

ciency. In both applications, accurate modeling of the power curve is essential, as it entrusts subsequent decision making.

In Chapter 5, we dedicate ourselves to various methods for power curve modeling and analysis. Chapter 6 discusses the relevance of power curves in turbine efficiency quantification. Chapter 7 focuses on one particular type of turbine change, known as turbine upgrade via retrofitting, and shows how data science methods can help quantify the change in power production due to an upgrade. Chapter 8 presents a study concerning how the wake effect affects a turbine's production performance.

5.1 IEC BINNING: SINGLE-DIMENSIONAL POWER CURVE

The current industrial practice of estimating the power curve relies on a non-parametric approach, known as the binning method, recommended by the International Electrotechnical Commission (IEC) [102]. The basic idea of the binning method is to discretize the domain of wind speed into a finite number of bins, say, using a bin width of 0.5 m/s. Then, the value to be used for representing the power output for a given bin is simply the sample average of all the data points falling within that specific bin, i.e.,

$$y_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{i,j}, \quad (5.1)$$

where $y_{i,j}$ is the power output of the j^{th} data point in bin i , and n_i is the number of data points in bin i .

The physical law of wind power generation [1, 15] states that:

$$y = \frac{1}{2} \cdot C_p \cdot \rho \cdot \pi R^2 \cdot V^3, \quad (5.2)$$

where R is the radius of the rotor and C_p is the power coefficient, which is believed to be a function of (at least) the blade pitch angle and the turbine's tip speed ratio. What else might affect C_p is still a matter under debate. Currently no formula exists to express C_p analytically in terms of its influencing factors. C_p is therefore empirically estimated. Turbine manufacturers provide for a specific turbine its nominal power curve with the corresponding C_p values under different combinations of wind speed, V , and air density, ρ . The above expression also provides the rationale why temperature, T , and air pressure, P , are converted into air density, ρ , to explain wind power, rather than used individually.

Even though the expression in Eq. 5.2 on the surface suggests that the electrical power that a wind turbine extracts from the wind is proportional to V^3 , an actual power curve may exhibit a different nonlinear relationship. This happens because the tip speed ratio is a function of wind speed, V , making C_p also a function of V and adding complexity to the functional relationship

between wind speed and wind power. Another complexity is brought by turbine controls. The power law in Eq. 5.2 governs the wind power generation before the rated wind speed, V_r . The use of the pitch control mechanism levels off, and ultimately caps, the power output when it reaches the rated power output, y_r . Recall the shape of the power curve shown in Fig. 1.2. The power curve has an inflection point somewhere near the rated wind speed, so that the whole curve consists of a convex segment, between V_{ci} and the inflection point and a concave segment, between the inflection point and V_{co} .

Given the physical relation expressed in Eq. 5.2, the wind industry recognizes the need to include air density as a factor in calculating the power output, and does so through a formula known as the air density correction. If V is the raw average wind speed measured in a 10-minute duration, the air density correction is to adjust the wind speed based on the measured average air density, ρ , in the same 10-minute duration, namely

$$V' = V \left(\frac{\rho}{\rho_0} \right)^{\frac{1}{3}}, \quad (5.3)$$

where ρ_0 is the sea-level dry air density ($=1.225 \text{ kg/m}^3$) per the International Organization for Standardization's atmosphere standard. The binning method with air density correction uses this corrected wind speed, V' , and the power output, y , to establish a power curve. In the subsequent numerical analysis, by "binning method" we refer to this air density corrected version, unless otherwise noted. To make the notation simpler, we continue using V to denote the wind speed even after the air density adjustment.

Another adjustment analysts practice in the wind industry is to identify the free sectors for a wind turbine. A free sector is a subset of wind directions under which a wind turbine is supposedly free of wake effect from its neighboring turbines. The use of a free sector is effectively a filtering action, which often removes as many as two-thirds of the raw data.

Please note that the IEC binning method only provides the estimation of the average power curve, not that of the conditional density. One simple way to get the conditional density is to assume a distribution type for a given bin, say, Gaussian, and use the data in that bin to estimate the parameters in the assumed distribution. In this way, the density estimation for the whole input region is the collection of a bunch of bin-based individual density estimations.

5.2 KERNEL-BASED MULTI-DIMENSIONAL POWER CURVE

Wind power production is apparently affected by more than just wind speed. The current IEC method, explained above, primarily considers wind speed, while using the air density information in an *ad hoc* manner. The IEC method does not actually use the wind direction information—it simply controls for that condition. The power curve established under the free sector has a poor predictive capability for wind power production under general wind conditions. In this sense, the IEC method is not created for power prediction or

turbine performance assessment purposes. Rather, the IEC's intention is to create a standardized condition when a turbine's power production can be compared and verified. Accomplishing this is important for activities like contracting, in which a manufacturer's claim of its wind turbine's production ability ought to be verified at the time of a transaction, under a condition agreed upon by both parties.

For the purposes of wind power prediction, turbine control, and turbine performance assessment, all under general wind directions, it is more desirable to have a multi-dimensional power curve that can account for the effects of as many environmental variables as possible. Some works [17, 109, 165, 191] study the impact of having wind direction incorporated as one of the input variables, or as an additional covariate, and find that much can be gained by this inclusion. Bessa et al. [17] also include in their power curve a third covariate, in addition to wind speed and wind direction, which is either the time of the day or a look-ahead time step. Lee et al. [132] present one of the first truly multi-dimensional power curve models, referred to as the additive-multiplicative kernel (AMK) method, for both mean estimation, namely $\mathbb{E}(y|\mathbf{x})$, and for the density estimation, namely $f(y|\mathbf{x})$. The AMK power curve model can, in principle, take as many inputs as possible, although the test cases included in [132] use up to seven covariates.

5.2.1 Need for Nonparametric Modeling Approach

The underlying physics of wind power generation expressed in Eq. 5.2 provides some clues concerning a preferable power curve model. The following summarizes the observations:

1. There appear to be at least three important factors that affect wind power generation: wind speed, V , wind direction, D , and air density, ρ . This does not exclude the possibility that other environmental factors may also influence the power output.
2. The functional relationships between the environmental factors and the power response are generally nonlinear. The complexity partially comes from the lack of understanding of C_p , which is affected by many environmental factors (V , D , and ρ included). As there is no analytical expression linking C_p to any of the influencing factors, the functional form of the power curve is unknown.
3. The environmental factors appear in a multiplicative relationship in the power law equation, Eq. 5.2, indicating interactions among the factors.

