My introduction to the field of wind energy started from a phone call taking place sometime in 2004. Dr. Jiong Tang of the University of Connecticut called and asked if I would be interested in tackling some wind turbine reliability problems.

I got to know Jiong when we were both mechanical engineering graduate students at the Pennsylvania State University. I later left Penn State for my doctoral study at the University of Michigan. My doctoral research was oriented towards a specialty area of data science—the quality science, which employs and develops statistical models and methods for quality improvement purpose. Prior to that phone call, my quality science applications were exclusively in manufacturing. I reminded Jiong that I knew almost nothing about wind turbines and wondered how I could be of any help. Jiong believed that data available from turbine operations had not been taken full advantage of and thought my data science expertise could be valuable. I was intrigued by the research challenges and decided to jump at the opportunity.

The first several years of my wind energy research, however, involved little data. Although the industry had gathered a large amount of operational data through the supervisory control and data acquisition systems of turbines, we had a hard time persuading any turbine manufacturer or owner/operator to share their data. Our luck turned around a few years later, after we aligned ourselves with national labs and several wind companies. Through the academia-government-industry partnership, my research group was able to collect over 100 gigabytes wind turbine testing data and wind farm operational data. Working with the vast amount of real-world data enabled me to build a rewarding career that developed data science methods to address wind energy challenges and it is still going strong.

While working in the wind energy area, I benefited from having a mechanical engineering background. The majority of wind energy research is carried out, for understandable reasons, by domain experts in aerospace, mechanical, civil, or electrical engineering. My engineering training allows me to communicate with domain experts with ease. Maybe this is why Jiong thought of involving me in his wind turbine project in the first place.

As I got involved more and more in the field of wind energy, I observed a disconnection between this typical engineering field and the emerging field of data science. Wind engineers or wind engineering researchers routinely handle data, but most of the domain experts are not exposed to systematic data science training while in schools because the engineering curricula, until very recently, offered only basic engineering statistics. This did not keep pace with the fast development of new ideas and methods introduced by data science in the past twenty years. On the other hand, wind engineering, like most other substantial engineering fields, finds a relatively small number of trained data scientists from computer science or statistics disciplines working in the area, probably because the entry barrier associated with mastering domain knowledge appears intimidating. This may explain that while there are plenty of generic data science and machine learning books, books that can bridge the two distinctive fields and offer specific and sophisticated data science solutions to wind energy problems are, in fact, scarce.

I had been thinking of writing a book filling precisely this void. I came to realize in early 2017 that I may have enough materials when I was leading a research team and preparing a National Science Foundation proposal to its BIG DATA program. In fact, the structure of this book closely mirrors the structure of that proposal, as it embodies three main parts discussing, respectively, wind field analysis, wind turbine performance analysis, and wind turbine load and reliability management. The 2017 NSF proposal was funded at the end of the summer, and, I decided to submit the book proposal to Chapman & Hall/CRC Press later in 2017.

I am grateful for the opportunities and privilege to work with many talented individuals on a problem of national importance. A few of those individuals played pivotal roles in my wind energy research career. The first is obviously Dr. Jiong Tang—without him, I wouldn't be writing this preface. Then, there is Dr. Eunshin Byon, a former Ph.D. student of mine and now a faculty member at the University of Michigan. Eunshin was the first student who worked with me on wind energy research. She came to my group during that aforementioned "data-light" period. Understandably, it was a difficult time for those of us who work with data. Eunshin was instrumental in sustaining our research at that time, finding data through public sources and testing innovative ideas that lay the foundation for the subsequent collaborations with several industry members. I am delighted to see that Eunshin becomes a recognized expert herself in the intersecting area of data science and wind energy.

I appreciate immensely Mr. Brian Hayes, Executive Vice President of EDP Renewables, North America, for his vision in starting the Texas A&M-EDP Renewables partnership and his generous support in funding our research and sharing their wind farm operational data. I am deeply grateful to Dr. Shuangwen (Shawn) Sheng at the National Renewable Energy Laboratory for engaging my research team at the national or international level and for countless hours of stimulating discussions that drive my research to new levels. Of course, I am indebted to my Ph.D. advisor, Dr. Jianjun Shi, then at the University of Michigan and now with the Georgia Institute of Technology, for bringing me to the data science world and for teaching me how to be an independent researcher.

Last but not least, I would like to thank my wife, Ying Li, and our daughter, Alexandra, for their love and support.

Yu Ding
Texas A&M University
College Station, Texas

June 2019

Acknowledgments

Special thanks goes to the following former and current students who help arrange the datasets and provide computer codes for producing many tables and figures or implementing certain algorithms:

- Hoon Hwangbo helped arrange most of the datasets used in this book, except the `Wind Spatial-Temporal Dataset1`, `Wind Spatial-Temporal Dataset2`, and `Simulated Bending Moment Dataset`, which were prepared by others. Hoon also provided the code for generating Tables 6.1–6.3, Table 7.4, Tables 8.2–8.5, and Tables 10.2–10.6, and for generating Figures 6.3–6.6, Figures 6.12–6.14, Figure 7.5, Figure 8.4, Figure 8.7, Figure 8.9, Figures 10.2–10.4, and Figures 10.6–10.10. Furthermore, Hoon created the illustrations in Figures 6.7, 6.9, and 6.11.

- Abhinav Prakash provided the code for generating Tables 2.1–2.8 and Tables 5.2–5.8, and for generating Figures 2.1–2.5, Figure 3.1 and Figure 3.2. Additionally, Abhinav created the illustrations in Figures 12.2, 12.4, and 12.6.

- Arash Pourhabib arranged the `Wind Spatial-Temporal Dataset1` and provided the code for generating Tables 3.1–3.3 and Figure 3.3.

- Ahmed Aziz Ezzat arranged the `Wind Spatial-Temporal Dataset2` and provided the code for generating Tables 3.4–3.8, Tables 4.4–4.6, and for generating Figures 3.4–3.7, Figure 4.4, and Figure 4.9. Aziz also created the illustrations in Figure 4.3 and Figures 4.5–4.7.

- Giwhyun Lee developed the original code for implementing the methods in Section 5.2, Section 7.3, and Chapter 10. Giwhyun also created the illustration in Figure 5.7.

- Eunshin Byon provided the `Simulated Bending Moment Dataset`, the code for Algorithm 9.1 (generating Figure 9.6), and the code for establishing the generalized additive model in Section 11.4.1 as the conditional probability of exceedance function. Eunshin also created the illustrations in Figures 9.2, 9.3, 9.7, 9.8, 10.5, and 11.6.

- Imtiaz Ahmed provided the code for Algorithm 12.3.

I would like to acknowledge the contribution of many people to the research work that forms the backbone of this book. Marc Genton, Jianhua Huang, Andrew Johnson, Mikyoung Jun, Bani K. Mallick, Lewis Ntaimo, Chanan Singh, Le Xie, and Li Zeng are faculty members who collaborated with me on wind energy research topics. Mithun P. Acharya, Daniel Cabezon-Martinez, Andrew Cordes, Aldo Dagnino, Oliver Eisele, Travis Galoppo, Ron Grife, Jaimeet Gulati, Ulrich Lang, Georgios Pechlivanoglou, and Guido Weinzierl were our industrial collaborators. Many of the former and current students, in addition to those mentioned above, contributed to various aspect of my wind energy research: Jason Lawley, Briana Niu, David Pérez, Eduardo Pérez, and Yei-Eun Shin. Randi Cohen and the team at Chapman & Hall/CRC Press have done a fantastic job in managing the book project and assisted me in numerous occasions. I also gratefully acknowledge NSF for its support of this work and Mike and Sugar Barnes for their generous endowment.

The author's following publications are reused, in part or in whole, in the respective chapters.

- **Chapter 3**
  Pourhabib, Huang, and Ding. "Short-term wind speed forecast using measurements from multiple turbines in a wind farm." *Technometrics*, 58:138–147, 2016.

  Ezzat, Jun, and Ding. "Spatio-temporal asymmetry of local wind fields and its impact on short-term wind forecasting." *IEEE Transactions on Sustainable Energy*, 9:1437–1447, 2018.

- **Chapter 4**
  Ezzat, Jun, and Ding. "Spatio-temporal short-term wind forecast: A calibrated regime-switching method." *The Annals of Applied Statistics*, in press, 2019.

- **Chapter 5**
  Lee, Ding, Genton, and Xie. "Power curve estimation with multivariate environmental factors for inland and offshore wind farms." *Journal of the American Statistical Association*, 110:56–67, 2015.

- **Chapter 6**
  Hwangbo, Johnson, and Ding. "A production economics analysis for quantifying the efficiency of wind turbines." *Wind Energy*, 20:1501–1513, 2017.

  Niu, Hwangbo, Zeng, and Ding. "Evaluation of alternative efficiency metrics for offshore wind turbines and farms." *Renewable Energy*, 128:81–90, 2018.

Hwangbo, Johnson, and Ding. “Power curve estimation: Functional estimation imposing the regular ultra passum law.” *Working Paper*, 2018. Available at `SSRN:http://ssrn.com/abstract=2621033`.

- **Chapter 7**
  Lee, Ding, Xie, and Genton. “Kernel Plus method for quantifying wind turbine upgrades.” *Wind Energy*, 18:1207–1219, 2015.

  Hwangbo, Ding, Eisele, Weinzierl, Lang, and Pechlivanoglou. “Quantifying the effect of vortex generator installation on wind power production: An academia-industry case study.” *Renewable Energy*, 113:1589–1597, 2017.

  Shin, Ding, and Huang. “Covariate matching methods for testing and quantifying wind turbine upgrades.” *The Annals of Applied Statistics*, 12:1271–1292, 2018.

- **Chapter 8**
  Hwangbo, Johnson, and Ding. “Spline model for wake effect analysis: Characteristics of single wake and its impacts on wind turbine power generation.” *IISE Transactions*, 50:112–125, 2018.

- **Chapter 9**
  Byon and Ding. “Season-dependent condition-based maintenance for a wind turbine using a partially observed Markov decision process.” *IEEE Transactions on Power Systems*, 25:1823–1834, 2010.

  Byon, Ntaimo, and Ding. “Optimal maintenance strategies for wind turbine systems under stochastic weather conditions.” *IEEE Transactions on Reliability*, 59:393–404, 2010.

  Byon, Pérez, Ding, and Ntaimo. “Simulation of wind farm operations and maintenance using DEVS.” *Simulation–Transactions of the Society for Modeling and Simulation International*, 87:1093–1117, 2011.

- **Chapter 10**
  Lee, Byon, Ntaimo, and Ding. “Bayesian spline method for assessing extreme loads on wind turbines.” *The Annals of Applied Statistics*, 7:2034–2061, 2013.

- **Chapter 12**
  Ahmed, Dagnino, and Ding. “Unsupervised anomaly detection based on minimum spanning tree approximated distance measures and its application to hydropower turbines.” *IEEE Transactions on Automation Science and Engineering*, 16: 654–667, 2019.

CHAPTER 1

Introduction

Wind energy has been used as far back as Roman Egypt [51] (or even earlier [194]). The well-preserved windmills that dotted the Dutch coastline or along the Rhine River have become symbols of usage before the modern age. Although outdated, those windmills are top tourist attractions nowadays. As widespread as those windmills were, wind energy played a rather minor role in commercial electricity generation until the end of the last century. In 2000, the wind power generation in the United States was 5.59 billion kilowatt-hours (kWh), accounting for about 0.15% of the total electricity generated by the US in that year [219]. In the past decade, however, wind energy witnessed a rapid development and deployment. By the end of 2016, the annual wind power production increased 40-fold relative to the amount of wind power in 2000, to nearly 227 billion kWh, and accounted for 5.6% of the total electricity generation in that year [220]. The US Department of Energy even contemplates scenarios in which wind may generate 10% of the nation's electricity by 2020, 20% by 2030, and 35% by 2050 [217].

Remarkable progress has been made in wind turbine technology, which enables the design and installation of larger turbines and allows wind farms to be built at locations where wind is more intermittent and maintenance equipment is less accessible. All these brought new challenges to operational reliability. In an effort to maintain high reliability, with the help of advancement in micro-electronics, modern wind farms are equipped with a large number and variety of sensors, including, at the turbine level, anemometers, tachometers, accelerometers, thermometers, strain sensors, and power meters, and at the farm level, anemometers, vanes, sonars, thermometers, humidity meters, pressure meters, among others. These sensors churn out a lot of data at a fast pace, presenting unprecedented opportunities for data science to play a crucial role in addressing technical challenges in wind energy.

Like solar energy, wind energy faces an intermittent nature of its source. People commonly refer to wind and solar energy as *variable* renewable energy sources. The intermittency makes wind and solar power different from most other types of energy, even hydropower, as reservoirs built for hydropower plants smooth out the impact of irregularity and variability in precipitation on hydropower production.

The intermittency in wind presents a number of challenges to wind energy operations. The non-steady mechanical load yields excessive wear in a turbine's drive train, especially the gearbox and bearings, and makes the wind turbines prone to fatigue failures—wind turbines operate just like a car being driven in a busy city with plenty of traffic lights and rarely any freeway. Meanwhile, the randomness in wind power output makes it difficult to accommodate a substantial level of wind power in the power grid. All these lead to an increased cost and a decreased market competitiveness for wind energy. No wonder that as of 2016, the federal production tax credit (PTC) for wind was still valued at 23 cents per kWh, roughly 30% of the levelized cost of energy for onshore wind. Undoubtedly, this tax credit considerably boosts the marketability of wind energy, but without it, the competitiveness of wind energy will be called into question.

As data continues to be accumulated, data science innovations, providing profound understanding of wind stochasticity and enabling the design of countermeasures, have the potential of generating ground-breaking advancements in the wind industry. The commercial competitiveness of wind energy can benefit a great deal from a good understanding of its production reliability, which is affected by the unpredictability of wind and the productivity of wind turbines. The latter, furthermore, depends on a turbine's ability to convert wind into power during its operation and the availability or reliability of wind turbines. Data science solutions are needed in all of these aspects.

1.1 WIND ENERGY BACKGROUND

The focus of this book is data analytics at the wind turbine and wind farm level. A thorough coverage of such a scope entails a wide variety of data and a broad array of research issues. While data analytics at the power grid level is also an important part of wind energy research, the author's research has yet to be extended to that area. Hence, the scope of this book does not include data analytics at the power grid level. Nevertheless, a great deal of the turbine-level and farm-level data analytics is related to grid-level data analytics. For example, power predictions have a significant impact on grid integration.

The wind turbines considered here are the utility-scale, horizontal axis turbines. As illustrated in Fig. 1.1, a turbine, comprising thousands of parts, has three main, visible components: the blades, the nacelle, and the tower. The drive train and control system, including the gearbox and the generator, are inside the nacelle. While the vast majority of horizontal axis wind turbines use a gearbox to speed up the rotor speed inside the generator, there are also direct drive wind turbines in which the gearbox is absent and the rotor directly drives the generator. An anemometer or a pair of them can be found sitting on top of the nacelle, towards its rear end, to measure wind speed, whereas a vane is for the measurement of wind direction. Responding to changes in wind direction, yaw control is to rotate and point the nacelle to where the wind comes from. Responding to changes in wind speed, pitch control turns the blades in relation to the direction of the incoming air flow, adjusting the capability of the turbine to absorb the kinetic energy in the wind or the turbine's efficiency in doing so.

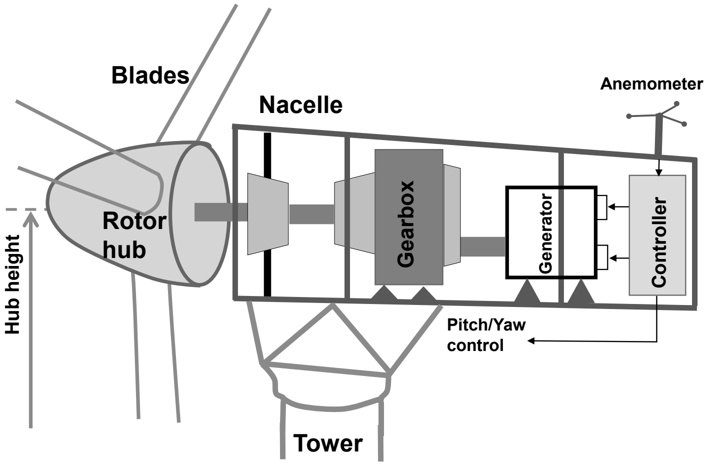

FIGURE 1.1 Schematic of major parts in a wind turbine.

A commercial wind farm can house several hundred wind turbines. For instance, the Roscoe Wind Farm, the largest wind farm in Texas as of this writing, houses 627 wind turbines. Other than turbines, meteorological masts are installed on a wind farm, known as the met towers or met masts. A number of instruments and sensors are installed on the met towers, measuring additional environmental conditions, such as temperature, air pressure, humidity, precipitation, among others. Anemometers and vanes are usually installed at multiple heights of a met tower. The multi-height measurements allow the calculation of vertical wind shear, which characterizes the change in wind speed with height, as well as the calculation of vertical wind veer, which characterizes the change in wind direction with height. The wind speed and direction measured at the nacelle during a commercial operation are typically only at the hub height.

Throughout the book, denote by $\boldsymbol{x}$ the input vector whose elements are the environmental variables, which obviously include wind speed, V, in the unit of meters per second (m/s), and wind direction, D, in degrees (°). The zero degree corresponds to due north. Sometimes analysts combine the speed and direction information of wind and express them in two wind velocities along the longitudinal and latitudinal directions, respectively. Other environmental variables include air density, ρ, humidity, H, turbulence intensity, I, and wind shear, S. Not all of these environmental variables are directly measured. Some of them are computed, such as,

- Turbulence intensity, I: first compute the standard deviation of the wind speeds in a short duration and denote it as $\hat{\sigma}$. Then, $I = \hat{\sigma}/\bar{V}$, where $\bar{V}$ is the average wind speed of the same duration. It is worth noting that the concept of turbulence intensity in air dynamics is similar to the coefficient of variation concept in statistics [58].

- Wind shear, S: wind speeds, V_1 and V_2, are measured at heights h_1 and h_2, respectively. Then, the vertical wind shear between the two heights is $S = \ln(V_2/V_1)/\ln(h_2/h_1)$ [175]. When anemometers are installed at locations both above and below the rotor hub, then two wind shears, the above-hub wind shear, S_a, and the below-hub wind shear, S_b, can be calculated.

- Air density, ρ, in the unit of kilograms per cubic meter (kg/m^3): given air temperature, T, expressed in Kelvin and air pressure, P, expressed in Newtons per square meter (N/m^2), $\rho = P/(\varrho \cdot T)$, where $\varrho = 287$ Joule/(kg·Kelvin) is the gas constant [216].

Using the above notation, the input vector to a turbine can be expressed as $\boldsymbol{x} = (V, D, \rho, H, I, S_a, S_b)^T$. But the input vector is not limited to the aforementioned variables. The hours in a day when a measurement is recorded, the power output of a nearby turbine, wind directional variation and wind veer if either or both are available, could also be included in the input vector, $\boldsymbol{x}$. On the other hand, while the wind speed, wind direction, and temperature measurements are commonly available on commercial wind farms, the availability of other measurements may not be.

Two types of output of a wind turbine are used in this book: one is the active power measured at a turbine, denoted by y and in the unit of kilowatts (kW) or megawatts (MW), and the other one is the bending moment, a type of mechanical load, measured at critical structural spots, denoted by z and in the unit of kiloNewtons-meter (kN-m) or million Newtons-meter (MN-m). The power output measures a turbine's power production capability, while the bending moment measurements are pertinent to a turbine's reliability and failure management. The power measurement is available for each and every turbine. Analysts may also aggregate the power outputs of all turbines in an entire wind farm when the whole farm is treated as a single power production unit. The bending moment measurements are currently not available on commercially operated turbines. They are more commonly collected on testing turbines and used for design purposes.

The input and output data can be paired into a data record. For the power response, it is the pair of $(\boldsymbol{x}, y)$, whereas for the mechanical load response, it is $(\boldsymbol{x}, z)$.

Turbine manufacturers provide a wind speed versus power functional curve, referred to as the *power curve*. Fig. 1.2 presents such a power curve. As shown in the power curve, a turbine starts to produce power after the wind reaches the cut-in speed, V_{ci}. A nonlinear relation between y and V then ensues, until the wind reaches the rated wind speed, V_r. When the wind speed is beyond V_r, the turbine's power output will be capped at the rated power output, y_r, also known as the nominal power capacity of the turbine, using control mechanisms such as pitch control and rotor speed regulation. The turbine will be halted when the wind reaches the cut-out speed, V_{co}, because high wind is deemed harmful to the safety of a turbine. The power curve shown here is an ideal power curve, also known as the nominal power curve. When the actual measurements of wind speed and power output are used, the V-versus-y plot will not appear as slim and smooth as the nominal power curve; rather, it will be a data scattering plot, showing considerable amount of noise and variability.

In order to protect the confidentiality of the data providers, the wind power data used in this book are normalized by the rated power, y_r, and expressed as a standardized value between 0 and 1.

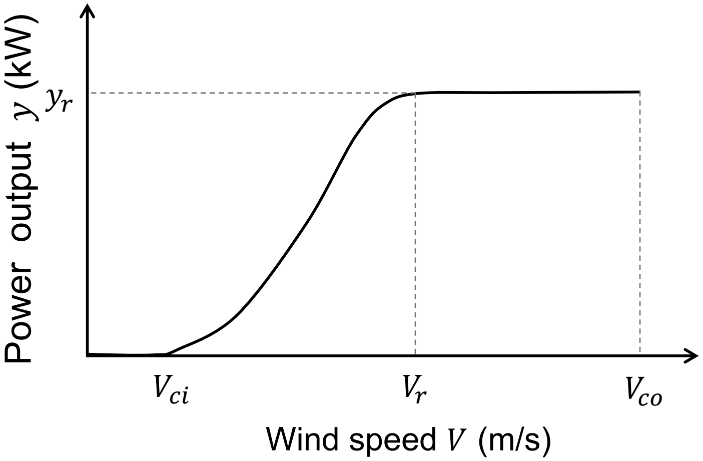

FIGURE 1.2 Nominal power curve of a wind turbine. (Reprinted with permission from Lee et al. [132].)

The raw data on wind turbines are recorded in a relatively fast frequency, in the range of a couple of data points per second to a data point per a couple of seconds. The raw data are stored in a database, referred to as the data historian. When the data are used in the turbine's supervisory control and data acquisition (SCADA) system, the current convention in the wind industry is to average the measurements over 10-minute time blocks because wind speed is assumed stationary over this 10-min duration and other environmental variables are assumed nearly constant. These assumptions are, of course, not always true. In this book, however, we choose to follow this industrial standard practice. With 10-min blocks, a year's worth of data has about

52,560 data pairs if there is no missing data at all. In reality, even with automated measurement devices, missing data is common, almost always making the actual data amount fewer than 50,000 for a year.

Even though the wind speed used is mostly a 10-min average, we decide to drop the overline while representing this average, for the sake of notational simplicity. That is to say, we use V, instead of $\bar{V}$, to denote the average wind speed in a 10-min block. When $\bar{V}$ is used, it refers to the average of 10-min averaged wind speeds.

Fig. 1.3 shows the arrangement of the multi-turbine, multi-year data for a wind farm. In the top panel, the whole dataset is shown as a cube, in which each cross section represents the spatial layout of turbines on a farm and the horizontal axis represents the time. The longitudinal data are the time-series of a turbine's power output, y, and environmental measurements, $\boldsymbol{x}$. The cross-sectional data, or the snapshot data, are of multiple turbines but are for a particular point in time. A cross section could be a short time period, for instance, a couple of days or weeks, during which the turbine's innate condition can be assumed unchanged. The power curve of a turbine is visualized as the light-colored (yellow) curve in the bottom panel (see also Color eBook), with the actual measurements in the background. As mentioned earlier, the actual measurements are noisy, and the nominal power curve averages out the noise.

1.2 ORGANIZATION OF THIS BOOK

We organize this book based on a fundamental data science formulation for wind power production:

$$f_t(y) = \int_{\boldsymbol{x}} f_t(y|\boldsymbol{x}) f_t(\boldsymbol{x}) d\boldsymbol{x}, \tag{1.1}$$

where $f(\cdot)$ denotes a probability density function and the subscript t, the time indicator, signifies the dynamic, time-varying aspect of the function.

This formulation implies that in order to understand $f_t(y)$, namely the stochasticity of power output y, it is necessary to understand the distribution of wind and other environmental variables, $f_t(\boldsymbol{x})$, as well as the turbine's power production conditioned on a given wind and environmental condition $\boldsymbol{x}$. We use a conditional density function, $f_t(y|\boldsymbol{x})$, to characterize the conditional distribution.

When the power output, y, is replaced by the mechanical load response (namely the bending moment), z, the above formulation is still meaningful, with $f(z|\boldsymbol{x})$ representing the conditional load response for a given environmental condition.

The use of conditional density functions is a natural result of wind intermittency. When the driving force to a turbine changes constantly, the turbine's response, regardless of being the power or the load, ought to be characterized under a given wind and environmental condition.

This book aims to address three aspects related to the aforementioned

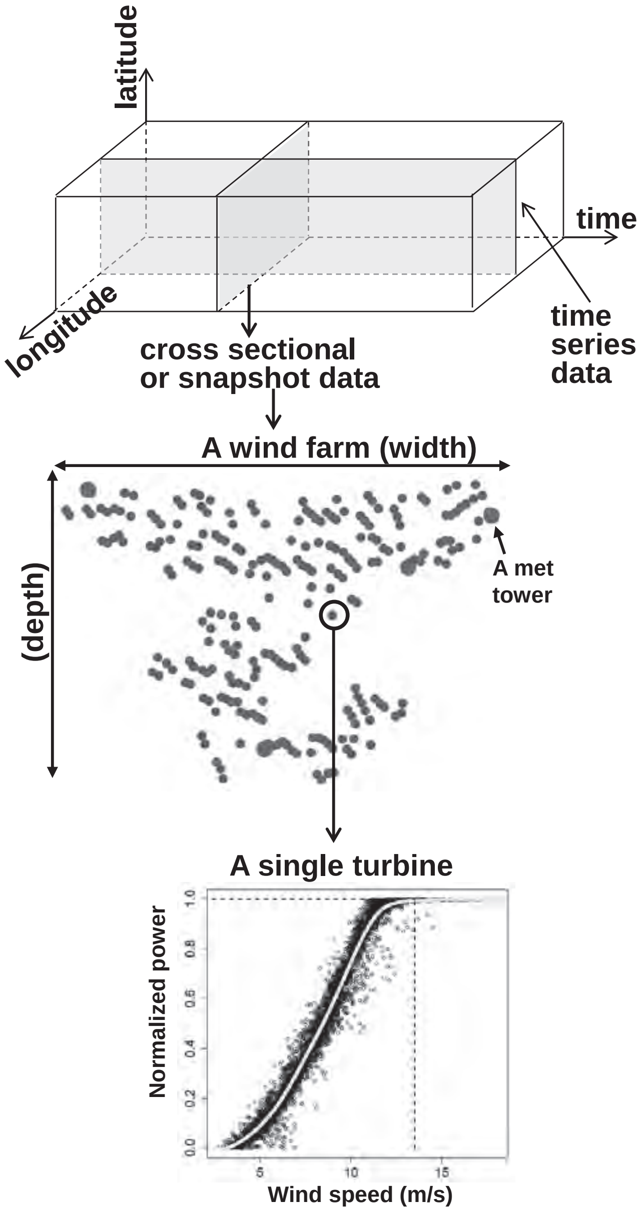

FIGURE 1.3 Arrangement of wind farm data. The top panel shows the spatio-temporal arrangement of wind farm data; the middle panel shows the layout of a wind farm, where the small dots are wind turbines and the big dots are the met towers; and the bottom panel presents the data from a single turbine, where the light-colored (yellow) curve (see Color eBook) is the nominal power curve and the circles in the background are the actual power measurements.

general formulation of wind power production. Thus, we divide the rest of this book into three parts:

1. The first part consists of three chapters. It is about the modeling of $f_t(\boldsymbol{x})$, which begets an analysis of the wind field. Based on the modeling and analysis of the wind field, a wind forecast can be made. If a whole wind farm is simplified as a single location, or the forecast at a single turbine is of concern, the need for a temporal analysis arises. If multiple turbines at different sites are to be studied, or multiple wind farms at different geographic locations are involved, the modeling of $f_t(\boldsymbol{x})$ becomes a spatio-temporal analysis. Both temporal and spatio-temporal methods will be described but the focus is on the spatio-temporal analysis.

2. The second part consists of four chapters. It discusses power response modeling and shows how the power response model can be used for performance evaluation of wind turbines. The general expression, $f(y|\boldsymbol{x})$, depicts a multivariate, probabilistic power response surface. The power curve is in fact the conditional expectation, $\mathbb{E}(y|\boldsymbol{x})$, when $\boldsymbol{x}$ is reduced to a univariate input, the wind speed, V. The modeling of $f(y|\boldsymbol{x})$ or $\mathbb{E}(y|\boldsymbol{x})$ falls into the area of density regression or nonparametric regression analysis.

3. The third part consists of four chapters. It provides a reliability and load analysis of wind turbines. Using Eq. 1.1 to assess power production assumes, implicitly, an up-running wind turbine, namely a non-zero $f_t(y|\boldsymbol{x})$. But wind turbines, under non-steady wind forces, are prone to failures and downtime. To factor in a turbine's reliability impact, it is important to assess a turbine's load response under various wind conditions. The statistical learning underlying the analysis in this part is related to sampling techniques, including importance sampling and Markov chain Monte Carlo sampling.

1.2.1 Who Should Use This Book

The book is intended to be a research monograph, but it can be used for teaching purposes as well. We expect our readers to have basic statistics and probability knowledge, and preferably a bachelor's degree in STEM (Science, Technology, Engineering, and Math). This book provides an in-depth discussion of how data science methods can improve decision making in several aspects of wind energy applications, from near-ground wind field analysis and wind forecast, turbine power curve fitting and performance analysis, turbine reliability assessment, to maintenance optimization for wind turbines and wind farms. A broad set of data science methods are covered, including time series models, spatio-temporal analysis, kernel regression, decision trees, splines, Bayesian inference, and random sampling. The data science methods are described in the context of wind energy applications with examples and case studies. Real data and case studies from wind energy research and industrial practices are used in this book. Readers who may benefit from reading this book include practitioners in the wind industry who look for data science solutions and faculty members and students who may be interested in the research of data science for wind energy in departments such as industrial and systems engineering, statistics, and power engineering.

There are a few books on renewable energy forecasting [117], which overlap, to a certain degree, with the content of Part I. A topic related to wind energy but left out in the book is about grid integration, for which interested readers can refer to the book by Morales et al. [148].

1.2.2 Note for Instructors

This book can be used as the textbook for a stand-alone course, with the course title the same as or similar to the title of this book. It can also be used to as a reference book that provides supplementary materials for certain segments of either a data science course (supplementing wind energy application examples) or a power engineering course (supplementing data science methods). These courses can come from the offerings of a broad set of departments, including Industrial Engineering, Electrical Engineering, Statistics, Aerospace Engineering, or Computer Science.

We recommend that the first chapter be read before later chapters are covered. The three parts after the first chapter are more or less independent of each other. It does not matter in which sequence the three parts are read or taught. Within each part, however, we recommend following the order of the chapters. It will take two semesters to teach the whole book. One can, nevertheless, sample one or two chapters from each part to form the basis for a one-semester course.

Most of the examples are solved using the `R` programming language, while some are solved using the `MATLAB`® programming language. At the end of a chapter, acronyms and abbreviations used in that chapter are summarized and explained in the Glossary section.

1.2.3 Datasets Used in the Book

In this book, the following datasets are used:

1. `Wind Time Series Dataset`. This dataset comes from a single turbine on an inland wind farm. The dataset covers the duration of one year, but data at some of the time instances are missing. Two time resolutions are included in the dataset: the 10-min data and the hourly data; the latter is the further average of the former. For each temporal resolution, the data is arranged in three columns. The first column is the time stamp, the second column is the wind speed, and the third column is the wind power.

2. `Wind Spatial Dataset`. This dataset comes from ten turbines in an offshore wind farm. Only the hourly wind speed data are included. The duration of the data covers two months. The longitudinal and latitudinal coordinates of each turbine are given, but those coordinates are shifted by an arbitrary constant, so that the actual locations of these turbines are protected. The relative positions of the turbines, however, remain truthful to the physical layout. The data is arranged in the following fashion. Under the header row, the next two rows are the coordinates of each turbine. The third row under the header is purposely left blank. From the fourth row onwards are the wind speed data. The first column is the time stamp. Columns 2-11 are the wind speed values measured in meters per second.

3. `Wind Spatio-Temporal Dataset1`. This dataset comprises the average and standard deviation of wind speed, collected from 120 turbines in an inland wind farm, for the years of 2009 and 2010. Missing data in the original dataset are imputed by using the iterative singular value decomposition [139]. Two data files are associated with each year—one contains the hourly average wind speed, used in Eq. 3.18, and the other contains the hourly standard deviation of wind speed, used in Eq. 3.25. The naming convention makes it clear which year a file is associated with and whether it is for the average speed (`Ave`) or for the standard deviation (`Stdev`). The data arrangement in these four files is as follows—the columns are the 120 turbines and the rows are times, starting from 12 a.m. on January 1 of a respective year as the first data row, followed by the subsequent hours in that year. The fifth file in this dataset contains the coordinates of the 120 turbines. To protect the wind farm's identity, the coordinates have been transformed by an undisclosed mapping, so that their absolute values are no longer meaningful but the turbine-to-turbine relative distances are maintained.

4. `Wind Spatio-Temporal Dataset2`. The data used in this study consists of one year of spatio-temporal measurements at 200 randomly selected turbines on a flat terrain inland wind farm, between 2010 and 2011. The data consists of turbine-specific hourly wind speeds measured by the anenometers mounted on each turbine. In addition, one year of hourly wind speed and direction measurements are available at three met masts on the same wind farm. Columns `B` through `OK` are the wind speed and wind power associated with each turbine, followed by Columns `OL` through `OQ`, which are for wind speed and wind direction associated with each mast. The coordinates of the turbines and masts are listed in the top rows, preceding the wind speed, direction, and power data. The coordinates are shifted by a constant, so that while the relative positions of the turbines and the met masts remain faithful to the actual layout, their true geographic information is kept confidential. This anemometer

network provides a coverage of a spatial resolution of one mile and a temporal resolution of one hour.

5. `Inland Wind Farm Dataset1` and `Offshore Wind Farm Dataset1`. Data included in these two datasets are generated from six wind turbines and three met masts and are arranged in six files, each of which is associated with a turbine. The six turbines are named WT1 through WT6, respectively. The layout of the turbines and the met masts is shown in Fig. 5.6. On the offshore wind farm, all seven environmental variables as mentioned above are available, namely $\boldsymbol{x} = (V, D, \rho, H, I, S_a, S_b)$, whereas on the inland wind farm, the humidity measurements are not available, nor is the above-hub wind shear, meaning that $\boldsymbol{x} = (V, D, \rho, I, S_b)$. Variables in $\boldsymbol{x}$ were measured by sensors on the met mast, whereas y was measured at the wind turbines. Each met mast has two wind turbines associated with it, meaning that the $\boldsymbol{x}$'s measured at a met mast are paired with the y's of two associated turbines. For WT1 and WT2, the data were collected from July 30, 2010 through July 31, 2011 and for WT3 and WT4, the data were collected from April 29, 2010 through April 30, 2011. For WT5 and WT6, the data were collected from January 1, 2009 through December 31, 2009.

6. `Inland Wind Farm Dataset2` and `Offshore Wind Farm Dataset2`. The wind turbine data in these two datasets include observations during the first four years of the turbines' operations. The inland turbine data are from 2008 to 2011, whereas the offshore data are from 2007 to 2010. The measurements for the inland wind farm include the same $\boldsymbol{x}$'s as in the `Inland Wind Farm Dataset1` and those for the offshore wind farm include the same $\boldsymbol{x}$'s as in the `Offshore Wind Farm Dataset1`. Most of the environmental measurements $\boldsymbol{x}$ are taken from the met mast closest to the turbine, with the exception of wind speed and turbulence intensity which are measured on the wind turbine. The mast measurements are used either because some variables are only measured at the mast (such as air pressure and ambient temperature, which are used to calculate air density) or because the mast measurements are considered more reliable (such as wind direction).

7. `Turbine Upgrade Dataset`. This dataset includes two sets, corresponding, respectively, to an actual vortex generator installation and an artificial pitch angle adjustment. Two pairs of wind turbines from the same inland wind farm, as used in Chapter 5, are chosen to provide the data, each pair consisting of two wind turbines, together with a nearby met mast. The turbine that undergoes an upgrade in a pair is referred to as the *experimental turbine*, the *reference turbine*, or the *test turbine*, whereas the one that does not have the upgrade is referred to as the *control turbine*. In both pairs, the test turbine and the control turbine

are practically identical and were put into service at the same time. This wind farm is on a reasonably flat terrain.

The power output, y, is measured on individual turbines, whereas the environmental variables in $\boldsymbol{x}$ (i.e., the weather covariates) are measured by sensors at the nearby mast. For this dataset, there are five variables in $\boldsymbol{x}$ and they are the same as those in the `Inland Wind Farm Dataset1`. For the vortex generator installation pair, there are 14 months' worth of data in the period before the upgrade and around eight weeks of data after the upgrade. For the pitch angle adjustment pair, there are about eight months of data before the upgrade and eight and a half weeks after the upgrade.

Note that the pitch angle adjustment is not physically carried out, but rather simulated on the respective test turbine. The following data modification is done to the test turbine data. The actual test turbine data, including both power production data and environmental measurements, are taken from the actual turbine pair operation. Then, the power production from the designated test turbine on the range of wind speed over 9 m/s is increased by 5%, namely multiplied by a factor of 1.05, while all other variables are kept the same. No data modification of any kind is done to the data affiliated with the control turbine in the pitch angle adjustment pair.

The third column of a respective dataset is the `upgrade status` variable, of which a zero means the test turbine is not modified yet, while a one means that the test turbine is modified. The `upgrade status` has no impact on the control turbine, as the control turbine remains unmodified throughout. The vortex generator installation takes effect on June 20, 2011, and the pitch angle adjustment takes effect on April 25, 2011.

8. `Wake Effect Dataset`. This dataset includes data from six pairs of wind turbines (or, 12 wind turbines in total) and three met masts. The turbine pairs are chosen such that no other turbines except the pair are located within 10 times the turbine's rotor diameter. Such arrangement is to find a pair of turbines that are free of other turbines' wake, so that the wake analysis result can be reasonably attributed to the wake of its pair turbine. The operational data for the six pairs of turbines are taken during roughly a yearlong period between 2010 and 2011. The datasets include wind power output, wind speed, wind direction, air pressure, and temperature, of which air pressure and temperature data are used to calculate air density. The wind power outputs and wind speeds are measured on the turbine, and all other variables are measured at the met masts. The data from Mast 1 are associated with the data for Turbine Pairs 1 and 2, Mast 2 with Pairs 3 and 4, and Mast 3 with Pairs 5 and 6. Fig. 8.6 shows the relative locations of the six pairs of turbines and three met masts.