The lack of precise physical understanding in power curve modeling presents an opportunity for data science methods. While developing data-driven power curve models, the second observation above makes it a compelling case for needing a nonparametric modeling approach to model a power curve, because the specific functional form of power curves is not known and

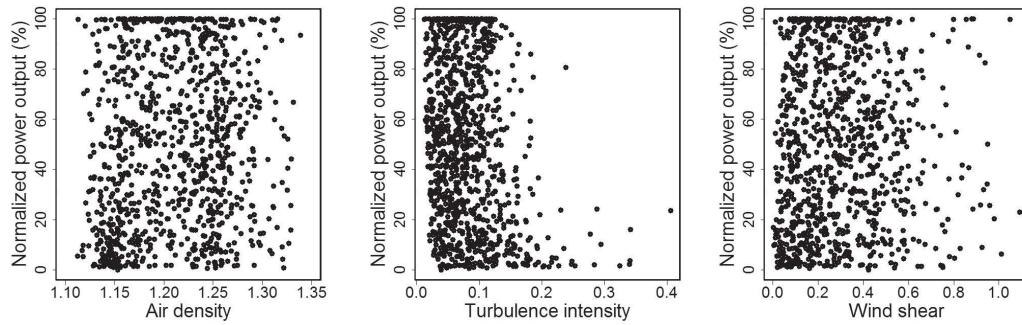


FIGURE 5.1 Scatter plots of the power output versus three environmental factors for $3.5 < V < 20$, $0^\circ < D < 360^\circ$. Left panel: power versus air density; middle panel: power versus turbulence intensity; right panel: power versus wind shear. (Reprinted with permission from Lee et al. [132].)

can be rather different under various circumstances. A parametric approach, by contrast, is to assume a function of a known form with a set of unknown parameters, say, a polynomial function to some degree, and then estimate the unknown parameters using the data. The major shortcoming of the parametric approaches is its lack of flexibility, as there is no guarantee that the assumed functional form captures the true relationship between the power and the environmental inputs. The nonparametric approach, on the other hand, follows the philosophy of “*let the data speak for itself*” and can be much more adaptive without making too many assumptions. The IEC binning method is in and by itself a nonparametric method.

The third observation above touches upon the issue of factor interactions. To see this aspect more pointedly, consider the scatter plots in Figs. 5.1 and 5.2. Fig. 5.1 presents the scatter plots between wind power and three environmental variables. These scatter plots are unconditional on wind speed and wind direction. Under this setting, these environmental factors show no obvious effect on the power output. Fig. 5.2 presents the scatter plots between the same variables but instead under different wind speeds and wind directions. One does observe nonlinear relationships in the conditional plots, and the relationships appear to be different depending on the wind conditions. This implies that interaction effects do exist among wind speed, wind direction, and other environmental factors. A power curve model should hence characterize not only the nonlinear effects of wind speed and wind direction, but also the interaction effects among the environmental factors. The existence of interaction effects suggests that purely additive models or generalized additive models (GAM) are unlikely to work well in modeling a power curve.

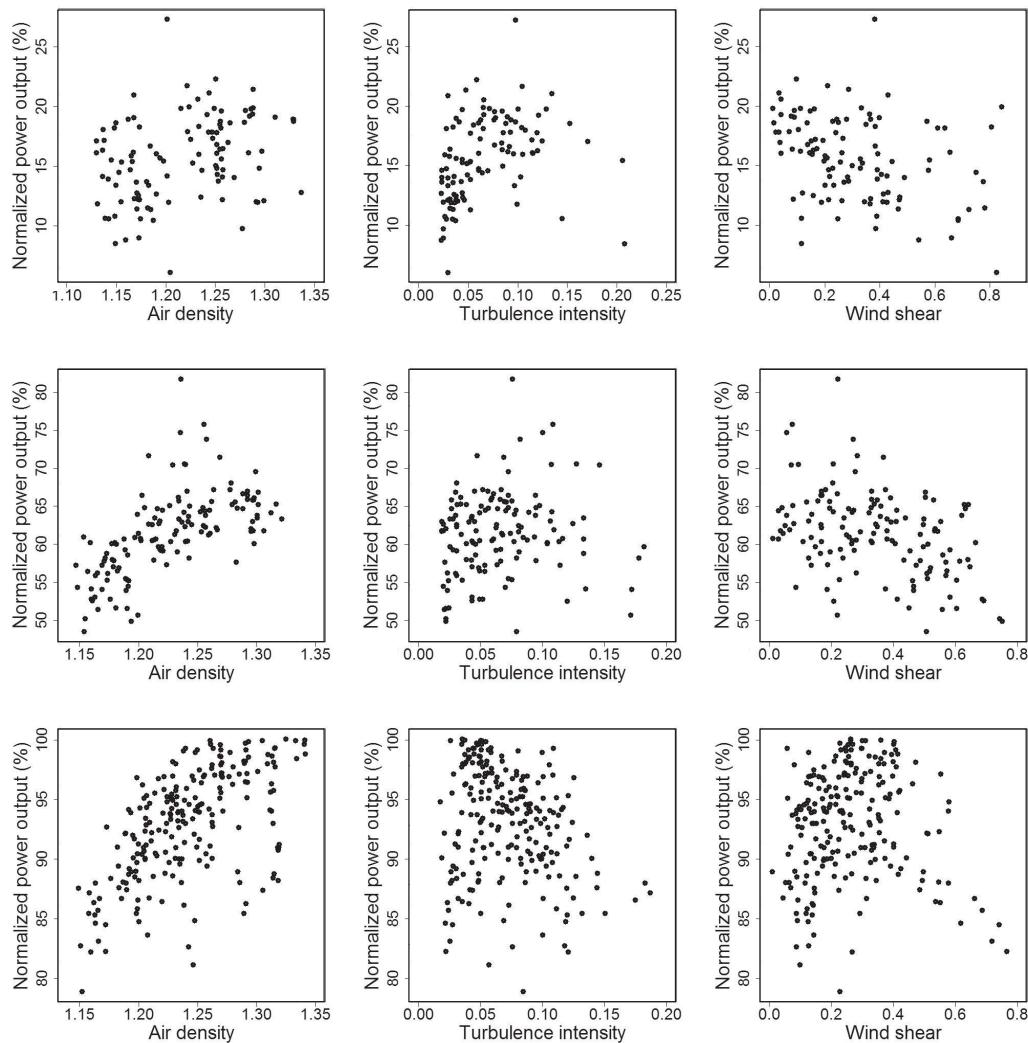


FIGURE 5.2 Scatter plots of power output versus environmental factors under specific wind speeds and wind directions. Top panel: $6.1 < V < 6.2$, $270^\circ < D < 300^\circ$; middle panel: $9.1 < V < 9.2$, $270^\circ < D < 300^\circ$; and bottom panel: $11.1 < V < 11.2$, $270^\circ < D < 300^\circ$. (Reprinted with permission from Lee et al. [132].)