9. `Turbine Bending Moment Dataset`. This dataset includes two parts. The first part is three sets of physically measured blade-root flapwise bending moments on three respective turbines, courtesy of Risø-DTU (Technical University of Denmark) [180]. The basic characteristics of the three turbines can be found in Table 10.1. These datasets include three columns. The first column is the 10-min average wind speed, the second column is the standard deviation of wind speed within a 10-min block, and the third column is the maximum bending moment, in the unit of MN-m, recorded in a 10-min block. The second part of the dataset is the simulated load data used in Section 10.6.5. This part has two sets. The first set is the training data that has 1,000 observations and is used to fit an extreme load model. The second set is the test data that consists of 100 subsets, each of which has 100,000 observations. In other words, the second dataset for testing has a total of 10,000,000 observations, which are used to verify the extreme load extrapolation made by a respective model. Both simulated datasets have two columns: the first is the 10-min average wind speed and the second is the maximum bending moment in the corresponding 10-min block. While all other datasets are saved in CSV file format, this simulated test dataset is saved in a text file format, due to its large size. The data simulation procedure is explained in Section 10.6.5.

10. `Simulated Bending Moment Dataset`. This dataset includes two sets. One set has 600 data records, corresponding to the training set referred to in Section 11.4.1, whereas the other set has 10,000 data records, which are used to produce Fig. 11.1. Each set has three columns of data (other than the serial number). The first column is the wind speed, simulated using a Rayleigh distribution, and the second and third columns are, respectively, the simulated flapwise and edgewise bending moments, in the unit of kN-m. The flapwise and edgewise bending moments are simulated from TurbSim [112] and FAST [113], following the procedure discussed in [149]. TurbSim and FAST are simulators developed at the National Renewable Energy Laboratory (NREL) of the United States.

GLOSSARY

CSV: Comma-separated values Excel file format

DTU: Technical University of Denmark

NREL: National Renewable Energy Laboratory

PTC: Production tax credit

SCADA: Supervisory control and data acquisition

STEM: Science, technology, engineering, and mathematics

US: United States of America

FURTHER READING

C. F. J. Wu. Statistics = Data Science? *Presentation at the H. C. Carver Professorship Lecture*, 1997. `https://www2.isye.gatech.edu/~jeffwu/presentations/datascience.pdf`

D. Donoho. 50 Years of Data Science. *Journal of Computational and Graphical Statistics*, 26: 745–766, 2017.

I

Wind Field Analysis

Taylor & Francis
Taylor & Francis Group
http://taylorandfrancis.com

CHAPTER 2

A Single Time Series Model

Part I of this book is to model $f_t(\boldsymbol{x})$. The focus is on wind speed, V, because wind speed is much more volatile and difficult to predict than other environmental variables such as air density or humidity. In light of this thought, $f_t(\boldsymbol{x})$ is simplified to $f_t(V)$.

A principal purpose of modeling $f_t(V)$ is to forecast wind speed or wind power. Because it is impossible to control wind, forecasting becomes an essential tool in turbine control and wind power production planning. Modeling the time-varying probability density function $f_t(V)$ directly, however, is difficult. In practice, what is typically done is to make a point forecast first and then assess the forecasting uncertainty, which is to attach a confidence interval to the point forecast. The point forecast is a single value used to represent the likely wind speed or power at a future time, corresponding, ideally but not necessarily, to the mean, median, or mode of the probability distribution of wind speed or power at that future time.

The forecasting can be performed either on wind speed or on wind power. Wind power forecasting can be done by forecasting wind speed first and then converting a speed forecast to a power forecast through the use of a simple power curve, as explained in Chapter 1, or the use of a more advanced power curve model, to be explained in Chapter 5. Wind power forecasting can also be done based purely on the historical observations of power output, without necessarily accounting for wind speed information. In the latter approach, the methods developed to forecast wind speed can be used, almost without any changes, to forecast wind power, so long as the wind speed data are replaced with the wind power data. For this reason, while our discussion in this chapter mainly refers to wind speed, please bear in mind its direct applicability to wind power forecast.

In Chapter 2, we consider models that ignore the spatial information and are purely based on the time series data. In Chapters 3 and 4, we discuss various types of spatial or spatio-temporal models.

2.1 TIME SCALE IN SHORT-TERM FORECASTING

One essential question in forecasting is concerning the time-scale requirements of forecast horizons. Turbine control typically requires an instantaneous response in seconds or sub-seconds. Production planning for grid integration and market response is in a longer time scale. Two energy markets, the real-time market and the day-ahead market, demand different response times. The real-time market updates every five minutes, requiring a response in the level of minutes, whereas the day-ahead market is for trading on the next day, requiring forecasting up to 24 hours ahead. Between these two time scales, there are other planning actions that may request a forecast from a few minutes to a few hours ahead. For instance, when the wind power supply is insufficient to meet the demand, the system operators would bring up reserve powers. The spinning reserve, which has been synchronized to the grid system, can be ready for dispatch within 10 minutes, whereas the delivery of contingency reserves may encounter a delay, up to an hour or more, thereby needing a longer lead time for notification. For various planning and scheduling purposes, a common practice for wind owners/operators is to create forecasts, for every hour looking ahead up to 24 hours, and then update that hourly ahead forecast at the next hour for the subsequent 24 hours, using the new set of data collected in between.

When it comes to wind forecasting, there are two major schools of thought. One is the physical model-based approach, collectively known as the Numerical Weather Prediction (NWP) [138], which is the same scientific method used behind our daily weather forecast, and the second is the data-driven, statistical modeling-based approach. By calling the second approach "data-driven," we do not want to leave readers with the impression that NWP is data free; both approaches use weather measurement data. The difference between the two approaches is that NWP involves physical atmospheric models, while the pure data-driven models do not.

Because NWP is based on physical models, it has, on the one hand, the capability to forecast into a relatively longer time horizon, from a few hours ahead to several days ahead. On the other hand, the intensive computation required to solve the complicated weather models limits the temporal and spatial resolutions for NWP, making analysts tend to believe that for a short-term forecast on a local wind field, the data-driven models are advantageous. There is, however, no precise definition of how short is a "short term." Giebel et al. [71] deem six hours as the partition, shorter than which, the data-driven models perform better, while longer than that, NWP ought to be used. Analysts do sometimes push the boundary of data-driven models and make forecasting over a longer horizon, but still, the horizon is generally shorter than 12 hours.

In this book, our interest is to make short-term forecasting on local wind fields. We follow the same limits for short-term as established in the literature, which is usually a few hours ahead and no more than 12 hours ahead.

2.2 SIMPLE FORECASTING MODELS

We first consider the situation that the historical wind data is arranged in a single time series, from time 1 to time t, denoted by $V_i, i = 1, \ldots, t$. The single time series is appropriate to describe the following application scenarios:

- The wind speed or power data measured on a single turbine is used to forecast future wind speed or power on the same turbine.
- The wind speed on a single met tower is used to forecast wind speed, and used as the representation of wind speed for a wind farm.
- The aggregated wind power of a wind farm, namely the summation of wind power output of all individual turbines on the farm, is used to forecast the future aggregated power output of the wind farm.
- Although wind speed is measured at multiple locations, the average wind speed over the locations is used to forecast the future average wind speed.

2.2.1 Forecasting Based on Persistence Model

The simplest point forecasting is based on the *persistence* (PER) model, which says the wind speed or power at any future time, $t+h, h > 0$, is simply the same as what is observed at the current time, t, namely,

$$\hat{V}_{t+h} = V_t, \quad h > 0, \tag{2.1}$$

where the hat notation ($\hat{}$) is used to indicate a forecast (or an estimate). The persistence forecast should, and can easily, be updated when a new observation of V arrives at the next time point.

When the persistence model is used, there is no uncertainty quantification procedure directly associated with it. In order to associate a confidence interval, one needs to establish a probability distribution for wind speed.

2.2.2 Weibull Distribution

Wind speeds are nonnegative and their distribution is right skewed. They do not strictly follow a normal distribution. Understandably, probability densities that are right skewed with nonnegative domain, such as Weibull, truncated normal, or Rayleigh distributions, are common choices for modeling wind speed; for a comprehensive list of distributions, please refer to a survey paper on this topic [32].

There is no consensus on which distribution best describes the data of wind speed, although Weibull distribution is arguably the most popular one. Analysts can try a few of the widely used distributions and test which one fits the data the best. This practice entails addressing two statistical problems—one is to estimate the parameters in the chosen distribution and the other is to assess the goodness-of-fit of the chosen distribution and see if the chosen distribution provides a satisfactory fit to the data.

Consider the Weibull distribution as an example. Its probability density function (pdf) is expressed as

$$f(x) = \begin{cases} \left(\frac{\beta}{\eta}\right)\left(\frac{x}{\eta}\right)^{\beta-1} \exp\left\{-\left(\frac{x}{\eta}\right)^{\beta}\right\} & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{2.2}$$

where $\beta > 0$ is the shape parameter, affecting the skewness of the distribution, and $\eta > 0$ is the scale parameter, affecting the concentration of the distribution. When $\beta \leq 1$, the Weibull density is a decaying function, monotonically going downwards from the origin. When $\beta > 1$, the Weibull density first rises up, passes a peak and then goes down. For commercial wind farms, it makes no practical sense to expect its wind speed to follow a Weibull distribution of $\beta \leq 1$, as what it suggests is that most frequent winds are all low-speed winds. If a wind farm planner does a reasonable job in selecting the farm's location, it is expected to see $\beta > 1$.

The probability density function in Eq. 2.2 is known as the two-parameter Weibull distribution, whose density curve starts at the origin on the x-axis. A more general version, the three-parameter Weibull distribution, is to replace x by $x - \nu$ in Eq. 2.2, where ν is the location parameter, deciding the starting point of the density function on the x-axis. When $\nu = 0$, the three-parameter Weibull density simplifies to the two-parameter Weibull density. The two-parameter Weibull is the default choice, unless one finds that there is an empty gap in the low wind speed measurements close to the origin.

2.2.3 Estimation of Parameters in Weibull Distribution

To estimate the parameters in the Weibull distribution, a popular method is the maximum likelihood estimation (MLE). Given a set of n wind speed measurements, $V_i, i = 1, \ldots, n$, the log-likelihood function, $\mathcal{L}(\beta, \eta | V)$, can be expressed as:

$$\mathcal{L}(\beta, \eta | V) = n \ln \beta - \beta n \ln \eta + (\beta - 1) \sum_{i=1}^{n} \ln V_i - \sum_{i=1}^{n} \left(\frac{V_i}{\eta} \right)^{\beta}. \tag{2.3}$$

Maximizing the log-likelihood function can be done by using an optimization solver in a commercial software, such as `nlm` in `R`. Because `nlm` is for minimization, one should multiply a (-1) to the returned values of the above log-likelihood function while using `nlm` or a similar minimization routine in other software packages. With the availability of the `MASS` package in `R`, fitting

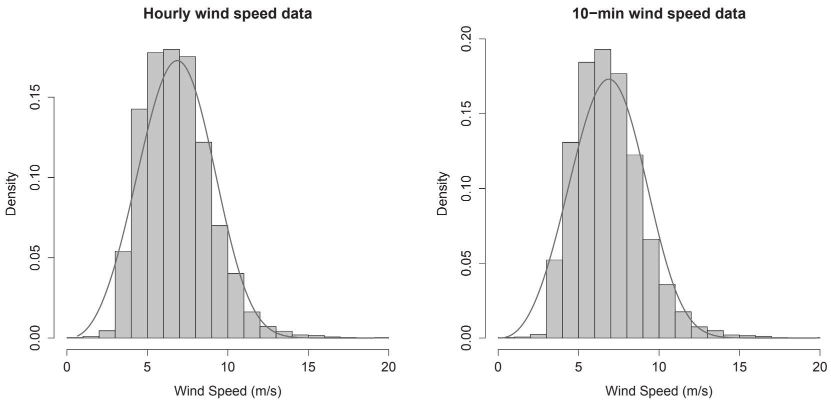

FIGURE 2.1 Fit a Weibell distribution to the wind speed data in the `Wind Time Series Dataset`. The left panel is the fit to the hourly data. The estimated parameters are: $\hat{\eta} = 7.60$, $\hat{\beta} = 3.40$, mean $= 6.84$, median $= 6.69$, mode $= 6.5$, and the standard deviation $= 2.09$. The right panel is the fit to the 10-min data. The estimated parameters are: $\hat{\eta} = 7.61$, $\hat{\beta} = 3.41$, mean $= 6.86$, median $= 6.67$, mode $= 6.5$, and the standard deviation $= 2.06$. The values of mean, median, mode and standard deviation are estimated directly from the data, rather than calculated using $\hat{\eta}$ and $\hat{\beta}$.

a Weibull distribution can be done more directly by using the `fitdistr` function. Suppose that the wind speed data is stored in the vector named `wsdata`. The following `R` command can be used for fitting a Weibull distribution,

```
fitdistr(wsdata, "weibull").
```

Fig. 2.1 presents an example of using a Weibull distribution to fit the wind speed data in the `Wind Time Series Dataset`. The Weibull distribution parameters are estimated by using the MLE. Fig. 2.1 presents the Weibull fit to the wind speed data of two time resolutions: the 10-min data and the hourly data. The estimates of the shape and scale parameters are rather similar despite the difference in time resolution.

2.2.4 Goodness of Fit

Once a Weibull distribution is fit to a set of data, how can we tell whether or not it is a good fit? This question is answered through a goodness-of-fit test, such as the χ^2 test. The idea of the χ^2 test is simple. It first bins the observed data, like in a histogram. For the j-th bin, one can count the number of actual observations falling into that bin; denote this as O_j. Should the data follow a specific type of distribution, the expected amount of data points in the same bin can be computed from the cumulative distribution function (cdf) of that distribution; denote this quantity as E_j. Suppose that we have a total of B bins. Then, the test statistic, defined below, follows a χ^2 distribution with a degree of freedom of $B - p - 1$, i.e.,

$$\chi^2 := \sum_{j=1}^{B} \frac{(O_j - E_j)^2}{E_j} \sim \chi^2_{B-p-1}, \tag{2.4}$$

where p is the number of parameters associated with the distribution.

The Weibull distribution has a closed form cdf. The fitted Weibull distribution function, by plugging in the estimated parameters, $\hat{\beta}$ and $\hat{\eta}$, is

$$F_{\hat{\beta},\hat{\eta}}(x) = 1 - \exp\left\{ - \left(\frac{x}{\hat{\eta}} \right)^{\hat{\beta}} \right\}. \tag{2.5}$$

Of the j-th wind speed bin, the left boundary wind speed value is V_{j-1} and the right boundary value is V_j, so E_j can be calculated by

$$E_j = n[F_{\hat{\beta},\hat{\eta}}(V_j) - F_{\hat{\beta},\hat{\eta}}(V_{j-1})]. \tag{2.6}$$

Once the χ^2 test statistic is calculated, one can compute the p-value of the test by using, for example, the `R` command, `1−pchisq(`χ^2`, ` $B-p-1$`)`. The null hypothesis says that the distribution under test provides a good fit. When the p-value is small enough, say, smaller than 0.05, analysts say that the null hypothesis is rejected at the significance level of 95%, implying that the theoretical distribution is less likely a good fit to the data. When the p-value is not small enough and the null hypothesis cannot be rejected, then the test implies a good fit.

We can apply the χ^2 test to one month of data of the `Wind Time Series Dataset` and the respective fitted Weibull distributions. The number of parameters in the two-parameter Weibull distribution is $p = 2$. While binning the wind speed data, one needs to be careful about some of the tail bins in which the expected data amount could be too few. The general guideline is that E_j should be no fewer than five; otherwise, several bins should be grouped into a single bin.

The test statistic and the corresponding p-values are shown in Table 2.1. As shown in the table, it looks like using the Weibull distribution to fit the wind speed data does not pass the goodness-of-fit test. This is particularly true when the data amount increases, as in the case of using the 10-min data. Nonetheless, the Weibull distribution still stays as one of the most popular distributions for modeling wind speed data. The visual inspection of Fig. 2.1 leaves analysts with the feeling of a reasonable fit. Passing the formal statistical test in the presence of abundant data appears tough. Analysts interested in a distribution alternative can refer to [32] for more choices.

TABLE 2.1 Goodness-of-fit test statistics and p-values.

	Hourly data	**10-min data**
Month selected	February	November
Data amount, n	455	3,192
Bin size	0.2 m/s	0.1 m/s
Number of bins, B	66	100
Test statistic	62.7	329.8
p-value	0.012	almost 0

2.2.5 Forecasting Based on Weibull Distribution

Assuming that the distribution of wind speed stays the same for the next time period, i.e., the underlying process is assumed stationary, analysts can use the mean as a point forecast, and then use the distribution to assess the uncertainty of the point forecast. We want to note that such approach is, in spirit, also a persistence forecasting, but it is conducted in the sense of an unchanging probability distribution.

The mean and the standard deviation of a Weibull distribution, if using the estimated distribution parameters, are

$$\begin{cases} \hat{\mu} &= \hat{\eta}\Gamma(1+\frac{1}{\hat{\beta}}), \\ \hat{\sigma} &= \hat{\eta}\sqrt{\Gamma(1+\frac{2}{\hat{\beta}}) - (\Gamma(1+\frac{1}{\hat{\beta}}))^2}, \end{cases} \tag{2.7}$$

where $\Gamma(\cdot)$ is the gamma function, defined such as $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$.

While the mean $\hat{\mu}$ is used as the point forecast, one can employ a normal approximation to obtain the $100(1-\alpha)\%$ confidence interval of the point forecast, as

$$[\hat{\mu} - z_{\alpha/2}\cdot\hat{\sigma}, \quad \hat{\mu} + z_{\alpha/2}\cdot\hat{\sigma}], \tag{2.8}$$

where z_α is the α-quantile point of a standard normal distribution. When $\alpha = 0.05$, $z_{0.05/2} = 1.96$.

Sometimes analysts think that using the mean may not make a good forecast, due to the skewness in the Weibull distribution. Alternatively, median and mode can be used. Their formulas, still using the estimated parameters, are

$$\begin{cases} \text{median} &= \hat{\eta}(\ln 2)^{1/\hat{\beta}}, \\ \text{mode} &= \hat{\eta}\left(1-\frac{1}{\hat{\beta}}\right)^{1/\hat{\beta}} \quad \text{for } \hat{\beta} > 1. \end{cases} \tag{2.9}$$

The mode of a Weibull distribution when $\beta \leq 1$ is zero. As mentioned earlier, the circumstances under which $\beta \leq 1$ are of little practical relevance in wind speed modeling at commercial wind farms.

Analysts may worry that using the normal approximation to obtain the confidence interval may not be accurate enough. If one has a sufficiently large

TABLE 2.2 Estimate of mean and 95% confidence interval of wind speed data. The total data amount is 7,265 for the hourly data and 39,195 for the 10-min data.

	Based on Eq. 2.8		**Directly from sample statistics**	
	Mean	C.I.	Mean	C.I.
Hourly data	6.83	[2.48, 11.47]	6.84	[3.54, 11.33]
10-min data	6.84	[2.50, 11.18]	6.86	[3.62, 11.49]

amount of wind speed data, say more than 1,000 data points, a simple way is to estimate the mean and its confidence interval directly from the data, following the two steps below.

1. Compute the sample average wind speed, $\bar{V}$,

$$\bar{V} = \frac{1}{n}\sum_{i=1}^{n} V_i,$$

   and use it as the point forecast.

2. Order the wind speed data from the smallest to the largest. Denote the ordered sequence as $V_{(1)}, V_{(2)}, \ldots, V_{(n)}$. Then, the $100(1-\alpha)\%$ confidence interval is estimated to be $[V_{[n\alpha/2]},\ V_{[n(1-\alpha/2)]}]$, where $[\cdot]$ returns the nearest integer number.

Table 2.2 presents the estimates of mean and confidence interval, either based on the Weibull distribution or directly from the data. One observes that the point forecasts are rather close, but the lower confidence intervals are noticeably different.

2.3 DATA TRANSFORMATION AND STANDARDIZATION

Before the wind speed data is fed into time series models, many of which assume Gaussianity, data preprocessing may be needed. Two common preprocessing tasks are: (1) normalizing the wind data, so that the transformed data behaves closer to a normal distribution, and (2) removing the diurnal nonstationarity or other seasonalities from the data.

A general power transformation is used [23] for the purpose of normalization, such as

$$V_t' = V_t^m, \forall i, \tag{2.10}$$

where V_t' is the transformed wind speed, and m is the power coefficient, with the convention that $m = 0$ refers to the logarithm transformation. Apparently, $m = 1$ means no transformation.

Suppose that the wind data indeed follow a Weibull distribution. A nice property of Weibull distribution is that a Weibull random variable remains Weibull when it is raised to a power $m \neq 0$, with its parameters becoming β/m and η^m, respectively. Dubey [52] points out that when the shape parameter is close to 3.6, a Weibull distribution is closer in shape to a normal distribution. The general advice is to estimate the shape parameter from the original wind data and then solve for m in the power transformation in Eq. 2.10 as

$$m = \frac{\hat{\beta}}{3.6}. \tag{2.11}$$

Alternatively, Hinkley [93] suggests checking the following measure of symmetry, based on sample statistics,

$$sym = \frac{\text{sample mean} - \text{sample median}}{\text{sample scale}}, \tag{2.12}$$

where the sample scale can be the sample standard deviation or the sample inter-quartile range; Hinkley himself prefers the latter. Given this symmetry measure, one could first choose a candidate set of m values (including $m = 0$) and apply the respective transformation on the wind data. Then, calculate the corresponding symmetry measure. To approximate the symmetric normal distribution, the symmetry value is desired to be zero. Whichever m produces a zero sym value is thus chosen as the power in the transformation. If no m in the candidate set produces a sym close to zero, then one can interpolate the computed (m, sym) points and find the m leading to a zero sym. One convenience allowed by Eq. 2.12 is that the logarithm transformation can be tested, together with other power transformations, whereas in using Eq. 2.11, $m = 0$ is not allowed.

Torres et al. [214] show that using Eq. 2.11 on wind data from multiple sites for every month in a whole year, the resulting m values are in the range of $[0.39, 0.70]$, but many of them are close to 0.5. Brown et al. [23] apply both aforementioned approaches on one set of wind data—Eq. 2.11 produces an $m = 0.45$, while the sym measure in Eq. 2.12 selects $m = 1/2$, implying a square-root transformation. It seems that the resulting m values are often not too far from 1/2. But this may not always be the case. When applying Eq. 2.11 to the data of each month in the `Wind Time Series Dataset` (see Table 2.3 for the corresponding m values), we find that most m's are around one. This is not surprising. The shape of the density curves in Fig. 2.1 looks rather normal-like, and the corresponding $\hat{\beta}$'s are close to 3.6. For the sake of convenience, analysts still use $m = 1/2$ as the default setting. This square-root transformation is in fact one of the popular normalizing transformations and applying it reduces the right skewness to make the resulting data closer to a normal distribution. When applied to wind data, the square-root transformation can take any wind speed values, since wind speed is supposedly non-negative. In contrast, if one applies the logarithm transformation, the zero wind speed values need to be removed first.

TABLE 2.3 Monthly values of m using Eq. 2.11.

	Jan	**Feb**	**Mar**	**Apr**	**May**	**Jun**
Hourly data	0.74	1.00	1.00	1.04	1.10	1.02
10-min data	0.74	0.94	1.02	1.05	1.10	0.98

	Jul	**Aug**	**Sep**	**Oct**	**Nov**	**Dec**
Hourly data	1.01	1.14	1.06	0.98	1.02	0.99
10-min data	1.01	1.13	1.06	0.99	1.03	1.00

Wind exhibits diurnal and seasonal nonstationarity. The seasonality is typically handled by carefully choosing the training period, making sure that the seasonal pattern of the training period is consistent with that in the forecasting period. This can be done by using the wind data in a short period of time immediately prior to the forecasting period, say, a few days or a couple of weeks, but usually no more than one month. To remove the diurnal nonstationarity, a simple treatment is to standardize the wind data by using its hourly average and standard deviation.

We show how this is done using the transformed wind data, V_t', but obviously the same procedure can be applied to the original wind data. We first arrange the data such that the time index t is in an hourly increment. If the raw data is in the 10-min format, then, one can get the hourly data by averaging the six 10-min wind data points within the same hourly block. For notational convenience, let us deem that $t = 0$ coincides with 12 a.m. (midnight) of the first day, $t = 1$ with one a.m., and so on. The time repeats itself as the same time on a different day in an increment of 24. We compute 24 hourly averages and standard deviations by pooling the data from the same time on different days in the training period. Suppose that there are a total of d days. Then, we can compute them as

$$\begin{cases} \bar{V}_\ell' &= \frac{1}{d}\sum_{j=0}^{d-1} V_{24j+\ell}', \\ s_\ell &= \sqrt{\frac{1}{d-1}\sum_{j=0}^{d-1}(V_{24j+\ell}' - \bar{V}_\ell')^2}. \end{cases} \quad \ell = 0, \ldots, 23. \tag{2.13}$$

The standardization of wind speed data is then carried out by

$$V_t'' = \frac{V_t' - \bar{V}_{(t \texttt{ mod } 24)}'}{s_{(t \texttt{ mod } 24)}}, \tag{2.14}$$

where `mod` means a modulo operation, so that (t `mod` 24) returns the remainder when t is divided by 24.

Fig. 2.2 presents the original hourly wind speed data and the standardized hourly data. Although the standardization is conducted for the whole year hourly data in the `Wind Time Series Dataset`, Fig. 2.2 plots only three

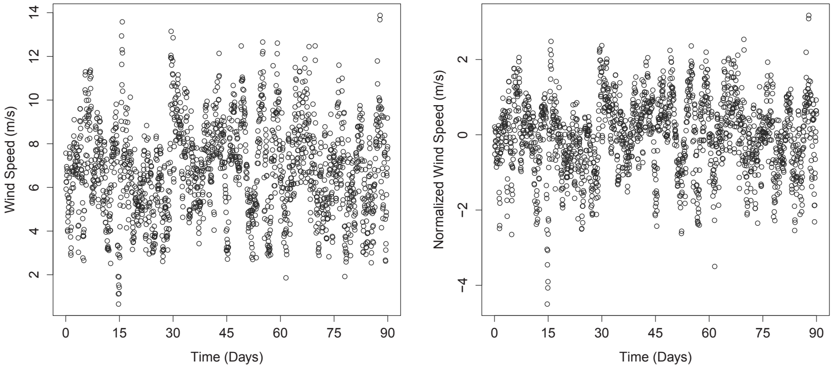

FIGURE 2.2 Left panel: original hourly wind speed data. Right panel: standardized hourly wind speed data. The data amount of the three months is $n = 1,811$. Eq. 2.10, with $m = 1/2$, and Eq. 2.14 are used for standardization.

months (October to December) for a good visualization effect. The difference between the two subplots is not very striking, because the original data, as we explained above, is already close to a normal distribution.

Gneiting et al. [75] introduce a trigonometric function to model the diurnal pattern, as in the following,

$$\Delta_t = c_0 + c_1 \sin\left(\frac{2\pi t}{24}\right) + c_2 \cos\left(\frac{2\pi t}{24}\right) + c_3 \sin\left(\frac{4\pi t}{24}\right) + c_4 \cos\left(\frac{4\pi t}{24}\right), \quad (2.15)$$

where $c_0, c_1, \ldots, c_4$ are the coefficients to be estimated from the data. The estimation is to assume $V'_t = \Delta_t + \varepsilon_t$, and then, use a least squares estimation to estimate the coefficients from the wind data. Subtracting the diurnal pattern from the original wind data produces the standardized wind speed,

$$V''_t = V'_t - \Delta_t. \quad (2.16)$$

2.4 AUTOREGRESSIVE MOVING AVERAGE MODELS

In this section, we apply a time series model like the autoregressive moving average (ARMA) model to the normalized and standardized wind data. For notational simplicity, we return to the original notation of wind speed, V_t, without the primes.

An autoregressive (AR) model of order p is to regress the wind variable on its own past values, up to p steps in the history, such as

$$V_t = a_0 + a_1 V_{t-1} + \ldots + a_p V_{t-p} + \varepsilon_t, \quad (2.17)$$

where $a_i, i = 1, \ldots, p$, are the AR coefficients and ε_t is the residual error, assumed to be a zero mean, identically, independently distributed (i.i.d) noise. Specifically, $\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

The autoregressive mechanism makes intuitive sense, as the inertia in air movement suggests that the wind speed at the present time is related to its immediate past. The actual relationship, however, may not necessarily be linear. The linear structure assumed in the AR model is for the sake of simplicity, making the model readily solvable.

A general ARMA model is to add a moving average (MA) part to the AR model, which is to model the residual as a linear combination of the i.i.d noises, going back in history for up to q steps. Including the MA part, the ARMA model reads

$$\begin{aligned} V_t &= a_0 + a_1 V_{t-1} + \ldots + a_p V_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + \ldots + b_q \varepsilon_{t-q} \\ &= a_0 + \sum_{i=1}^{p} a_i V_{t-i} + \varepsilon_t + \sum_{j=1}^{q} b_j \varepsilon_{t-j}, \end{aligned} \tag{2.18}$$

where $b_j, j = 1, \ldots, q$, are the MA coefficients. The overall model in Eq. 2.18 is referred to as ARMA(p, q), where p is the AR order and q is the MA order.

2.4.1 Parameter Estimation

For the model in Eq. 2.17, the AR parameters can be estimated through a least squares estimation, expressed in a closed form. Suppose that we have the historical data going back n steps. For each step in the past, one can write down an AR model. The following are the n equations,

$$\begin{aligned} V_t &= a_0 + a_1 V_{t-1} + \ldots + a_p V_{t-p} + \varepsilon_t, \\ V_{t-1} &= a_0 + a_1 V_{t-2} + \ldots + a_p V_{t-1-p} + \varepsilon_{t-1}, \\ &\cdots \quad \cdots \quad \cdots \quad \cdots \\ V_{t-n} &= a_0 + a_1 V_{t-n-1} + \ldots + a_p V_{t-n-p} + \varepsilon_{t-n}. \end{aligned} \tag{2.19}$$

Express $\mathbf{V} = (V_t, V_{t-1}, \ldots, V_{t-n})^T_{n\times 1}$, $\mathbf{a} = (a_0, a_1, \ldots, a_p)^T_{(p+1)\times 1}$, $\boldsymbol{\varepsilon} = (\varepsilon_t, \ldots, \varepsilon_{t-n})^T_{n\times 1}$, and

$$\mathbf{W} = \begin{pmatrix} 1 & V_{t-1} & \cdots & V_{t-p} \\ 1 & V_{t-2} & \cdots & V_{t-1-p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & V_{t-n} & \cdots & V_{t-n-p} \end{pmatrix}_{n\times(p+1)}.$$

Then, Eq. 2.19 can be written in a matrix form, such as

$$\mathbf{V} = \mathbf{W} \cdot \mathbf{a} + \boldsymbol{\varepsilon}. \tag{2.20}$$

As such, the least squares estimate of the parameter vector, $\mathbf{a}$, is

$$\hat{\mathbf{a}} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{V}.$$

The fitted wind speed value, $\hat{\mathbf{V}}$, is therefore $\hat{\mathbf{V}} = \mathbf{W}\hat{\mathbf{a}}$. The variance of the residual error term can be estimated by

$$\hat{\sigma}_{\varepsilon}^2 = \frac{(\mathbf{V} - \hat{\mathbf{V}})^T(\mathbf{V} - \hat{\mathbf{V}})}{n-p-1} = \frac{(\mathbf{V} - \mathbf{W}\hat{\mathbf{a}})^T(\mathbf{V} - \mathbf{W}\hat{\mathbf{a}})}{n-p-1}. \tag{2.21}$$

With the MA part included in a general ARMA(p, q) model, the least squares estimation of both AR and MA coefficients does not have a closed form expression anymore. The estimation problem needs to be solved iteratively through a numerical procedure. Analysts use the maximum likelihood estimation method to estimate the parameters. Denote by $\mathbf{b} = (b_1, b_2, \ldots, b_q)_{q\times 1}^T$. The log-likelihood function of an ARMA model, denoted as $\mathcal{L}(\mathbf{a}, \mathbf{b}, \sigma_{\varepsilon}^2|\mathbf{V})$, is a bit involved. We choose not to write down its expression here. In practice, it is advised to use the `arima` function in `R`'s `stats` package to carry out the estimation task. The `arima` function is named after the autoregressive integrated moving average model, considered as a generalization of the ARMA model and expressed as ARIMA(p, k, q), which has one extra parameter than an ARMA model has. To handle an ARMA(p, q) model using the three-parameter `arima` function, one can simply set $k = 0$. By default, the `arima` uses the maximum likelihood method for parameter estimation.

To use the `arima` function, one needs to specify p and q. For instance, the command,

```
fit<-arima(wsdata, order = c(3,0,1)),
```

fits an ARMA(3,1) model. Typing `fit` in the `R` program displays the values of $\hat{a}_0$, $\hat{a}_1$, $\hat{a}_2$, $\hat{a}_3$, $\hat{b}_1$, the standard deviations of the respective estimates, as well as $\hat{\sigma}_{\varepsilon}^2$. It also displays a few other things, such as the log-likelihood value and AIC, which we explain next.

2.4.2 Decide Model Order

When using the `arima` function, the model orders p and q need to be specified. In the `forecast` package, there is an `auto.arima` function, which can decide the model order on its own. If one is curious about how `auto.arima` selects its model order or wants to have more control on model selection by oneself, this section explains the thought process.

Popular model selection criteria used for time series models include the Akaike Information Criterion (AIC) [7] and Bayesian Information Criterion (BIC) [197]. Both criteria follow the same philosophy, which is to trade off between a model's training error and its complexity, in order to select a simple enough model that in the meanwhile fits well enough to the training data. The difference between AIC and BIC is in the specific weighting used to trade off the two objectives, which is going to be clear below.

The AIC is defined as

$$\text{AIC} = 2 \times \text{number of parameters} - 2\hat{\mathcal{L}}, \tag{2.22}$$

where $\hat{\mathcal{L}}$ is the log-likelihood value of the ARMA model, evaluated at the estimated parameters. The log-likelihood value is one of the outputs from the `arima` function. The number of parameters in an ARMA(p, q) model is $p + q + 1$. Hence, AIC $= 2(p + q + 1) - 2\hat{\mathcal{L}}$ for an ARMA(p, q) model.

The BIC is defined as

$$\begin{aligned} \text{BIC} &= \ln(n) \times \text{number of parameters} - 2\hat{\mathcal{L}} \\ &= \ln(n) \cdot (p + q + 1) - 2\hat{\mathcal{L}}. \end{aligned} \tag{2.23}$$

Using AIC or BIC, one would select the model that minimizes either of the criteria.

The log-likelihood value indicates how well an ARMA model fits the training data—the greater, the better. Because the data are noisy, a model that fits too well to the training data could have read too much into the noise part, a problem known as *overfitting* [86]. An overfit model loses its predictive ability and has actually a worse forecasting accuracy. Analysts come to realize that an effective way to avoid overfitting is to select a simpler model. The number of parameters in an ARMA model measures its model complexity—the fewer the parameters, the simpler a model is.

AIC deems that one unit increase in the model complexity, namely one more parameter included in the model, is equivalent to one unit decrease in the log-likelihood. In using AIC, this trade-off is independent of the data amount, n. BIC, instead, considers the weighting coefficient to be dependent on the data amount. Specifically, it uses $\ln(n)$ to quantify the model complexity. When $n = 7.4$, meaning the training data points are seven or eight, $\ln(n) = 2$, making AIC and BIC equivalent. When $n \geq 8$, BIC tends to choose a simpler model than AIC. In practical situations, n is much greater than eight, suggesting that BIC yields a simpler ARMA model that tends to forecast more accurately on future data.

Aware of the shortcoming of the original AIC, analysts propose a corrected AIC [34], referred to as AICc and defined in the context of ARMA(p, q) as

$$\text{AICc} = \text{AIC} + 2 \times \frac{(p + q + 1)^2 + (p + q + 1)}{n - p - q}. \tag{2.24}$$

AICc is virtually AIC with an extra penalty term for model complexity. When n is far greater than the square of the number of parameters in a model, AIC and AICc behave almost the same.

The `arima` function returns the values of AIC. One can use the `BIC` function to compute the BIC value, and use the formula in Eq. 2.24 to calculate AICc. When using `auto.arima`, one can set its argument `ic` to be either `aicc`, `aic`, or `bic`, so that the respective information criterion is used in selecting p and q in the model. For instance,

```
fit<-auto.arima(wsdata, ic=c('bic'))
```

uses the BIC for model selection. The default setting in `auto.arima` is AICc.

TABLE 2.4 The log-likelihood, BIC, AIC, and AICc values of 18 candidate models, up to ARMA(6, 3), based on the hourly data of April in the `Wind Time Series Dataset`, where $n = 433$. Boldface values are either the largest log-likelihood or the smallest values of a respective information criterion.

Model	**Log-likelihood**	**BIC**	**AIC**	**AICc**
ARMA (1 , 1)	−293.7	**605.5**	593.3	593.4
ARMA (1 , 2)	−293.0	610.4	594.1	594.2
ARMA (1 , 3)	−292.9	616.1	595.8	595.9
ARMA (2 , 1)	−293.3	610.9	594.7	594.7
ARMA (2 , 2)	−292.7	615.8	595.4	595.6
ARMA (2 , 3)	−292.8	622.0	597.6	597.8
ARMA (3 , 1)	−292.7	615.8	595.4	595.6
ARMA (3 , 2)	−290.5	617.3	**593.0**	**593.2**
ARMA (3 , 3)	−289.7	622.0	593.5	593.8
ARMA (4 , 1)	−293.0	622.3	597.9	598.1
ARMA (4 , 2)	−289.8	622.0	593.5	593.8
ARMA (4 , 3)	−289.7	628.0	595.5	595.8
ARMA (5 , 1)	−293.0	628.4	599.9	600.2
ARMA (5 , 2)	−289.7	627.9	595.4	595.7
ARMA (5 , 3)	−289.1	632.9	596.2	596.7
ARMA (6 , 1)	−290.7	630.1	597.5	597.8
ARMA (6 , 2)	−289.0	632.7	596.1	596.5
ARMA (6 , 3)	**−288.6**	638.0	597.3	597.8

We want to note that certain software packages, like these in `R`, count the variance estimate, $\hat{\sigma}^2_\varepsilon$, as a parameter estimated. Hence, the number of parameters in an ARMA(p, q) model becomes $p + q + 2$. Using this parameter number does change the AIC and BIC values but they do not change the model selection outcome, as all AIC's or BIC's are basically offset by a constant. When this new number of parameters is used with AICc, however, it could end up choosing a different model.

When applying to one month (April) of hourly data in the `Wind Time Series Dataset`, the BIC produces the simplest ARMA model, which is ARMA(1,1), namely $p = q = 1$. Had AIC or AICc been used on the same set of data, ARMA(3,2) would have been chosen, which is more complicated than ARMA(1,1). For the detailed information, please refer to Table 2.4. The estimated parameters for this ARMA(1,1) model are: $\hat{a}_0 = 0.0727$, $\hat{a}_1 = 0.8496$, $\hat{b}_1 = 0.0871$, and $\hat{\sigma}^2_\varepsilon = 0.2265$.

2.4.3 Model Diagnostics

In addition to using the information criteria, described above, to choose an appropriate time series model, analysts are encouraged to use graphical plots to check the model's fitting quality—this is referred to as model diagnostics or diagnostic checking. For ARMA models, the two most commonly used plots are the autocorrelation function (ACF) plot and the partial autocorrelation function (PACF) plot.

The model diagnostics is performed on the residuals after a model is fitted. The purpose is to check whether the model assumptions regarding the error term hold. The plots are supposed to show that the residuals, after the model part is removed from the data, appear random and contain no systematic patterns; otherwise, it suggests the model fitting is not properly done. Some diagnostics also tests if the residual follows a normal distribution.

Based on Eq. 2.18, we can compute the residuals recursively, using the estimated parameters, such as

$$\begin{aligned} \hat{\varepsilon}_t =& V_t - \hat{a}_0 - \sum_{i=1}^{p} \hat{a}_i V_{t-i} - \sum_{j=1}^{q} \hat{b}_j \hat{\varepsilon}_{t-j}, \quad t = 1, \ldots, n, \\ & V_\ell = 0, \ \hat{\varepsilon}_\ell = 0, \quad \forall \ell \leq 0. \end{aligned} \tag{2.25}$$

The autocorrelation function of ε_t is just the correlation function of the random variable with its own past. Denote by $Cov(X, Y)$ the covariance of two random variables, X and Y. Then, the autocovariance function between two time points, t and $t-h$, in the stochastic process of ε_t, is denoted as $Cov(\varepsilon_t, \varepsilon_{t-h})$. When $h = 0$, $Cov(\varepsilon_t, \varepsilon_t) = \sigma_\varepsilon^2$ is the variance of the underlying process. Define by $\rho(X, Y)$ the correlation between two random variables, X and Y. Then, the autocorrelation function of ε_t is

$$\rho(\varepsilon_t, \varepsilon_{t-h}) = \frac{Cov(\varepsilon_t, \varepsilon_{t-h})}{Cov(\varepsilon_t, \varepsilon_t)} = \frac{Cov(\varepsilon_t, \varepsilon_{t-h})}{\sigma_\varepsilon^2}.$$

Considering that the residuals should be stationary (after all these modeling steps), then the autocorrelation function does not depend on the starting point in time but only on the time lag h. As such, its notation can be simplified as ρ_h. With the residuals computed in Eq. 2.25, the sample autocorrelation can be computed through

$$\hat{\rho}_h = \frac{\sum_{t=h+1}^{n} (\hat{\varepsilon}_t - \bar{\hat{\varepsilon}})(\hat{\varepsilon}_{t-h} - \bar{\hat{\varepsilon}})}{\sum_{t=1}^{n} (\hat{\varepsilon}_t - \bar{\hat{\varepsilon}})^2} \approx \frac{\sum_{t=h+1}^{n} \hat{\varepsilon}_t \hat{\varepsilon}_{t-h}}{\sum_{t=1}^{n} \hat{\varepsilon}_t^2}, \tag{2.26}$$

where $\bar{\hat{\varepsilon}}$ is the sample mean of the residuals, which is supposed to be zero (or near zero), so that they can be omitted from the equation. Applying Bartlett's formula [20, Eq. 6.2.2], the standard error (se) for testing the significance of $\hat{\rho}_h$ is approximated by

$$\text{se}_\rho = \sqrt{\frac{1 + 2\sum_{i=1}^{h-1} \hat{\rho}_i^2}{n}}.$$

The 95% confidence interval for $\hat{\rho}_h$ is approximated by $\pm 1.96 \cdot \text{se}_\rho$. Under the null hypothesis that the residuals are uncorrelated, meaning $\hat{\rho}_h = 0, \forall h > 0$, the standard error is then simplified to $\mathrm{se}_\rho = \sqrt{1/n}$, and correspondingly, the 95% confidence interval becomes simply $\pm 1.96/\sqrt{n}$.

One could plot $\hat{\rho}_h$ against a series of time lags, h, and observe how much, if any at all, the residuals are still correlated with their own past. This can be done by using the `R` function `acf` in the `forecast` package. The default setting in `acf` draws an autocorrelation plot, on which there are two dashed lines (blue in color print). These lines correspond to the 95% confidence interval under the null hypothesis, which are at the values of $\pm 1.96/\sqrt{n}$, as explained above. With an autocorrelation plot, analysts can quickly inspect if there is any $\hat{\rho}_h$ exceeding the line of $\pm 1.96/\sqrt{n}$, and if yes, that suggests still strong enough autocorrelation.

The autocorrelation between ε_t and ε_{t-2} presumably comes from two sources—one is a lag-1 propagation via the correlation between ε_t and ε_{t-1} and then the correlation between ε_{t-1} and ε_{t-2}, while the other is the correlation directly between ε_t and ε_{t-2}. The autocorrelation, ρ_2, as defined and computed above, is the summation of the two sources. When one sees a large ρ_2, one may wonder if its large value is caused by a large lag-1 autocorrelation and its propagation or if it is caused by the direct correlation. The concept of partial autocorrelation is therefore introduced to quantify this direct correlation, which is the amount of correlation between a variable and a lag of itself that is not explained by correlations at all lower-order lags.

Consider the AR model of order p in Eq. 2.17. Applying the correlation operation with V_{t-1} on each term in both sides gives us the following equation, where we replace the coefficient, a_i, in Eq. 2.17 by ϕ_{pi}, such as

$$\rho_1 = \phi_{p1} + \phi_{p2}\rho_1 + \ldots + \phi_{pp}\rho_{p-1}. \tag{2.27}$$

In the above equation, we replace ρ_0 by its value, which is one. Here we use a double index subscript on ϕ to signify that this set of coefficients is obtained when we use an AR model of order p. Do the correlation operation with V_{t-j}, for $j = 1, \ldots, p$. We end up with the set of Yule-Walker equations [20] as,

$$\underbrace{\begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_p \end{pmatrix}}_{\boldsymbol{\rho}} = \underbrace{\begin{pmatrix} 1 & \rho_1 & \cdots & \rho_{p-1} \\ \rho_1 & 1 & \cdots & \rho_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p-1} & \rho_{p-2} & \cdots & 1 \end{pmatrix}}_{\mathbf{R}} \underbrace{\begin{pmatrix} \phi_{p1} \\ \phi_{p2} \\ \vdots \\ \phi_{pp} \end{pmatrix}}_{\boldsymbol{\phi}}, \tag{2.28}$$

or in the matrix format,

$$\boldsymbol{\rho} = \mathbf{R}\boldsymbol{\phi}.$$

Because $\mathbf{R}$ is a full-rank and symmetric matrix, we can solve for $\boldsymbol{\phi}$ as

$$\hat{\boldsymbol{\phi}} = \mathbf{R}^{-1}\boldsymbol{\rho}.$$

The partial autocorrelation function is estimated by the sequence of

$\hat{\phi}_{11}, \hat{\phi}_{22}, \ldots$, which can be obtained by solving the Yule-Walker equations for $p = 1, 2, \ldots$. Here it becomes apparent why we replace the single index coefficient, a_i, in the AR model with the double index coefficient, ϕ_{pi}, in the Yule-Walker equations; it is otherwise difficult to express the partial autocorrelation function.

The Yule-Walker equations can be solved recursively using the Levinson-Durbin formula [54],

$$\begin{aligned}
\hat{\phi}_{pp} &= \frac{\hat{\rho}_p - \sum_{j=1}^{p-1} \hat{\phi}_{(p-1)j}\hat{\rho}_{p-j}}{1 - \sum_{j=1}^{p-1} \hat{\phi}_{(p-1)j}\hat{\rho}_j}, \quad p = 2, 3, \ldots \\
\hat{\phi}_{pj} &= \hat{\phi}_{(p-1)j} - \hat{\phi}_{pp}\hat{\phi}_{(p-1)(p-j)}, \\
\hat{\phi}_{11} &= \hat{\rho}_1.
\end{aligned} \tag{2.29}$$

Using the above equations, we figure out that the partial autocorrelation of lag 2, $\hat{\phi}_{22}$, is

$$\hat{\phi}_{22} = \frac{\hat{\rho}_2 - \hat{\rho}_1^2}{1 - \hat{\rho}_1^2}. \tag{2.30}$$

Recall the example mentioned earlier about the autocorrelation between ε_t and ε_{t-2}. The two-step sequential propagation of the lag-1 autocorrelation is $\hat{\rho}_1^2$, whereas $\hat{\rho}_2$ is the full lag-2 autocorrelation. If $\hat{\rho}_2 = \hat{\rho}_1^2$, the correlation between ε_t and ε_{t-2} that is not explained by correlations at the lower lag-1 order is zero. As such, it is reflected in the partial autocorrelation function as $\hat{\phi}_{22} = 0$, according to Eq. 2.30. If $\hat{\rho}_2 \neq \hat{\rho}_1^2$, their difference, scaled by $1 - \hat{\rho}_1^2$, is the partial autocorrelation of lag 2, or the direct correlation between ε_t and ε_{t-2}.

The partial autocorrelation function is useful in identifying the model order of an autoregressive process. If the original process is autoregressive of order k, then for $p > k$, we should have $\phi_{pp} = 0$. This can again be done in a partial autocorrelation function plot by inspecting, up to which order, PACF becomes zero or near zero. By setting `type=c('partial')` in one of its arguments, the `acf` function computes PACF values and draws a PACF plot. Alternatively, the `pacf` function in the `tseries` package can do the same. The dashed line on a PACF plot bears the same value as the same line on an ACF plot.

Fig. 2.3 presents the ACF and PACF plots using the April data in the `Wind Time Series Dataset`. The ARMA(1,1) model is fit to the set of data, and the model residuals are thus computed. The ACF and PACF plots of the residuals are presented in Fig. 2.3 as well.

2.4.4 Forecasting Based on ARMA Model

Suppose that our final model selected is an ARMA(p, q) and their parameters are estimated using the training data. Then, for the h-step ahead point

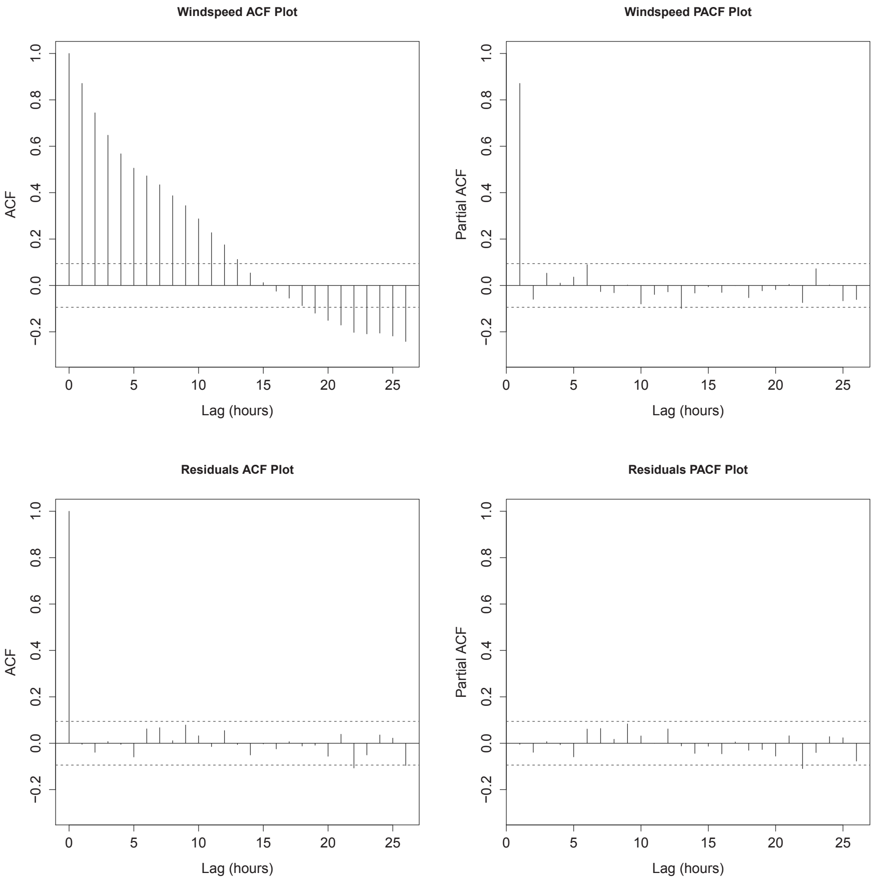

FIGURE 2.3 Top panel: ACF and PACF plots of the original hourly wind data; bottom panel: ACF and PACF plots of the residuals after an ARMA(1,1) model is fit.

forecasting, which is to obtain $\hat{V}_{t+h}$, we use the following formula,

$$\begin{aligned}\hat{V}_{t+h} &:= \mathbb{E}(V_{t+h}|V_1, V_2, \ldots, V_n) \\ &= \hat{a}_0 + \sum_{i=1}^{p} \hat{a}_i \hat{V}_{t-i+h} + \sum_{j=1}^{q} \hat{b}_j \hat{\varepsilon}_{t-j+h}.\end{aligned} \tag{2.31}$$

In the above equation, when the time index on a $\hat{V}$ is prior to t, meaning that the wind data has been observed, then $\hat{V}$ is replaced by its observed value at that time and $\hat{\varepsilon}$ is estimated in Eq. 2.25, whereas when the time index on a $\hat{V}$ is posterior to t, then $\hat{V}$ is the forecasted value at that time and $\mathbb{E}(\hat{\varepsilon}) = 0$.

To assess the uncertainty of the forecast, we need to calculate the variance of the forecasting error. For that, we use the Wold decomposition [8]. The Wold decomposition says that the ARMA model in Eq. 2.18 can be expressed as an infinite summation of all the error terms, such as

$$V_{t+h} = a_0 + \varepsilon_{t+h} + \psi_1 \varepsilon_{t+h-1} + \ldots \psi_{h-1}\varepsilon_{t+1} + \psi_h \varepsilon_t + \psi_{h+1}\varepsilon_{t-1} + \ldots, \tag{2.32}$$

where ψ_i's can be decided from a_i's and b_j's in Eq. 2.18. We here omit the detailed expression for ψ_i's.

With the expression in Eq. 2.32, the h-step ahead forecast is

$$\hat{V}_{t+h} := \mathbb{E}(V_{t+h}|V_1, V_2, \ldots, V_n) = \hat{a}_0 + \psi_h \hat{\varepsilon}_t + \psi_{h+1}\hat{\varepsilon}_{t-1} + \ldots. \tag{2.33}$$

Therefore, the h-step ahead forecast error at time t, denoted by $e_t(h)$, can be expressed as

$$e_t(h) = V_{t+h} - \hat{V}_{t+h} = \varepsilon_{t+h} + \psi_1 \varepsilon_{t+h-1} + \ldots + \psi_{h-1}\varepsilon_{t+1}. \tag{2.34}$$

The expectation of $e_t(h)$ is zero, namely $\mathbb{E}(e_t(h)) = 0$, and its variance is expressed as

$$Var(e_t(h)) = Var\left(\sum_{\ell=0}^{h-1} \psi_\ell \varepsilon_{t+h-\ell}\right) = \sigma_\varepsilon^2 \sum_{\ell=0}^{h-1} \psi_\ell^2, \tag{2.35}$$

where we define $\psi_0 = 1$. Combining the point forecast and the variance, the $100(1-\alpha)\%$ prediction interval for the h-step ahead forecasting is

$$\hat{V}_{t+h} \pm z_{\alpha/2} \cdot \sqrt{Var(e_t(h))} = \hat{V}_{t+h} \pm z_{\alpha/2} \cdot \sigma_\varepsilon \cdot \sqrt{\sum_{\ell=0}^{h-1} \psi_\ell^2}. \tag{2.36}$$

From the above formula, it is apparent that farther in the future the forecast is, the greater the forecasting variance becomes.

In R, one can use the function `forecast` in the `forecast` package to make forecasting. The basic syntax is `forecast(wsdata, `h`, model = fit)`, which

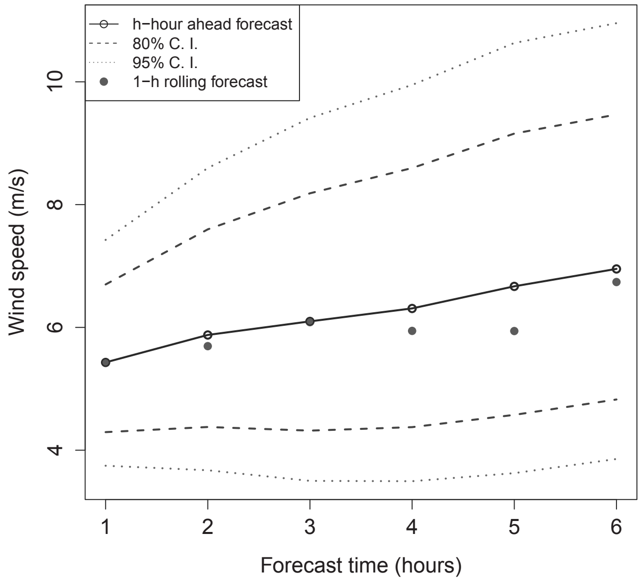

FIGURE 2.4 Wind speed forecasting based on the ARMA(1,1) model. $h = 1, 2, \ldots, 6$.

makes an h-step ahead forecasting using the fitted ARMA model whose parameters are stored in `fit`. The `forecast` function plots both the point forecast and the confidence intervals. By default, the `forecast` function draws two confidence intervals, which are the 80% and 95% confidence intervals. The confidence levels can be adjusted by setting the input argument `level` to other values. For example, `level = c(95, 99)` sets the two confidence intervals at 95% and 99%, respectively.

Fig. 2.4 presents the forecasting outcome based on the ARMA(1,1) model estimated in the previous subsections and using the hourly data of April. The solid line is the h-hour ahead forecast, assuming that the data is available only up to time t. The solid dots represent a one-hour ahead rolling forward forecasting by using the new wind speed observation, at $t+1$, $t+2$, …, $t+5$, respectively. For the rolling forward forecasting, the ARMA(1,1) model is refit every time. It is understandable that the two forecasts are the same at $t+1$ but they differ starting from $t+2$ when the one-hour ahead rolling forward forecasting uses the actual wind speed observations V_{t+h} at $h > 0$, while the h-hour ahead forecasting uses the forecasted wind speed $\hat{V}_{t+h}$ at $h > 0$.

2.5 OTHER METHODS

Several other methods, some of machine learning flavor, have been developed for short-term forecasting. In this section, we discuss the use of the Kalman filter (KF), support vector machine (SVM) and artificial neural network (ANN). We defer discussion on regime switching techniques [6, 75] to Chapter 4.

2.5.1 Kalman Filter

The Kalman filter [116] was initially developed for linear dynamic systems described by a state space model. Using the notations introduced in this chapter, the state space model for wind speed forecasting can be expressed as,

$$\begin{aligned} \text{state equation} \quad & \mathbf{a}_t = \mathbf{\Phi} \mathbf{a}_{t-1} + \boldsymbol{\omega}_{t-1}, \\ \text{observation equation} \quad & V_t = \mathbf{h}_t^T \mathbf{a}_t + \varepsilon_t, \end{aligned} \tag{2.37}$$

where $\mathbf{\Phi}$ is known as the state matrix, $\mathbf{a}_t = (a_{1,t}, \ldots, a_{p,t})^T$ is the state vector, $\mathbf{h}_t = (V_{t-1}, \ldots, V_{t-p})^T$ is the observation vector, and, ε_t and $\boldsymbol{\omega}_t$ are random noises. The first equation is referred to as the state equation, whereas the second equation is referred to as the observation equation. The observation equation is essentially an AR model, which is to predict the future wind speed (or power) as a linear combination of its past observations. Unlike the AR model, the Kalman filter model treats the coefficients, $\mathbf{a}_t$, as variables rather than constants, and updates them as new observations arrive, so as to catch up with the dynamics in the wind data. The two noise terms are often assumed to be normal variables, namely $\varepsilon_t \sim \mathcal{N}(0, (\sigma_\varepsilon^2)_t)$ and $\boldsymbol{\omega}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$, where $(\sigma_\varepsilon^2)_t$ is the time-varying variance of ε_t and $\mathbf{Q}_t$ is the time-varying covariance matrix of $\boldsymbol{\omega}_t$. The state vector, $\mathbf{a}_t$, is a random vector. It also has a covariance matrix, which we define by $\mathbf{P}_t$.

In the wind application, the observation vector, as expressed in Eq. 2.37, is the past n observations of wind speed, immediately before the current time t. But some analysts use the output from an NWP [44, 136] as their observation vector, and in this way, the Kalman filter serves to enhance the predictive resolution and accuracy of the heavy-computing, slow-running NWP.

The state matrix, $\mathbf{\Phi}$, is often assumed an identity matrix, namely $\mathbf{\Phi} = \mathbf{I}$, unless the underlying process dictates a different evolution dynamics of the state vector, $\mathbf{a}_t$. A further simplification is to assume that $\mathbf{Q}_t$ is a diagonal matrix—random variables in $\boldsymbol{\omega}_t$ are uncorrelated—and has an equal variance. As such, we can express $\mathbf{Q}_t = (\sigma_\omega^2)_t \cdot \mathbf{I}$, where $(\sigma_\omega^2)_t$ is known as the variance of the *system noise*, whereas the $(\sigma_\varepsilon^2)_t$ is known as the variance of the *observation noise*.

Before introducing the Kalman filter prediction and updating mechanism, we need to articulate the meaning of time instance t here. When we say “at time t,” we mean that we have observed the wind data at that time. The Kalman filter has an update step between two time instances, $t-1$ and t, or more specifically, after the wind data at $t-1$ has been observed but before the observation at t. To denote this update step, analysts use the notation, $t|t-1$. For example, $\mathbf{a}_{t|t-1}$ is the predicted value of the state vector after the observations up to $t-1$ but before the observation at t.

The Kalman filter runs through two major steps in iteration—prediction and update. Suppose that we stand between $t-1$ and t, and have the historical observations in $\mathbf{h}_t$ as well as previous estimations, $\hat{\mathbf{a}}_{t-1}$ and $\mathbf{P}_{t-1}$. At this moment, before we observe V_t, we can predict

$$\begin{aligned} \hat{\mathbf{a}}_{t|t-1} &= \mathbf{\Phi}\hat{\mathbf{a}}_{t-1}, && (2.38)\\ \mathbf{P}_{t|t-1} &= \mathbf{\Phi}\mathbf{P}_{t-1}\mathbf{\Phi}^T + (\sigma_\omega^2)_{t-1}\cdot\mathbf{I}, && (2.39)\\ \hat{V}_{t|t-1} &= \mathbf{h}_t^T\hat{\mathbf{a}}_{t|t-1}, && (2.40)\\ (\hat{\sigma}_V^2)_{t|t-1} &= \mathbf{h}_t^T\mathbf{P}_{t|t-1}\mathbf{h}_t + (\sigma_\varepsilon^2)_t. && (2.41) \end{aligned}$$

The last two equations are used to make a one-step ahead forecasting. The $100(1-\alpha)\%$ predictive confidence interval for V_t, before V_t is observed, is

$$[\hat{V}_{t|t-1} - z_{\alpha/2}\cdot(\hat{\sigma}_V^2)_{t|t-1}, \quad \hat{V}_{t|t-1} + z_{\alpha/2}\cdot(\hat{\sigma}_V^2)_{t|t-1}].$$

If the desire is to make multiple-hour ahead forecasting, then the state space model should be built on a coarse temporal granularity. The default temporal resolution is an hour, meaning that one hour passes from $t-1$ to t. If we increase the temporal granularity to two hours, meaning that two hours pass from $t-1$ to t, then, the above one-step ahead forecasting makes a 2-hour ahead forecast. The downside is that the historical data is thinned and the data point between the two chosen time instances for the Kalman filter are ignored—this apparently is a drawback.

At time t, after V_t is observed, $\hat{\mathbf{a}}_t$ and $\mathbf{P}_t$ get an update through the following steps,

$$\begin{aligned} \mathbf{K}_t &= \frac{1}{(\hat{\sigma}_V^2)_{t|t-1}}\mathbf{P}_{t|t-1}\mathbf{h}_t, && (2.42)\\ \hat{\mathbf{a}}_t &= \hat{\mathbf{a}}_{t|t-1} + \mathbf{K}_t(V_t - \hat{V}_{t|t-1}), && (2.43)\\ \mathbf{P}_t &= (\mathbf{I} - \mathbf{K}_t\mathbf{h}_t^T)\mathbf{P}_{t|t-1}, && (2.44) \end{aligned}$$

where $\mathbf{K}_t$ is known as the *Kalman gain.* To start the process, analysts can set the initial values for $\hat{\mathbf{a}}_t$ and $\mathbf{P}_t$ as

$$\mathbf{a}_0 = (1, 0, \ldots, 0)^T \quad \text{and} \quad \mathbf{P}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

The above $\mathbf{a}_0$ means that at the beginning, the prediction uses only the immediate past observation. Another parameter to be decided in the Kalman filter is p, the size of the state vector. This p can be decided by fitting an AR model and choosing the best p based on BIC.

To run the above Kalman filter, the variances of the observation noise and system noise are also needed. Crochet [44] suggests using the Smith algorithm and Jazwinski algorithm to dynamically estimate $(\sigma_\varepsilon^2)_t$ and $(\sigma_\omega^2)_t$, respectively. The basic idea for estimating the observation noise, $(\sigma_\varepsilon^2)_t$, is to treat it as the product of a nominal value, $(\sigma_\varepsilon^2)_0$, and a coefficient, ζ_t, where ζ_t is further assumed to follow an inverse gamma distribution with a shape parameter κ_t. Then, the Smith algorithm [199] updates the observation noise variance by

$$\begin{aligned} (\sigma_\varepsilon^2)_t &= \zeta_{t-1} \cdot (\sigma_\varepsilon^2)_0, \\ \zeta_t &= \frac{\zeta_{t-1}}{\kappa_{t-1}+1} \left(\kappa_{t-1} + \frac{(V_t - \hat{V}_{t|t-1})^2}{(\sigma_V^2)_{t|t-1}} \right), \\ \kappa_t &= \kappa_{t-1} + 1. \end{aligned} \tag{2.45}$$

The variance of the system noise, $(\sigma_\omega^2)_t$, can be estimated through the Jazwinski algorithm [107] as

$$(\sigma_\omega^2)_t = \left(\frac{(V_t - \hat{V}_{t|t-1})^2 - \mathbf{h}_t^T \mathbf{\Phi} \mathbf{P}_{t-1} \mathbf{\Phi}^T \mathbf{h}_t - (\sigma_\varepsilon^2)_t}{\mathbf{h}_t^T \mathbf{h}_t} \right)_+ , \tag{2.46}$$

where $(\cdot)_+$ returns the value in the parenthesis if it is positive, or zero otherwise. The initial values used in Eq. 2.45 are set as $(\sigma_\varepsilon^2)_0 = 1$, $\zeta_0 = 1$, and $\kappa_0 = 0$. The initial value, $(\sigma_\omega^2)_0$, is also set to zero.

Fig. 2.5 presents an illustrative example, which compares the Kalman filter forecast with AR(1) model forecast, when both are applied to the hourly data of April. The order of the AR model is chosen based on BIC. The best order, corresponding to the smallest BIC, is $p = 1$. Because the Kalman filter updates its one-hour ahead forecast with the new observation, to make a fair comparison, we use the AR(1) model to conduct a one-hour ahead forecast on a rolling forward basis from $t+1$ to $t+6$, the same as what is done for the solid dots in Fig. 2.4. The difference is that the model used then is ARMA(1,1), whereas the model used here is AR(1). The actual difference is, however, negligible, because $\hat{b}_1 = 0.0871$ in the ARMA(1,1) model, and as such, ARMA(1,1) behaves nearly identically to AR(1) with the same autoregressive coefficient. The point forecast of both methods are similar here, but the confidence interval of the Kalman filter is narrowing as more data are accumulated, while the confidence interval of the AR(1) one-hour ahead forecast stays much flatter.

2.5.2 Support Vector Machine

Support vector machine is one of the machine learning methods that are employed in wind speed forecasting. Support vector machine was initially developed for the purpose of classification, following and extending the work of optimal separating hyperplane. Its development is largely credited to Vladimir Vapnik [221].

Two important ideas are employed in a support vector machine. The first

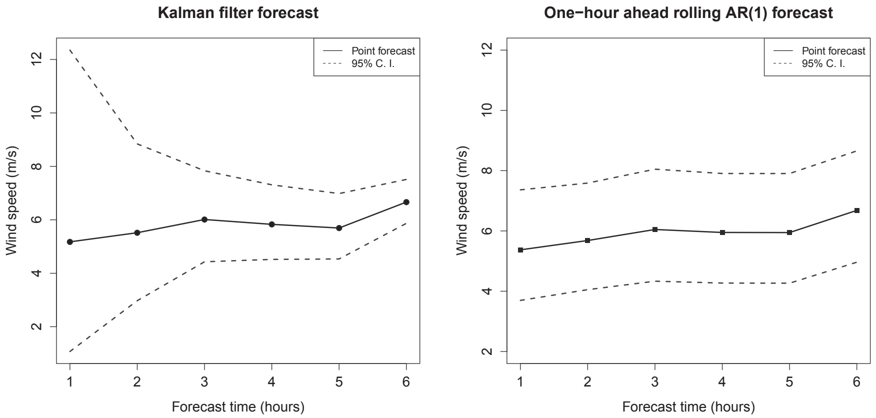

FIGURE 2.5 One-hour ahead forecasting plots from $t+1$ to $t+6$: Kalman filter (left panel) and AR(1) model (right panel).

is to use a small subset of the training data, rather than the whole set, in the task of learning. This subset of data points was called the *support vector* by its original developers, namely Vapnik and his co-authors. This is where the name Support Vector Machine comes from. In the case of a two-class classification, the data points constituting the support vector are those close to the boundary separating the two competing classes. The data points that are more interior to a data class and farther away from the separating boundary do not affect the classification outcome.

The second idea is to transform the data from its original data space to a potentially high-dimensional space for a better modeling ability. This type of transformation is nonlinear, so that a complicated response surface or a complex feature in the original space may become simpler and easier to model in the transformed space. The theoretical foundation for such transformation lies in the theory of reproducing kernel Hilbert space (RKHS) [86].

The use of the first idea helps the use of the second idea. One key reason for SVM to do well in a higher dimensional space without imposing too much computational burden is because the actual number of data points involved in its learning task, which is the size of the support vector, is relatively small.

The application of SVM to wind speed data is to solve a regression problem, in which the response is a real value, albeit nonnegative, instead of a categorical value. SVM is applicable to regression problems but a different loss function ought to be used. We will discuss those next.

Support vector machine falls into the class of supervised machine learning methods, in which a set of data pairs, $\{\boldsymbol{x}_i, y_i\}_{i=1}^{n}$, is collected and used to train a model (model training is the same as to decide the model order and estimate the model parameters). In the data pairs, $\boldsymbol{x}_i$ is the input and y_i

is the corresponding output. In the context of wind speed forecasting, what analysts use to forecast a future value is the historical observations. At time t, the input vector comprises the wind speed data p-step back in the history, and the response y is the h-step ahead to be forecasted. In other words, $\boldsymbol{x}_t$ and y_t can be expressed as

$$\boldsymbol{x}_t = (V_t, \ldots, V_{t-p+1})^T \quad \text{and} \quad y_t = V_{t+h}.$$

This $\boldsymbol{x}_t$ is essentially the same as the observation coefficient vector, $\mathbf{h}_{t+1}$, in the Kalman filter. We group the data in the collection of historical observations running from time 1 to time $n + h$ and label them as $\boldsymbol{x}$'s and y's accordingly. Wind speed V_ℓ for $\ell \leq 0$ is set to zero. Like in the Kalman filter, p can be chosen by fitting an AR model to the wind data.

SVM finds the relationship between $\boldsymbol{x}$ and y, so that a forecast can be made for h-step ahead whenever a new set of wind speed observations are available. Unlike in AR models and the Kalman filter, the y-to-$\boldsymbol{x}$ relationship found by SVM is not necessarily linear. In fact, it is generally nonlinear. Analysts believe that a nonlinear functional relationship is more flexible and capable, and could hence lead to an enhanced forecasting capability. When using SVM, for a different h, a different SVM predictive model needs to be built, or needs to be trained. This aspect appears different from the recursive updating nature of the Kalman filter or the ARMA model.

The general learning problem of SVM can be formulated as

$$\hat{\boldsymbol{\alpha}} = \arg\min \left\{ L(\mathbf{y}, \mathbf{K}\boldsymbol{\alpha}) + \frac{\gamma}{2}\boldsymbol{\alpha}^T \mathbf{K}\boldsymbol{\alpha} \right\}, \tag{2.47}$$

where $L(\cdot, \cdot)$ is a loss function that can take different forms, depending on whether this is a regression problem or a classification problem, $\mathbf{y} = (y_1, \ldots, y_n)^T$ is the output vector, $\mathbf{K}$ is the Gram matrix (or the kernel matrix), to be explained below, $\boldsymbol{\alpha}$ is the model parameters to be learned in the training period, using the training dataset, $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, and γ is the penalty constant to regulate the complexity of the learned functional relationship. A large γ forces a simpler, smooth function, while a small γ allows a complicated, more wiggly function. Recall the overfitting issue discussed in Section 2.4.2. An overly complicated function leads to overfitting, which in turn harms a model's predictive capability. The inclusion of γ is to help select a simple enough model that has good predictive performances.

The above formulation appears to be different from many of the SVM formulations presented in the literature. This is because the above SVM formulation is expressed under the reproducing kernel Hilbert space framework. The RKHS theory is too involved to be included here—after all, the main purpose of this book is not machine learning fundamentals. The benefit to invoke this RKHS framework is that doing so allows the SVM formulation to be presented in a clean and unified way and also be connected easily with other learning methods, such as Gaussian process regression [173] or smoothing splines [86].

In the kernel space formulation, one key element is the Gram matrix $\mathbf{K}$, which is created by a kernel function $K(\cdot,\cdot)$, such that the (i,j)-th element of $\mathbf{K}$ is $(\mathbf{K})_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This is how the input vector information $\boldsymbol{x}$'s get incorporated in the learning equation of Eq. 2.47; otherwise, it may appear strange that SVM learns $\boldsymbol{\alpha}$ by using $\mathbf{y}$ only, as on the surface, $\boldsymbol{x}$ does not appear in Eq. 2.47.

There are several commonly used kernel functions in SVM. A popular one is the radial basis function kernel, defined as

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left\{-\phi\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2\right\}, \tag{2.48}$$

where $\|\cdot\|_2$ defines a 2-norm; for more discussions on norm, please refer to Section 12.3.1. The radial basis kernel is also known as the Gaussian kernel, as its function form resembles the density function of a Gaussian (normal) distribution. Using the radial basis kernel, it introduces one extra parameter, ϕ, which will be decided in a similar fashion as how γ in Eq. 2.47 is decided. This is to be discussed later.

Once the parameters in $\boldsymbol{\alpha}$ are learned, analysts can use the resulting SVM to make forecasting. For instance, we train an SVM using data from 1 to n. Then, with a new observation, V_{n+1}, we would like to make a forecast of V_{n+h+1}. We first form a new input vector, denoted by $\boldsymbol{x}_{\text{new}} = (V_{n+1}, V_n, \ldots, V_{n-p+2})^T$. Then, the forecasting model is

$$\hat{V}_{n+h+1}(\boldsymbol{x}_{\text{new}}) = \sum_{i=1}^{n} \hat{\alpha}_i K(\boldsymbol{x}_{\text{new}}, \boldsymbol{x}_i). \tag{2.49}$$

For a general h-step ahead forecasting where $h > 1$, it is important to make sure that $\boldsymbol{x}_{\text{new}}$ properly includes the new observations that matter to the forecasting. Then, the same formula can be used to obtain a general h-step ahead forecast $\hat{V}_{t+h}$.

SVM for classification and SVM for regression use different loss functions. First, let us define a general prediction function for SVM as $g(\boldsymbol{x})$. Similar to the prediction expressed in Eq. 2.49, the general prediction function takes the form of

$$g(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i K(\boldsymbol{x}, \boldsymbol{x}_i). \tag{2.50}$$

The loss function can be denoted by $L(y, g(\boldsymbol{x}))$. For classification, a hinge loss function,

$$L(y, g(\boldsymbol{x})) = \sum_{i=1}^{n} \left(1 - y_i g(x_i)\right)_+, \tag{2.51}$$

is used. As illustrated in Fig. 2.6, left panel, this loss function, expressed in yg, looks like a hinge comprising two straight lines. For regression, an ϵ-sensitive error loss function,

$$L(y, g(\boldsymbol{x})) = \sum_{i=1}^{n} \left(|y_i - g(x_i)| - \epsilon\right)_+, \tag{2.52}$$

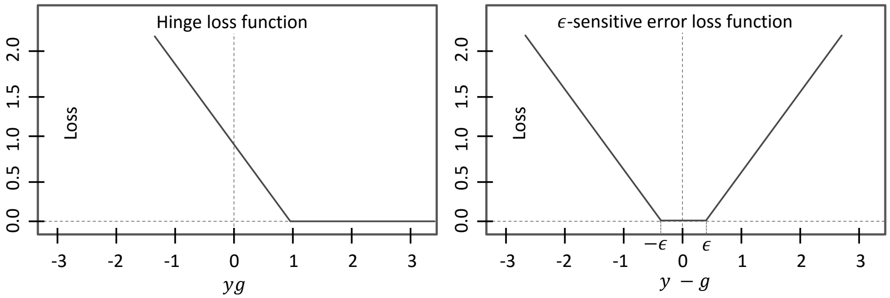

FIGURE 2.6 The loss functions used by support vector machine in classification (left panel) and in regression (right panel).

is used, which is illustrated in Fig. 2.6, right panel.

SVM regression can be made equivalent to Gaussian process regression, if (a) the loss function uses a squared error loss function, (b) $\gamma/2$ is set to σ_ε^2, which is the variance of the i.i.d noise term, (c) when the kernel function, $K(\cdot,\cdot)$, is set to be a covariance function. This connection becomes clearer after we discuss the Gaussian process regression in Section 3.1.3 (see also Exercise 3.2).

To run an SVM regression, the needed input is the training dataset $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$ and three exogenous parameters, γ in Eq. 2.47, ϕ in Eq. 2.48, and ϵ in Eq. 2.52 (not to confuse this ϵ with the i.i.d noise term ε). These exogenous parameters can be decided by using a cross-validation strategy [86]. A five-fold cross validation is carried out through the steps in Algorithm 2.1.