5.2.2 Kernel Regression and Kernel Density Estimation

Kernel regression or kernel density estimation methods have been used for modeling power curves [17, 109, 132]. The kernel-based methods appear to be a capable statistical modeling tool, not only capturing the complicated higher-order interaction effects but also avoiding the need to specify a functional form of the power curve relationship.

A kernel regression [86] is a type of localized regression method, which is to make an estimation, \hat{y} , at a target input value, x_0 , by using observed data points close to x_0 . This can be accomplished by a weighted average of the data points falling into a local neighborhood, such as

$$\begin{aligned}\hat{y}(x_0) &= \sum_{x_i \in \mathfrak{N}(x_0)} w_i y_i, \\ \text{and } \sum_i w_i &= 1,\end{aligned}\tag{5.4}$$

where $\mathfrak{N}(x_0)$ is the neighborhood of x_0 , however it is defined, y_i is the observed response corresponding to input x_i , w_i is the weighting coefficient associated with y_i , and the constraint, $\sum_i w_i = 1$, is to ensure that the magnitude of \hat{y} is consistent with that of y .

In the kernel regression, the localization is achieved by employing a weighting function symmetric with respect to x_0 , known as the kernel function and denoted by $K(x_0, x_i)$. A kernel function is supposed to be integrable to one, following the same rationale above of requiring the summation of w_i 's to be one. The kernel function has a bandwidth parameter λ that controls how fast the function decays from its peak towards zero and effectively defines the neighborhood, $\mathfrak{N}(x_0)$. Consider the one-dimensional Gaussian kernel function, taking the form of a normal probability density function,

$$K_\lambda(x_0, x_i) = \frac{1}{\sqrt{2\pi\lambda^2}} \exp\left(-\frac{\|x_0 - x_i\|^2}{2\lambda^2}\right),\tag{5.5}$$

where λ is equivalent to the standard deviation in a normal pdf. This kernel function is mathematically equivalent to the kernel function used in the support vector machine in Section 2.5.2. The term, $1/\sqrt{2\pi\lambda^2}$, in the above equation is the normalization constant to ensure that $K_\lambda(x_0, x)$ is integrable to one. When $K(\cdot, \cdot)$ is used to define the weighting coefficient, w_i , the normalization constant appears in both the numerator and denominator, so that it is cancelled out. For this reason, analysts have practically omitted the normalization constant and simply write the Gaussian kernel as

$$K_\lambda(x_0, x_i) = \exp\left(-\frac{\|x_0 - x_i\|^2}{2\lambda^2}\right).\tag{5.6}$$

This expression looks the same as that in Eq. 2.48 if one lets $\phi = 1/(2\lambda^2)$.

What matters in the kernel function is the difference, or the distance,

between x_0 and x_i , just like in a stationary covariance function. Therefore, analysts choose to simplify the input arguments in $K(\cdot, \cdot)$ to be a single variable, say, $u = x_0 - x_i$. As such, the kernel function in kernel regression is often denoted by one of the following interchangeable expressions: $K_\lambda(u)$ or $K_\lambda(\|x_0 - x_i\|)$ or $K_\lambda(x_0, x_i)$.

Another way, arguably more common, to write the Gaussian kernel function is to first express it with a unit bandwidth, namely $\lambda = 1$, as

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\|u\|^2}{2}\right). \quad (5.7)$$

This $K(u)$ is integrable to one. It is then used as the building block for kernel function, $K_\lambda(u)$, with an arbitrary bandwidth λ — $K_\lambda(u)$ is referred to as the scaled kernel. Using $K(u)$, $K_\lambda(u)$ is written as

$$K_\lambda(u) = \frac{1}{\lambda} K\left(\frac{u}{\lambda}\right), \quad (5.8)$$

which gives back the expression in Eq. 5.5. The Gaussian kernel expression in Eq. 5.7 and Eq. 5.8 is used throughout the book when a kernel regression or a kernel density estimation is concerned.

Fig. 5.3, left panel, presents an illustration of a Gaussian kernel function. Suppose that one only has three data points, marked as #1, #2, and #3, respectively, and the corresponding data pair is $\{x_i, y_i\}$, $i = 1, 2, 3$. One wants to assess the response, $\hat{y}(x_0)$, at x_0 . The weighting coefficient associated with each one of the data points is decided through the kernel function. Specifically,

$$w_i(x_0) = \frac{K_\lambda(\|x_0 - x_i\|)}{\sum_{j=1}^3 K_\lambda(\|x_0 - x_j\|)}, \quad i = 1, 2, 3,$$

and using this weighting coefficient function, one has

$$\hat{y}(x_0) = w_1(x_0)y_1 + w_2(x_0)y_2 + w_3(x_0)y_3.$$

In this example, as illustrated in Fig. 5.3, left panel, points #1 and #2 have positive weights associated with them, whereas point #3 has a virtually zero weight, so that $\hat{y}(x_0)$ is effectively the weighted average of y_1 and y_2 at points #1 and #2, respectively. Other data points that are even farther away from x_0 than #3 hardly affect the estimation of $\hat{y}(x_0)$ at all. One may consider that the neighborhood of x_0 , $\mathfrak{N}(x_0)$, covers a certain distance from x_0 on either side and contains #1 and #2 but not #3. Eq. 5.4, once factoring in the neighborhood constraint for this three-point case, can be expressed as

$$\hat{y}(x_0) = w_1(x_0)y_1 + w_2(x_0)y_2, \quad w_1(x_0) + w_2(x_0) = 1.$$

If one moves x_0 continuously from one end of the input domain to the other end and estimates $\hat{y}(x_0)$ at every x_0 using the same kernel-based localized

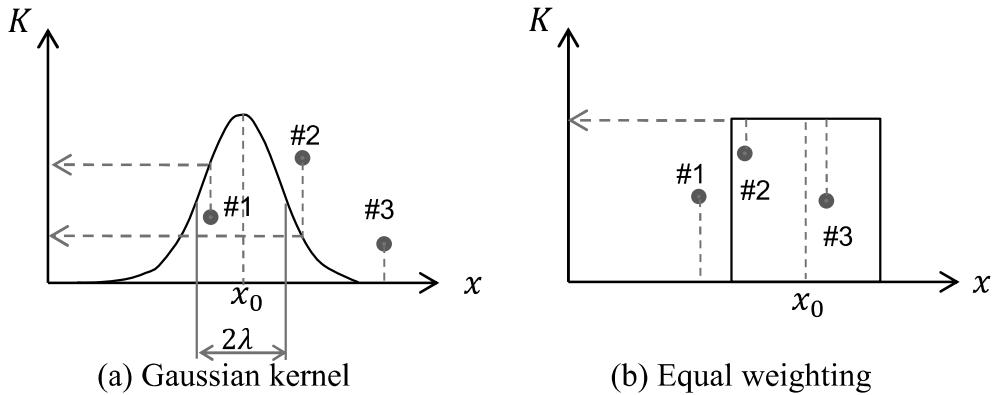


FIGURE 5.3 Gaussian kernel function versus rectangular window function used in IEC binning. (Reprinted with permission from Ding et al. [50].)

regression, one practically estimates the input-output functional relationship between x and y . Let us drop the subscript “0” from the input variable, as it is now a generic input variable. The estimated relationship is denoted by $\hat{y}(x)$. The kernel regression leading to $\hat{y}(x)$, based on n pairs of data points, $\{(x_1, y_1), \dots, (x_n, y_n)\}$, is

$$\hat{y}(x) = \sum_{i=1}^n w_i(x)y_i, \quad (5.9)$$

where $w_i(x) = \frac{K_\lambda(\|x - x_i\|)}{\sum_{j=1}^n K_\lambda(\|x - x_j\|)}$.

This estimator in Eq. 5.9 is in fact the Nadaraya-Watson kernel regression estimator [152, 226]. The kernel function used therein does not have to be always the Gaussian kernel. The Gaussian kernel function does not go exactly to zero unless $\|x - x_i\| \rightarrow \infty$. There are other kernel functions, for instance, the Epanechnikov kernel function, that defines a window, so that the kernel function value is precisely zero outside the window.

The kernel regression is considered a nonparametric approach not because the kernel function does not have parameters or not use a functional form; in fact it does have a parameter, which is the bandwidth. Being a nonparametric approach, the kernel function is different from the target function, $y(x)$, that it aims at estimating. While the target functions may vary drastically from one application to another, the kernel function used in the estimation can remain more or less the same. The non-changing kernel regression is able to adapt to the ever-changing target functions, as long as there are enough data. The parameter in the kernel function serves a role differing substantially from the coefficients in a linear regression. The coefficients in a linear regression play a

direct role in connecting an input x to the response y , while the bandwidth parameter in kernel functions, by contrast, defines the neighborhood and plays a rather indirect role in connecting x to y .

Comparing the kernel regression with the binning method, one notices that the binning method can be considered as a special kernel regression that uses a uniform kernel function; see the illustration in Fig. 5.3, right panel. A uniform kernel function is a rectangular window function, giving equal weights to all data points within the window, regardless of how far away they are from x_0 . Once a data point is outside the function window (point #1), its weight is zero. The final estimate at x_0 is a simple average of all y 's associated with the data points within the window (points #2 and #3 in this case). Of course, the IEC binning method is not really a kernel model due to another important reason: in the kernel regression, the kernel function moves continuously along the x -axis, producing a continuous, smooth curve, while the window functions in the binning method are disjoint, so that the resulting function response from the binning method, if magnified enough, is discretized.

When analysts use the weighted average of y values of the data points falling under the kernel function, the resulting outcome is a point estimate of the function value at a given input x . The kernel method, nonetheless, is capable of producing the estimate of a probability density function of y , conditioned on x , namely $f(y|x)$. The way of doing this is very similar to that in Eq. 5.9 but notice a different requirement here—for a density estimation, the left-hand side is supposed to be a density function, rather than a point estimate. One idea is to replace y_i with a density function, centered at y_i , so that the weighted averaging acts now on a series of density functions and thus results in a density function as well. Recall that the Gaussian kernel function is in fact a density function, and as such, a conditional density estimation [99, 184] can be obtained through a formula like

$$\hat{f}(y|x) = \sum_{i=1}^n w_i(x) K_{\lambda_y}(y - y_i), \quad (5.10)$$

where $w_i(x)$ is likewise defined as in Eq. 5.9, but λ is to be replaced with λ_x , as the bandwidth parameters associated with x may differ from that with y .

5.2.3 Additive Multiplicative Kernel Model

The AMK method [132] is a kernel-based approach. Prior to AMK, a bivariate conditional kernel density including wind speed and direction is used in [109] and a trivariate kernel density is used in [17]. AMK goes beyond the plain version of a kernel density estimation or a kernel regression and employs a special model structure that allows it to handle the multi-dimensional inputs in power curve modeling.

Recall the conditional density estimate $\hat{f}(y|x)$ in Eq. 5.10. For a multivari-

ate input, denoted by \mathbf{x} , let us re-write Eq. 5.10 as

$$\hat{f}(y|\mathbf{x}) = \sum_{i=1}^n w_i(\mathbf{x}) K_{\lambda_y}(y - y_i), \quad (5.11)$$

where

$$w_i(\mathbf{x}) = \frac{K_{\lambda_x}(\|\mathbf{x} - \mathbf{x}_i\|)}{\sum_{j=1}^n K_{\lambda_x}(\|\mathbf{x} - \mathbf{x}_j\|)}, \quad (5.12)$$

$\lambda_x = (\lambda_1, \dots, \lambda_q)$ is the vector of the bandwidth parameters associated with the environmental factors in \mathbf{x} , and q is the number of explanatory variables in \mathbf{x} . In the above formulation, $K_{\lambda_x}(\|\mathbf{u}\|)$, where $\mathbf{u} = \mathbf{x} - \mathbf{x}_i$, is a multivariate kernel function and is composed of a product kernel that is a multiplication of univariate kernel functions, such as

$$K_{\lambda_x}(\|\mathbf{u}\|) := K_{\lambda_1}(u_1) K_{\lambda_2}(u_2) \cdots K_{\lambda_q}(u_q). \quad (5.13)$$