In `R`'s `e1071` package, a number of functions can help execute the SVM regression and make forecast. The `svm` function performs both classification and regression. It performs a regression, if it detects real values in y. By default, the `svm` uses a radial basis function and sets $\gamma = 1$, $\phi = 1/p$, and $\epsilon = 0.1$. Please note that γ in our formulation is the reciprocal of the `cost` argument used in the standard SVM package in `R`, and the radial kernel coefficient, ϕ, is called `gamma` in the SVM package.

The following command can be used to perform a model training,

```
svm.model <- svm(Y~ X, data = trainset).
```

To apply the SVM to the test dataset,

```
svm.pred <- predict(svm.model, testset).
```

To select the exogenous parameters, analysts can use the `tune` function to run a grid search. Suppose that we have fixed $\phi = 1$ but want to see which combination of γ and ϵ produces a better model, we may use

```
outcome<-tune(svm, Y~ X, data = trainset, ranges =
  list(epsilon = seq(0,1,0.1), cost = 10^(-4:4)).
```

Algorithm 2.1 A five-fold cross-validation procedure.

1. Choose a value for γ, ϕ, and ϵ, respectively.
2. Split the whole training dataset into five subsets of nearly equal data amount.
3. Use four subsets of the data to train an SVM regression model.
4. Use the remaining unused data subset to evaluate the performance of the model, using one of the performance metrics that are to be discussed in Section 2.6.
5. Repeat Steps 3 and 4 five times. Each time, always use four subsets to train a model and use the unused fifth subset to evaluate the model's forecasting performance.
6. Use the average of the performance metric values as the final model performance.
7. Repeat from Step 1 by trying other combinations of γ, ϕ, and ϵ. Select whichever combination produces the best forecasting model.

Because the `tune` function runs an exhaustive search, it could take a long time, especially if all three parameters are to be optimized. To speed up, analysts can optimize one factor at a time or employ a meta-heuristic optimization routine such as the genetic algorithm.

Using the same April wind data as used in the previous subsections, we explore which parameter combination produces the best SVM. Here, a radial basis kernel is used, $p = 1$ as in the Kalman filter example, and $\phi = 1$. To ease the computation, we use a greedy search strategy, which is to fix the value of $\epsilon = 0.1$, vary `cost` in a broad range. It turns out that `cost` $= 1$ is preferred. Then, fix `cost` $= 1$ and vary ϵ from 0 to 1. This process chooses $\epsilon = 0.2$.

2.5.3 Artificial Neural Network

Artificial neural network is another machine learning method that is widely employed in wind speed forecasting. ANN can be used for both classification and regression, too. Like in the case of SVM, the application of ANN to wind speed forecasting is a regression problem. The problem setting is similar to that described in the SVM section:

- A set of training data points, $\{\boldsymbol{x}_i, y_i\}_{i=1}^n$, is collected, where $\boldsymbol{x}_i$ and y_i are defined likewise as in SVM.
- ANN aims to find the relationship between $\boldsymbol{x}$ and y, and the resulting

relationship is nonlinear, as in the case of SVM and unlike the linear relationship assumed in AR models and the Kalman filter.

- To make a forecast at $t+h$, one chooses V_{t+h} as the corresponding y_i. ANN can train a model with multiple outputs, meaning that the outputs of an ANN can make forecasts, all at once, at a number of h-step ahead times with different h's. This is, at least conceptually, a convenience provided by ANN. On the flip side, training a multi-output model takes more care than training a single-output model.

Neural networks consist of an input layer and an output layer, which are connected through one or many hidden layers in between. Fig. 2.7, left panel, presents a multiple-input and single-output neural network, which has only one hidden layer. Each layer comprises a number of nodes. The nodes on the input layer are basically the input variables, whereas the nodes on the output layer are the response variables, namely the forecast to be made in the wind applications. By letting $y = V_{t+h}$, the neural net in Fig. 2.7 is to make an h-step ahead forecast for the given h. As mentioned above, it is straightforward for an ANN to have multiple outputs, so as to make simultaneous forecasts at multiple future time instances.

The information flow in an ANN goes as follows. The input layer takes in the input data. The connection between the input nodes and a node on the hidden layer feeds a linear combination of the inputs to the hidden node and outputs a value after a nonlinear transformation. The final output of the network is a linear combination of the values of the hidden nodes. Denote by Z the node on the hidden layer and assume that there are M hidden nodes, i.e., $Z_1, \ldots, Z_M$. As such, a neural net is described mathematically as,

$$Z_m(\boldsymbol{x}) = \sigma(\alpha_{0m} + \boldsymbol{\alpha}_m^T \boldsymbol{x}), m = 1, \ldots, M, \tag{2.53}$$

$$\hat{y} = g(\boldsymbol{x}) = \beta_0 + \sum_{m=1}^{M} \beta_m Z_m(\boldsymbol{x}), \tag{2.54}$$

where α_{0m}, $\boldsymbol{\alpha}_m$, and $\beta_i, i = 0, 1, \ldots, M$ are the model parameters to be learned from the training data, and $\sigma(\cdot)$ is the sigmoid function, taking the form $\sigma(x) = 1/(1+e^{-x})$. For an illustration, please take a look at Fig. 2.6, right panel. This sigmoid function is the nonlinear transformation, referred to a short while ago, that takes place at the hidden nodes. Because of this nonlinear transformation, the resulting ANN model is inherently nonlinear. This sigmoid function is called an *activation* function, as what it does is to tame an input if its value is negative, but let the input pass if its value is positive. This function is adopted to mimic the activation of a biological neuron responding to a stimulus—this analogy earns the method its name. In Eq. 2.53, analysts sometimes use the radial basis function as the $\sigma(\cdot)$ function, instead of a sigmoid function. If so, the resulting ANN is referred to as a radial basis function neural net.

If we choose an identity function as $\sigma(\cdot)$, namely $\sigma(x) = x$, then the ANN

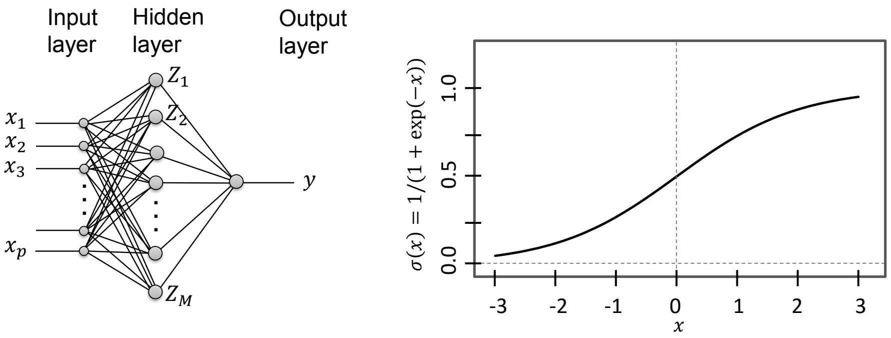

FIGURE 2.7 Left panel: a single hidden layer, a single-output neural network. Right panel: a sigmoid function.

model simplifies to a linear model. In this way, an ANN can be thought of as a two-stage, nonlinear generalization of the linear model. A general ANN also has multiple layers. It has been long believed that having multiple layers increases the data modeling capability of the resulting neural net, but the difficulty surrounding the optimization for parameter estimation made a multiple-layer neural net initially less practical. This optimization problem was addressed about a decade ago, and consequently, the many-layered neural nets become popular nowadays. The many-layered neural nets are referred to as deep neural nets, or commonly, *deep learning* models. The single layer one, by contrast, is called a shallow neural net.

An ANN is parameterized by α_{0m}, $\boldsymbol{\alpha}_m$, and $\beta_i, i = 0, 1, \ldots, M$, known as the *weights* in the language of neural nets, as they can be viewed as the weights associated with the links between an input node and a hidden node, or between a hidden node and an output node. For regression, the loss function used in an ANN training is the squared error loss, i.e., $\sum_{i=1}^{n}(y_i - g(\boldsymbol{x}_i))^2$.

For a single-layer, single-output ANN, the number of inputs and that of the hidden nodes need to be decided before the training stage. Concerning the number of inputs, we recommend using the same number of inputs as in the Kalman filter or the support vector machine, for which the choice of p can be hinted by fitting an AR model. Please be aware that the inputs to an ANN can be easily expanded. Analysts have included wind power, time in a day, temperature, among other things, as inputs. Due to the flexibility of an ANN, the training is supposed to take care of the y-to-$\boldsymbol{x}$ relationship, depending much less on the nature of the inputs. Concerning the number of hidden nodes, Hastie et al. [86] recommend the range of 5 to 100 and using more hidden nodes if there are more input nodes, and offer the following rule of thumb—"*Generally speaking it is better to have too many hidden units [nodes] than too few.*"

When training a neural net, the starting values of the parameters are typically chosen to be random values near zero [86]. When inputs of different physical units are used, it is advised to standardize the inputs to have a zero mean and a standard deviation of one.

The R package neuralnet can facilitate the process of building a neural net. Suppose that we choose to have 10 hidden nodes on a single hidden layer. The following R command can be used,

```
nn <- neuralnet(Y~ X,data=trainset, hidden=10, linear.output=T),
```

where linear.output=T means that this is a regression problem. By default, the neuralnet function uses the resilient back-propagation with weight backtracking algorithm [178] to solve the optimization problem and estimate the parameters. If one chooses a multi-layer neural net, then the hidden argument needs to be set accordingly. For instance, setting hidden = c(5, 4, 3) means that the resulting ANN has three hidden layers, having 5, 4, and 3 nodes, respectively. To visualize the resulting neural net, one can use plot(nn). To test the resulting ANN on a set of test data, one can use

```
test.nn <- compute(nn, testset).
```

Using the April wind data and $p = 1$, we test a single hidden layer ANN with four different choices for the number of the hidden nodes, which are 5, 10, 15, and 30. A ten-fold cross validation settles at five hidden nodes.

2.6 PERFORMANCE METRICS

In order to assess the forecasting quality, a number of performance metrics are used. Consider the case that we have a set of n_{test} test data points, V_i, $i = 1, \ldots, n_{\text{test}}$, the corresponding forecast of each of which is $\hat{V}_i$. The most popular two metrics are the root mean squared error (RMSE) and the mean absolute error (MAE), defined, respectively, as

$$\text{RMSE} = \sqrt{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\hat{V}_i - V_i)^2}, \quad \text{and} \tag{2.55}$$

$$\text{MAE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} |\hat{V}_i - V_i|. \tag{2.56}$$

Both metrics evaluate the performance of a point forecast. RMSE is based on the squared error loss function, and thus sensitive to the existence of outliers, whereas MAE is based on the absolute error loss, and thus less sensitive to outliers.

Both RMSE and MAE count the absolute amount of forecasting error, regardless of the base value to be predicted. Some may argue that an error of 1 m/s, when predicting at the base wind speed of 3 m/s versus predicting at 15 m/s, has different impacts. To measure the relative error, the mean absolute percentage error (MAPE) is used. MAPE is defined as

$$\text{MAPE} = \frac{100}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \left| \frac{\hat{V}_i - V_i}{V_i} \right| . \tag{2.57}$$

Note that MAPE is given as a percentage quantity but its value can exceed 100%.

We want to point out that in some literature, for instance, in [75], MAE is called the mean absolute prediction error, the acronym of which is also MAPE. This confusion can be cleared in the context by looking at the spelled-out version of the acronym or the definition.

Hering and Genton [91] favor measuring the impact on the final power response affected by wind speed forecast. This is because the impact of a forecast error in wind speed on wind power is not uniform. Recall the power curve in Fig. 1.2. For a wind speed smaller than the cut-in wind speed or larger than the rated wind speed, an error in wind speed forecast has a smaller impact on wind power than the same amount of forecasting error has when the wind speed is between the cut-in speed and the rated speed, where the power curve has a steeper slope. To factor in the impact on a turbine's power response, Hering and Genton [91] propose the following power curve error (PCE), defined as

$$\text{PCE}_i = \begin{cases} \xi \left(g(V_i) - g(\hat{V}_i) \right) & \text{if} \quad \hat{V}_i \leq V, \\ (1-\xi) \left(g(\hat{V}_i) - g(V_i) \right) & \text{if} \quad \hat{V}_i > V, \end{cases} \tag{2.58}$$

$$\text{PCE} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \text{PCE}_i, \tag{2.59}$$

where $g(\cdot)$ is the power curve function and $\xi \in (0,1)$ is introduced to penalize underestimation and overestimation differently. In practice, underestimating incurs more cost than overestimating. Therefore, for practical purposes $\xi > 0.5$. Hering and Genton [91] recommend setting $\xi = 0.73$. Generally speaking, using the PCE ensures that the optimal forecast is a ξ-quantile [73]. If $\xi = 0.5$, PCE is the same as MAE.

The above metrics all measure the quality of a point forecast. If the forecasting is a probability density, to measure the quality of a density estimation or prediction, we use the mean continuous ranked probability score (CRPS) [76]. CRPS compares the estimated cumulative distribution function with the observations, and it is computed as

$$\text{CRPS} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \int \left(\hat{F}(V) - \mathbb{1}(V > V_i) \right)^2 dV, \tag{2.60}$$

where $\hat{F}(V)$ is the estimated cdf and $\mathbb{1}(\cdot)$ is an indicator function, such that $\mathbb{1}(\texttt{logic}) = 1$ if `logic` is true and zero otherwise. When the cdf, $F(\cdot)$, is replaced by a point forecast, CRPS reduces to MAE [75].

TABLE 2.5 Model parameters of SVM and ANN selected by cross validation. The ϕ parameter in SVM is set to be the reciprocal of p.

	SVM			**ANN**	
h	p	`cost`	ϵ	p	# of hidden nodes
1	1	100	0.3	1	10
2	1	10	0.4	1	5
3	4	1	0.5	1	5
4	4	1	0.6	1	10
5	3	1	0.3	1	5
6	3	1	0.5	1	5

2.7 COMPARING WIND FORECASTING METHODS

In this section, we conduct a comparison study using the yearlong hourly data in the `Wind Time Series Dataset` and see how individual forecasting models work.

For each month, we split the wind speed data into two portions as follows. We reserve the last six hours of data points as one of the test sets and take the remaining data in that month as one of the training datasets. We then group all 12 monthly training sets into an aggregated training set for the whole year.

Five different forecasting methods are considered—the persistence model, forecasting based on Weibull distribution (WEB), ARMA model, SVM, and ANN. For ARMA, BIC is used to decide the best model order. When training SVM and ANN, the cross-validation strategy is used to decide the exogenous parameters. For the first four models, the training data in the yearlong dataset are used to find the best model order, if applicable, and estimate the respective model parameters. For ANN, the convergence of the `R` package while using the yearlong dataset is very slow. We instead use only one month of data (April) in a cross validation to decide the number of hidden nodes for a single-layer neural net. Once that is decided, the remaining parameters in the ANN model are estimated still based on the yearlong training data.

For WEB, the mean of the estimated distribution is used as the forecast for all six h-hour ahead forecasts. For ARMA, an ARMA(2,2) model is chosen for making h-hour ahead forecasts at $h = 1, 2, \ldots, 6$. For SVM and ANN, six different models of each kind are trained to cover all six h values. For instance, when $h = 1$, we train an SVM and an ANN for one-hour ahead forecasting; when $h = 2$, we will train another SVM or ANN model for two-hour ahead forecasting; and so forth. Recall that this is a feature of the machine learning methods mentioned on page 42. The parameters of the selected SVM and ANN models are presented in Table 2.5.

The trained models are used to make forecasts at each month's test data. For each h, there are 12 test data points, i.e., $n_{\text{test}} = 12$, one per month. The

TABLE 2.6 RMSE (m/s) of five different forecasting methods.

Method	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$
PER	0.826	1.597	2.055	2.336	2.659	3.005
WEB	3.237	3.439	3.177	3.474	2.703	2.322
ARMA(2,2)	0.984	1.541	1.777	2.394	2.348	2.488
SVM	1.065	1.504	2.661	2.487	2.154	2.905
ANN	1.074	1.727	1.857	2.666	2.595	2.429

TABLE 2.7 MAE (m/s) of five different forecasting methods.

Method	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$
PER	0.631	1.194	1.626	1.744	2.227	2.442
WEB	2.405	2.452	2.457	2.839	2.099	1.813
ARMA(2,2)	0.769	1.163	1.303	1.962	2.055	2.024
SVM	0.864	1.258	2.007	1.959	1.780	2.235
ANN	0.856	1.452	1.441	2.125	2.148	2.010

12 test data points are used to compute three performance metrics—RMSE, MAE, and MAPE—for each forecasting method.

Tables 2.6–2.8 present the three metrics for the five methods. We observe the following:

1. For very short terms, like $h = 1$ or $h = 2$, the persistence model and ARMA model are clear winners. The method based on Weibull distribution is the worst by a noticeable margin.

2. Despite the bad performance for very near-term forecasting, WEB holds steady its performance as the forecasting horizon projects into the future, while the performances of all other methods deteriorate quickly. Eventually, WEB becomes the best forecasting at $h = 6$. PER suffers the greatest performance degradation when h increases from one hour to six hours.

3. The two machine learning methods, SVM and a single-layer ANN in this comparison, perform rather similarly. It is difficult to conclude which method is better. A many-layered ANN, or a deep neural net might, however, win over SVM. That remains to be studied.

4. If this study is used as a guide, then analysts are advised to use PER for one-hour or two-hour ahead forecasting, WEB for six-hour ahead or longer forecasting (before switching to NWP), and use ARMA models or machine learning methods for forecasting in between.

TABLE 2.8 MAPE (percentage) of five different forecasting methods.

Method	$h = 1$	$h = 2$	$h = 3$	$h = 4$	$h = 5$	$h = 6$
PER	8.2	16.0	21.4	19.4	27.9	30.3
WEB	26.6	27.0	29.8	29.4	24.6	21.7
ARMA(2,2)	9.3	16.2	18.1	20.9	25.6	27.0
SVM	9.8	16.5	23.7	20.7	23.0	26.8
ANN	9.6	18.1	19.1	21.9	26.0	26.6

GLOSSARY

ACF: Autocorrelation function

AIC: Akaike information criterion

AICc: Akaike information criterion corrected

ANN: Artificial neural network

AR: Autoregressive

ARMA: Autoregressive moving average

BIC: Bayesian information criterion

cdf: Cumulative distribution function

CRPS: Continuous ranked probability score

i.i.d: Identically, independently distributed

KF: Kalman filter

MA: Moving average

MAE: Mean absolute error

MAPE: Mean absolute percentage error

MLE: Maximum likelihood estimation

NWP: Numeric weather prediction

PACF: Partial autocorrelation function

PCE: Power curve error

pdf: Probability density function

PER: Persistence model or forecasting

RKHS: Reproducing kernel Hilbert space

RMSE: Root mean squared error

SVM: Support vector machine

WEB: Weibull distribution-based forecasting

EXERCISES

2.1 Find the probability density function for a three-parameter Weibull distribution.

a. Derive the corresponding log-likelihood function.

b. Use the three-parameter Weibull distribution to fit the hourly data in the `Wind Time Series Dataset` and report the estimated parameters.

c. Suppose the turbine cut-in speed is 4 m/s. Remove the wind speed data below the cut-in speed and fit both the two-parameter Weibull distribution and the three-parameter distribution. Please discuss the differences in your estimation outcomes.

2.2 Evaluate what impact different bin widths may have on the χ^2 goodness-of-fit test.

a. Use one month of the hourly data in the `Wind Time Series Dataset` and try the following bin widths: 0.2, 0.5, 1, 2 m/s.

b. Switch to one week of the 10-min data and try the same set of bin widths.

2.3 Use Hinkley's method to select the power transformation coefficient, m.

a. Try this on the hourly data in the `Wind Time Series Dataset` and try the following m values: 0, 0.5, 1, 2. Which m produces a $sym = 0$? Interpolation may be needed.

b. Switch to the 10-min data and try the same set of m values.

2.4 Remove the diurnal trend in the hourly data in the `Wind Time Series Dataset` by using Gneiting's trigonometric function in Eq. 2.15. Plot the original time series and the standardized time series. Compare them with the standardization using Eq. 2.14 and note any difference that you may have observed.

2.5 For the linear model in Eq. 2.20, the objective function leading to a least-squares estimation is

$$\min\left\{(\mathbf{V}-\hat{\mathbf{V}})^T(\mathbf{V}-\hat{\mathbf{V}}) = (\mathbf{V}-\mathbf{W}\hat{\mathbf{a}})^T(\mathbf{V}-\mathbf{W}\hat{\mathbf{a}})\right\}.$$

The least-squares estimation can be attained by taking the first derivative of this objective function, with respect to $\hat{\mathbf{a}}$, and setting it to zero. Please derive the least-squares estimation formula.

2.6 Use the hourly data in the `Wind Time Series Dataset` and conduct an ARMA modeling exercise. First, select the data from one of its months, and use this specific month data and do the following.

a. Fit a series of AR models, with $p = 1, 2, \ldots, 6$, respectively. When applying the three information criteria, do they select the same model order? Which criterion selects the simplest model?

b. Use the simplest AR model order selected in (a) and denote it as p_0. Compare the model AR(p_0) with ARMA(p_0, q) for $q = 1, 2, 3$, and select the model order q in a similar fashion as in (a) that selects p_0. Denote the resulting MA model order as q_0.

c. Conduct some model diagnostics of this ARMA(p_0, q_0) model by plotting its ACF and PACF. Do the ACF and PACF plots confirm a good model fit?

2.7 Derive Eq. 2.27 and Eq. 2.28 from Eq. 2.17.

2.8 When the loss function is a squared error loss function in Eq. 2.47, find the closed-form expression for the optimal $\hat{\boldsymbol{\alpha}}$.

2.9 Take the January hourly data from the `Wind Time Series Dataset` and use that as the historical training data. In the presence of missing data, please simply skip time stamps where data are missing and continue with the next available data.

a. Fit a series of AR models, with $p = 1, 2, \ldots, 6$, respectively. Use BIC to select the best model order p.

b. Use the resulting AR model to do an h-hour ahead forecast, for $h = 1, 2, \ldots, 100$. One hundred hours is a little bit over four days. Call this forecast 1.

c. Use the resulting AR model in (a) to do a one-hour ahead forecast. Shift the data sequence in (a) by one hour, namely that adding one new observation and dropping the oldest observation. Repeat, for the next 100 time instances, both the model fitting (including the determination of p) and the forecasting. Call this forecast 2.

d. Use a Kalman filter to do the one-hour forecasting but continue running the Kalman filter for the next 100 time instances. Set the p in the Kalman filter as that found in (a). Call this forecast 3.

e. For each forecasting, record both the forecasting result and the corresponding wind speed observation at every time instance. Compute RMSE and MAE for each forecast. Compare the performance metrics for all three forecasts and discuss pros and cons of each approach.

2.10 For the hourly data in the `Wind Time Series Dataset`, take wind power data, instead of wind speed data, and repeat the comparison study conducted in Section 2.7. Compute the three performance metrics for five different methods for each $h = 1, 2, \ldots, 6$.

CHAPTER 3

Spatio-temporal Models

When building predictive models for short-term wind forecast, spatial information is less frequently used than temporal information. Chapter 2 uses data obtained from a single turbine on a wind farm, which can also be applied to a single time-series data aggregating wind power outputs from the whole farm. Analysts have noticed that valuable information may be elicited by considering spatial measurements in a local region, as wind characteristics at a site may resemble those at neighboring sites. This gives rise to the idea of developing spatio-temporal methods to model the random wind field evolving through space and time.

Recall that we denote the wind speed data in Chapter 2 by V_t, which has only the time index. To model a spatio-temporal process, we expand the input variable set to include both the location variable, denoted by $\mathbf{s} \in \mathbb{R}^2$, and the time variable, denoted still by $t \in \mathbb{R}$, so that the spatio-temporal random wind field is represented by $V(\mathbf{s}, t)$. In this chapter, unless otherwise noted, N is used to denote the number of sites, whereas n is used to denote the number of time instances in the training set.

One of the key aspects in spatio-temporal modeling is to model the covariance structure of V through a positive-definite parametric covariance function, $Cov[V(\mathbf{s}, t), V(\mathbf{s}', t')]$.

3.1 COVARIANCE FUNCTIONS AND KRIGING

In this section, we focus on spatial covariance. For the time being, $V(\mathbf{s}, t)$ is simplified to be $V(\mathbf{s})$. Recall that the temporal covariance, also known as autocovariance, is discussed in Section 2.4.3.

We use $C(\mathbf{s}, \mathbf{s}'; t, t')$ to represent a covariance function, namely

$$C(\mathbf{s}, \mathbf{s}'; t, t') := Cov[V(\mathbf{s}, t), V(\mathbf{s}', t')].$$

When the time is held still and only the spatial covariance is concerned, the covariance function $C(\mathbf{s}, \mathbf{s}'; t, t')$ can be simplified to $C(\mathbf{s}, \mathbf{s}') := Cov[V(\mathbf{s}), V(\mathbf{s}')]$, after dropping the time index.

Given a set of N locations, $\mathbf{s}_1, \ldots, \mathbf{s}_N$, we can compute the corresponding covariance matrix $\mathbf{C}$, whose (i, j)-th entry is $C_{ij} = C(\mathbf{s}_i, \mathbf{s}_j)$. The covariance matrix is positive definite if all its eigenvalues are strictly positive, or positive semidefinite if some of its eigenvalues are zeros while the rest are positive. It is not difficult to notice that the covariance function is related to the kernel function mentioned in Section 2.5.2 and the covariance matrix is related to the Gram matrix (or the kernel matrix). A covariance function is referred to as a covariance kernel in a general machine learning context, and it can be shown that a positive definite kernel can be obtained as a covariance kernel in which the distribution has a particular form [94].

3.1.1 Properties of Covariance Functions

We start with the discussion of some general properties of the covariance functions.

Stationarity. A covariance function can be used to characterize both stationary and nonstationary stochastic processes. We primarily consider the stationary covariance function in this book, which has the property

$$C(\mathbf{s}, \mathbf{s}') = g(\mathbf{s} - \mathbf{s}'), \tag{3.1}$$

where $g(\cdot)$ is a function to be specified. The stationarity means that the covariance does not depend on the start location of a stochastic process but only depends on the distance and orientation between two points in that process. The variance of a stationary stochastic process can be expressed as

$$Var[V(\mathbf{s})] = g(\mathbf{0}) = \sigma_V^2. \tag{3.2}$$

For a stationary function, σ_V^2 is a constant, so that the stationary covariance matrix can be further factorized as

$$\mathbf{C} = \sigma_V^2 \cdot \mathbf{R}, \tag{3.3}$$

where $\mathbf{R}$ is a correlation matrix whose (i, j)-th entry is $\rho_{ij} = C_{ij}/\sigma_V^2$.

The concept of stationarity extends to the spatio-temporal covariance functions. By assuming stationarity, the covariance function only depends on the spatial lag, $\mathbf{u} = \mathbf{s} - \mathbf{s}'$, and the time lag, $h = t - t'$, such that the general function form $C(\mathbf{s}, \mathbf{s}'; t, t')$ can be expressed as $C(\mathbf{u}; h)$.

Isotropy: A stationary covariance function is isotropic, provided that

$$C(\mathbf{s}, \mathbf{s}') = g(\|\mathbf{s} - \mathbf{s}'\|_2), \tag{3.4}$$

where $\|\mathbf{s} - \mathbf{s}'\|_2$ is the Euclidean distance between the two locations $\mathbf{s}$ and $\mathbf{s}'$. When it does not cause any ambiguity, the subscript "2" is dropped hereinafter. Isotropy is to require invariance under rotation. This is to say, every pair of data points at $\mathbf{s}$ and $\mathbf{s}'$, respectively, having a common interpoint distance, must have the same covariance regardless of their orientation. Apparently, isotropy is a stronger condition than stationarity.

Smoothness: Smoothness (continuity and differentiability) is a property associated with sample functions, which are the realization of the stochastic process under a specified covariance function. The smoothness requirement is an important consideration in choosing a covariance function. The general relationship between the smoothness of sample functions and the covariance function is not straightforward. It is easier to talk about smoothness of sample functions when a specific covariance function is considered.

3.1.2 Power Exponential Covariance Function

A popular family of covariance functions is the power exponential function,

$$C(\mathbf{s}, \mathbf{s}') = \sigma_V^2 \exp\left\{ -\frac{1}{2} \sum_{j=1}^{d} \left| \frac{s_j - s_j'}{\theta_j} \right|^{p_j} \right\}, \tag{3.5}$$

where d is the dimension of $\mathbf{s}$, $0 < p_j \leq 2$ is the shape parameter, and θ_j is the scale parameter. Usually $d = 2$ in spatial statistics.

A special form of the power exponential covariance function is the isotropic squared exponential (SE) covariance function (the phrase "isotropic" is often omitted), whose parameters are $\theta_1 = \cdots = \theta_d = \theta$, and $p_1 = \cdots = p_d = p = 2$, so that

$$C_{\mathsf{SE}}(u) = \sigma_V^2 \exp\left\{ -\frac{u^2}{2\theta^2} \right\}, \tag{3.6}$$

where $u = \|\mathbf{u}\| = \|\mathbf{s} - \mathbf{s}'\| = \sqrt{\sum_{j=1}^{d} (s_j - s_j')^2}$. This function is also called the Gaussian covariance function. Recall the radial basis kernel in Eq. 2.48. The $C_{\mathsf{SE}}(\cdot)$ is the same as $K(\cdot, \cdot)$ if $\phi = 1/2\theta^2$ and $\sigma_V^2 = 1$.

An anisotropic form of the squared exponential covariance function is where the scale parameters are different along different input directions while its shape parameter is fixed at 2, namely $p_1 = \cdots = p_d = p = 2$. This anisotropic form is also known as the *automatic relevance determination* (ARD). The corresponding covariance function reads as,

$$C_{\text{SE-ARD}}(\mathbf{s}, \mathbf{s}') = \sigma_V^2 \exp\left\{ -\frac{1}{2} \sum_{j=1}^{d} \left| \frac{s_j - s_j'}{\theta_j} \right|^{2} \right\}. \tag{3.7}$$

The impact of the three types of parameters in the power exponential covariance function can be understood as follows, and Fig. 3.1 presents a few examples of the sample function under different parameter combinations.

- The variance term, σ_V^2, is referred to as the *amplitude*, because it is related to the amplitude of a sample function.

- The shape parameter, p, determines the smoothness of the sample functions. In the above two special cases of the power exponential function,

$p = 2$. Analysts like this choice because the corresponding sample functions are infinitely differentiable, meaning that the sample paths are smooth. For the power exponential family, $p = 2$ is the only shape parameter choice under which the sample functions are differentiable. When $p = 1$, the corresponding covariance function is known as the *exponential covariance function.* This choice is less popular because its sample functions are not smooth.

- The scale parameter, θ, referred to as the *length scale*, determines how quickly the correlation decays as the between-point distance increases. When θ decreases, the correlation between a pair of points of a fixed distance decreases, and thus, the sample functions have an increasing number of local optima. As a result, the sample function exhibits fast changing patterns and a short wavelength, where as θ increases, the correlation between a fixed pair of points increases, and the sample function hence exhibits slow changing patterns and a long wavelength.

Another popular family of the covariance function is the Matérn covariance function, which has a smoothness parameter, υ, that can control the smoothness of sample functions more precisely. Specifically, the sample functions are almost surely continuously differentiable of order $\lceil \upsilon \rceil - 1$, where $\lceil \cdot \rceil$ rounds up to the next integer. We choose to omit the presentation of the Matérn covariance function because we do not use it in this book. Interested readers can refer to [173] for more information.

3.1.3 Kriging

Kriging is the method commonly used to make spatial predictions. The method is named after the South African mining engineer, D. G. Krige. In spatial statistics and machine learning, kriging is generally referred to as the Gaussian process regression [41, 173]. The problem setting is as follows. Consider sites, $\mathbf{s}_1, \ldots, \mathbf{s}_N$, and the wind speeds at these locations, denoted by $V(\mathbf{s}_1), \ldots, V(\mathbf{s}_N)$.The N sites can be the turbine sites in a wind farm, and the wind speeds at $\{\mathbf{s}_1, \ldots, \mathbf{s}_N\}$ can be the wind speed measurements obtained by the respective nacelle anemometers. Analysts express the sites and respective measurements as data pairs, such as $\{\mathbf{s}_i, V(\mathbf{s}_i)\}_{i=1}^{N}$. The objective is to make a prediction at a site, say, $\mathbf{s}_0$, where no measurements are taken.

Two popular versions of kriging are the ordinary kriging and universal kriging. The ordinary kriging uses the following model,

$$V(\mathbf{s}_i) = \beta_0 + \delta(\mathbf{s}_i) + \varepsilon_i, \quad i = 1, ..., N, \tag{3.8}$$

where β_0 is an unknown constant, $\delta(\cdot)$ is the term modeling the underlying random field via the spatial correlation among sites, and ε is the zero mean, i.i.d Gaussian noise, such that $\varepsilon_i \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. The i.i.d Gaussian noise, ε, is also known as the *nugget effect.*

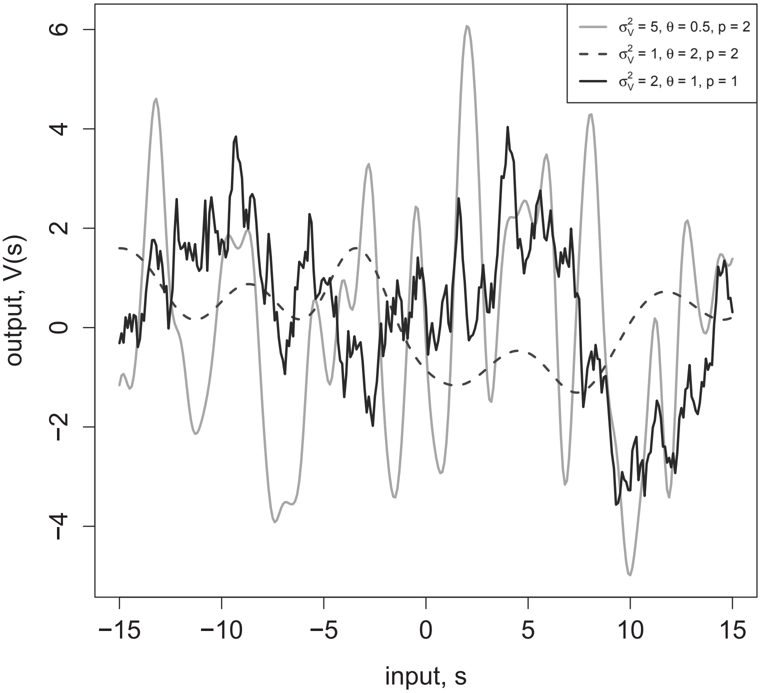

FIGURE 3.1 Three sample functions using a squared exponential covariance function with different parameter choices.

The random field term $\delta(\cdot)$ is assumed to be a zero-mean Gaussian process whose covariance structure is characterized, for instance, by a power exponential covariance function (other covariance functions can be used, too, but the power exponential family is a popular choice). Suppose that the squared exponential covariance function in Eq. 3.6 is used. It means that

$$C(\delta(\mathbf{s}), \delta(\mathbf{s}')) = \sigma_\delta^2 \exp\left\{-\frac{\|\mathbf{s}-\mathbf{s}'\|^2}{2\theta^2}\right\}, \tag{3.9}$$

where σ_δ^2 is the variance term associated with the random field function $\delta(\cdot)$. As such, the variance of wind speed is the summation of the variance associated with the random field and that of the i.i.d random noise, namely $\sigma_V^2 = \sigma_\delta^2 + \sigma_\varepsilon^2$.

What the ordinary kriging model implies is that the wind speed over a spatial field is centered around a grand average, β_0. The random fluctuation consists of two portions—the first depends on specific sites and is characterized by the spatial correlation between site $\mathbf{s}_0$ and the sites where observations are made or measurements are taken, and the second is the pure random noise, resulting from, for instance, the measurement errors.