Here $K_{\lambda_j}(u_j)$ is generally a univariate Gaussian kernel, except for wind direction, D . The kernel function for D is chosen to be the von Mises kernel [212], because D is a circular variable that may cause trouble in numerical computation, had a Gaussian kernel been used. For more discussion regarding the handling of circular variables, please refer to [143, 144, 145]. The von Mises kernel function can characterize the directionality of a circular variable and takes the form of

$$K_\nu(D - D_i) = \frac{\exp\{\nu \cos(D - D_i)\}}{2\pi I_0(\nu)}, \quad (5.14)$$

where $I_0(\cdot)$ is the modified Bessel function of order 0, and ν is the concentration parameter of the von Mises kernel, which has now taken the role of the inverse of the bandwidth parameter λ_D .

In addition, the mean of the conditional density estimator in Eq. 5.11 provides an estimation of the conditional expectation, $\hat{y}(\mathbf{x}) := \mathbb{E}(y|\mathbf{x})$, as

$$\hat{y}(\mathbf{x}) = \int y \hat{f}(y|\mathbf{x}) dy. \quad (5.15)$$

Hydman et al. [99] note that the estimator in Eq. 5.15 is equivalent to the Nadaraya-Watson regression estimator in Eq. 5.9 with the input variable in Eq. 5.9 replaced by its multivariate counterpart, \mathbf{x} .

The AMK method [132] does not simply use the multivariate kernel as is. The reason is two-fold. One is concerning data scarcity in a multi-dimensional space. With wind data arranged in 10-minute blocks, one year's worth of data translates to slightly over 52,000 data pairs, which could still become scarce in a multi-dimensional factor space. The dimensionality of the input space, if using the list of elements in \mathbf{x} given in Section 1.1, is seven. When 52,000 data points are dispersed into the seven-dimensional space, certain combinations of environmental conditions could have very little data or even no

data at all, thereby deteriorating the performance of the resulting multivariate kernel model. In the future, technology innovation will almost surely make additional measurements available, so that a truly multi-dimensional power curve model should be able to entertain as many input variables as realistically possible. The scalability issue is also the reason why the IEC binning is not used for multi-dimensional cases. The second difficulty is that running a multi-dimensional kernel-based conditional density estimation takes longer computational times than analysts typically prefer. It is thus desirable to use fewer input variables to form the multivariate product kernels if possible.

Lee et al. [132] tailor the power curve modeling to an additive-multivariate model structure, which gives their kernel model the name AMK. The idea is to form a series of product kernel functions taking three input variables each, allowing up to three-factor interactions to be modeled. The use of trivariate kernels helps alleviate the data scarcity concern, as a trivariate kernel only needs to handle a three-dimensional space, low enough to avoid the curse of dimensionality. For high-dimensional covariates (more than three), AMK pools multiple trivariate product kernels together in an additive structure.

For notation simplicity, let us designate the first two elements of \mathbf{x} , namely x_1 and x_2 , as V and D , respectively. Other environmental variables are denoted by x_j , $j = 3, \dots, q$. AMK employs the following model structure,

$$\begin{aligned}\hat{f}(y|\mathbf{x}) &= \sum_{i=1}^n \frac{1}{(q-2)} [w_i(x_1, x_2, x_3) + \dots + w_i(x_1, x_2, x_q)] K_{\lambda_y}(y - y_i), \\ \hat{y}(\mathbf{x}) &= \frac{1}{(q-2)} [\hat{y}(x_1, x_2, x_3) + \dots + \hat{y}(x_1, x_2, x_q)].\end{aligned}\tag{5.16}$$

In the above expression, AMK keeps the multivariate kernels but limits them to be product kernels of three inputs. Based on the observations from Fig. 5.2, it is believed that it is important to include V and D , incorporating wind speed and direction information, in every multivariate kernel so that the trivariate kernels can capture the interaction effect between the third environmental factor with wind speed and wind direction. AMK can be used for high-dimensional data without causing computational or data sparsity problems. When additional explanatory variables become available, AMK would include extra additive terms, each of which has the same structure as the current terms, namely a trivariate kernel having inputs of V , D , and a third explanatory variable.

5.2.4 Bandwidth Selection

The key parameters in AMK are the bandwidth parameters, λ_y and λ_x . Lee et al. [132] employ a data-driven selection criterion, known as the integrated

squared error (ISE) criterion [61, 83], as follows,

$$\begin{aligned}
\text{ISE}(\lambda_x, \lambda_y) &= \int \int \left(f(y|\boldsymbol{x}) - \hat{f}(y|\boldsymbol{x}) \right)^2 f(\boldsymbol{x}) dy d\boldsymbol{x} \\
&= \int \int \hat{f}(y|\boldsymbol{x})^2 f(\boldsymbol{x}) dy d\boldsymbol{x} - 2 \int \int \hat{f}(y|\boldsymbol{x}) f(y|\boldsymbol{x}) f(\boldsymbol{x}) dy d\boldsymbol{x} \\
&\quad + \int \int f(y|\boldsymbol{x})^2 f(\boldsymbol{x}) dy d\boldsymbol{x} \\
&= I_1 - 2I_2 + I_3.
\end{aligned} \tag{5.17}$$

With this criterion, one would choose the bandwidths that minimize the ISE. Because I_3 in the ISE expression does not depend on the bandwidth selection, it can be omitted during the minimization of ISE. For I_1 and I_2 , Fan and Yim [61] suggest using a leave-one-out cross-validation estimator as

$$\begin{aligned}
\hat{I}_1 &= \frac{1}{n} \sum_{i=1}^n \int \left(\hat{f}_{-i}(y|\boldsymbol{x}_i) \right)^2 dy, \quad \text{and} \\
\hat{I}_2 &= \frac{1}{n} \sum_{i=1}^n \hat{f}_{-i}(y_i|\boldsymbol{x}_i),
\end{aligned} \tag{5.18}$$

where $\hat{f}_{-i}(y|\boldsymbol{x}_i)$ is the estimator $\hat{f}(y|\boldsymbol{x}_i)$ with the i -th data pair $\{\boldsymbol{x}_i, y_i\}$ omitted. The data-driven bandwidth selection is simply to choose the bandwidths λ_x and λ_y that minimize $\hat{I}_1 - 2\hat{I}_2$.