Re-write Eq. 3.8 into a matrix form, i.e.,

$$\underbrace{\begin{pmatrix} V(\mathbf{s}_1) \\ V(\mathbf{s}_2) \\ \vdots \\ V(\mathbf{s}_N) \end{pmatrix}}_{\mathbf{V}} = \beta_0 \cdot \mathbf{1}_N + \underbrace{\begin{pmatrix} \delta(\mathbf{s}_1) \\ \delta(\mathbf{s}_2) \\ \vdots \\ \delta(\mathbf{s}_N) \end{pmatrix}}_{\boldsymbol{\delta}} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_N \end{pmatrix}}_{\boldsymbol{\varepsilon}}, \tag{3.10}$$

where $\mathbf{1}_N$ is an $N \times 1$ vector of all ones. Denote the covariance matrix of $\boldsymbol{\delta}$ by $\mathbf{C}_{NN} = (C_{ij})_{N \times N}$, where the subscript "$NN$" means that this is a covariance matrix for the N sites. Recall that δ and ε are two normal random variables having different covariance structures. This suggests that $\mathbf{V}$ follows a multivariate normal distribution, such as,

$$f(\mathbf{V}) = \mathcal{N}(\beta_0 \cdot \mathbf{1}_N, \mathbf{C}_{NN} + \sigma_\varepsilon^2 \mathbf{I}). \tag{3.11}$$

For the new site $\mathbf{s}_0$, the wind speed to be measured there, whose notation is simplified to V_0, has covariances with the existing N sites. The covariances can be characterized using the same covariance function, such as $C_{0j} = C(\mathbf{s}_0, \mathbf{s}_j)$, for $j = 1, \ldots, N$. Introduce a new $1 \times N$ row vector,

$$\mathbf{c}_{0N} := (C_{01}, \ldots, C_{0N}).$$

Then, the multivariate joint distribution of $(V_0, \mathbf{V}^T)^T$ is

$$f\left(\begin{bmatrix} V_0 \\ \mathbf{V} \end{bmatrix}\right) = \mathcal{N}\left(\beta_0 \cdot \mathbf{1}_{N+1}, \begin{bmatrix} \sigma_\delta^2 + \sigma_\varepsilon^2 & \mathbf{c}_{0N} \\ \mathbf{c}_{0N}^T & \mathbf{C}_{NN} + \sigma_\varepsilon^2 \mathbf{I} \end{bmatrix}\right), \tag{3.12}$$

where $\sigma_\delta^2 + \sigma_\varepsilon^2$ is the variance of V_0, namely σ_V^2, which is also known as the prior variance at the unseen site $\mathbf{s}_0$ before the prediction. Invoke the conditional Gaussian distribution formula, which says that if $\mathbf{x}$ and $\mathbf{y}$ are jointly Gaussian, i.e.,

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_x \\ \boldsymbol{\mu}_y \end{bmatrix}, \begin{bmatrix} \mathbf{A} & \mathbf{D} \\ \mathbf{D}^T & \mathbf{B} \end{bmatrix} \right),$$

then, the condition distribution $f(\mathbf{x}|\mathbf{y})$ is

$$f(\mathbf{x}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_x + \mathbf{D}\mathbf{B}^{-1}(\mathbf{y} - \boldsymbol{\mu}_y), \mathbf{A} - \mathbf{D}\mathbf{B}^{-1}\mathbf{D}^T). \tag{3.13}$$

By using this conditional Gaussian distribution formula, we can express

$$\begin{aligned} f(V_0|\mathbf{V}) = \mathcal{N}(\beta_0 + \mathbf{c}_{0N}(\sigma_\varepsilon^2\mathbf{I} + \mathbf{C}_{NN})^{-1}(\mathbf{V} - \beta_0 \cdot \mathbf{1}), \\ \sigma_V^2 - \mathbf{c}_{0N}(\sigma_\varepsilon^2\mathbf{I} + \mathbf{C}_{NN})^{-1}\mathbf{c}_{0N}^T). \end{aligned} \tag{3.14}$$

This conditional distribution leads to the predictive distribution of V_0, once the observations on the existing N sites are obtained. We can write the predictive mean and predictive variance, respectively, as

$$\begin{aligned} \hat{V}_0 &:= \hat{\mu}_0 = \hat{\beta}_0 + \mathbf{c}_{0N}(\hat{\sigma}_\varepsilon^2\mathbf{I} + \mathbf{C}_{NN})^{-1}(\mathbf{V} - \hat{\beta}_0 \cdot \mathbf{1}), \\ Var(\hat{V}_0) &:= \hat{\sigma}_0^2 = \sigma_V^2 - \mathbf{c}_{0N}(\hat{\sigma}_\varepsilon^2\mathbf{I} + \mathbf{C}_{NN})^{-1}\mathbf{c}_{0N}^T. \end{aligned} \tag{3.15}$$

The first equation is the *kriging predictor*, which is a linear combination of the observed wind speeds in $\mathbf{V}$. The linear coefficients (the weights) depend on the correlation between the unseen site, $\mathbf{s}_0$, and the N training sites as well as the variance in the training data. The coefficients are bigger, namely the weights are greater, if the correlation is strong and the training data have small variances. The predictive variance is reduced from the prior variance σ_V^2 at the unseen site. The reduced amount depends also on the correlation between the unseen site and the training sites as well as the variance in the training data. The $100(1 - \alpha)\%$ confidence interval for the prediction at $\mathbf{s}_0$ can be obtained as

$$[\hat{V}_0 - z_{\alpha/2}\hat{\sigma}_0, \qquad \hat{V}_0 + z_{\alpha/2}\hat{\sigma}_0].$$

In the ordinary kriging model, Eq. 3.8, where an SE covariance function is used, there are four parameters, $\{\beta_0, \sigma_\delta^2, \theta, \sigma_\varepsilon^2\}$. These parameters can be estimated by maximizing a log-likelihood function, which is the density function in Eq. 3.11. Specifically, the log-likelihood function reads

$$\begin{aligned} \mathcal{L}(\mathbf{V}|\beta_0, \sigma_\delta^2, \theta, \sigma_\varepsilon^2) = -\frac{1}{2}(\mathbf{V} - \beta_0 \cdot \mathbf{1})^T \left(\sigma_\varepsilon^2\mathbf{I} + \mathbf{C}_{NN}\right)^{-1} (\mathbf{V} - \beta_0 \cdot \mathbf{1}) \\ - \frac{1}{2}\log\left|\sigma_\varepsilon^2\mathbf{I} + \mathbf{C}_{NN}\right| - \frac{N}{2}\log(2\pi). \end{aligned} \tag{3.16}$$

Alternatively, one can first estimate β_0 by using the average of $\{V_i\}_{i=1}^N$ and then center the raw wind speed data by subtracting its average. After that, one can use the centered wind speed data and the maximum likelihood estimation to estimate the remaining three parameters (replace β_0 by $\bar{V}$ in Eq. 3.16).

Conceptually, the universal kriging is not much different from the ordinary kriging. The main extension is to make the mean component in Eq. 3.8 and Eq. 3.11 more flexible. In the ordinary kriging, the mean component is assumed a constant, β_0. In the universal kriging, the mean component is assumed as a polynomial model, $\beta_0 + \mathbf{g}^T(\mathbf{s})\boldsymbol{\beta}$, so that the universal kriging model can be expressed as

$$V(\mathbf{s}_i) = \beta_0 + \mathbf{g}^T(\mathbf{s}_i)\boldsymbol{\beta} + \delta(\mathbf{s}_i) + \varepsilon_i, \quad i = 1, ..., N, \tag{3.17}$$

where $\mathbf{g}(\cdot) = (g_1(\cdot), \ldots, g_q(\cdot))^T$ is a set of basis functions, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)^T$ is the coefficient vector, and q is the number of terms in the polynomial model in addition to the grand average, β_0. There are many different choices for the basis function $\mathbf{g}(\cdot)$ but it can be simply that $g_1(\mathbf{s}) = s_1$ and $g_2(\mathbf{s}) = s_2$ (in this case, $q = d = 2$). By expanding the mean component, the parameters in a universal kriging are $\{\beta_0, \beta_1, \ldots, \beta_q, \sigma_\delta^2, \theta, \sigma_\varepsilon^2\}$, and the number of parameters is $q + 4$, which is q more than that in the ordinary kriging. Nonetheless, the maximum likelihood estimation can still be used to estimate all the parameters after adjusting the log-likelihood function properly.

Kriging can be assisted by using the `geoR` package in `R`. Using the `likfit` function to estimate the parameters, such as

```
para <- likfit(spatialdata, ini.cov.pars = c(1,1), nugget =
                            0.5).
```

Here, `spatialdata` is the wind spatial data object holding the $\{\mathbf{s}_i, V_i\}_{i=1}^N$ data pairs, `ini.cov.pars` provides the initial value for σ_δ^2 and θ, respectively, `nugget` provides the initial value for σ_ε^2. By default, the `likfit` uses the SE covariance function and estimates the parameters by the maximum likelihood estimation in an ordinary kriging. To make a prediction at $\mathbf{s}_0$, one can use

```
V0 <- krige.conv(spatialdata, locations = s0, krige =
          krige.control(obj.model = para)),
```

where `obj.model = para` in the `krige.control` function passes the parameters just estimated to the prediction function. The default setting is an ordinary kriging.

Fig. 3.2 presents an example of applying the ordinary kriging predictor to the wind speed data from ten turbines in the `Wind Spatial Dataset`. An ordinary kriging model is established based on the wind speed data collected in July at ten turbine sites. Then the kriging model is used to predict the wind speed at site #6 using the observed wind speed at the other nine sites for the month of August. Fig. 3.2, right panel, shows that the spatially predicted wind speed at site #6 closely matches the observed wind speed at the same site.

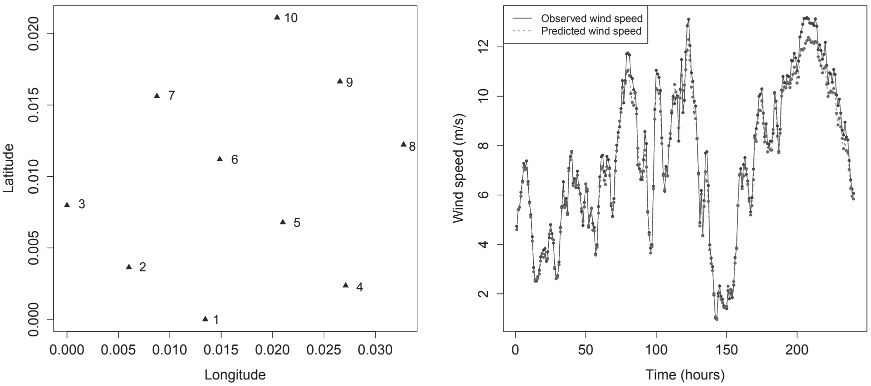

FIGURE 3.2 Left panel: the layout of the ten turbines. Right panel: the predicted and observed wind speeds of the first ten days in August at site #6.

3.2 SPATIO-TEMPORAL AUTOREGRESSIVE MODELS

The previous section considers purely the spatial correlation. This section presents a method that combines the spatial model feature and time series model feature in a method known as the Gaussian spatio-temporal autoregressive model (GSTAR) [166].

3.2.1 Gaussian Spatio-temporal Autoregressive Model

The wind speed in GSTAR model is assumed to follow a truncated normal distribution, a distribution choice as popular as Weibull used for modeling wind speed [75]. For notational simplicity, the site notation, $\mathbf{s}_i$, is shortened as site i, and consequently, $V(\mathbf{s}_i;t)$ is simplified to $V_i(t)$. To handle the wind speed nonstationarity over time, the time in a day is split into a number of *epochs*, during which the wind speed is assumed stationary [88]. For instance, 6 a.m. to 12 p.m. in a day can be treated as one epoch. With these notations, Pourhabib et al. [166] express $V_i(t) \sim \mathcal{N}^+(\mu_i(e_t), \sigma_i^2(e_t))$, where $i = 1, \ldots, N$, and e_t denotes the "epoch" at time t.

The GSTAR model assumes that the mean of wind speed at site i is a function of the past wind speeds at not only the target site but also other sites in its neighborhood, such that

$$\mu_i(e_t) = \beta_0 + \sum_{\ell=1}^{p} \sum_{j \in J_i} a_{ij\ell} V_j(t-\ell), \qquad \text{for} \quad i = 1, 2, \ldots, N, \tag{3.18}$$

where β_0 is an unknown constant, p is the autoregressive model order, $J_i \subset$

$\{1, 2, \ldots, N\}$ denotes the set of neighborhood sites whose wind speeds have a strong enough correlation with the wind speed at the target site i, and $a_{ij\ell}$ are the coefficients that quantify the spatio-temporal dependency. Note that Eq. 3.18 is a model for the expectation, so that the zero-mean, i.i.d noise term, ε, disappears.

GSTAR relies on one important assumption, which is to assume the spatio-temporal parameters, $a_{ij\ell}$, can be factorized into the respective spatial and temporal parts, such that

$$a_{ij\ell} = a^s_{ij} a^t_{i\ell} \qquad \text{for} \quad i = 1, 2, \ldots, N, \quad j \in J_i, \quad \ell = 1, 2, \ldots, p, \tag{3.19}$$

and GSTAR models the spatial part, a^s_{ij}, and the temporal part, $a^t_{i\ell}$, individually. GSTAR models its spatial dependency coefficient, a^s_{ij}, through a Gaussian kernel,

$$a^s_{ij} = \exp\left\{ -\left(\mathbf{s}_i - \mathbf{s}_j\right)^T \mathbf{\Lambda}_i \left(\mathbf{s}_i - \mathbf{s}_j\right)\right\}, \quad i = 1, 2, \ldots, N, \quad j \in J_i, \tag{3.20}$$

where $\mathbf{\Lambda}_i = \text{diag}\{\lambda_{i1}, \lambda_{i2}\}$, and λ_{i1} and λ_{i2} characterize the spatial decay in the longitudinal and latitudinal directions, respectively. Differing from that in Eq. 2.48, the Gaussian kernel in Eq. 3.20 has different scale parameters along the two spatial directions, whereas Eq. 2.48 has a single scale parameter ϕ for all directions. In this sense, this Gaussian kernel is the counterpart of the $C_{\text{SE-ARD}}$ covariance function in Eq. 3.7, whereas Eq. 2.48 is the counterpart of the C_{SE} covariance function in Eq. 3.6.

GSTAR models its temporal dependency, $a^t_{i\ell}$, through an exponential decay in terms of time distance, such as

$$a^t_{i\ell} = \exp\left\{-\lambda_{i3}\ell\right\}, \quad i = 1, 2, \ldots, N, \quad \ell = 1, \ldots, p \tag{3.21}$$

where λ_{i3} characterizes the temporal decay. Using Eq. 3.19–3.21, the otherwise large number of spatio-temporal parameters for site i is reduced to the three parameters, λ_{i1}, λ_{i2}, and λ_{i3}.

Let $\mathbf{A}_i$ denote an $N \times p$ matrix of spatial dependency for site i, of which the (j, ℓ)-th entry, $(\mathbf{A}_i)_{j\ell}$, is a^s_{ij}. Because a^s_{ij} does not have the ℓ index, all the entries are the same for the j-th row. For instance, the elements in the first row are all a^s_{i1}. If $j \notin J_i$, the corresponding row of $\mathbf{A}_i$ is entirely zero. Let $\mathbf{D}_i$ denote a $p \times p$ diagonal matrix whose (ℓ, ℓ)-th entry is $a^t_{i\ell}$. Let $\mathbf{V}_i(t) = (V_i(t-1), \ldots, V_i(t-p))^T$ be the time series data vector at site i, and $\mathcal{V}(t)$ be the $N \times p$ time series data matrix for all sites, namely

$$\mathcal{V}(t) = \begin{pmatrix} \mathbf{V}^T_1(t) \\ \mathbf{V}^T_2(t) \\ \vdots \\ \mathbf{V}^T_N(t) \end{pmatrix}_{N \times p}. \tag{3.22}$$

With the above notations, Eq. 3.18 can be expressed in a matrix form as,

$$\mu_i(e_t) = \beta_0 + \text{tr}\left(\mathbf{A}_i \mathbf{D}_i \mathcal{V}^T(t)\right), \qquad i = 1, 2, \ldots, N. \tag{3.23}$$

This model is referred to as the GSTAR of order p, or, simply GSTAR(p).

To estimate the parameters in Eq. 3.23, GSTAR uses a regularized least-squares estimation procedure as,

$$\min_{\lambda_{i1},\lambda_{i2},\lambda_{i3}} \sum_{\ell=1}^{n} \Big\{ L\left[V_i(\ell+h) - \bar{V}_i,\, \mathrm{tr}\left(\mathbf{A}_i \mathbf{D}_i \mathcal{V}^T(\ell)\right)\right] + \gamma \mathrm{Pen}\left(\mathbf{A}_i\right) \Big\}, \tag{3.24}$$

where h is the look-ahead time at which the GSTAR model is trained for making a forecast, n is the number of time stamps in the training set, $\bar{V}_i = \frac{1}{n}\sum_{\ell=1}^{n} V_i(\ell)$, $L[\cdot,\cdot]$ is a loss function (see Section 2.6 for various choices), γ is the penalty coefficient, and $\mathrm{Pen}\left(\mathbf{A}_i\right)$ is the penalty term that controls the size of the neighborhood, to be discussed in the next section. This optimization problem is solved using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [64], which belongs to the class of quasi-Newton methods.

Following the approach in [75], GSTAR models the standard deviation of wind speed as a linear combination of volatility, which measures the magnitude of recent changes in wind speed, such as,

$$\hat{\sigma}_i(e_t) = b_0 + b_1 \hat{\nu}_i(t), \qquad i = 1, 2, \ldots, N, \tag{3.25}$$

where

$$\hat{\nu}_i(t) = \left[\frac{1}{2|J_i|} \sum_{j \in J_i} \sum_{\ell=0}^{1} \Big\{ \left(V_j(t-\ell) - V_j(t-\ell-1)\right)^2 \Big\} \right]^{\frac{1}{2}}, \tag{3.26}$$

and $|J_i|$ is the number of elements in J_i. In the above equation, only the immediately past two moving range values, i.e., the difference between wind speed at t and at $t-1$ and that between wind speed at $t-1$ and at $t-2$, are used to estimate the volatility, $\hat{\nu}$. The two coefficients, b_0 and b_1, can be estimated by regressing the sample standard deviation in the left-hand side of Eq. 3.25 on $\hat{\nu}_i(t)$.

GSTAR makes an h-step ahead forecast at site i based on the α-quantile of the truncated normal distribution, such as

$$\hat{V}_i(t+h) = \hat{\mu}_i(t+h) + \hat{\sigma}_i(t+h) \cdot \Phi^{-1}\left[\alpha + (1-\alpha)\Phi\left(-\frac{\hat{\mu}_i(t+h)}{\hat{\sigma}_i(t+h)}\right)\right], \tag{3.27}$$

where $\Phi(\cdot)$ is the cdf of the standard normal distribution, $\hat{\mu}_i(\cdot)$ is the estimated mean found through Eq. 3.23, in which $t+h$ denotes a forecasting time that falls in the epoch e_t, and $\hat{\sigma}_i(\cdot)$ is the estimated standard deviation, decided through Eq. 3.25. The value of α should be decided based on the choice of the loss function. Using MAE or RMSE, $\alpha = 0.5$. Using PCE, α should be chosen consistently with ξ in Eq. 2.58.

3.2.2 Informative Neighborhood

GSTAR only uses the sites within a neighborhood to make forecasts at the target site. This neighborhood of site i, denoted by J_i, is much smaller than the whole wind farm. The rationale of this treatment is that not every single site on the farm has strong enough correlation with the target site to provide meaningful information and hence facilitate forecasting. The use of the Gaussian kernel essentially means that when the distance grows to a certain extent, the turbine sites lying beyond would have very little impact. For this reason, this neighborhood is referred to as an *informative neighborhood* for the purpose of forecasting. An obvious benefit of using an informative neighborhood is the reduced computational burden in the solution procedure.

Unlike the traditional wisdom that uses a time-invariant distance-based criterion [88, 128], leading to a disc-like neighborhood with a fixed radius, GSTAR uses the correlation among the rate of change in wind speed to determine the spatial dependency. Pourhabib et al. [166] discover through their analysis that two locations are informative to each other if the two sites have similar rates of change in wind speed for a given period, which explains why a pure distance-based criterion alone could be ineffective. Employing this criterion to find the informative neighborhood is done by designing a special penalty term in Eq. 3.24.

Denote by $Z_i(t) = dV_i'(t)/dt \approx V_i'(t) - V_i'(t-1)$ the first derivative of wind speed (the change rate), where $V_i' = V_i/\max\{V_i(t)\}$ is the wind speed normalized by the maximum wind speed for the whole farm during the training period. Then, compute the $N \times N$ sample covariance matrix for $\mathbf{Z}(\ell) = [Z_1(\ell), Z_2(\ell), \ldots, Z_N(\ell)]^T$ as,

$$\mathbf{C}_Z = \frac{1}{n}\sum_{\ell=1}^{n}\left(\mathbf{Z}(\ell) - \bar{\mathbf{Z}}\right)\left(\mathbf{Z}(\ell) - \bar{\mathbf{Z}}\right)^T, \tag{3.28}$$

where $\bar{\mathbf{Z}} = \sum_{\ell=1}^{n}\mathbf{Z}(\ell)/n$. To create the penalty term, Pen $(\mathbf{A}_i)$, it goes through three steps of action:

(a) Set an entry in $\mathbf{C}_Z$ to zero if its value is smaller than a prescribed threshold $\kappa \in [0, 1]$;

(b) Create a new matrix whose entries are the element-wise inverse of the entries of the matrix obtained in step (a) (with the convention that the inverse of zero is ∞); and

(c) Calculate the Frobenius norm of the product between the matrix obtained after step (b) and $\mathbf{A}_i$ in Eq. 3.24, with the convention that $0 \times \infty = 0$.

The specific mathematical steps are as follows. Let $\mathbf{C}_Z^{\kappa}$ denote the matrix after $\mathbf{C}_Z$ is truncated using κ, i.e.,

$$C_{Z,jk}^{\kappa} = C_{Z,jk} \qquad \text{if} \quad C_{Z,jk} \geq \kappa, \quad \text{otherwise} \quad C_{Z,jk}^{\kappa} = 0, \tag{3.29}$$

where $C^{\kappa}_{Z,jk}$ and $C_{Z,jk}$ are the (j,k)-th entry of $\mathbf{C}^{\kappa}_{Z}$ and $\mathbf{C}_{Z}$, respectively. Then, let $(\mathbf{C}^{\kappa}_{Z})^{-}$ denote the entry-wise inverse of $\mathbf{C}^{\kappa}_{Z}$. As such, the penalty term is defined as,

$$\text{Pen}\,(\mathbf{A}_i) = \|\mathbf{A}_i^T(\mathbf{C}^{\kappa}_{Z})^{-}\|_F, \tag{3.30}$$

where $\|\cdot\|_F$ represents the Frobenius norm and, inside $\text{Pen}\,(\mathbf{A}_i)$, one uses the notational convention that $0 \times \infty = 0$.

What this penalty term does is to associate each spatial dependency term, a^s_{ij}, with the inverse of a $C^{\kappa}_{Z,jk}$, in a fashion that can be loosely expressed as $a^s_{ij}/C^{\kappa}_{Z,jk}$. To reduce the cost resulting from the penalty term, one apparently wants to keep $a^s_{ij}/C^{\kappa}_{Z,jk}$ as small as possible. If $C^{\kappa}_{Z,jk} = 0$, meaning that the sample covariance of the first derivative of wind speed is smaller than the threshold, κ, then the corresponding a^s_{ij} is forced to zero. If $C^{\kappa}_{Z,jk}$ is not zero but small, indicating a weak correlation between the two first derivatives, then the corresponding a^s_{ij} is penalized more, whereas if $C^{\kappa}_{Z,jk}$ is large, indicating a strong correlation, then a^s_{ij} is penalized less. The informative neighborhood $J_i = \{j : C^{\kappa}_{Z,ij} \neq 0\}$ is then selected through this penalizing scheme.

Fig. 3.3 presents an example of the informative neighborhoods selected for three different target sites. Note that informative neighborhoods are irregularly shaped, rather than disc-like, and they are different when the target site is at a different location. The shape and size of the informative neighborhoods are time varying, and they will be updated through the learning process as the new wind data arrives. This informative neighborhood concept and method is more flexible and versatile in terms of capturing the spatial relevance.

Concerning the choice for the threshold, κ, the general understanding is that a smaller κ leads to a larger neighborhood, because it causes $\mathbf{C}^{\kappa}_{Z}$ to have fewer zero entries, whereas a large κ creates a smaller informative neighborhood, because the resulting $\mathbf{C}^{\kappa}_{Z}$ has more zero entries. Here GSTAR sets the κ value at 0.85 for all forecast horizons. Analysts can certainly conduct fine-scale adjustments by, say, setting a lower and an upper threshold for the size of an informative neighborhood. If the number of turbines in the neighborhood is below the lower threshold, the κ value is to be reduced, which in turn makes the neighborhood bigger to accommodate more turbines. If the number of turbines is above the upper threshold, then the κ is to be increased, to make the neighborhood smaller. In the numerical analysis in Section 3.2.3, the lower and upper bounds are set as 2 and 15, respectively.

3.2.3 Forecasting and Comparison

This section applies the GSTAR method to the `Wind Spatial-Temporal Dataset1`. In this application, GSTAR defines four epochs for each day in a calendar month: (1) 12:00 am to 6:00 am, (2) 6:00 am to 12:00 pm, (3) 12:00 pm to 6:00 pm, and (4) 6:00 pm to 12:00 am. Consequently, an individual GSTAR model for each epoch is fit, which is used to make forecasts for the horizon belonging to the same epoch. Each GSTAR model is trained using one month of data and then makes h-hour ahead forecasts for $h = 2, 3, 4, \text{and}\, 5$.

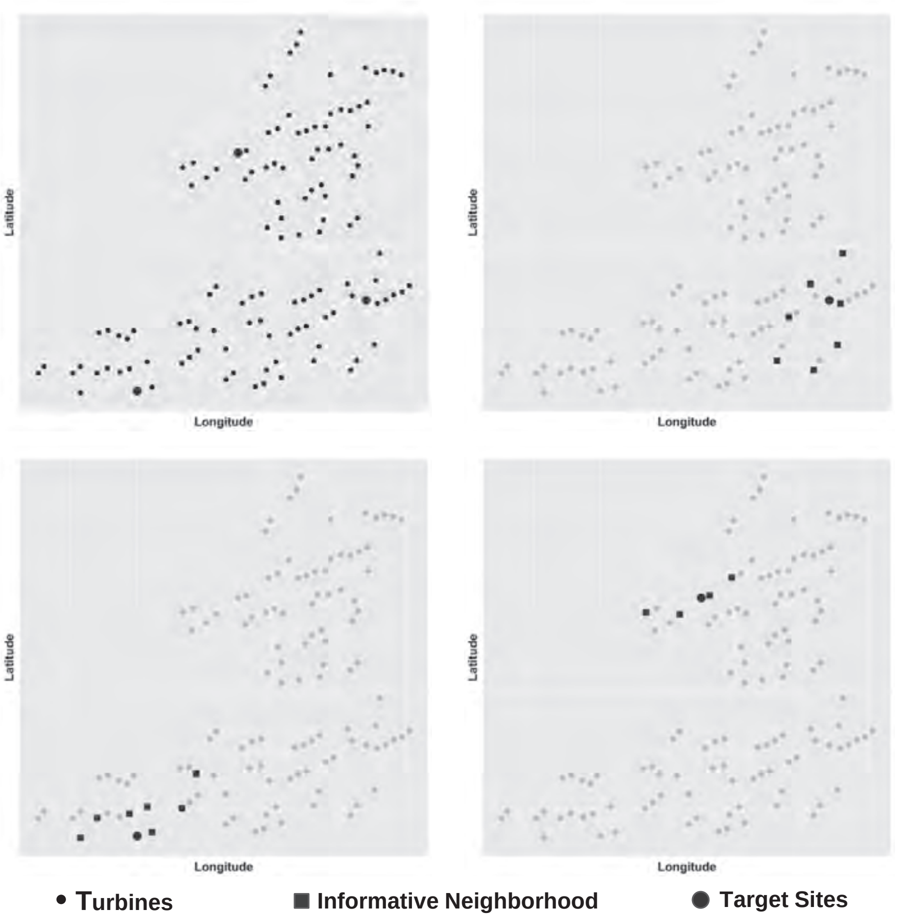

FIGURE 3.3 Neighborhoods selected by GSTAR based on one month of data in the `Wind Spatio-Temporal Dataset1`. Top-left: three turbine sites and the surrounding turbines; top-right, bottom-left and bottom-right: informative neighborhood selected for each site.

Pourhabib et al. [166] choose the PCE loss function as $L[\cdot,\cdot]$ in Eq. 3.24. When using PCE, a power curve function is needed as $g(\cdot)$ in Eq. 2.58. Using a nonlinear power curve function complicates the optimization in Eq. 3.24. Because of that, Pourhabib et al. simplify the power curve function to a piecewise linear function, such that

$$g(V) = \begin{cases} 0, & V \leq 3.5; \\ 0.1053(V - 3.5), & 3.5 < V \leq 13; \\ 1, & 13 < V. \end{cases}$$

This piecewise linear power curve function does not differ that much from the nonlinear power curve function. The ξ parameter used in PCE is set to 0.73. To ease the computational burden to go through the sizeable combinations of turbines, months, and epochs, each of the $N = 120$ turbine cases is randomly assigned to evaluate one of the epochs for a given month. The forecast error for a given month, averaged over roughly 30 evaluation cases, is then reported.

The competing models used in this comparison are ARMA(p, q), ARMA*(p, q), and the persistence model. ARMA*(p, q) is the same as ARMA(p, q), except that the analysis is performed on the residuals after removing a diurnal trend using Eq. 2.15. As seen in Chapter 2, a small time lag usually suffices to capture the temporal dependency. For the datasets used in this section, the partial autocorrelation of lag 1 is dominant, suggesting $p = 1$. Using BIC would select $p = 1$ and $q = 2$ for most of the cases. So the model order in the ARMA model is set as $p = 1$ and $q = 2$. When evaluating the ARMA models, another random sampling is applied to the 30 evaluation cases mentioned above, further reducing the number of runs to about 25% of what is used for GSTAR.

Table 3.1 presents the forecasting results of GSTAR and the comparison with the two versions of ARMA models and the persistence model. GSTAR, on average, outperforms the other three methods, indicating the benefit of incorporating the spatial dependency information. Interestingly, in this comparison, the persistence model wins over the ARMA models.

Table 3.2 shows some results using CRPS to give a sense of the quality of predictive distribution. Forty turbines are randomly chosen, to which the GSTAR and ARMA(1,2) are applied. Please note that here CRPS is computed for power response, meaning that the integration is conducted over y; please refer to Eq. 5.23.

In practice, the optimal value of ξ used in PCE may change over time and a variation of ξ around 0.73 can be expected. A sensitivity analysis is conducted, which is to change ξ between 0.6 and 0.8, and then average the PCE over this range. One hundred turbines are randomly chosen and the 2009 data are used in this analysis. Table 3.3 shows that the performance of the GSTAR model is reasonably robust when ξ is around 0.73.

TABLE 3.1 Forecasting results for 2009 and 2010 using PCE. The values in parentheses are the standard deviations of the corresponding forecasting. The row of "Imp. over PER" shows the improvement of GSTAR over PER in percentage.

	2-hour	**3-hour**	**4-hour**	**5-hour**
		2009		
PER	0.0614(0.0159)	0.0741(0.0184)	0.0857(0.0215)	0.0943(0.0212)
ARMA(1,2)	0.0663(0.0375)	0.0826(0.0386)	0.0844(0.0473)	0.0991(0.0463)
ARMA*(1,2)	0.0752(0.0366)	0.0917(0.0421)	0.1002(0.0485)	0.1038(0.0486)
GSTAR(1)	0.0608(0.0297)	0.0716(0.0318)	0.0816(0.0327)	0.0884(0.0321)
Imp. over PER	1.1%	3.3%	4.8%	6.3%
		2010		
PER	0.0484(0.0137)	0.0572(0.0160)	0.0644(0.0185)	0.0698(0.0208)
ARMA(1,2)	0.0650(0.0398)	0.0779(0.0437)	0.0783(0.0394)	0.0794(0.0400)
ARMA*(1,2)	0.0690(0.0386)	0.0823(0.0418)	0.0838(0.0460)	0.0857(0.0380)
GSTAR(1)	0.0477(0.0212)	0.0569(0.0231)	0.0630(0.0260)	0.0692(0.0277)
Imp. over PER	1.5%	0.5%	2.1%	0.8%

TABLE 3.2 CRPS values using forty randomly selected turbines and 2009 data.

	2-hour	**3-hour**	**4-hour**	**5-hour**
ARMA(1,2)	0.1538	0.1452	0.1496	0.1559
GSTAR	0.1243	0.1299	0.1378	0.1467

TABLE 3.3 Average PCE while ξ varying in $[0.6, 0.8]$ for 100 turbines using the data of 2009. The values in parentheses are the standard deviations.

	2-hour	**3-hour**	**4-hour**	**5-hour**
PER	0.0616(0.0122)	0.0731(0.0220)	0.0855(0.0327)	0.0937(0.0286)
GSTAR	0.0628(0.0235)	0.0723(0.0332)	0.0835(0.0364)	0.0900(0.0357)

3.3 SPATIO-TEMPORAL ASYMMETRY AND SEPARABILITY

3.3.1 Definition and Quantification

One of the key assumptions made in the GSTAR model is that the spatio-temporal dependency structure, $a_{ij\ell}$, can be expressed as the product of a spatial part and a temporal part; please refer to Eq. 3.19. Generally, a covariance structure is said to be *separable* if its covariance function can be factored into the product of purely spatial and purely temporal components such that $C(\mathbf{u}, h) = C^s(\mathbf{u}) \cdot C^t(h)$. This assumption of spatio-temporal separability is in fact rather common in spatio-temporal analysis [43] because separable spatio-temporal models are easier to be dealt with mathematically.

Assuming separability suggests the lack of interaction between the spatial and temporal components and implies full symmetry in the spatio-temporal covariance structure, which brings up the concept of *spatio-temporal symmetry*. A covariance structure is symmetric if

$$C(\mathbf{s}_1, \mathbf{s}_2; t_1, t_2) = C(\mathbf{s}_1, \mathbf{s}_2; t_2, t_1). \tag{3.31}$$

This is to say that the correlation between sites $\mathbf{s}_1$ and $\mathbf{s}_2$ at times t_1 and t_2 is the same as that between $\mathbf{s}_1$ and $\mathbf{s}_2$ at times t_2 and t_1. For a stationary covariance function, this can be written as $C(\mathbf{u}, h) = C(-\mathbf{u}, h) = C(\mathbf{u}, -h) = C(-\mathbf{u}, -h)$ [72]. Separability is a stronger condition. It can be shown that a separable spatio-temporal covariance structure must have symmetry but the converse is not necessarily true, meaning that a symmetric covariance structure may or may not be separable [74].

To quantify asymmetry, Stein [204] proposes a metric in terms of spatio-temporal semi-variograms. The spatio-temporal empirical semi-variogram of $V_i(t)$ between site $\mathbf{s}_1$ and site $\mathbf{s}_2$ at time lag h is defined as,

$$\varpi(\mathbf{s}_1, \mathbf{s}_2; h) = \frac{1}{2(n-h-1)} \sum_{j=1}^{n-h-1} [V_1(t_j + h) - V_2(t_j)]^2. \tag{3.32}$$

Then, introduce two semi-variograms between $\mathbf{s}_1$ and $\mathbf{s}_2$: $\varpi(\mathbf{s}_1, \mathbf{s}_2, h)$ and $\varpi(\mathbf{s}_2, \mathbf{s}_1, h)$. Both of them represent the dissimilarity between the two spatial sites, but $\varpi(\mathbf{s}_1, \mathbf{s}_2, h)$ means that measurements taken at $\mathbf{s}_2$ are h time lag behind that at $\mathbf{s}_1$, whereas $\varpi(\mathbf{s}_2, \mathbf{s}_1, h)$ means that measurements at $\mathbf{s}_1$ are behind those at $\mathbf{s}_2$. A quantitative asymmetry metric can be thus defined as the difference between the two semi-variograms, namely

$$asym(\mathbf{s}_1, \mathbf{s}_2, h) := \varpi(\mathbf{s}_1, \mathbf{s}_2, h) - \varpi(\mathbf{s}_2, \mathbf{s}_1, h). \tag{3.33}$$

When the two semi-variograms are the same, the wind field is said to be symmetric. But when there is a dominant wind blowing from $\mathbf{s}_1$ towards $\mathbf{s}_2$, the propagation of wind from $\mathbf{s}_1$ towards $\mathbf{s}_2$ would generate a significantly positive value for $asym$, indicating a lack of symmetry. To signify the dominant wind direction, denoted by ϑ, the asymmetric metric is also expressed as $asym(\mathbf{s}_1, \mathbf{s}_2, h, \vartheta)$.

3.3.2 Asymmetry of Local Wind Field

The space-time symmetry assumption is not universally valid, and it is especially not true in many geophysical processes, such as wind fields, in which the prevailing air flow, if existing, causes the correlation in space and time stronger in one direction than other directions, thus invalidating the symmetry assumption. To see this, let us look at Fig. 3.4, left panel. Consider two sites and a wind flow primarily from $\mathbf{s}_1$ to $\mathbf{s}_2$. Let $t_1 = t$ and $t_2 = t{+}k$, $k > 0$. Were the assumption of symmetry true, it meant that $C(\mathbf{s}_1, \mathbf{s}_2; t, t{+}k) = C(\mathbf{s}_1, \mathbf{s}_2; t{+}k, t)$. The left-hand side covariance, $C(\mathbf{s}_1, \mathbf{s}_2; t, t + k)$, dictates how much information at $\mathbf{s}_1$ and t is there to help make predictions at a down-wind site $\mathbf{s}_2$ and a future time $t + k$. A significant $C(\mathbf{s}_1, \mathbf{s}_2; t, t + k)$ suggests that the upstream wind measurements at t help with the downstream wind prediction at $t + k$. This makes perfect sense, considering that wind goes from $\mathbf{s}_1$ to $\mathbf{s}_2$. The assumption of symmetry, were it true, says that the right-hand side covariance, $C(\mathbf{s}_1, \mathbf{s}_2; t + k, t)$, is equally significant, meaning that the downstream wind measurements at t could help with the upstream prediction at $t + k$, as much as the upstream helps the downstream. This no longer makes sense.

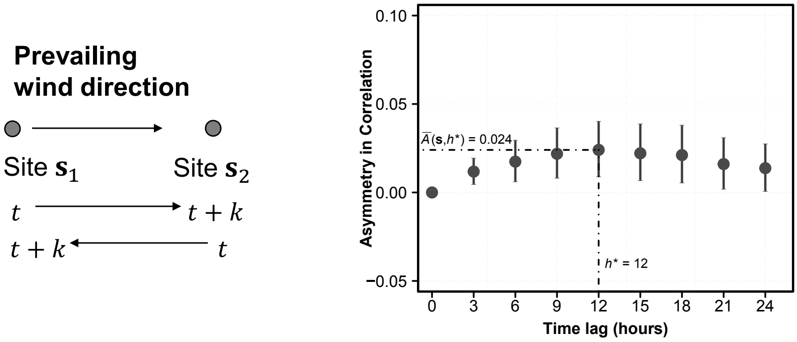

FIGURE 3.4 Under a dominant air flow, the covariance structure of the underlying wind field may become asymmetric. (Right panel reprinted with permission from Ezzat et al. [59].)

While studying large-scale atmospheric processes, analysts have in fact noted that when there exists a dominant air or water flow in the processes, the resulting random field does not have a symmetric covariance structure [42, 72, 114, 204, 225]. The question is—does this lack of symmetry phenomenon also take place on a small-scale wind field as compact as a wind farm?

Ezzat et al. [59] set out to investigate this question for the wind field on a farm. In their analysis, the diurnal trend for wind speed is first fitted using Eq. 2.15 to remove nonstationarity in the wind data. The fitted trend is then subtracted from the actual wind speed data and the residuals are subsequently used for quantifying asymmetry. Using the `Wind Spatio-Temporal Dataset2`, the yearly average wind direction is estimated as $\bar{\vartheta} = 264.24°$ (due west is 270°). Because of this, for every pair of turbines i and j such that $\mathbf{s}_i$ is west of $\mathbf{s}_j$, Ezzat et al. compute $\varpi(\mathbf{s}_i, \mathbf{s}_j, h) - \varpi(\mathbf{s}_j, \mathbf{s}_i, h)$ using the residuals in place of V in Eq. 3.32. This computation is repeated for every pair of turbines and for different time lags ranging from 0 to 24 hours. All of the computed quantities are then transformed into the correlation scale. For the ℓ-th pair of turbines, the resulting quantity at each temporal lag h is the spatio-temporal asymmetry, $asym^{\ell}(\mathbf{s}_i, \mathbf{s}_j, h, \bar{\vartheta})$.

Denote the collection of asymmetry values at each temporal lag by $A(\mathbf{s}, h) = \{asym^{\ell}(\mathbf{s}_i, \mathbf{s}_j, h, \bar{\vartheta})\}_{\ell=1}^{\mathfrak{L}}$, where $\mathfrak{L}$ is the total number of turbine pairs. Represent by $\bar{A}(\mathbf{s}, h)$ the 50-th percentile of this collection. Fig. 3.4, right panel, presents the 25-th, 50-th and 75-th percentiles of $A(\mathbf{s}, h)$ for $h \in \{0, \ldots, 24\}$ with a three-hour increment. On the one hand, all median asymmetry values in Fig. 3.4, right panel, are slightly positive, indicating a potential tendency towards spatio-temporal asymmetry. On the other hand, the largest median occurs at $h^* = 12$ and is approximately 0.024 on the correlation scale. To put this value in perspective, please note that Gneiting [72] reports a value of 0.12 for asymmetric large-scale wind flow over Ireland. The values of asymmetry reported in [74] range between 0.04 and 0.14, and are averaged at 0.11. Relative to those levels, an asymmetry of 0.024 appears to be rather weak to justify the existence of asymmetry in the local wind field. Analysts would understandably trade such weak asymmetry for computational efficiency and model simplicity gained by making the symmetry assumption. This may explain why separable, symmetric models are dominant in the wind application literature.

On the surface, the above analysis appears to indicate that there does not exist significant asymmetry in a local wind field within an area as compact as a wind farm. Ezzat et al. [59] believe that the weak asymmetry is due to the non-optimal handling of wind farm data, especially in terms of its temporal handling. When producing the right panel of Fig 3.4, the wind data is grouped for the whole year. Ezzat et al. test different temporal resolutions like monthly or weekly. Under the finer temporal resolutions, the asymmetric level indeed increases but still not much. Ezzat et al. hypothesize that a special spatio-temporal "lens" is needed to observe the wind data in order to detect strong degrees of asymmetry in a local wind field. This makes intuitive sense. In a large-scale atmospheric process, a dominant wind can persist for a sustained period of time and travel a substantial distance. These patterns can be pre-identified through climatological expertise over a region of interest, and as such, regular calendar decompositions, like weekly, monthly, seasonal, or yearly, appear to be reasonable choices. For a local wind field, however, observational data suggest that alternations in local winds occur at a relatively high rate, resulting in several distinct wind characteristics at each wind alternation. In such settings, regular calendar periods rarely contain a single dominant wind scenario. Rather, they contain various dominant winds that create multiple asymmetries having distinct directions and magnitudes. Consequently, aggregating the heterogeneous, and perhaps opposite, asymmetries leads to an underestimation of the true asymmetry level.

3.3.3 Asymmetry Quantification

The physical differences between local wind fields and large-scale atmospheric processes require special adjustments to the spatio-temporal resolution used to analyze wind measurements, in order to reveal the underlying asymmetry pattern. Ezzat et al. [59] devise a special lens consisting of two components—a temporal adjustment and a spatial adjustment.

The main reason that temporal aggregations based purely on calendar periods are not going to be effective is because such decomposition intervals are created arbitrarily. Hence, one key step for a successful temporal adjustment is to isolate the time intervals in which a unique dominant wind persists—such intervals are referred to as the *prevailing periods*, and detecting them is basically solving a change-point detection problem.

A binary segmentation version of the circular change-point detection [106] is used to detect the change points in wind direction. The `R` package `circular` is used to facilitate the task. The change-point detection method is applied to the wind direction data measured at one of the masts. Fig. 3.5 presents the detected change points for two weeks of the wind direction data, for the sake of illustration. For the whole year, a dominant wind direction lasts, on average, for 3.04 days with a standard deviation of 2.46 days. For 50% of the prevailing periods, the wind direction alternates in less than 2.27 days. The maximum interval of time in which a dominant wind direction is found to be persistent is 15.5 days, while the shortest length of a prevailing period is found to be 6 hours. These statistics indicate a fast dynamics and unpredictable nature in wind direction change, explaining why a typical calendar period-based approach is ineffective. A total of 119 change points are detected in the yearlong wind direction data, leading to 120 prevailing periods identified over the year. For the ℓ-th prevailing period, the dominant wind direction is denoted by ϑ_ℓ.

On the spatial level, the relative position of the turbines on a wind farm is another factor that affects the asymmetry level at a given time. Physically, asymmetry exists when wind propagates from an upstream turbine to a downstream one, implying that the latter is in the along-wind direction with respect to the former. Therefore, the spatial adjustment is to select only the along-wind turbines for asymmetry quantification.

A spatial bandwidth, denoted by b_ℓ, is to be selected for the ℓ-th prevailing period. The specific procedure is executed as follows: vary the bandwidth in the range $[2.5^\circ, 45^\circ]$ in increments of 2.5° and then select the bandwidth that maximizes the median asymmetry and denote that choice as the optimal bandwidth b^*_ℓ. With the spatial adjustment, the asymmetry metric, $asym(\cdot)$, is now denoted as $asym(\mathbf{s}_1, \mathbf{s}_2, h_\ell, \vartheta_\ell, b_\ell)$. Finally, an optimal time lag h^*_ℓ is

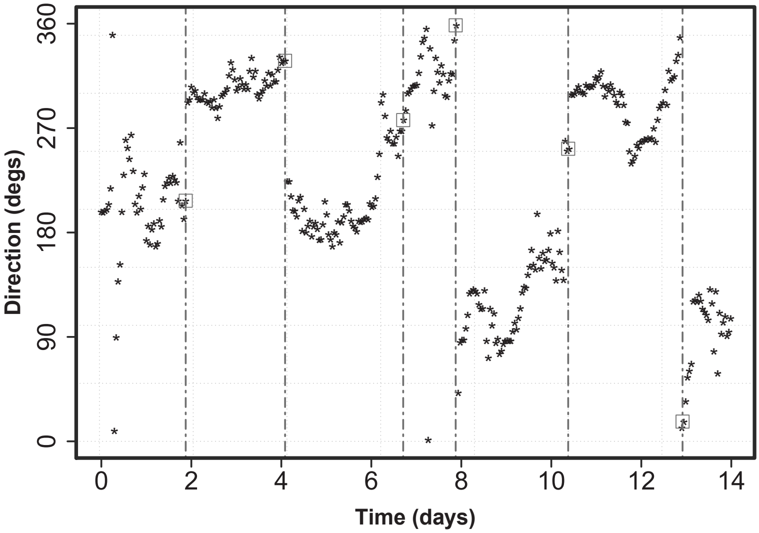

FIGURE 3.5 Change points detected in the first two weeks of wind direction data. The vertical dashed lines indicate the change points. (Reprinted with permission from Ezzat et al. [59].)

chosen to maximize the median asymmetry level in each prevailing period when spatial and temporal parameters are set at b_ℓ^* and ϑ_ℓ, respectively.

Under the parameter setting, $h_\ell^*, \vartheta_\ell, b_\ell^*$, the asymmetric metric $asym(\cdot)$ is computed using the `Wind Spatio-Temporal Dataset2`. Fig. 3.6 presents the 25-th, 50-th and 75-th percentiles of the asymmetry level versus the separating distance subgroups for the different scenarios thus considered: yearly, seasonal, monthly, weekly, temporal-only lens scenario, and spatio-temporal lens scenario. It is apparent that applying the spatio-temporal lens detects much higher asymmetry levels. For instance, at separating distances greater than 20 km, all of the turbine pairs exhibit positive asymmetry and 50% of them exhibit an asymmetry level higher than 0.2 on the correlation scale, a level considered significant in the past study [72] and nearly an order of magnitude greater than the median asymmetry of 0.024 detected earlier on the yearly data.

Table 3.4 classifies the median asymmetry values of all distance subgroups, where 93% of the prevailing periods exhibit positive median asymmetry, nearly a quarter of them exhibit a greater than 0.2 median asymmetry, and more than 41% of them exhibit a median asymmetry larger than 0.1, the level of asymmetry previously reported in [72, 74] for signaling the existence of ap-

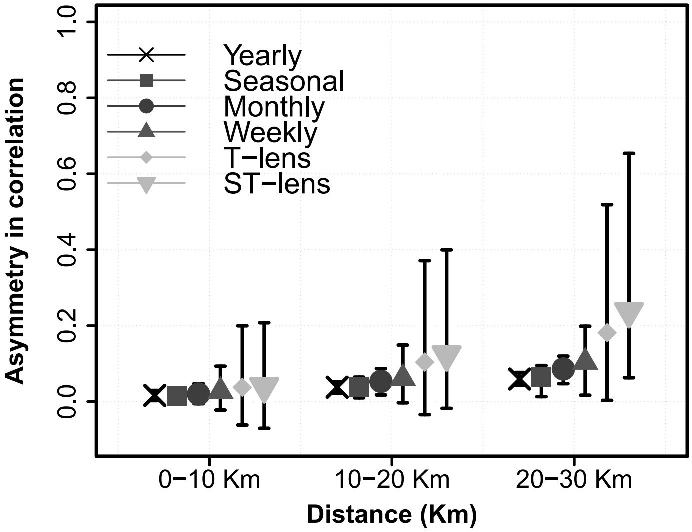

FIGURE 3.6 The 25-th, 50-th and 75-th percentiles of asymmetry values of different scenarios versus separating distance in kilometers. "T-lens" means temporal adjustment only, whereas "ST-lens" means spatio-temporal adjustments. (Reprinted with permission from Ezzat et al. [59].)

preciable asymmetric behavior in the large-scale atmospheric processes. The findings suggest that not only does strong asymmetry exist in local wind fields, but also the discovered asymmetry appears to fluctuate spatially and temporally in both magnitude and direction. Each prevailing period appears to have a unique asymmetry pattern, creating a temporal fluctuation of asymmetry throughout the year.

3.3.4 Asymmetry and Wake Effect

The implication of capturing the asymmetry in a local wind field can enrich the understanding of complex physical phenomena on a wind farm such as the wake effect. The spatio-temporal dynamics within a wind farm are affected by the wake effect because the rotating turbine blades cause changes in the speed, direction and turbulence intensity of the propagating wind [40]. For each prevailing period, Ezzat et al. [59] divide the whole farm, based on the wind direction, into two regions having approximately the same number of turbines. The first region is the set of wake-free wind turbines that receive

TABLE 3.4 Classification of prevailing periods according to the median asymmetry level.

Group	Range	Percentage
1.	$\bar{A}(\mathbf{s}_1, \mathbf{s}_2, h_\ell^*, \vartheta_\ell, b_\ell^*) \leq 0$	7%
2.	$0 < \bar{A}(\mathbf{s}_1, \mathbf{s}_2, h_\ell^*, \vartheta_\ell, b_\ell^*) < 0.05$	27%
3.	$0.05 \leq \bar{A}(\mathbf{s}_1, \mathbf{s}_2, h_\ell^*, \vartheta_\ell, b_\ell^*) < 0.1$	25%
4.	$0.1 \leq \bar{A}(\mathbf{s}_1, \mathbf{s}_2, h_\ell^*, \vartheta_\ell, b_\ell^*) < 0.2$	20%
5.	$0.2 \leq \bar{A}(\mathbf{s}_1, \mathbf{s}_2, h_\ell^*, \vartheta_\ell, b_\ell^*)$	21%

Source: Ezzat et al. [59]. With permission.

less turbulent wind, whereas the second region is the set of wind turbines which are in the wake of other turbines and receive the disturbed, turbulent wind. Fig. 3.7 plots the medians of the asymmetry for each region. The wake-free region appears to exhibit stronger asymmetry, which is consistent with the physical understanding since the less-turbulent wind is the driving force creating the asymmetry. This analysis indicates that the asymmetry level spatially varies on a wind farm due to the wake effect. Incorporating such patterns in a spatio-temporal model could benefit modeling and prediction, as well as aid research in wake characterization.

3.4 ASYMMETRIC SPATIO-TEMPORAL MODELS

3.4.1 Asymmetric Non-separable Spatio-temporal Model

Consider a simple spatio-temporal model, the counterpart of the ordinary kriging in Eq. 3.8, such as

$$V_i(\ell) = \beta_0 + \delta_i(\ell), \quad i = 1, \ldots, N, \text{ and } \ell = t, t-1, \ldots, t-n, \tag{3.34}$$

where β_0 is the unknown constant, like in Eq. 3.8. Unlike Eq. 3.8, which has two random terms, the i.i.d noise term ε is absorbed into the spatio-temporal random field term $\delta_i(\ell)$ here.

The key in spatio-temporal modeling, as mentioned at the beginning of this chapter, is to specify the covariance function for the spatio-temporal random field term, $\delta_i(\ell)$. The specific asymmetric non-separable spatio-temporal model presented here is a modified version of that proposed in [74], in which the asymmetric, non-separable covariance function is expressed as follows,

$$C_{\text{ASYM}}(\mathbf{u}, h) = \sigma_{\text{ST}}^2 \Big\{ (1-\varphi)\rho_{\text{NS}}(\mathbf{u}, h) + \varphi \rho_{\text{A}}(\mathbf{u}, h) \Big\} + \eta \mathbb{1}_{\{\|\mathbf{u}\|=|h|=0\}}, \tag{3.35}$$

where ρ_A is an asymmetric correlation function to be given below and ρ_{NS} is a non-separable symmetric correlation function such that

$$\rho_{\text{NS}}(\mathbf{u}, h) = \frac{1-\tau}{1+\zeta|h|^2} \left(\exp\left[-\frac{\phi\|\mathbf{u}\|}{(1+\zeta|h|^2)^{\frac{\beta}{2}}} \right] + \frac{\tau}{1-\tau} \mathbb{1}_{\{\|\mathbf{u}\|=0\}} \right). \tag{3.36}$$

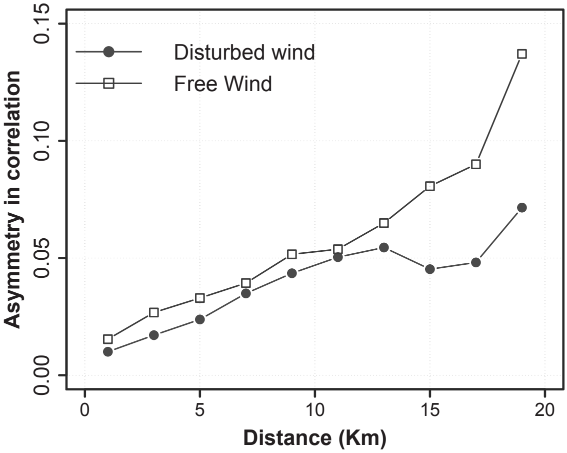

FIGURE 3.7 Wake effect and its implication on spatio-temporal asymmetry. (Reprinted with permission from Ezzat et al. [59].)

In Eq. 3.35 and Eq. 3.36,

- ζ and ϕ are, respectively, the temporal and spatial scale parameters,
- τ and η are, respectively, the spatial and spatio-temporal nugget effects (i.e., i.i.d random noise),
- σ^2_{ST} is the spatio-temporal variance,
- β is the non-separability parameter, characterizing the strength of the spatio-temporal interaction, and
- φ is the asymmetry parameter, characterizing the lack of symmetry.
- The valid ranges of these parameters are: $\tau \in [0, 1)$, $\beta \in [0, 1]$, $\varphi \in [0, 1]$, $\sigma^2_{\mathrm{ST}} > 0$, and ϕ, ζ and η are all non-negative.

The $\rho_{\mathrm{A}}(\cdot, \cdot)$ defined in [74] is a Lagrangian compactly supported function,

$$\rho_{\mathrm{A}}(\mathbf{u}, h) = \left(1 - \frac{1}{2\|\mathbf{U}\|}\|\mathbf{u} - \mathbf{U}h\|\right)_+, \tag{3.37}$$

where $\mathbf{U} = (U_1, U_2)^T$ is the two-dimensional velocity vector having a longitudinal component and a latitudinal component and to be defined based on the knowledge of the weather system. For example, if the dominant wind is known to be strictly westerly, then $\mathbf{U}$ is chosen to be $(U_1, 0)^T$, namely a non-zero longitudinal wind velocity reflecting the traveling of the wind along the longitudinal axis. A generalized version of ρ_{A} is proposed by Schlater in [195]. Instead of using a constant vector, Schlater defines $\mathbf{U}$ as a random variable that follows a multivariate normal distribution, i.e., $\mathbf{U} \sim \mathcal{N}(\boldsymbol{\mu}, \frac{\mathbf{D}}{2})$. As such, ρ_{A} is defined as,

$$\rho_{\text{A}}(\mathbf{u}, h) = \frac{1}{\sqrt{|\mathbf{1}_{2\times 2} + h^2\mathbf{D}|}} \exp\left\{-(\mathbf{u}-\boldsymbol{\mu}h)^T(\mathbf{1}_{2\times 2}+h^2\mathbf{D})^{-1}(\mathbf{u}-\boldsymbol{\mu}h)\right\}, \quad (3.38)$$

where $|\cdot|$ denotes the matrix determinant.

The asymmetric non-separable model used by Ezzat et al. [59] consists of the modeling components in Eq. 3.35, Eq. 3.36, and Eq. 3.38, and it is referred to hereinafter as ASYM.

3.4.2 Separable Spatio-temporal Models

By setting $\beta = \varphi = 0$ in ASYM, the asymmetric, non-separable model is reduced to a symmetric, separable model. Analysts could entertain two variants of the symmetric, separable model. The first variant is to take the parameters of ASYM after all of them are estimated but simply reset $\beta = \varphi = 0$. The second variant is to first set $\beta = \varphi = 0$ before parameter estimation and then freely estimate the remaining parameters from the data. Understandably, the second variant generally works better and is what is used in Section 3.5. This symmetric, separable model is referred to as SEP.

3.4.3 Forecasting Using Spatio-temporal Model

The short-term wind forecasting may benefit from using an asymmetric, separable spatio-temporal covariance structure. Once the covariance function is specified, the forecasting is conducted similarly as in the kriging method of Section 3.1.3.

Let us arrange the spatio-temporal wind speed, $V_i(t)$, into an $Nn\times 1$ vector, such as

$$\mathbf{V} = (V_1(t), \cdots, V_N(t), V_1(t-1), \cdots, V_N(t-1), \cdots, V_1(t-n), \cdots, V_N(t-n))^T.$$

The objective is to make a forecast at site $\mathbf{s}_0$ and time $t+h$, denoted by $V_0(t+h)$, which is an h-hour ahead forecast at $\mathbf{s}_0$.

A covariance matrix corresponding to $\mathbf{V}$ can be constructed by using the covariance function C_{ASYM} and is hence denoted by $\mathbf{C}_{\text{ASYM}}$. A covariance row vector, $\mathbf{c}_0$, can be constructed by treating its i-th element $(\mathbf{c}_0)_i$ as the covariance between $V_0(t+h)$ with the i-th element in $\mathbf{V}$. The notation of $\mathbf{c}_0$ bears the same meaning as the notation of $\mathbf{c}_{0N}$ used earlier in Section 3.1.3. Here we drop the subscript "N" because the size of $\mathbf{V}$ for this spatio-temporal process is no longer $N \times 1$ but $Nn \times 1$. Denote by $\sigma_0^2 := C_{\text{ASYM}}(0,0)$ the prior variance of the underlying spatio-temporal process. Similar to the kriging forecasting in Eq. 3.15, the forecast of $V_0(t+h)$ can be obtained as

$$\begin{aligned} \hat{V}_0(t+h) &= \hat{\beta}_0 + \mathbf{c}_0 \mathbf{C}_{\text{ASYM}}^{-1}(\mathbf{V} - \hat{\beta}_0 \cdot \mathbf{1}), \\ Var(\hat{V}_0(t+h)) &= \sigma_0^2 - \mathbf{c}_0 \mathbf{C}_{\text{ASYM}}^{-1} \mathbf{c}_0^T. \end{aligned} \tag{3.39}$$

The flowchart in Fig. 3.8 presents the steps of the forecasting procedure. To perform an h-hour ahead forecast, only the data in the preceding prevailing period that share similar wind asymmetry characteristics are used for model training. This implies that a small subset of data relevant to the current prevailing period is used in the model training stage. The benefit of such an approach is two-fold. First, it eliminates the computational burden in fitting a complicated asymmetric, non-separable spatio-temporal model, because the data in the preceding prevailing period are usually limited to from a few hours to a few tens of hours, rather than weeks or months. Second, this approach makes use of a local informative spatio-temporal neighborhood that is most relevant to the short-term forecasting horizon. In this sense, it bears the similarity with the spatial informative neighborhood discussed in Section 3.2.2 or the temporal neighborhood used in [231].

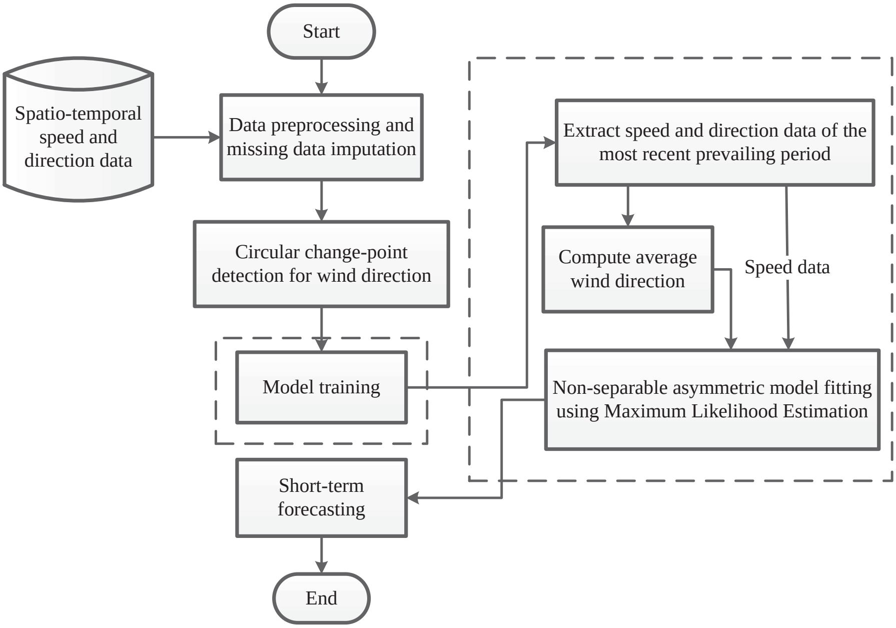
Start
Spatio-temporal speed and direction data
Data preprocessing and missing data imputation
Circular change-point detection for wind direction
Model training
Short-term forecasting
End
Extract speed and direction data of the most recent prevailing period
Compute average wind direction
Speed data
Non-separable asymmetric model fitting using Maximum Likelihood Estimation

FIGURE 3.8 A flowchart that outlines the short-term forecasting based on a spatio-temporal model. (Reprinted with permission from Ezzat et al. [59].)

3.4.4 Hybrid of Asymmetric Model and SVM

A spatio-temporal model can be used together with some machine learning models to improve further the forecasting capability. Here, an ASYM is fit to the spatio-temporal wind training data, and then an SVM model is fit to the residuals obtained by ASYM, in order to capture any nonlinearities that are not covered by the base ASYM model. The final hybrid model has an additive form as

$$V_i(t) = V_i^{\text{ASYM}}(t) + \mathcal{E}_i^{\text{SVM}}(t) + \tilde{\varepsilon}_i(t), \tag{3.40}$$

where $V_i^{\text{ASYM}}(t)$ is the ASYM model fit, $\mathcal{E}_i^{\text{SVM}}(t)$ represents the SVM model fit to the spatio-temporal residuals after the ASYM model fit, and $\tilde{\varepsilon}_i(t)$ is the final residual term. This hybrid forecasting model is referred to as HYB.

3.5 CASE STUDY

In contrast to the situations where wind measurements come from a small number of locations spread over large areas, as in [75, 91, 208], the within-farm local wind field is much denser. Recall that the spatial and temporal resolutions of wind data in the `Wind Spatio-Temporal Dataset2` are one mile and one hour, respectively. The purpose of this case study is to demonstrate the existence of an asymmetric wind pattern in certain time periods and the benefit that a non-separable model may render in terms of short-term wind forecasting on such a compact wind field.

Four periods are chosen from different times in the `Wind Spatio-Temporal Dataset2`. For each of the four periods, six hours of data are used for model training. The choice for this short training period is motivated by observing that the shortest prevailing period length, as shown in Section 3.3.3, is about six hours. As such, a training period of six hours ensures temporal homogeneity and stationarity in the training data, allowing for reliable model estimation. Furthermore, for short-term wind forecasting, using a longer history of wind measurements is not necessarily helpful, as evident by the low time lag order used in the time series models in Chapter 2 or in the GSTAR model in Section 3.2.3.

In this study, forecasting is made for up to four hours ahead, i.e., $h=$ 1, 2, 3, or 4. A variety of forecasting models are studied and compared, including ASYM, SEP, the persistence model, a time-series model chosen as ARMA(1,1), an SVM model using a radial basis kernel function and the wind speeds measured at $t-1$, as well as an HYB that combines ASYM and SVM.

Although we by and large follow the numerical analysis conducted in Ezzat et al. [59], there are a couple of differences in treatment here leading to different numerical outcomes. But the main messages stay consistent with those advocated in Ezzat et al. [59].

This section employs a missing data imputation procedure, and as a result, the `Wind Spatio-Temporal Dataset2` does not have any missing data for wind speed. The power curve used here is a turbine-specific power curve,

TABLE 3.5 Log-likelihoods of asymmetric versus separable spatio-temporal models.

Period	ASYM	SEP
1.	−2090.900	−2091.294
2.	−2033.135	−2033.140
3.	−1800.352	−1800.702
4.	−2181.999	−2185.815

rather than a single power curve averaged for all the turbines. To specify $\boldsymbol{\mu}$ and $\mathbf{D}$ in ASYM, the most recent period is used as the training dataset. The speed and direction time-series data, recorded at one of the masts, is used to compute a time-series vector of wind velocities, along the longitudinal and latitudinal directions, respectively. The estimate of $\boldsymbol{\mu}$ is the sample average of the wind velocity vector, whereas the estimate of 2×2 matrix $\mathbf{D}$ is the sample covariance matrix of the horizontal and vertical velocities. This estimate of $\mathbf{D}$ is different from that in Ezzat et al. [59] but is the same as what is used in Chapter 4. This new estimate of $\mathbf{D}$ is used so that the ASYM model and its parameter estimation are consistent between Chapter 3 and Chapter 4.

The rest of the parameters in ASYM are estimated through a maximum likelihood estimation, implemented in `R` using the routine `nlm`. To appreciate the space-time coupling and asymmetry, take the prevailing period in January 2011 as an example. The non-separability parameter, $\hat{\beta} = 0.840$, and the asymmetry parameter, $\hat{\varphi} = 0.102$. These estimated values suggest that the underlying spatio-temporal process has space and time coupling and is asymmetric. When fitting the asymmetric model and its separable counterpart, i.e., ASYM and SEP, analysts can compare the respective log-likelihood values and observe which modeling option provides a better fit. Table 3.5 presents the log-likelihood values for ASYM and SEP model fits for all four periods. The numerical results show that ASYM has a higher log-likelihood value, albeit sometimes marginally so, than that of SEP.

In this study, two performance metrics are used—RMSE and MAE; for their definitions, please refer to Section 2.6. Tables 3.6 and 3.7 present the RMSE and MAE values for up to a 4-hour ahead forecast using the aforementioned temporal or spatio-temporal models. The aggregate measure reported is the average over all 4-hour ahead forecasts.

The results presented in Tables 3.6 and 3.7 show that the forecasts based on the asymmetric non-separable model outperform the competing methods considered in the study. The improvement of ASYM over the separable models is due to ASYM's capturing of the strong asymmetries, whereas its improvement over ARMA and SVM is mostly due to the characterization of spatial correlations as well as asymmetry, both of which the ARMA and SVM models fail to capture. Hybridizing ASYM with SVM (the HYB model) appears to achieve a further enhancement in forecasting accuracy over the ASYM only

TABLE 3.6 RMSE of wind speed forecasting. The percentage improvements are the error inflation rate relative to HYB.

Period	**Method**	$h = 1$	$h = 2$	$h = 3$	$h = 4$	**Average**	**% Imp.**
1	ASYM	0.993	1.441	2.853	3.122	2.289	3%
	SEP	1.070	1.727	3.242	3.469	2.582	14%
	PER	1.287	1.719	2.984	3.161	2.424	8%
	ARMA(1,1)	1.627	2.056	3.480	3.622	2.833	22%
	SVM	1.611	1.912	3.335	3.437	2.701	18%
	HYB	1.019	1.441	2.784	2.981	2.222	
2	ASYM	1.618	2.747	2.573	2.093	2.300	5%
	SEP	1.616	2.743	2.569	2.090	2.297	5%
	PER	1.832	2.877	2.569	2.075	2.374	8%
	ARMA(1,1)	1.986	3.054	2.781	2.222	2.547	14%
	SVM	2.543	3.777	3.531	2.977	3.243	33%
	HYB	1.585	2.667	2.438	1.874	2.184	
3	ASYM	0.897	0.946	1.078	1.390	1.095	0.2%
	SEP	0.900	1.184	1.269	1.654	1.281	15%
	PER	1.007	1.067	1.358	1.510	1.253	13%
	ARMA(1,1)	1.114	1.316	1.303	1.648	1.359	20%
	SVM	1.035	1.155	1.340	1.683	1.326	18%
	HYB	0.894	0.944	1.077	1.388	1.093	
4	ASYM	1.319	1.521	1.934	3.745	2.336	6%
	SEP	1.415	1.630	2.028	3.681	2.362	7%
	PER	1.880	2.096	2.526	5.281	3.248	33%
	ARMA(1,1)	2.070	1.769	2.144	3.809	2.575	15%
	SVM	1.806	1.859	2.392	4.375	2.810	22%
	HYB	1.239	1.422	1.942	3.446	2.191	

approach, demonstrating the additional benefit brought by the machine learning method. The improvements of HYB over ASYM for wind speed forecast range from 0.2% to 6%, and on average, 3.6%. Combining the strength of the asymmetrical modeling and machine learning, in terms of wind speed forecast, HYB improves, based on the average of the four periods, 10% in RMSE (12% in MAE, same below) over SEP, 16% (14%) over PER, 18% (20%) over ARMA(1,1), and 23% (24%) over SVM.

Measuring the performance metrics in terms of wind power, analysts can first make a wind speed forecast and then convert the wind speed to wind power, using the power curve as explained in Fig. 1.2. The nominal power curve is usually provided by the turbine manufacturer. To get more accurate representation of the actual power curve, the site-specific wind speed and wind power data can be used to estimate the turbine-specific power curve. The topic of estimating a power curve is the focus of Chapter 5. The specific procedure used here for power curve estimation is the binning method, the standard nonparametric method used in the wind industry [102]; for more details about the binning method, please refer to Chapter 5. Using the estimated power curves of individual turbines, analysts can predict the wind power generated at each turbine given the wind speed forecasts.

TABLE 3.7 MAE of wind speed forecasting. The percentage improvements are the error inflation rate relative to HYB.

Period	**Method**	$h = 1$	$h = 2$	$h = 3$	$h = 4$	**Average**	**% Imp.**
1	ASYM	0.846	1.248	2.636	2.912	1.911	4%
	SEP	0.919	1.540	3.046	3.273	2.194	16%
	PER	1.048	1.491	2.803	2.901	2.061	11%
	ARMA(1,1)	1.379	1.833	3.292	3.395	2.475	26%
	SVM	1.404	1.694	3.142	3.175	2.354	22%
	HYB	0.879	1.236	2.533	2.712	1.840	
2	ASYM	1.268	2.526	2.379	1.813	1.997	6%
	SEP	1.266	2.522	2.375	1.810	1.993	6%
	PER	1.489	2.552	2.265	1.749	2.013	7%
	ARMA(1,1)	1.615	2.806	2.520	1.894	2.209	15%
	SVM	2.308	3.485	3.211	2.610	2.904	35%
	HYB	1.232	2.442	2.240	1.599	1.878	
3	ASYM	0.729	0.773	0.906	1.224	0.908	0.4%
	SEP	0.736	1.017	1.110	1.476	1.085	17%
	PER	0.807	0.840	1.054	1.203	0.976	7%
	ARMA(1,1)	0.930	1.151	1.136	1.429	1.161	22%
	SVM	0.835	0.937	1.065	1.403	1.060	15%
	HYB	0.722	0.771	0.904	1.222	0.905	
4	ASYM	1.049	1.267	1.578	3.538	1.858	7%
	SEP	1.129	1.361	1.671	3.470	1.908	9%
	PER	1.488	1.711	2.060	4.782	2.510	31%
	ARMA(1,1)	1.668	1.437	1.757	3.503	2.091	17%
	SVM	1.469	1.525	1.934	3.968	2.224	22%
	HYB	0.968	1.180	1.566	3.226	1.735	

TABLE 3.8 RMSE of wind power forecasting. The percentage improvements are the error inflation rate relative to HYB.

Period	Method	$h = 1$	$h = 2$	$h = 3$	$h = 4$	Average	% Imp.
1	ASYM	0.090	0.140	0.326	0.383	0.265	3%
	SEP	0.092	0.171	0.372	0.415	0.295	13%
	PER	0.111	0.161	0.333	0.370	0.267	4%
	ARMA(1,1)	0.138	0.201	0.396	0.430	0.317	19%
	SVM	0.133	0.186	0.376	0.405	0.299	14%
	HYB	0.090	0.142	0.321	0.363	0.256	
2	ASYM	0.221	0.354	0.356	0.312	0.315	6%
	SEP	0.221	0.353	0.355	0.310	0.314	6%
	PER	0.252	0.368	0.346	0.293	0.318	7%
	ARMA(1,1)	0.282	0.404	0.387	0.325	0.353	16%
	SVM	0.389	0.525	0.509	0.450	0.471	37%
	HYB	0.216	0.341	0.332	0.276	0.295	
3	ASYM	0.116	0.094	0.105	0.146	0.117	1%
	SEP	0.102	0.116	0.124	0.177	0.133	13%
	PER	0.137	0.122	0.155	0.172	0.148	21%
	ARMA(1,1)	0.126	0.127	0.130	0.170	0.140	17%
	SVM	0.111	0.115	0.135	0.176	0.137	15%
	HYB	0.114	0.092	0.105	0.146	0.116	
4	ASYM	0.168	0.171	0.212	0.442	0.255	−0.3%
	SEP	0.170	0.175	0.226	0.431	0.256	0%
	PER	0.255	0.279	0.331	0.660	0.389	34%
	ARMA(1,1)	0.251	0.205	0.248	0.482	0.299	14%
	SVM	0.225	0.239	0.298	0.568	0.339	24%
	HYB	0.193	0.175	0.216	0.425	0.256	

Table 3.8 compares the competing models in terms of the RMSE of wind power prediction. Similar degrees of improvement of using the asymmetric, nonseparable model are observed in wind power prediction as in wind speed forecast. Specifically, the improvement of HYB over ASYM is up to 6%, and on average, 2.4%. Compared to other methods, HYB on average improves, in terms of reduction in RMSE, 8% over SEP, 17% over PER, 17% over ARMA(1,1), and 23% over SVM. These results are aligned with the findings made in Section 3.3 that local wind fields can be strongly asymmetric at the fine-scale spatio-temporal resolutions. Spatio-temporal models that capture such physical phenomena are expected to enhance short-term forecasting.

GLOSSARY

ARD: Automatic relevance determination

ARMA: Autoregressive moving average

ASYM: Asymmetric, non-separable spatio-temporal model

BFGS: Broyden-Fletcher-Goldfarb-Shanno optimization algorithm

cdf: Cumulative distribution function

CRPS: Continuous ranked probability score

GSTAR: Gaussian spatio-temporal autoregressive model

HYB: Hybrid model combining ASYM and support vector machine

i.i.d: Identically, independently distributed

MAE: Mean absolute error

PCE: Power curve error

PER: Persistence forecasting

RMSE: Root mean squared error

SE: Squared exponential covariance function

SEP: Separable spatio-temporal model

SVM: Support vector machine

EXERCISES

3.1 In the machine learning literature, if a prediction mechanism can be expressed as $\hat{\mathbf{V}} = \mathbf{SV}$, it is called a linear smoother, where $\mathbf{S}$ is the smoother matrix. It is also established that the effective number of parameters in a linear smoother is $\text{tr}(\mathbf{S})$. In the following, to make things simpler, assume $\beta_0 = 0$. Consider a total of N data pairs in the training set:

a. Show that the kriging predictor in Eq. 3.15 is a linear smoother.

b. Show that the effective number of parameters in a kriging predictor is

$$\sum_{i=1}^{N} \frac{\lambda_i}{\lambda_i + \hat{\sigma}_\varepsilon^2},$$

where λ_i's, $i = 1, \ldots, N$, are the eigenvalues of $\mathbf{C}_{NN}$.

c. Show that for a kriging predictor without the nugget effect, its effective number of parameters is N, the same as that of the data points in the training set. What does this tell you about the difference between a linear regression predictor and a kriging predictor (i.e., a Gaussian process regression)?

3.2 When we discuss the support vector machine formulation (2.47), we state (page 44) that "SVM regression can be made equivalent to Gaussian process regression, if (a) the loss function uses a squared error loss function, (b) $\gamma/2$ is set to σ_ε^2, which is the variance of the i.i.d noise term, (c) when the kernel function, $K(\cdot, \cdot)$, is set to be a covariance function." Please show that this is true.

3.3 When the kriging model in Eq. 3.8 has no nugget effect term, then it is said that the process has noise-free observations. Under that circumstance, the kriging predictor has an interpolation property, which means $\hat{V}(\mathbf{s}_i) = V(\mathbf{s}_i)$, if $\mathbf{s}_i$ is in the training set.

a. Prove the interpolation property.

b. Suppose that an underlying true function is $g(x) = e^{-1.4x}\cos(7\pi x/2)$, and seven training data pairs $\{x, y\}$ are taken from the curve, which are, respectively,

$$\{0.069, 0.659\}, \{0.212, -0.512\}, \{0.355, -0.440\}, \{0.498, 0.344\},$$
$$\{0.641, 0.294\}, \{0.783, -0.229\}, \{0.926, -0.199\}.$$

Please use this set of data and the ordinary kriging model without the nugget effect to construct the predictive function $\hat{g}(x)$. Plot both $g(x)$ and $\hat{g}(x)$ with the seven data points marked. Observe whether the kriging predictor interpolates the training data points.

3.4 Take one month of 10-min wind speed data and wind power data from the `Wind Time Series Dataset`. Treat the wind speed data as x and the wind power data as y. Fit an ordinary kriging model. Use the squared exponential covariance function. Please generate a plot with the original data points, the mean prediction line, and the two standard deviation lines.

3.5 Please generate one-dimensional sample functions using a power exponential function for the following parameter combinations:

a. $\theta = 5$, $\sigma_V^2 = 1$, $p = 2$.

b. $\theta = 1$, $\sigma_V^2 = 0.1$, $p = 2$.

c. $\theta = 5$, $\sigma_V^2 = 1$, $p = 1$.

d. $\theta = 1$, $\sigma_V^2 = 0.1$, $p = 1$.

e. $\theta = 5$, $\sigma_V^2 = 1$, $p = 1.5$.

f. $\theta = 1$, $\sigma_V^2 = 0.1$, $p = 1.5$.

3.6 Complete the following:

a. Derive Eq. 3.12.

b. Derive Eq. 3.14.

c. Derive the log-likelihood function in Eq. 3.16.

d. Given the universal kriging model in Eq. 3.17, find its log-likelihood function.

3.7 Use the 2009 data in the `Wind Spatio-Temporal Dataset1` and compute the pairwise sample correlation between any two turbines. Then plot the correlation against the distance between the two turbines, in which the horizontal axis is the between-turbine distance and the vertical axis is the correlation in its absolute value in $[0, 1]$.