Using this cross-validation algorithm could, however, take a long time. In order to have a faster bandwidth selection for practical purposes, Lee et al. choose to employ a simpler, greedy procedure to select the bandwidth parameters one at a time, as described in Algorithm 5.1.

In the algorithm, to handle the von Mises kernel, Lee et al. [132] follow an approach suggested in [212] that ties the concentration parameter ν to the bandwidth parameter λ_2 as $\nu = 1/\lambda_2^2$. Then, λ_2 can be selected together with other bandwidth parameters for the Gaussian kernels.

In R, the package `kernplus` implements the kernel regression, i.e., the mean function estimation $\hat{y}(\boldsymbol{x})$ in Eq. 5.16. Suppose that wind data is stored in the matrix `windpw`. The syntax to fit a multi-dimensional power curve is

```
pc.est <- kp.pwcurv(windpw$y, windpw[, c('V', 'D', 'rho', 'I',
                                         'Sb')], id.spd = 1, id.dir = 2),
```

where the two arguments, `id.spd=1` and `id.dir=2`, indicate the first two columns in the data matrix are, respectively, the wind speed and wind direction data. Five covariates are included in this example, i.e., $\boldsymbol{x} = (V, D, \rho, I, S_b)$.

5.3 OTHER DATA SCIENCE METHODS

Addressing the multi-dimensional power curve problem is essentially a regression problem. For this matter, other data science methods, especially those

Algorithm 5.1 Greedy kernel bandwidth selection.

1. Consider only a simple univariate kernel regression corresponding to individual environmental variables in \mathbf{x} .
2. Calculate the bandwidth for each univariate kernel following the direct plug-in (DPI) approach suggested in [187]. This DPI estimator can be obtained by using the `dpiill` function in the `KernSmooth` package and performing it on one input variable at a time, such as

$$\hat{\lambda}_j \leftarrow \text{dpiill}(x_j, y).$$

3. Denote the resulting bandwidths as $(\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_q)$;
 4. Use a basic power curve model that includes only the wind speed, V and wind direction, D as inputs, and fix the bandwidths for the two univariate kernels corresponding to V and D as $\hat{\lambda}_1$ and $\hat{\lambda}_2$, respectively. Then, estimate the bandwidth $\hat{\lambda}_y$ that minimizes $\hat{I}_1 - 2\hat{I}_2$.
-

of semi-parametric or nonparametric nature, can be employed as well. Two methods introduced previously, the support vector machine in Section 2.5.2 and artificial neural network in Section 2.5.3, can certainly be applicable. As argued in Section 5.2.1, parametric regression methods are less effective and not robust in power curve modeling.

In this section, we would like to introduce three more data science methods: k -nearest neighborhood (kNN), tree-based methods, and spline-based methods; please also refer to [86] for the basics about these methods. Most of the methods produce only the mean estimation, $\hat{y}(\mathbf{x})$, but Bayesian additive regression trees (BART) [36], being a Bayesian method, naturally produces the posterior distribution, leading to the density estimation, $\hat{f}(y|\mathbf{x})$.

5.3.1 k -Nearest Neighborhood Regression

The idea of kNN is fairly simple. For a prediction at any target site, the method uses the average of the closest k data points. Suppose that we have n data points in the training set and want to make a prediction at \mathbf{x}_0 . Then, the kNN regression at \mathbf{x}_0 is

$$\hat{y}(\mathbf{x}_0) = \frac{1}{k} \sum_{\mathbf{x}_i \in \mathfrak{N}_k(\mathbf{x}_0)} y_i, \quad (5.19)$$

where the subscript k in the neighborhood notation, \mathfrak{N}_k , signifies that this neighborhood contains exactly k data points. The parameter in kNN is the neighborhood size, k , which needs to be selected *a priori*, usually through cross validations. In the power curve modeling, kNN is used for regression.

It can also be used for classification in other applications. For classification, a data instance will be assigned to the class to which the majority of data instances belong in the neighborhood.

One may notice that the kNN above looks awfully similar to the kernel regression in Eq. 5.4, especially once letting $w_i = 1/k$ for all i 's. But there are a couple of differences. The kNN regression uses a simple average of all data points in its neighborhood, whereas the kernel regression uses a weighted average. In the terms of averaging, kNN is the same as the binning method that uses the uniform kernel.

A more important difference between kNN and kernel regression is in the definition of the neighborhood. The neighborhood in the kernel regression is decided through the use of a specific kernel function and its bandwidth parameter λ . Kernel regression does not directly control the number of data points in its neighborhood but once λ is chosen, the size of the neighborhood is more or less decided. By contrast, the kNN regression decides its neighborhood through setting a specific amount on the data points closest to the target site. The closeness metric used in kNN is usually based on the Euclidean distance, although it could be other distance alternatives.

To appreciate the difference of neighborhood definition between kNN and kernel regression, consider the following analogy. Pretend that two kids in a kindergarten want to decide who can be their friends. Alexandra declares whoever is similar enough to her (based on her definition of similarity) is her friends, and that it does not matter how many friends she ends up with. Nicholas says that he wants precisely k friends and those k kids who live closest in distance to him are his friends, regardless of how different they may be from each other. Alexandra uses the same approach as in the kernel regression, whereas Nicholas uses the kNN approach.

To use kNN for regression, analysts can call the `knn.reg` function in the `FNN` package. The syntax is

```
knn.reg(training inputs, test inputs, y, k).
```

If k is not specified, the default value is three.

5.3.2 Tree-based Regression

A tree-based model is commonly known as the Classification and Regression Trees (CART) [86]. Consider the objective of regression. Recall the learning formulation of SVM in Eq. 2.47. In fact, that formulation is extendable to a general class of learning problems, known as machine learning through regularization, which entails three key components: a loss function, $L(\cdot, \cdot)$, a penalty function, $\text{Penalty}(\cdot)$, and the structure of the hypothesis space, \mathcal{H} , in which the optimization is conducted. Given a set of training data, $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$, the learning problem can be loosely formu-

lated as,

$$\hat{g} = \underset{g \in \mathcal{H}}{\operatorname{argmin}} \left[\sum_{i=1}^n L(y_i, g(\mathbf{x}_i)) + \gamma \cdot \operatorname{Penalty}(g) \right], \quad (5.20)$$

where γ is the penalty coefficient, trading off between the loss function and the penalty function. Note that Eq. 2.47 uses $\gamma/2$ as the penalty coefficient but the inclusion of the constant, “2,” is for mathematical convenience and does not fundamentally change the learning outcome.