3.8 Derive Eq. 3.23.

3.9 Use the data of January 2009 from the `Wind Spatio-Temporal Dataset1` and select a target site. Try different values of κ and see how it affects the resulting informative neighborhood.

3.10 Derive the α-quantile of the truncated normal distribution in Eq. 3.27.

3.11 Use the `Wind Spatio-Temporal Dataset2` and group the data for a month. Compute the asymmetry level for any pair of turbines for that month under its specific average wind direction. Repeat this for each month in the yearlong dataset and group the asymmetry values based on their corresponding time lags. Create a plot similar to the right panel of Fig. 3.4.

3.12 In Eq. 3.5, when $p = 1$, we say that the resulting covariance function is an exponential covariance function, which reads, if assuming isotropy,

$$C_{\text{Exp}}(\mathbf{u}) = \sigma_V^2 \exp\left\{-\frac{\|\mathbf{u}\|_1}{2\theta}\right\} = \sigma_V^2 \exp\left\{-\frac{|u_1| + \cdots + |u_d|}{2\theta}\right\},$$

where $\mathbf{u}$ is assumed to have d elements. But there is another definition of the exponential covariance function, which uses a 2-norm inside the exponential to measure distances, namely

$$C_{\text{Exp}}(\mathbf{u}) = \sigma_V^2 \exp\left\{-\frac{\|\mathbf{u}\|_2}{2\theta}\right\} = \sigma_V^2 \exp\left\{-\frac{\sqrt{u_1^2 + \cdots + u_d^2}}{2\theta}\right\}.$$

a. Explain under what condition the covariance function, $C_{\text{ASYM}}(\mathbf{u}, h)$, is the same as $C_{\text{Exp}}(\mathbf{u})$ with the 2-norm distance.

b. Consider a separable spatio-temporal covariance function, $C(\mathbf{u}, h)$, that is constructed by the product of exponential covariance functions for both the spatial and temporal components, i.e.,

$$C(\mathbf{u}, h) = C_{\text{Exp}}(\mathbf{u}) \cdot C_{\text{Exp}}(h).$$

How is this separable covariance function, $C(\mathbf{u}, h)$, different from $C_{\text{ASYM}}(\mathbf{u}, h)$ when β, φ, and τ are set to zero?

3.13 Use the `Wind Spatio-Temporal Dataset2` to conduct the following studies.

a. Use the `circular` package to conduct a change-point detection on the yearlong wind direction measured on one of the met masts, and see how many change points you detect. Suppose that k change points are detected, then it leads to $k+1$ prevailing periods.

b. Calculate the asymmetry level for each one of the prevailing periods and tabulate the results in a fashion similar to Table 3.4.

c. Select a period in which the asymmetry is weak (smaller than 0.05) and make sure that its overall duration is longer than ten hours. Then, fit an ASYM model and an SEP model using the first six hours of data. Compare the common model parameters and the log-likelihood of the two models.

d. Use the next four hours of data to conduct an h-hour ahead forecasting for $h = 1, 2, 3, 4$. Compare ASYM and SEP using both RMSE and MAE.

CHAPTER 4

Regime-switching Methods for Forecasting

One particular class of wind forecasting methods worth special attention is the regime-switching approach. We hence dedicate this chapter to the discussion of regime-switching methods.

The motivation behind the regime-switching approach is to deal with nonstationarity in wind dynamics—in wind speed, in wind direction, or in spatial correlation. Recall that the spatio-temporal covariance structures introduced in Chapter 3 are all stationary in nature. While nonstationary covariance structures do exist, using them is not easy. Analysts find that a simpler approach is to compartmentalize the nonstationary variables into a finite number of disjoint intervals, each of which is referred to as a regime. Within a regime, the underlying wind process is assumed stationary. To account for the overall nonstationarity, a mechanism is needed for the forecasting model to transition from one regime to another, as the underlying wind process is progressing. The resulting approach is called regime-switching. In essence, a regime-switching method is a collection of distinct, and most often linear, models.

The regime-switching mechanism can be used with a temporal only process, considering only nonstationarity in time, or with a spatio-temporal process, considering nonstationarity in both space and time.

4.1 REGIME-SWITCHING AUTOREGRESSIVE MODEL

Suppose that analysts pre-define a number of regimes, indexed from 1 to R, and denote the wind regime at time t by $r(t) \in \{1, ..., R\}$, which is known as the *regime variable*. The regime-switching autoregressive (RSAR) model [234] is a collection of R autoregressive models, each of which is associated with a wind regime and thus uses a set of parameters peculiar to that regime to produce regime-dependent forecasts.

In an RSAR, the wind speed, $V(t)$, at time t and in regime $r(t)$ is modeled as an AR model of order $p^{r(t)}$ using a set of regime-dependent parameters $\{a_0^{r(t)}, a_1^{r(t)}, \ldots, a_j^{r(t)}, \ldots\}$, such as

$$V(t) = a_0^{r(t)} + \sum_{j=1}^{p^{r(t)}} a_j^{r(t)} V(t-j) + \varepsilon(t), \tag{4.1}$$

where $\varepsilon(t)$ is a zero-mean, normally distributed, i.i.d random noise whose variance can be regime-dependent. In this section, the value of regime variable, $r(t)$, is determined based on the observed values of wind speed. Be aware that $r(t)$ can be decided using other explanatory variables, including, but not limited to, wind direction or temperature [75, 176].

Estimating the parameters for a regime-switching autoregressive model is usually conducted for each individual AR model separately. The procedure, model selection criteria, and model diagnostics, as explained in Section 2.4, can be used here without much modification. Zwiers and von Storch [234] note a number of differences in handling a bunch of AR models, as opposed to handling a single AR model, summarized below.

- One word of caution is on ensuring that each regime should have a sufficient amount of data for parameter estimation. This aspect is less problematic nowadays with much advanced data collection capability in commercial wind operations. Data appear to be more than enough even after being divided into a number of regimes. The data amount sufficiency could have been an issue 30 years ago.

- An analyst can choose to use an aggregated AIC to decide the overall model order for the regime-switching method. This practice becomes less popular, as analysts nowadays rely more on computational procedures that split the data into training and test sets, like in cross validation, to test on a model's forecasting performance and to adjust respective modeling decisions.

- As mentioned above, $\varepsilon(t)$ could have different variances in different regimes. An implication is that analysts should pay attention to the heteroscedasticity issue (i.e., different variances) when devising a statistical test. For more discussion, please refer to page 1351 in [234].

The use of an RSAR for forecasting is fairly straightforward. Analysts first identify either the current wind regime, per definition given below, or the regime anticipated in the forecasting horizon, select the AR model corresponding to the target regime, and then make forecasts using the selected AR model, as one would have while using a single AR model.

4.1.1 Physically Motivated Regime Definition

In a regime-switching method, here as well as in the methods introduced in the sequel, one crucial question is how to decide the number of wind regimes

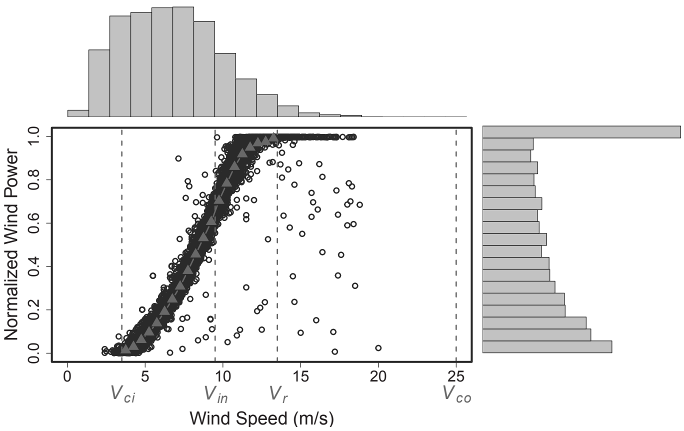

FIGURE 4.1 Normalized wind power versus wind speed. V_{ci}: cut-in speed, V_{in}: inflection point, V_r: rated speed and V_{co}: cut-out speed. On the top and right sides are the histograms of wind speed and power, respectively. The circle dots are raw wind data. (Reprinted with permission from Ezzat et al. [60].)

and the boundaries dividing these regimes. Consider R disjoint wind speed regimes, denoted by $\{r_1, r_2, \ldots, r_R\}$, such that $V(t)$ belongs to one and only one of the R wind regimes. Each regime, r_k, is defined by an interval $[u_k, v_k)$, such that u_k and v_k are the boundary values for r_k, with $u_1 = 0$ and $u_{k+1} = v_k$.

One approach is to pre-define the wind regimes based on physical understanding. We guide the selection of wind speed regimes in light of the regions associated with a wind power curve. Fig. 4.1 plots the wind speed against the normalized wind power recorded at one of the turbines for one year's worth of data in the `Wind Spatio-Temporal Dataset2`. The power curve is estimated by using the binning method [102] as mentioned in Chapter 3 and to be detailed in Chapter 5. The binning estimates are shown in Fig. 4.1 as the triangles.

Four physically meaningful values of wind speed are critical to defining a wind power curve, which are—the cut-in speed, V_{ci}, the inflection point, V_{in}, the rated speed, V_r, and the cut-out speed, V_{co}. We have explained in Section 1.1 the meanings of the cut-in speed, the rated speed, and the cut-out speed. A turbine manufacturer provides the values of V_{ci}, V_r, and V_{co} for a specific turbine. Their typical values are, respectively, 3.5, 13.5, and 25 m/s, although some turbines have their cut-out speed at 20 m/s. Between V_{ci} and V_r, the power curve follows a nonlinear relationship, with an inflection point separating the convex and concave regions. This inflection point, denoted by V_{in}, marks the start when the turbine control mechanism is used to regulate the power production. Hwangbo et al. [96] estimate V_{in} for modern wind turbines to be around 9.5 m/s. These physically meaningful values induced by the power curve motivate analysts to define a total of $R = 4$ regimes, with the regime boundaries set at V_{ci}, V_{in}, V_r, and V_{co}. We advocate using these values as a starting point and make necessary adjustment when needed.

For the `Wind Spatio-Temporal Dataset2` specifically, only around 3% of wind speed data are higher than V_r. It makes sense to merge the last two wind speed regimes by eliminating the threshold at V_r. Moreover, V_{co} is in fact 20 m/s for the `Wind Spatio-Temporal Dataset2`, and adjusting the end point of the wind speed spectrum from 25 m/s to 20 m/s does not affect the above wind speed regime definition.

Wind regimes can also be defined by using wind direction, to be seen in Section 4.2, or by using the combination of wind speed regimes and wind direction regimes, to be seen in Section 4.4, where we define three wind speed regimes and two wind direction regimes, the combination of which produces a total of six wind regimes.

4.1.2 Data-driven Regime Determination

Another approach to identify the number of wind regimes is data-driven. Kazor and Hering [119] present a regime determination approach based on the Gaussian mixture model (GMM). The idea is to use a GMM to model the wind variable from the R regimes, each of which is treated as a stationary random process. Kazor and Hering use the 2×1 wind velocity vector, $\mathbf{v} = (V_1, V_2)^T$, where V_1 and V_2 are, respectively, the wind velocity along the longitudinal and latitudinal directions. Each regime is modeled as a bivariate normal density, i.e., $\mathbf{v} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, $k = 1, \ldots, R$. Denote by τ_k the proportion of observations available under the k-th regime. Then, the Gaussian mixture density function of the R regimes is expressed as

$$f(\mathbf{v}|\Theta) = \sum_{k=1}^{R} \tau_k \mathcal{N}(\mathbf{v}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4.2}$$

where $\Theta := \{\tau_1, \ldots, \tau_R; \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_R; \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_R\}$ is the set of parameters in this GMM. Kazor and Hering further simplify the covariance matrices by assuming their off-diagonal elements all zeros, leaving only two variance terms per covariance matrix to be estimated for this bivariate distribution. This assumption implies that the two wind velocity variables are uncorrelated. Under this assumption, there are five parameters per regime—one τ, two mean terms, and two variance terms—or a total of $5R$ parameters for R regimes. The parameters can be estimated by using a maximum likelihood estimation.

To determine the number of regimes, R, Kazor and Hering suggest computing the BIC for the GMM for a range of regime numbers. They specifically recommend computing the BIC for models with between one and five regimes. Recall the definition of BIC in Eq. 2.23, it can be expressed for this GMM model as

$$\mathrm{BIC}(R) = \ln(n)\cdot(5R) - 2\ln(\hat{f}(\mathbf{v}|\hat{\Theta})),$$

where n is the amount of data used to estimate the parameters, $5R$ is the number of parameters with the presence of R regimes, and $\ln(\hat{f}(\mathbf{v}|\hat{\Theta}))$ is the log-likelihood evaluated at the estimated parameters. For selecting the number of regimes, one can plot the BIC values against the number of regimes and then choose the elbow point and its corresponding number of regimes, similar to how analysts select the significant principal components using a scree plot [111].

This GMM approach does not need to define the boundaries of the regimes explicitly. Each regime is represented by its mean and variance parameters, which are in turn estimated from the data. Upon a new wind observation, $\mathbf{v}_{\text{new}}$, analysts can compute the likelihood of each individual regime, which is $\hat{\tau}_k\mathcal{N}(\mathbf{v}_{\text{new}}|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)$, for $k = 1,\ldots,R$, and then select the regime corresponding to the largest likelihood. This treatment is called *hard thresholding*, implying that one regime is chosen while all other regimes are discarded. By contrast, the *soft thresholding* treatment is to compute the normalized weighting to be given to each regime model as

$$w_k = \frac{\hat{\tau}_k\mathcal{N}(\mathbf{v}_{\text{new}}|\hat{\boldsymbol{\mu}}_k, \hat{\boldsymbol{\Sigma}}_k)}{\sum_{i=1}^{R}\hat{\tau}_i\mathcal{N}(\mathbf{v}_{\text{new}}|\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i)}, \quad k = 1,\ldots,R, \tag{4.3}$$

and then the forecasting is made by using all R models and by associating each model with the corresponding weight w_k.

4.1.3 Smooth Transition between Regimes

Analysts recognize that abrupt changes between regimes may not be desirable. The concept of smooth transition between regimes is therefore introduced. The soft-thresholding GMM is a type of smooth transition approach, as there are no rigid boundaries between regimes, and for each forecast, all regime-dependent models are used with their respective weights.

Pinson et al. [164] introduce another smooth transition autoregressive model (STAR, not to be confused with GSTAR in Section 3.2). The model takes the form of

$$\begin{aligned} V(t) = \sum_{i=1}^{R-1}\Bigg(& \left[a_0^{r_i} + \sum_{j=1}^{p^{r_i}} a_j^{r_i} V_{t-j}\right]\tilde{G}_i(\hat{V}^{r(t)}) \\ & + \left[a_0^{r_{i+1}} + \sum_{j=1}^{p^{r_{i+1}}} a_j^{r_{i+1}} V_{t-j}\right] G_i(\hat{V}^{r(t)})\Bigg) + \varepsilon(t), \end{aligned} \tag{4.4}$$

where $\tilde{G}_i(\cdot) = 1 - G_i(\cdot)$ is the smooth transition function that assigns weights to the AR models associated with the i-th and $(i+1)$-th regimes, and $\hat{V}^{r(t)}$ is the estimated wind speed corresponding to the regime at time t. Pinson et al. suggest using the d-step lagged wind speed, $V(t-d)$, as $\hat{V}^{r(t)}$, and then, using a logistic function to create a soft-thresholding transition, such as

$$G_k(V(t-d)) = \frac{1}{1+\exp\{-\varphi_k(V(t-d)-c_k)\}}, \quad k=1,\ldots,R, \tag{4.5}$$

where $\varphi_k > 0$ and c_k are the parameters in the transition function. The set of parameters in the smooth transition model includes those of the AR models as well as these for the transition functions. Typically, the AR model parameters can be estimated separately for each regime, following the approach outlined for ARMA models in Section 2.4. Then, the parameters for the transition functions, $\{\varphi_k, c_k\}$, are decided by using a cross-validation approach.

4.1.4 Markov Switching between Regimes

A Markov-switching autoregressive (MSAR) model [6, 164, 201] uses a group of AR models, similar to those expressed in Eq. 4.1, but MSAR assumes that the switch between the regimes is triggered by a Markov chain and thus employs a transition probability matrix to govern regime changes.

The one-step ahead transition probability matrix, $\mathbf{\Pi}_{R\times R}$, is expressed as

$$\mathbf{\Pi}_{R\times R} = \begin{pmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1R} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2R} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_{R1} & \pi_{R2} & \cdots & \pi_{RR} \end{pmatrix}, \tag{4.6}$$

where the (i,j)-th element, π_{ij}, is defined as

$$\pi_{ij} = P[r(t+1) = r_j | r(t) = r_i].$$

In the above definition, the Markovian property is invoked, which says that the probability of a regime at time $t+1$ only depends on the regime status at the previous time, t, rather than on the entire history of regimes. Mathematically, what this means is

$$P[r(t+1)|r(t), r(t-1), \ldots, r(1)] = P[r(t+1)|r(t)]. \tag{4.7}$$

The transition matrix provides the probabilistic information for switching between regimes for one step ahead. The i-th row in $\mathbf{\Pi}$ represents the probabilities for the i-th regime to switch to other regimes, including itself (unchanged). The summation of all the probabilities per row should be one, i.e., $\sum_{j=1}^{R} \pi_{ij} = 1, \forall i$. The transition matrix can be estimated by using the data in a training period, namely that each π is estimated by the empirical probability based on the training data.

Once the one-step ahead transition matrix is estimated, its use mirrors that in the GMM-based approach described in Section 4.1.2. Individual AR models are fit using the data peculiar to specific regimes. The forecast at time $t+1$ is the weighted average of the forecasts made by individual AR models. Suppose that the forecast at $t+1$ made by the AR model in regime r_k is denoted by $\hat{V}^{(r_k)}(t+1)$. The weights to be used with $\hat{V}^{(r_k)}(t+1)$ come from the transition matrix. Here, again, analysts can use either the hard thresholding approach or the soft thresholding approach. Assuming the current regime is r_k, the final forecast while using the soft thresholding is

$$\hat{V}(t+1) = \sum_{j=1}^{R} \hat{\pi}_{kj} \hat{V}^{(r_j)}(t+1). \tag{4.8}$$

For the hard thresholding, one identifies the largest $\hat{\pi}_{kj}$ for $j = 1, \ldots, R$, and suppose it is $\hat{\pi}_{kj^*}$. Then the final forecast is simply to use the AR model corresponding to regime j^*, namely $\hat{V}(t+1) = \hat{V}^{(r_{j^*})}(t+1)$.

For h-step ahead forecasts, $h > 1$, a formula similar to Eq. 4.8 can be used, but one needs to replace $\hat{\pi}_{kj}$ with an h-step ahead transition probability and replace $\hat{V}^{(r_j)}(t+1)$ with the raw forecast at $t+h$, $\hat{V}^{(r_k)}(t+h)$, which can be made by the regime-specific AR model for h steps ahead. The h-step transition probability is denoted as

$$\pi_{ij}^{(h)} = P[r(t+h) = r_j | r(t) = r_i],$$

which can be recursively computed using the one-step ahead transition matrix, $\mathbf{\Pi}$, once per step. Apparently, $\pi_{ij}^{(1)} = \pi_{ij}$. Using the soft thresholding approach, the h-step ahead can be made by

$$\hat{V}(t+h) = \sum_{j=1}^{R} \hat{\pi}_{kj}^{(h)} \hat{V}^{(r_j)}(t+h).$$

The hard thresholding forecast can be attained similarly.

4.2 REGIME-SWITCHING SPACE-TIME MODEL

The previous section discusses how the regime-switching mechanism works with time series data or temporal only models. This section discusses the regime-switching space-time models, primarily based on the work reported in [75].

For a spatio-temporal wind process, the wind speed is denoted by $V_i(t)$, following the same notational convention used in Chapter 3, where the subscript i is the site index and t is the time index. Recall that we use n to indicate the data amount along the time axis and N to represent the number of sites. A generic spatio-temporal regime-dependent model [163] can be expressed as

$$V_*(t) = a_0^{r(t)} + \sum_{i=1}^{N} \sum_{\ell=1}^{p^{r(t)}} a_{i\ell}^{r(t)} V_i(t-\ell) + \varepsilon_*(t), \tag{4.9}$$

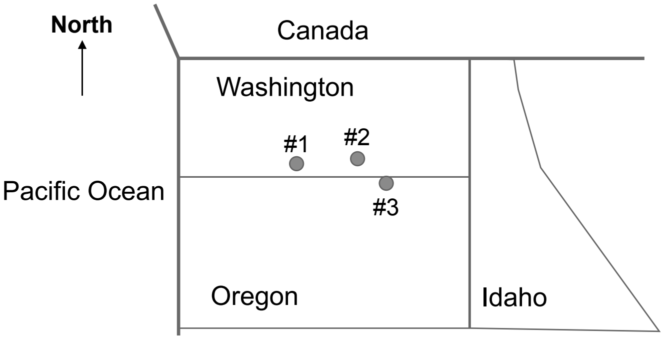

FIGURE 4.2 Geographic layout of the three sites in the border area of the states of Washington and Oregon. The actual state boundaries are not strictly parallel or vertical and the shapes of the states are approximated.

where $a_{i\ell}^{r(t)}$ is the spatio-temporal coefficient peculiar to the regime represented by $r(t)$ and '$*$' indicates the target site.

Gneiting et al. [75] consider a specific regime-switching spatio-temporal model. The setting in their study includes three geographical locations in the border area of the states of Washington and Oregon—see Fig. 4.2 for an illustration. The three sites are more or less on the same latitude but spread along a west-east line. The westernmost site is about 146 km from the middle site, which is in turn 39 km west of the easternmost site. The easternmost site is in the vicinity of the Stateline wind energy center, which is the target site for wind forecasting. The three sites are labeled as #1, #2, and #3, respectively, from the westernmost to the easternmost.

The regime is determined by the observed wind direction. The prevailing wind in that area, due to the pressure difference between the Pacific Ocean and the continental interior, is largely west-eastward. Gneiting et al. [75] predefine their space-time regimes based on this physical understanding. They define two regimes—the westerly regime when the wind blows from the west and the easterly regime when the wind blows from the east, and then fit a space-time model for each regime.

The model used in [75] assumes a truncated normal predictive distribution at time $t+h$ and the target site, i.e., $\mathcal{N}^+(\mu_3(t+h), \sigma_3^2(t+h))$, where the subscript "3" indicates site #3, the target site for forecasting. This treatment resembles what is used in the GSTAR model in Section 3.2. In fact, the GSTAR model borrows this approach from [75], as [75] was published earlier, but their presentation order in this book may have left the readers with the opposite impression.

Gneiting et al. [75] propose a space-time model specific for each of the two regimes. For the westerly regime, the mean forecasting model is

$$\begin{aligned}\mu_3(t+h) = a_0^{\mathrm{W}} &+ a_1^{\mathrm{W}} V_3(t) + a_2^{\mathrm{W}} V_3(t-1) \\ &+ a_3^{\mathrm{W}} V_2(t) + a_4^{\mathrm{W}} V_2(t-1) + a_5^{\mathrm{W}} V_1(t),\end{aligned} \tag{4.10}$$

where a_i^{W}, $i = 0, 1, \ldots, 5$, are the model coefficients to be estimated by using the data in the westerly regime. Note that in the above model, a low temporal order is used, only going back in history for two steps, i.e., t and $t-1$. For the westernmost site (site #1), Gneiting et al. find that it is only beneficial enough to include the time history at t, not even that at $t-1$.

For the easterly regime, the mean forecasting model is

$$\mu_3(t+h) = a_0^{\mathrm{E}} + a_1^{\mathrm{E}} V_3(t) + a_2^{\mathrm{E}} V_2(t). \tag{4.11}$$

Here, Gneiting et al. [75] find that it is not beneficial to use the wind speed measurements at site #1 (westernmost) to make forecasts at site #3 (easternmost), because while the westerly wind creates a much stronger correlation between the two sites, the correlation is multi-fold weaker under the easterly wind. Another difference of the model in the easterly regime is that its temporal order is one lower than that used in the westerly regime.

The predictive standard deviation at $t+h$, $\sigma_3(t+h)$, is modeled similarly to that in Eq. 3.25, i.e.,

$$\sigma_3(t+h) = b_0 + b_1 \nu_3(t), \tag{4.12}$$

where in this specific case,

$$\nu_3(t) = \sqrt{\frac{1}{6} \sum_{i=1}^{3} \sum_{\ell=0}^{1} (V_i(t-\ell) - V_i(t-\ell-1))^2},$$

and b_0, b_1 take different values in the two different regimes, although we drop the regime-indicating superscript for a clean presentation.

Gneiting et al. [75] further suggest removing the diurnal pattern from the data using Eq. 2.15 and then fitting the above space-time model to the residuals, corresponding to V'' in Eq. 2.16. But Gneiting et al. only recommend doing so for the westerly regime while leaving the easterly regime to use the original data. The dominant westerly wind, from the ocean to land, creates a special pattern causing all these differences in the above treatments.

The aforementioned models are supposed to be established for the respective regimes using the data collected in the corresponding regime. When making forecasts, the wind direction measured at site #1 is used to invoke one of the regimes and hence the corresponding AR model. In [75], Gneiting et al. are only concerned with making a forecast at $h = 2$, i.e., a two-hour ahead forecast, but the model above can be used for other h's in its current form. If the mean of the predictive distribution is used as the point forecast at $h = 2$ and site #3, then

$$\hat{V}_3(t+2) = \hat{\mu}_3(t+2) + \hat{\sigma}_3(t+2)\frac{\phi\left(\frac{\hat{\mu}_3(t+2)}{\hat{\sigma}_3(t+2)}\right)}{\Phi\left(\frac{\hat{\mu}_3(t+2)}{\hat{\sigma}_3(t+2)}\right)},$$

where $\phi(\cdot)$ is the pdf of the standard normal distribution. If the median, or more generally, the α-quantile of the predictive distribution is used as the point forecast, then Eq. 3.27 is to be used; for median, i.e., the 0.5-quantile, set $\alpha = 0.5$.

For parameter estimation, Gneiting et al. [75] use the CRPS criterion, to be consistent with their probabilistic modeling approach. For a truncated normal distribution with its distribution parameter estimated as $\hat{\mu}$ and $\hat{\sigma}$, Gneiting et al. show that the CRPS can be expressed as

$$\begin{aligned} \text{CRPS}_{\text{TN}} = \frac{1}{n}\sum_{t=1}^{n} \hat{\sigma}\cdot\Phi\left(\frac{\hat{\mu}}{\hat{\sigma}}\right)^{-2} & \left\{ \frac{V_3(t)-\hat{\mu}}{\hat{\sigma}}\Phi\left(\frac{\hat{\mu}}{\hat{\sigma}}\right) \right. \\ & \times \left[2\Phi\left(\frac{V_3(t)-\hat{\mu}}{\hat{\sigma}}\right) + \Phi\left(\frac{\hat{\mu}}{\hat{\sigma}}\right) - 2\right] \\ & \left. + 2\phi\left(\frac{V_3(t)-\hat{\mu}}{\hat{\sigma}}\right)\Phi\left(\frac{\hat{\mu}}{\hat{\sigma}}\right) - \frac{1}{\sqrt{\pi}}\Phi\left(\sqrt{2}\frac{\hat{\mu}}{\hat{\sigma}}\right) \right\}, \end{aligned} \tag{4.13}$$

where π is the circumference constant, not to be confused with the transition probability variable used in Eq. 4.6. The smaller the CRPS, the better. Minimizing the CRPS may run into numerical issues, especially as $\hat{\mu}/\hat{\sigma} \to -\infty$. Gneiting et al. recommend setting the CRPS to a large positive number when $\hat{\mu}/\hat{\sigma} \leq -4$ to resolve this issue.

Gneiting et al. [75] admit that the characteristics of this geographical area make the choice of regimes easier. Under other circumstances, the identification of forecast regimes may not be so obvious. Motivated to extend the regime-switching space-time model to a general setting, Hering and Genton [91] propose to include the wind direction as a circular variable in the model formulation to relax the model's dependence on arbitrary regime selections. Denote $\vartheta_i(t)$ as the wind direction measured at site i and time t, and the model in Eq. 4.10 now becomes

$$\begin{aligned} \mu_3(t+h) = a_0 &+ a_1 V_3(t) + a_2 V_3(t-1) + a_3 V_2(t) + a_4 V_2(t-1) + a_5 V_1(t) \\ &+ a_6 \sin(\vartheta_3(t)) + a_7 \cos(\vartheta_3(t)) + a_8 \sin(\vartheta_2(t)) + a_9 \cos(\vartheta_2(t)) \\ &+ a_{10} \sin(\vartheta_1(t)) + a_{11} \cos(\vartheta_1(t)). \end{aligned} \tag{4.14}$$

Hering and Genton [91] recommend fitting the model in Eq. 4.14 to the residuals after removing the diurnal pattern using Eq. 2.15 and refer to the resulting

TABLE 4.1 RMSE for 2-hour ahead point forecasts for wind speed at site #3 in May to November 2003. Boldface values indicate the best performance.

	May	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**
PER	2.14	1.97	2.37	2.27	2.17	2.38	2.11
AR-N	2.04	1.92	2.19	2.13	2.10	2.31	2.08
AR-D	2.01	1.85	2.00	2.03	2.03	2.30	2.08
RST-N	1.76	1.58	1.78	1.83	1.81	2.08	**1.87**
RST-D	**1.73**	**1.56**	**1.69**	**1.78**	**1.77**	**2.07**	**1.87**

Source: Gneiting et al. [75]. With permission.

TABLE 4.2 CRPS for probabilistic 2-hour ahead forecasts for wind speed at site #3 in May to November 2003. Boldface values indicate the best performance.

	May	**Jun**	**Jul**	**Aug**	**Sep**	**Oct**	**Nov**
AR-N	1.12	1.04	1.19	1.16	1.13	1.22	1.10
AR-D	1.11	1.01	1.10	1.11	1.10	1.22	1.10
RST-N	0.97	0.86	0.99	0.99	0.99	**1.08**	**1.00**
RST-D	**0.95**	**0.85**	**0.94**	**0.95**	**0.96**	**1.08**	**1.00**

Source: Gneiting et al. [75]. With permission.

method the trigonometric direction diurnal (TDD) model. For TDD, analysts do not need to estimate the model coefficients, $a_0, \ldots, a_{11}$, separately for the respective pre-defined regimes. The wind direction variable, ϑ, is supposed to adjust the model automatically based on the prevailing wind direction observed at the relevant sites. Pourhabib et al. [166] combine this regime switching idea with their GSTAR model and create a regime-switching version of the GSTAR model, which is called RSGSTAR. But the numerical results in [166] show that RSGSTAR produces only a marginal benefit as compared to the plain version of GSTAR.

Table 4.1 presents the comparison between the regime-switching space-time model with the AR model and the persistence model in terms of RMSE, whereas Table 4.2 presents the comparison in terms of CRPS. The persistence model is not included in Table 4.2 because it only provides point forecasts and no probabilistic forecasts. Here the regime-switching space-time model uses the pre-defined two regimes, i.e., the models in Eq. 4.10 and Eq. 4.11.

In the tables, the autoregressive model uses the acronym AR and the regime-switching space-time model uses the acronym RST. The suffix '-N' means that the respective model is fit to the original data, where the suffix '-D' means that the model is fit to the residual data after removing the diurnal pattern.

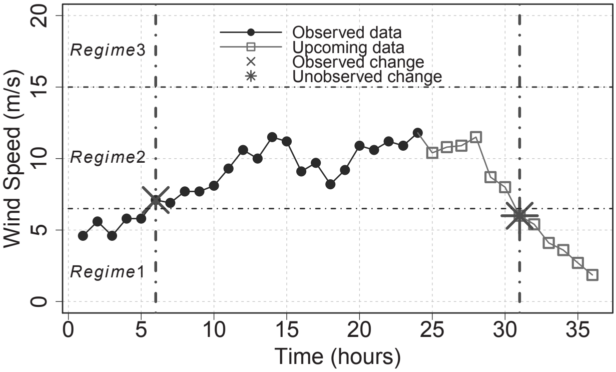

FIGURE 4.3 Wind speed at one of the turbines for a 36-hour duration. Two regime changes are identified: one in-sample and the other out-of-sample.

4.3 CALIBRATION IN REGIME-SWITCHING METHOD

The regime-switching autoregressive model and the regime-switching space-time method can be perceived as a "reactive" approach. Plainly speaking, a reactive model observes a regime change or a manifestation of it, and then adapts itself accordingly to accommodate it. In other words, the regime switching *reacts* to the regime observed and uses the forecasting model peculiar to the current wind regime to produce regime-dependent forecasts. The GMM-based approach, the smooth transition, and the Markov switching add flexibility to account for multiple possible wind regimes in the upcoming forecast period.

Ezzat et al. [60] argue that one key shortcoming of the reactive regime-switching approaches is their lack of anticipation of the upcoming regime changes in the forecast horizon. Fig. 4.3 plots the wind speeds recorded at one of the turbines in the `Wind Spatio-Temporal Dataset2` for a 36-hour duration. In practice, forecasting is often carried out in a rolling forward fashion. One could run into a situation where the goal is to obtain predictions for the next 12 hours, based on the past 24-hour data. Assume the number of regimes and regime boundaries have been pre-specified as shown in Fig. 4.3. Two regime changes are identified in the 36-hour duration, one of which takes place in the unobserved forecasting horizon. Reactive approaches may have the ability to deal with the in-sample change, but do not in their current treatment handle the unobserved, out-of-sample change. Extrapolating the characteristics learned from the training data, which are obviously not representative of the near future, could lead to negative learning and poor predictive performance. Note that the in-sample change in Fig. 4.3 is from Regime 1 to Regime 2, while the out-of-sample change is the opposite.

In the near ground wind fields like those on a wind farm, wind patterns can change rather frequently. Standing at any time point, an out-of-sample regime change could be imminent. Our analysis using the first 30 days of data in the `Wind Spatio-Temporal Dataset2` shows that the minimum-time-to-change and the median-time-to-change in wind speed are 5 hours and 15 hours, respectively, while those in wind direction are 11 hours and 33 hours, respectively. On average, a change in wind speed or wind direction takes place every 10 hours. Ignoring the occurrence of out-of-sample regime changes can seriously undermine the extrapolation ability of a regime-switching forecasting model.

Fig. 4.4 illustrates the change points detected in both wind speed and wind direction, using the first 30 days of data in the `Wind Spatio-Temporal Dataset2`. The wind direction data are from one of the met masts on the wind farm. The wind speed data are from the turbine anemometers but to facilitate a univariate detection, the wind speeds at all 200 turbines are spatially averaged to produce a single time series. Given that the hourly data are used, both wind speed and wind direction data vectors for one month are of the size 720 × 1. One may have noticed that the first half portion of the change points in the wind direction plot (bottom panel) is the same as that in Fig. 3.5. The specific change-point detection methods used are: for wind speed, a binary segmentation for multiple change detection based on the package `changepoint` in `R` [122], while for wind direction, a binary segmentation version of the circular change-point detection [106] based on the package `circular`. Recall that the circular change-point detection method is also used in Section 3.3.3 when producing Fig. 3.5.

Prompted by this observation, Ezzat et al. [60] contemplate a more proactive approach for short-term wind forecasting, which involves an action of wind speed calibration, referred to as the calibrated regime-switching (CRS) method. The CRS approach distinguishes between the in-sample regime changes taking place in the observed portion of the data and the out-of-sample regime changes occurring in the unobserved forecasting horizon. Next we take a closer look at the two types of changes. Hereinafter in this chapter, unless otherwise noted, the time index, t, is used to indicate the present time, while ℓ denotes an arbitrary time index. A forecast is to be made at $t + h$ for $h = 1, 2, \ldots, H$, i.e., the forecast horizon could be as far as H hours ahead of the present time.

4.3.1 Observed Regime Changes

An observed, in-sample regime change takes place in the observed portion of the data. Formally, an in-sample regime change occurs at time $\ell^* \in (1, t]$, when $r(\ell^* - 1) = r_k$, while $r(\ell^*) = r_{k'}$, such that $k \neq k'$ and $k, k' \in \{1, \cdots, R\}$. The CRS method signals an observed change in wind regimes by monitoring the most recent history of wind speed and wind direction. In practice, the retrospective searching for a regime change usually goes no further back than one month.

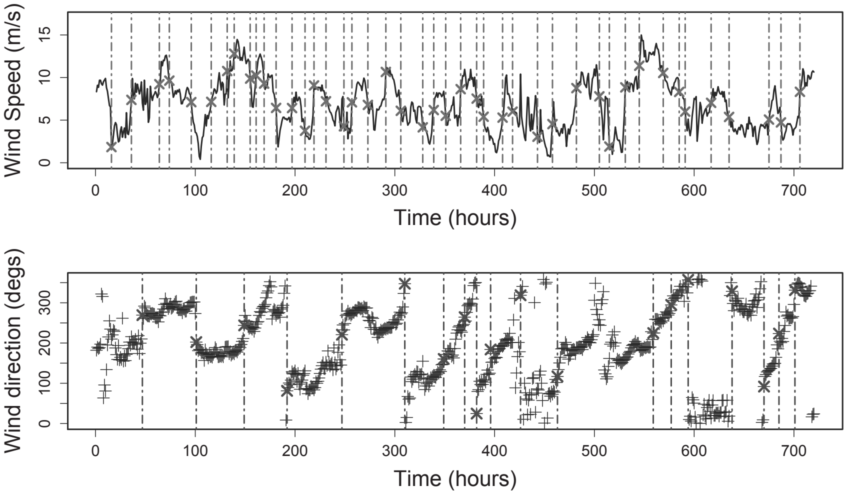

FIGURE 4.4 Top panel: change points in one month of spatially aggregated wind speed data. Bottom panel: change points in one month of wind direction data. The span of the x-axis is a month, or 720 hours. (Reprinted with permission from Ezzat et al. [60].)

4.3.2 Unobserved Regime Changes

An unobserved, out-of-sample regime change takes place in the forecasting horizon, $[t+1, t+H]$. In other words, a future regime change may occur at $t+h$, where $r(t+h-1) = r_k$, while $r(t+h) = r_{k'}$, such that $k \neq k'$ and $k, k' \in \{1, \cdots, R\}$.

Anticipating the out-of-sample regime changes is understandably much more challenging. It is important to identify certain *change indicator* variables that are thought to be able predictors of out-of-sample changes and whose values can be extracted from the observed data. Ezzat et al. [60] identify two principal change indicators: the current observed wind regime, i.e., $r(t)$, and the *runlength*, denoted by $x(t+h)$, which is to be explained below.

The current wind regime, $r(t)$, is naturally a useful indicator of upcoming wind regimes at $t+h$. For instance, in windy seasons, it is more likely to transit from low-speed to high-speed regimes, and the converse holds true for calmer seasons. This, in fact, is the essence of using Markov switching autoregressive models which translate the current regime information into transition probabilities for connections with the upcoming regimes.

Given the frequent changes in wind speed and direction as observed in Fig. 4.4, the current regime information alone is not sufficient to confidently inform about when and how out-of-sample changes occur. An additional input is required to make a good inference. Ezzat et al. [60] conclude that the runlength, which is the time elapsed since the most recent change point in the response of interest, is a far more potent indicator of upcoming changes than many other alternatives—other alternatives include the rate of change in wind speed, or that in wind direction, turbulence intensity and volatility measures. The use of runlength is first proposed in the online change-point detection literature [188].

The value of the runlength at any arbitrary time index ℓ is defined as $x(\ell) = \ell - \ell^*$, where ℓ^* is the time at which the most recent regime change is observed such that $\ell^* < \min(\ell, t)$. For a time point in the forecast horizon, i.e., $\ell = t + h$, Ezzat et al. [60] define the runlength in the forecast horizon as $x(t+h) = t + h - \ell^*$.

To appreciate the relevance of the runlength variable more intuitively, let us run a simple analysis using the change test results on the first 30 days of wind speed data, as shown in Fig. 4.4. Understandably, the change points in Fig. 4.4 are not exactly the regime change points, because the regime change points are defined using a set of prescribed wind speed or wind direction thresholds, whereas the change points in Fig. 4.4 are identified through a statistical significance test. Nevertheless, both types of changes serve a similar purpose, which is to identify a segment of time series data for which either the wind speed or the wind direction or both can be assumed relatively stationary. If the runlength is relevant to one, it ought to be relevant to the other.

The change test results in Fig. 4.4 suggest that there exist 43 change points in wind speed out of the 720 data points. For each of the 720 observations, one can compute the corresponding runlength, forming a 720 × 1 vector, namely $[x(1), \ldots, x(720)]^T$, where $x(1) = 0$. For instance, if the first change point was observed at $\ell = 16$, then $x(15) = 15$, $x(16) = 16$, but $x(17) = 1$, and so forth. Fig. 4.5, left panel, illustrates the runlength values for the first 100 points, where change points are marked by the crosses. Note how the runlength grows linearly with time, reaches its peak at change points, and then resets to one right after the change.

The 720 data points are subsequently grouped into two classes: the time points deemed as "not a change point," like at $\ell = 15$ and $\ell = 17$ as mentioned above, versus the "change points," like at $\ell = 16$. Fig. 4.5, right panel, presents the boxplots of the runlength values associated, respectively, with the two classes. The difference is remarkable: the median runlengths are 8.0 and 16.0 hours for the two classes, respectively. This means that for a given time point, which could be in the forecasting horizon, say at $t+h$, the larger its runlength $x(t+h)$, the more likely a change will occur. On the contrary, a small runlength makes it more likely that the wind follows the most recently observed pattern.

4.3.3 Framework of Calibrated Regime-switching

The basic idea of the CRS approach is as follows. Assume that a *base model*, $\mathcal{M}$, can produce a spatio-temporal forecast, $\hat{V}_i(t + h)$, at the i-th site and time $t + h$. This base model, $\mathcal{M}$, could be a spatio-temporal model yield-

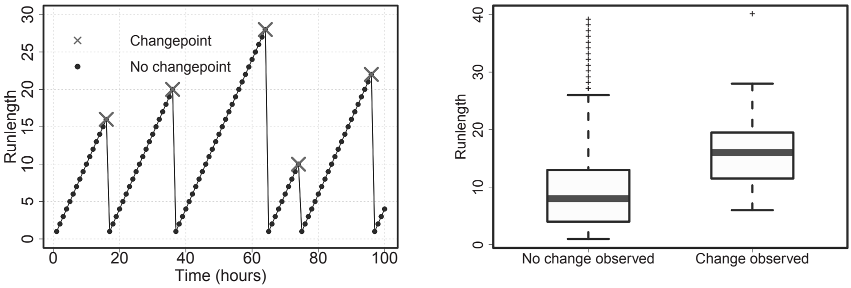

FIGURE 4.5 Left panel: runlength as a function of time. Right panel: boxplots of the runlengths for data points at which no change was observed versus those for data points at which a change was observed.

ing kriging-based forecasts, as we discuss in Chapter 3. Admittedly, this base model produces reactive, albeit regime-specific, forecasts. CRS seeks to calibrate the reactive forecasts to safeguard against upcoming, out-of-sample regime changes, by adding a regime-dependent term, $c_i^{r(t)}(t+h) \in \mathbb{R}$, to the raw forecast, $\hat{V}_i(t+h)$. This additional term, $c_i^{r(t)}(t+h)$, is referred to as the regime-dependent *forecast calibration*, and the quantity $\hat{V}_i(t+h)+c_i^{r(t)}(t+h)$ as the *calibrated forecast*. The idea behind CRS is illustrated in Fig. 4.6, where the goal of the calibration is to adjust the forecast at $t+h$ in anticipation of a regime change.

Determining the sign and magnitude of $c_i^{r(t)}(t+h)$ is arguably the most critical aspect of the CRS approach. Ezzat et al. [60] assume that the sign and magnitude of the forecasting calibration, $c_i^{r(t)}(t+h)$, can be informed by the observed data up to time t, denoted by $\mathbb{D}_t$. The dependence on $\mathbb{D}_t$ is signified by the notation, $c_i^{r(t)}(t+h|\mathbb{D}_t)$. For simplicity, $c_i^{r(t)}(t+h|\mathbb{D}_t)$ is assumed to only vary over time but be fixed across space, that is, $c_i^{r(t)}(t+h|\mathbb{D}_t) = c^{r(t)}(t+h|\mathbb{D}_t)$, for $i = 1, \cdots, N$. A general formulation to infer $c^{r(t)}(\cdot)$ can be expressed as

$$\min_{c^{r(t)} \in \mathcal{C}} \quad L\big[\hat{V}_i(t+h) + c^{r(t)}(t+h|\mathbb{D}_t), V_i(t+h)\big], \tag{4.15}$$

where $\mathcal{C}$ is some class of functions to which $c^{r(t)}(\cdot)$ belongs, and $L[\cdot,\cdot]$ is a loss function that defines a discrepancy measure. To solve Eq. 4.15, $c^{r(t)}(\cdot|\mathbb{D}_t)$ ought to be parameterized.

Based on the discussion in Section 4.3.2, the sign and magnitude of a forecasting calibration is determined through the observed values of the two change indicators, $r(t)$ and $x(t+h)$. Ezzat et al. [60] further propose to use a log-normal cdf to characterize $c^{r(t)}(\cdot)$'s relationship with the two inputs. The choice of the lognormal cdf as a calibration function is motivated by its flexibility to model a wide spectrum of regime-switching behavior, ranging

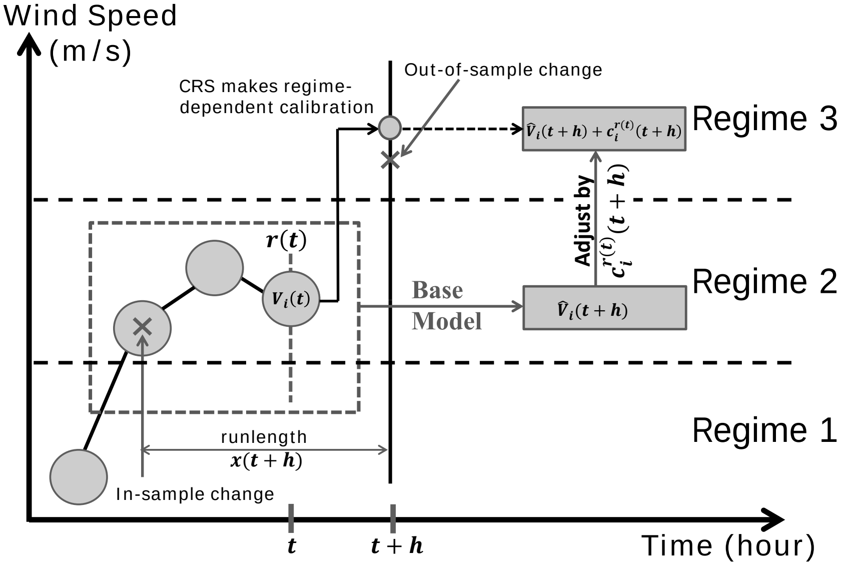

FIGURE 4.6 Illustration of the forecasting calibration for out-of-sample changes.

from abrupt shifts to gradual drifts, depending on the values of its parameters that are learned from the data.

Given R pre-defined wind regimes, $c^{r(t)}(\cdot)$ is modeled individually in each of them. The current regime information, $r(t)$, is then implicitly incorporated by the regime partition, as $c^{r(t)}(\cdot)$ uses the parameters specific to that particular regime. Consequently, the characterization of $c^{r(t)}(\cdot)$ has only the runlength variable, $x(t+h)$, as an explicit input. For the k-th regime, let us denote the regime-dependent parameters by $\Psi^k = \{\psi_1^k, \psi_2^k, \psi_3^k\}$, so that the regime-specific calibration function can be denoted as $c(x(t+h); \Psi^k|\mathbb{D}_t)$ and the superscript $r(t)$ is dropped. The log-normal cdf has the form of

$$c(x(t+h); \Psi^k) = \psi_1^k \Phi\left(\frac{\ln(x(t+h)) - \psi_2^k}{\psi_3^k}\right).$$

CRS aims to learn Ψ^k for each regime using the historical training data and continuously update them during the rolling forward forecasting.

The estimation procedure goes as follows. Assume that an analyst has at hand a sequence of forecasts obtained via a base model, $\mathcal{M}$, and their corresponding true observations. These forecasts are obtained in a rolling forward fashion, such that for the ℓ-th roll, the data observed up to time t^ℓ are used to obtain forecasts from $t^\ell + 1$ till $t^\ell + H$. Then, the window is slid by a specified interval, say s, so that the "present time" for the next forecasting roll is $t^{\ell+1} = t^\ell + s$. Suppose that there are $\mathfrak{L}$ forecasting rolls in the training set. For the ℓ-th forecasting roll, $\ell = 1, \ldots, \mathfrak{L}$, the following information is saved—the observed wind regime at the time of forecasting, $r(t)$, the associated runlength, $x^\ell(t+h)$, the raw forecast via $\mathcal{M}$, $\hat{V}_i^\ell(t+h)$, and the actual observation at $t+h$, $V_i^\ell(t+h)$. By employing a squared error loss function, the optimization problem of Eq. 4.15 can be re-written as,

$$\min_{\Psi^k} \frac{1}{\mathfrak{L}^k \times N \times H} \sum_{\ell=1}^{\mathfrak{L}^k} \sum_{i=1}^{N} \sum_{h=1}^{H} \left[\hat{V}_i^\ell(t+h) + c(x^\ell(t+h); \Psi^k) - V_i^\ell(t+h)\right]^2 \tag{4.16}$$

where $\mathfrak{L}^k$ denotes the number of forecasting rolls relevant to regime k. Solving Eq. 4.16 for each regime individually, i.e., for $k = 1, \ldots, R$, gives the least-squared estimate of the parameters in $\{\Psi^k\}_{k=1}^R$.

Table 4.3 presents the features of various forecasting models. A checkmark "✓" means the presence of that feature, whereas a cross "X" means absence. The last column indicates the piece of information on which a method is actively invoked as a forecasting indicator. Please note that methods like ASYM, SEP and PER do not explicitly consider a wind regime and they are usually not included as a regime-switching approach. Nevertheless, they can be considered as a special case of reactive regime-switching, which has always a single regime and assume that the same regime continues in the forecast horizon. For this reason, ASYM, SEP, PER, RSAR, and RST are collectively referred to as the reactive methods.

TABLE 4.3 Features of various forecasting models.

Method	Temporal	Spatial	Asym- metry	In sample	Out of sample	Regime indicators
PER	X	X	X	X	X	X
SEP	✓	✓	X	X	X	X
ASYM	✓	✓	✓	X	X	X
RSAR	✓	X	X	✓	X	$r(t)$
MSAR	✓	X	X	✓	✓	$\{r(t), \mathbf{\Pi}\}$
RST	✓	✓	✓	✓	X	$r(t)$
CRS	✓	✓	✓	✓	✓	$\{r(t), x(t+h)\}$

4.3.4 Implementation Procedure

To run a CRS comprises three sequential phases: (1) Phase I: generating the raw forecasts (via the base model $\mathcal{M}$) in the initialization period, (2) Phase II: learning the forecasting calibration function based on the raw forecasts and the actual observations solicited in the initialization period, (3) Phase III: making continuous rolling-forward forecasting and updating. Phases I and II use a subset of the data, say, the first month of data, to set up the CRS model. In Phase III, the actual forecasting and testing are carried out on the remaining months in the dataset. Fig. 4.7 presents a diagram for understanding the implementation of CRS.

Phases I and II are the training stage. Without loss of generality, the base spatio-temporal model, $\mathcal{M}$, is assumed to be parameterized by a set of parameters in Θ and thus denoted as $\mathcal{M}(\Theta)$.

The rolling mechanism in Phase I goes as follows. The first roll of training data is the first 12-hour data. Using the 12-hour data, the model parameters Θ are estimated and the raw forecasts from $t+1$ till $t+H$ are made. The regime information, $r(t)$, and the forecasts, $\hat{V}_i(t+h)$, $h = 1, \ldots, H$, are saved for subsequent training. Then, the window is slid by a pre-specified interval s and all data points within that sliding interval are revealed, so that the runlength, $x(t+h)$, and the actual wind speed, $V_i(t+h)$, can be recorded and saved, too. Next, one is ready to make a new forecast, and for that, one needs to re-estimate Θ using the newly revealed data. One thing to bear in mind is that if the sliding interval contains any change points, one should use only the "relevant" data for estimating Θ. The "relevant" data refer to those from the most recent stationary data segment leading to the present time. For instance, Ezzat et al. [60] consider temporal lags for up to 4 hours into history. If the immediate past regime change happens within four time lags from the present time, Ezzat et al. use data with an even shorter time history, which is since the immediate past regime change. This rolling mechanism is continued until all data in the initialization period is exhausted, supposedly resulting in $\mathfrak{L}$ rolls.

Once Phase I is finished, the goal of Phase II is to learn the calibration

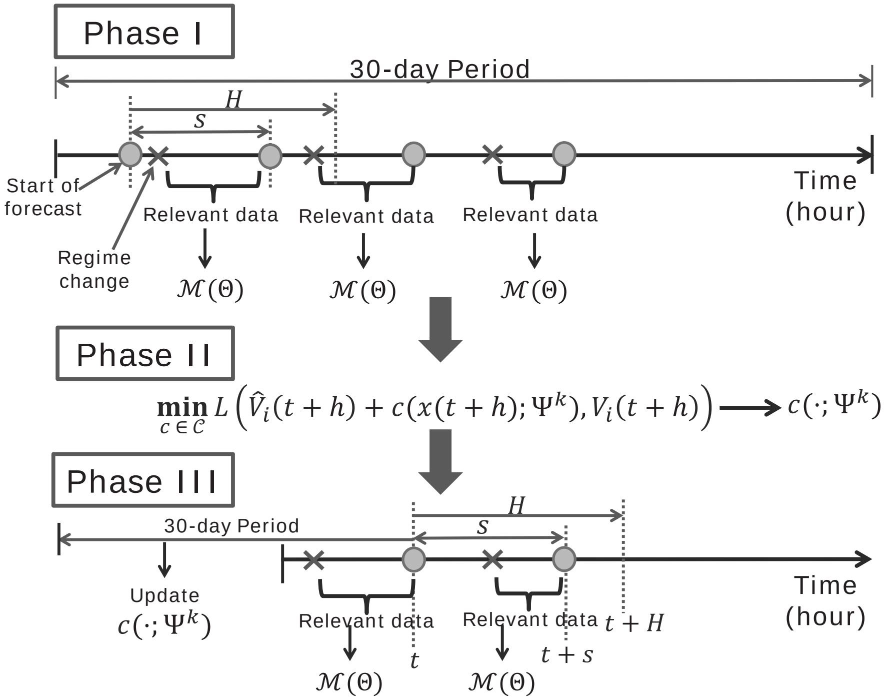

FIGURE 4.7 Steps and notations in the execution of the calibrated regime-switching approach.

function, $c(x(t+h);\Psi^k)$, using the Phase I data, where Eq. 4.16 is solved for each regime individually to estimate the regime-dependent parameters Ψ^k.

Then, proceed to Phase III, where rolling forecasts are performed. At the present time t, one should first look back and search for the most recent in-sample change point. Again, only the "relevant data," defined the same as before, are used to estimate the base model parameters in Θ. The base model is used to make the raw forecasts. The $c(x(t+h);\Psi^k)$, $h = \{1,\ldots,H\}$, is calculated based on the knowledge of the current wind regime, the runlength, and Ψ^k. The resulting $c(x(t+h);\Psi^k)$ is used to calibrate the raw forecasts.

The window is then slid by s. At $t+s$, first use the last 30 days of data to update Ψ^k by re-solving Eq. 4.16 for $k = 1,\ldots,R$, given the newly revealed observations, then estimate the base model parameters in Θ using the "relevant data," and finally, make forecasts for $t+s+h$, $h = 1,\ldots,H$. The cycle is repeated until the forecasts for all the remaining months are produced.

4.4 CASE STUDY

This section applies the calibrated regime-switching method, together with a few alternatives, to the yearlong `Wind Spatio-Temporal Dataset2`. The performances of the respective methods are illustrated and compared.

4.4.1 Modeling Choices and Practical Considerations

In this analysis, the forecast horizon is up to $H = 12$. The sliding interval is set to $s = 6$ hours, meaning that after each roll, the first six hours of the forecast horizon are revealed, and the horizon is shifted by another six hours. This value appears reasonable considering the frequency at which forecasts are updated in practice.

The base model used in CRS is the non-separable, asymmetric spatio-temporal model presented in Section 3.4.1 and the corresponding forecasting model is the kriging method presented in Section 3.4.3. Same as in Section 3.4, by setting the asymmetry and separability parameters to zero, a separable version of the general spatio-temporal model can be obtained.

The base spatio-temporal model used is stationary, but wind fields have been reported to exhibit signs of nonstationarity [69, 166]. By considering only the most recent history of wind speed and direction for model training, it helps overcome the temporal nonstationarity, as the assumption of temporal stationarity is sufficiently reasonable in the short time window since the latest change point. Ezzat et al. [60] account for spatial nonstationarity by assuming local spatial stationarity within a subregion on the wind farm. Three subregions of wind turbines based on their proximity to the three masts are defined, and a region-specific stationary spatio-temporal model is fit and subsequently used for forecasting.

The physically motivated regime definition, as explained in Section 4.1.1, is used here for defining three wind speed regimes. Ezzat et al. [60] also define two wind direction regimes upon observing a dominant east-westward directional wind in the dataset. The combination of the wind speed regimes and wind direction regimes produces a total of $R = 6$ wind regimes.

A further fine-tuning is conducted to adjust the boundaries of the resulting regimes for boosting the performance of the CRS approach. Using the first month of data, the fine-tuning is conducted on 112 different combinations of regime thresholds, chosen as follows: $u_1 = 0$, vary v_1 from V_{ci} to $V_{ci} + 1.5$ with increments of 0.5 m/s, v_2 from $V_{in} - 1.5$ to V_{in} with increments of 0.5 m/s, D_1 from $180° - 45°$ to $180° + 45°$ with 15° increments, and set $D_2 = 360° - D_1$, where D_1 and D_2 are the wind direction thresholds. The fine-tuning based on the `Wind Spatio-Temporal Dataset2` yields the final regime thresholds at 4.5 and 9.0 m/s for wind speed and 45° and 225° for wind direction.

Fig. 4.8 illustrates the learned calibration functions for the six regimes as functions of the runlength. It appears that the wind speed variable is the main factor alluding to the upcoming out-of-sample changes. For instance,

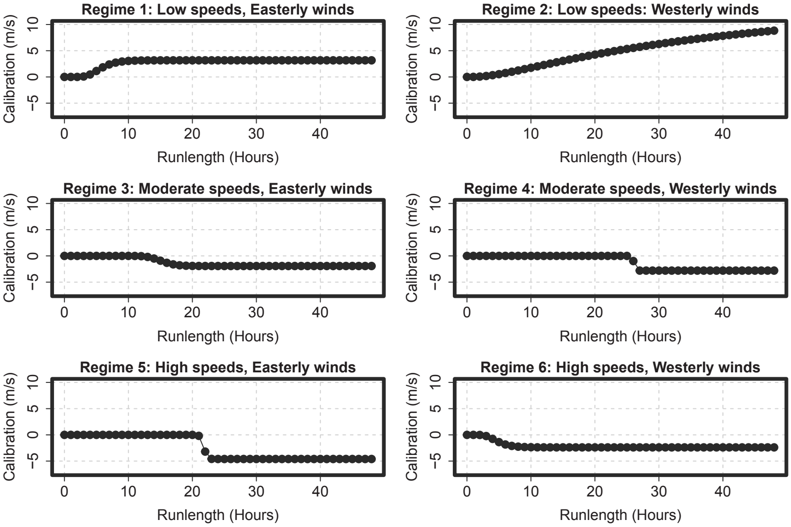

FIGURE 4.8 Learned forecasting calibration functions, $c(x(t+h);\Psi^k)$, using Phase I data for the six regimes. (Reprinted with permission from Ezzat et al. [60].)

the first two regimes (top row), which share the same wind speed profile (low wind speeds), both transit to higher wind speed regimes. Regimes 3 and 4, both with moderate wind speeds, and regimes 5 and 6, both with high wind speeds, likewise have a calibration function of the same pattern to their respective group. The wind direction appears to have a secondary, yet still important, relationship with the magnitude of the out-of-sample change, as well as its timing. For instance, it appears that the magnitude of change is larger in regime 2 (westerly) than in regime 1 (easterly), and larger in regime 4 (westerly) than in regime 3 (easterly). The opposite happens in regimes 5 (easterly) and 6 (westerly). The switching behavior difference between gradual shifts like in regimes 1, 2, 3, and 6 and abrupt shifts like in regimes 4 and 5 also implies a certain degree of interaction between the two factors. These functions may change with time and they are continuously re-estimated in Phase III.

Finally, during the actual testing in Phase III, Ezzat et al. [60] decide to impose maximal and minimal thresholds on the magnitude of forecast calibration to avoid over-calibrating the forecasts when extrapolating. Some numerical experiments indicate that restricting the magnitude of the calibration quantities to the range $(-3, 3)$ m/s yields satisfactory performance. Empirical evidence also suggests that, on average, forecast calibration does not offer much benefit in the very short-term horizon, like less than three hours ahead. For this reason, CRS only calibrates the forecasting for more than three hours ahead (three hours ahead included). This is understandable, since at very short time horizons, wind conditions are more likely to persist than to change drastically.

4.4.2 Forecasting Results

This subsection presents the numerical results comparing CRS with the following approaches: persistence forecast, the asymmetric model, the separable model, the regime-switching autoregressive model, a soft-thresholding Markov-switching model, and a Markov-switch vector autoregressive model (MSVAR) [119]. MSVAR generalizes the MSAR model to further account for the spatial dependence. The temporal order used in RSAR and MSAR (both versions) is one, i.e., $p = 1$.

The aforementioned models are compared in terms of both wind speed and wind power forecasting performances. The forecast accuracy is evaluated using MAE for each h. Specifically, the MAE used in this comparison study is expressed as

$$\text{MAE}(h) = \frac{1}{\mathfrak{L} \times N} \sum_{\ell=1}^{\mathfrak{L}} \sum_{i=1}^{N} \left| \hat{V}_i^{\ell}(t+h) - V_i^{\ell}(t+h) \right|, \tag{4.17}$$

where $V_i^{\ell}(t+h)$ and $\hat{V}_i^{\ell}(t+h)$ are, respectively, the observed and forecasted responses from a forecasting model, obtained at the i-th site and for h-hour ahead forecasting during the ℓ-th forecasting roll, $\ell = 1, ..., \mathfrak{L}$. For each h, MAE is computed as an average over all turbines and forecasting rolls for the eleven-month test data. The MAE values are presented in Tables 4.4 and 4.5, for wind speed and power, respectively. Please note that when computing the MAE for CRS (as well as the PCE below), $\hat{V}_i(t+h)$ is substituted by the calibrated forecast, i.e., $\hat{V}_i(t+h) + c(x(t+h); \Psi^k)$.

The results in Table 4.4 demonstrate that, in terms of wind speed, CRS outperforms the competing models in most forecasting horizons. For $h \geq 2$, the CRS approach renders the best performance among all competing models. This improvement is mainly due to the use of regime-specific calibration functions, which help anticipate the out-of-sample regime changes hinted by runlength. Additional benefits over temporal-only and separable spatio-temporal models come from the incorporation of comprehensive spatio-temporal correlations and flow-dependent asymmetries. For the very short-term horizon, $h = 1$, PER offers the best performance, with CRS slightly behind, but still enjoying a competitive performance.

Fig. 4.9, upper panel, presents the percentage improvements, in terms of MAE and wind speed forecast, that the CRS approach has over the competing models at different forecast horizons. The percentage improvement over reactive methods such as ASYM, SEP, RSAR and PER is more substantial as the look-ahead horizon increases. This does not come as a surprise since the farther the look-ahead horizon is, the more likely a change will occur in that horizon, and hence, the benefit of using CRS is more pronounced.

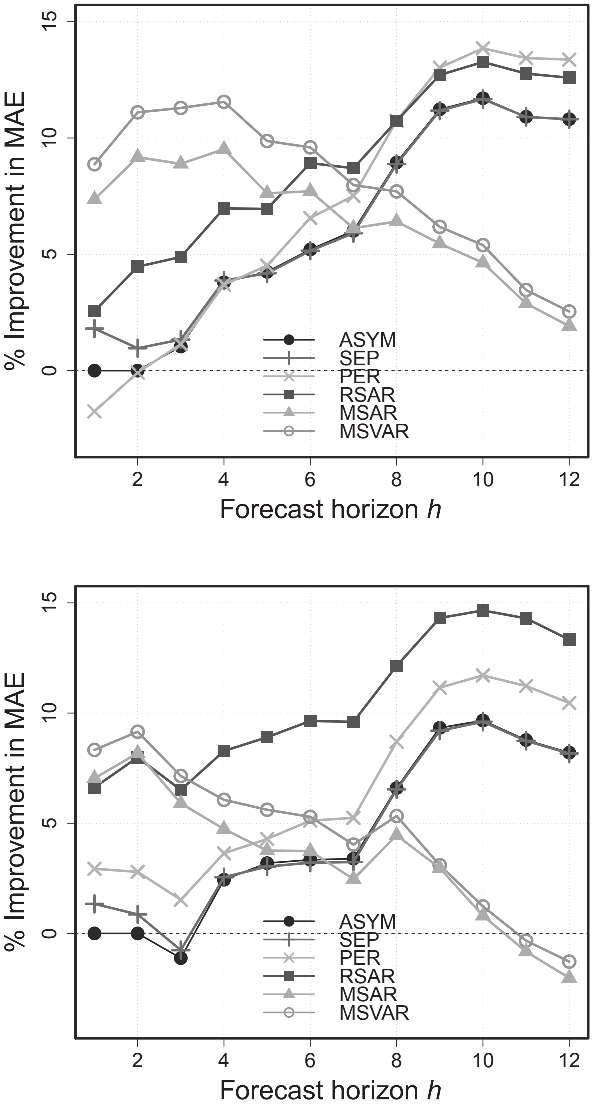

FIGURE 4.9 Percentage improvements in terms of MAE that CRS has over the competing approaches in wind speed (upper panel) and in wind power (lower panel). (Reprinted with permission from Ezzat et al. [60].)

TABLE 4.4 MAE for wind speed forecasting for h-hour ahead, $h = 1, 2, \ldots, 12$. Bold-faced values indicate best performance.

Method	1	2	3	4	5	6
ASYM	1.12	**1.45**	1.72	1.96	2.15	2.27
SEP	1.15	1.47	1.74	1.97	2.15	2.27
PER	**1.11**	1.46	1.73	1.97	2.16	2.31
RSAR	1.16	1.53	1.79	2.03	2.21	2.36
MSAR	1.23	1.64	1.92	2.14	2.28	2.38
MSVAR	1.21	1.60	1.87	2.09	2.23	2.33
CRS	1.12	**1.45**	**1.71**	**1.89**	**2.06**	**2.15**
	7	8	9	10	11	12
ASYM	2.39	2.51	2.68	2.77	2.83	2.87
SEP	2.40	2.52	2.68	2.77	2.84	2.87
PER	2.44	2.57	2.74	2.84	2.92	2.96
RSAR	2.46	2.56	2.73	2.82	2.89	2.93
MSAR	2.45	2.48	2.54	2.59	2.62	2.63
MSVAR	2.40	2.45	2.52	2.57	2.60	2.61
CRS	**2.25**	**2.29**	**2.37**	**2.44**	**2.52**	**2.56**

Source: Ezzat et al. [60]. With permission.

The trend of the improvement of CRS over the Markov-switching approaches, i.e., MSAR and MSVAR, is different. The Markov-switching approaches anticipate regime changes in the look-ahead forecast horizon, too, but use a different mechanism (the transition probabilities). For short-term horizons, the performance of CRS is remarkably better than the Markov-switching approaches. As the look-ahead horizon increases, the advantage of CRS over the Markov-switching models reaches a peak around $h = 4$ hours, and after that, the performance of the Markov-switching approaches gradually catches up with that of CRS. The difference between CRS and the Markov-switching approaches highlights the merit of using the runlength to anticipate the out-of-sample changes. The inclusion of runlength and regime information in CRS appears to offer higher sensitivity, and thus more proactivity, to out-of-sample changes than that offered by the transition probabilities in the Markov-switching approaches.

Similar findings are extended to the power prediction results in Table 4.5, in which CRS is shown to outperform the competing models for most forecasting horizons. Its improvement over the reactive methods is also higher as the look-ahead horizon increases, whereas its improvement over the Markov-switching approaches is best in the shorter forecast horizons. The percentage improvements shown in Fig. 4.9, lower panel, are somewhat different from

TABLE 4.5 MAE values for wind power forecasting for h-hour ahead, $h = 1, 2, \ldots, 12$. Bold-faced values indicate best performance.

Method	1	2	3	4	5	6
ASYM	**0.121**	**0.156**	**0.184**	0.212	0.227	0.236
SEP	0.123	0.158	0.185	0.212	0.227	0.236
PER	0.125	0.161	0.189	0.215	0.230	0.241
RSAR	0.129	0.169	0.199	0.226	0.241	0.253
MSAR	0.132	0.171	0.200	0.220	0.233	0.242
MSVAR	0.131	0.170	0.198	0.217	0.228	0.238
CRS	**0.121**	**0.156**	0.186	**0.207**	**0.220**	**0.229**
	7	8	9	10	11	12
ASYM	0.247	0.261	0.280	0.291	0.294	0.296
SEP	0.247	0.261	0.280	0.292	0.295	0.296
PER	0.253	0.268	0.286	0.299	0.303	0.304
RSAR	0.264	0.278	0.297	0.309	0.314	0.314
MSAR	0.249	0.258	0.263	0.267	0.268	.269
MSVAR	0.245	0.256	0.262	0.266	**0.267**	**0.267**
CRS	**0.239**	**0.244**	**0.254**	**0.263**	0.268	0.271

Source: Ezzat et al. [60]. With permission.

their counterparts in the upper panel. The difference is mainly due to the nonlinear speed-power conversion used in computing wind power.

In addition to MAE, Table 4.6 presents the average PCE errors across all forecasting horizons, for values of ξ ranging between 0.5 and 0.8 with a 0.1 increment, as well as $\xi = 0.73$, which is the value recommended in [91]. It appears that the improvement of CRS over the competing models is also realizable in terms of PCE. The CRS approach performs well when under-estimation is penalized more severely than over-estimation (namely $\xi > 0.5$), which describes the more realistic cost trade-off in power systems.

TABLE 4.6 Average PCE for competing models across all horizons. Bold-faced values indicate best performance.

Method	$\xi = 0.5$	$\xi = 0.6$	$\xi = 0.7$	$\xi = 0.73^*$	$\xi = 0.8$
ASYM	0.116	0.117	0.114	0.111	0.104
SEP	0.116	0.118	0.114	0.112	0.105
PER	0.118	0.121	0.124	0.125	0.127
RSAR	0.123	0.123	0.120	0.117	0.110
MSAR	0.113	0.123	0.127	0.124	0.126
MSVAR	0.112	0.118	0.122	0.118	0.119
CRS	**0.109**	**0.110**	**0.107**	**0.105**	**0.097**

Source: Ezzat et al. [60]. With permission.

GLOSSARY

AR: Autoregressive model

AR-D: Autoregressive model fit after the diurnal pattern is removed

AR-N: Autoregressive model fit to the original data

ARMA: Autoregressive moving average

BIC: Bayesian information criterion

cdf: Cumulative distribution function

CRPS: Continuous ranked probability score

CRS: Calibrated regime switching

GMM: Gaussian mixture model

GSTAR: Gaussian spatio-temporal autoregressive model

MAE: Mean absolute error

MSAR: Markov-switching autoregressive model

MSVAR: Markov-switching vector autoregressive model

PCE: Power curve error

pdf: Probability density function

PER: Persistence forecasting

RMSE: Root mean squared error

RSAR: Regime-switching autoregressive model

RSGSTAR: Regime-switching Gaussian spatio-temporal autoregressive model

RST: Regime-switching space time model

RST-D: Regime-switching space time model fit after the diurnal pattern is removed

RST-N: Regime-switching space time model fit to the original data

SEP: Separable spatio-temporal model

STAR: Smooth transition autoregressive model

TDD: Trigonometric direction diurnal model

EXERCISES

4.1 Use the `Wind Time Series Dataset` and conduct the following exercise.

a. Use the three pre-defined wind speed regimes, $[0, 4.5)$, $[4.5, 9.0)$ and $[9.0, 20)$, and fit three AR models to the hourly data of April and May. To select the model order for the AR models, please use BIC.

b. Use the hourly data of April and May to fit a single AR model. Still use BIC to decide the model order. Compare the AR model in (b) with the three AR models in (a).

c. Use the AR models in (a) to make one-hour ahead rolling forward forecasts for the next ten hours. The regime for one hour ahead is assumed the same as the current regime. Compute the MAE of the ten one-hour ahead forecasts.

d. Use the AR models in (b) to make one-hour ahead rolling forward forecasts for the next ten hours. Compute the MAE of the ten one-hour ahead forecasts. Compare the MAEs obtained in (c) and (d). What do you observe?

4.2 Use the `Wind Time Series Dataset` and fit a Gaussian mixture model to the yearlong hourly data. Here you do not have the wind direction data. So instead of fitting a bivariate Gaussian distribution, like in Eq. 4.2, you will fit a univariate Gaussian distribution.

a. Explore the number of regimes between one and five. Use the BIC to decide the best number of regimes.

b. Using the R decided in (a) and the associated GMM parameters, compute the weight w_k in Eq. 4.3 for wind speed between 0 m/s and 20 m/s with an increment of 1 m/s. Do this for $k = 1, \ldots, R$ and make a plot of w_k to demonstrate how each regime model is weighted differently as the wind speed changes.

4.3 Use the hourly data in `Wind Time Series Dataset` and assume three pre-defined wind speed regimes, $[0, 4.5)$, $[4.5, 9.0)$ and $[9.0, 20)$. Conduct the following exercise.

a. Go through the first half year's data, i.e., January through June. At any data point, label the wind speed's current regime (namely, at t) as well as the regime at the next hour (namely, at $t+1$). For the entire half year of data, count the regime switching numbers between the three regimes, including the case of remaining in the same regime. Note that the regime switching from 1 to 2 and that from 2 to 1 are counted as different regime switchings. Then, divide each count by the total number of switchings. The relative frequency provides the empirical estimate of π_{ij}. Please write down the 3×3 transition probability matrix $\mathbf{\Pi}$. Verify if each row sums to one.

b. Do the same for the second half year's data, i.e., July through December. Compare the new $\mathbf{\Pi}$ with that obtained in (a). Do you find any noticeable difference between the two $\mathbf{\Pi}$'s?

4.4 If $F(\cdot)$ is the predictive cdf and V is the materialized wind speed, the continuous ranked probability score is defined as

$$\text{crps}(F, V) = \int_{-\infty}^{\infty} \left(F(x) - \mathbb{1}(x \geq V)\right)^2 dx.$$

The expression in Eq. 2.60 is the sample average based on n_{test} observations, namely

$$\text{CRPS} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \text{crps}(\hat{F}, V_i).$$

Please derive the closed-form expression of $\text{crps}(F, V)$ when $F(\cdot)$ is a normal distribution.

4.5 The cdf of the truncated normal distribution, $\mathcal{N}^+(\mu, \sigma^2)$, is

$$F(x) = \frac{\Phi(\frac{x-\mu}{\sigma}) - \Phi(-\frac{\mu}{\sigma})}{1 - \Phi(-\frac{\mu}{\sigma})} \tag{P4.1}$$

when $x \geq 0$, and $F(x) = 0$ when $x < 0$. Please drive the closed-form expression of $\text{crps}(F, V)$ for the truncated normal distribution, which is the expression inside the summation in Eq. 4.13.

4.6 Use the wind speed data in `Wind Spatio-Temporal Dataset2`. Select three turbines from the wind farm, the west-most turbine, the east-most turbine, and a turbine roughly halfway from the two turbines on the periphery. If possible, try to select the turbines on a similar latitude. Use the average of the wind directions measured on the three met masts to represent the wind direction for the wind farm. Create four wind regimes—the easterly, southerly, westerly, northerly regimes of which the wind direction ranges are, respectively, $(45°, 135°)$, $(135°, 225°)$, $(225°, 315°)$, and $(315°, 45°)$. Use the first two months of data associated with the three turbines to fit four separate AR models, each of which has the same structure as in Eq. 4.10. Doing this yields a four-regime RST method. Use this RST method to make forecasts at the east-most turbine for $h = 2$. Shift the data by one month and repeat the above actions, and then, repeat for the whole year. One gets eleven 2-hour ahead forecasts. Compute the MAE and RMSE for these $h = 2$ forecasts.

4.7 Take the first month of wind direction data from a met mast and implement the circular variable detection algorithm to detect the change points. How many change points are there? Are the minimum-time-to-change and median-time-to-change different from those values reported on page 105?

4.8 Use the change-point detection results from the previous problem and produce boxplots similar to that in Fig. 4.5, right panel. Is there a noticeable difference between the two resulting boxplots? How do you feel using the runlength as a change indicator for a wind direction-based regime-switching method?

4.9 Test the sensitivity of the CRS approach by comparing the following competing alternatives:

a. No forecasting calibration for $h = 1$ and $h = 2$ versus conducting calibration for $h = 1$ and $h = 2$.

b. Cap the magnitude of the calibration quantities to the range $[-3, 3]$ versus $[-2, 2]$, or $[-5, 5]$, or no restriction at all.

c. Three wind speed regimes, with boundary values at 4.5 and 9.0 m/s, versus four wind speed regimes, with boundary values at 3.5, 9.5, and 13.5 m/s.