The learned function, $\hat{g}(\cdot)$, is used to make a prediction at a target site, say, \mathbf{x}_* , such that $\hat{y}(\mathbf{x}_*) = \hat{g}(\mathbf{x}_*)$. In the SVM regression, the loss function is the ϵ -sensitive error loss function, the penalty is a squared function, and \mathcal{H} is the reproducing kernel Hilbert space, so that $g(\mathbf{x}) = \sum_{i=1}^n \alpha_i K(\mathbf{x}, \mathbf{x}_i)$. In CART, the loss function is a squared error loss function, the penalty function is the tree size, and the most important difference is that the hypothesis space, \mathcal{H} , is piecewise constant.

Basically, a CART partitions the input data space into J regions, each of them denoted by R_j , $j = 1, \dots, J$. For each region, CART uses a single constant, c_j , to represent it. As such, a CART model is represented by the following parameters: $\{J; R_1, \dots, R_J; c_1, \dots, c_J\}$. Then, the prediction using CART can be expressed as

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^J c_j \cdot \mathbb{1}(\mathbf{x} \in R_j).$$

Since CART uses a squared error loss function, the representation for each region is the sample average of all data points falling into that region, meaning that the estimate of c_j is \bar{y}_j for R_j .

Practically, a CART is built through a greedy algorithm, which is to perform a binary splitting on a variable, one at a time. The action every time splits an input domain into subregions. Once a splitting is carried out, it will not be revisited, even if it may not be the optimal splitting in hindsight. This greedy algorithm runs efficiently, in the complexity of $O(pn \log n)$, where p is the dimension of the input space and n is the data amount.

If one carries out this binary partition process, the process can be visualized through the growing of a binary tree—this is how the method gets its name. One such example in a two-dimensional space is presented in Fig. 5.4. In the tree, the whole region corresponds to the root node and the final subregions correspond to the terminal nodes. When one region is split to two subregions, the two subregions correspond to the two children nodes of the same parent node. Two nodes sharing the same parent node are called sibling nodes. The tree size is decided by the number of terminal nodes. Once a tree is established, the data points scattered in the original space, i.e., in the root node, are now dispersed into the respective terminal nodes.

The tree growing process starts with all the data. At each step, it considers a splitting variable j and split point s and defines the pair of the half-planes

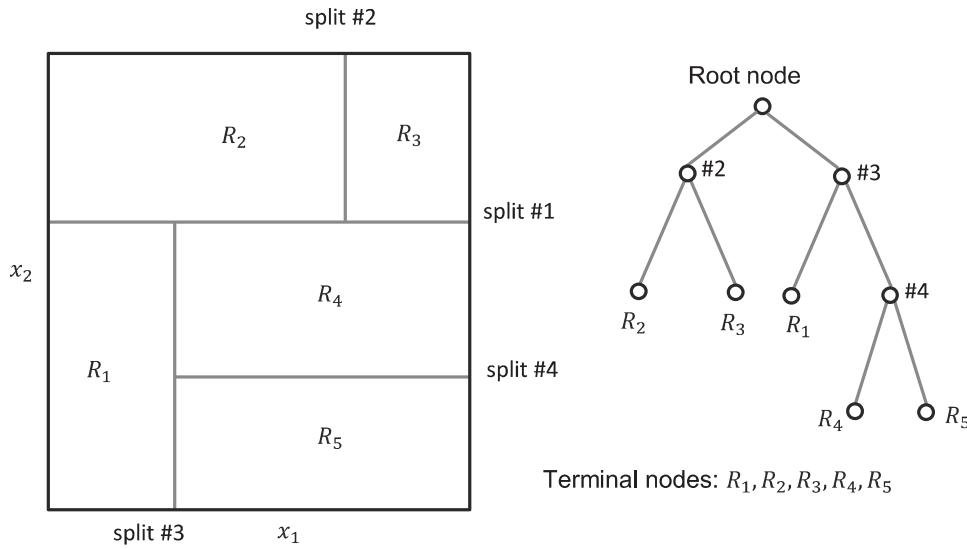


FIGURE 5.4 The two-dimensional example of binary splitting in the creation of a tree model.

as,

$$R_1(j, s) = \{\boldsymbol{x} | x_j \leq s\} \text{ and } R_2(j, s) = \{\boldsymbol{x} | x_j > s\}.$$

Note that R_1 and R_2 are the generic notations for any half planes at a splitting action and do not correspond to those same symbols used in Fig. 5.4. The tree model is built by searching all p elements in \boldsymbol{x} , one at a time, and decide the best splitting variable and the corresponding split point at each step, by minimizing the following objective function,

$$\min_{j,s} \left[\sum_{\boldsymbol{x}_i \in R_1(j,s)} (y_i - \hat{c}_1)^2 + \sum_{\boldsymbol{x}_i \in R_2(j,s)} (y_i - \hat{c}_2)^2 \right] \quad (5.21)$$

where \hat{c}_j is the sample average of all data points in R_j .

The above is done by treating that the tree size, J , is fixed. The tree size represents the model complexity of CART. If the resulting tree is too small (too simple a model), the piecewise constant approximation is crude, causing a high bias in prediction, whereas if the resulting tree is too large (too complicated a model), then it fits the training data too hard, causing overfitting and again leading to poor prediction. Selecting an appropriate tree size is therefore important. A practical procedure of deciding the tree size is the bottom-up pruning. The basic idea is to build the largest possible tree first and then prune the large tree to an appropriate size. Pruning is the action of merging terminal nodes to reduce the size of the tree, as outlined in Algorithm 5.2.

Algorithm 5.2 Tree pruning procedure.

1. Start with the largest possible tree (split until some pre-specified minimum node size is reached);
 2. Identify terminal siblings, which are sibling terminal nodes having the same parent node;
 3. *Provisionally* merge each terminal sibling pair making the respective parent a terminal node (i.e., remove split);
 4. Find the pair that, when merged, increases the fitting error of tree on training data the least;
 5. Remove the corresponding split, making the parent node terminal, and creating a tree with one fewer terminal node;
 6. Go to Step 2 until no splits left (i.e., only one root node).
-

The final tree size is chosen as J^* that minimizes the estimate of the test error using either an independent validation dataset or the cross-validation method.

Suppose that a tree method is applied to wind power data with a single input (wind speed). It could produce an outcome that looks like from the binning method. Using a tree, the final multi-bin result comes out of the iterations of binary splitting. One thing different, though, is that unlike the IEC binning using bins of equal width, the tree-based method less likely produces bins of equal size, because the actual split points depend on the solution to the optimization problem in Eq. 5.21.

According to Hastie et al. [86], Table 10.1, despite many appealing characteristics, CART has a relatively poor predictive power. This understanding motivates analysts to enhance the predictive power of a tree-based method through ensembling a set of trees. The general thought process is that a weak base learner like CART can be made much capable, or appreciably stronger, when many weak learners are made to work together. Specific ensembling mechanisms used include bagging, leading to bagged trees or the random forest (RF), or boosting, leading to the multiple additive regression trees (MART). Bayesian additive regression trees [36], or BART, a Bayesian version of sum of trees, is also an ensemble of trees, each of which explains a small and different portion of the predictive function. Conceptually, BART is closer to boosting than to bagging.

The technical details in BART are rather involved. For practitioners, it is advised to use the `bart` function in the `BayesTree` package. The syntax of using BART is

```
output<-bart(x.train, y.train, x.test),
```

where `output` is an R object and `output$yhat.test` contains the samples from the estimated conditional density function, $\hat{f}(y|\mathbf{x})$. Each column is a vector of samples drawn from $\hat{f}(y|\mathbf{x})$ for a specific \mathbf{x} corresponding to a row in the input argument `x.test`, and the average of all the samples in that column is the corresponding conditional expectation, namely the point estimation, $\hat{y}(\mathbf{x})$.

5.3.3 Spline-based Regression

The spline-based regression is to use piecewise polynomials to model a nonlinear response. One of the popular spline functions used is the cubic spline [86]; see Fig. 5.5, middle panel, for an illustration. A cubic spline partitions the input domain into a few segments, which is in fact an action of binning, and models each segment using a cubic polynomial. In order to produce a smooth, coherent model for the whole domain, a cubic spline imposes continuity and smoothness constraints at the partition points, known as *knots*. In Fig. 5.5, two knots are used and denoted as ξ_1 and ξ_2 , respectively. Although ξ_1 and ξ_2 partition the input domain in Fig. 5.5 into three roughly equal parts, knots in general do not have to be evenly spaced. Each cubic polynomial is specified by four parameters, producing a total of 12 parameters for the three piecewise cubic polynomials. The constraints imposed at the partition points, however, reduce the number of actual parameters that need to be estimated. For the cubic spline in Fig. 5.5, there are three constraints at each knot, which require, respectively, the equality of the function value, that of its first-order derivative and that of the second-order derivative, at each of the partition points. With the six constraints considered, the number of actual parameters to be estimated for the cubic spline is six.

One may have noticed that the spline method in fact injects the idea of binning into its action of modeling. If only using the idea of binning without the boundary constraints, however, the response looks like the plot in the right-most panel of Fig. 5.5. The three unconstrained piecewise cubic polynomials need a total of 12 parameters to specify. When a single global cubic polynomial is used to model a response, it uses four parameters, but its modeling adaptivity to local features is far worse than the other two alternatives. With only a slight increase in model complexity (measured by the number of parameters), the cubic spline is endowed with the level of modeling adaptivity as a binning method allows.

Analysts may argue that the binning method can use a single constant for each bin, so that the number of parameters for the right-most example in Fig. 5.5 can be three, instead of 12. The problem of this argument is that when using a constant to model a bin, the bin width needs to be much smaller, or equivalently, the number of bins needs to be much greater, so that a piecewise constant function can approximate a nonlinear response with sufficient accuracy. It is not unusual that with one single input variable such as wind speed, analysts need to use 20 bins to model the whole response. With 20 bins, the

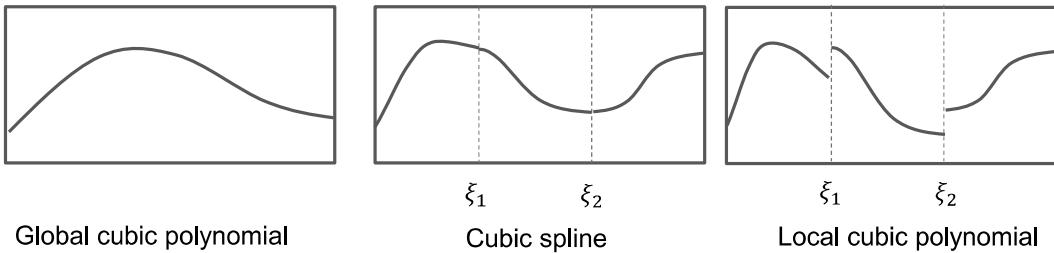


FIGURE 5.5 Global cubic polynomial, cubic spline, and local cubic polynomials. (Reprinted with permission from Ding et al. [50].)

number of parameters cannot be fewer than 20, already producing a model that is unnecessarily complicated.

To use spline-based regression on actual applications, there are a few technical problems to be resolved. One of the problems is that the behavior of polynomial fit to data tends to be erratic near the boundaries. To fix the problem, analysts introduce the *natural cubic spline*, which postulates that the response function be linear beyond the boundary knots. This constraint translates to the continuous and continuous first derivative requirements at each boundary knot but it no longer requires continuous second derivatives, because the linear function outside the boundary knot does not have a second derivative.

Another technical problem is how many knots one should choose and where to position them. For addressing the knot selection problem, analysts introduce the *smoothing spline*. The smoothing spline finds a function that minimizes the following objective function,

$$\hat{g} = \operatorname{argmin} \left\{ \sum_{i=1}^n [y_i - g(x_i)]^2 + \gamma \int_t \{g''(t)\}^2 dt \right\}. \quad (5.22)$$

This optimization formulation is consistent with the general regularization learning problem expressed in Eq. 5.20. Here the loss function is a squared error loss, the penalty function is the integration of the second derivative on $g(\cdot)$, and the hypothesis space \mathcal{H} is that $g(\cdot)$ should have two continuous derivatives. When $\gamma = 0$, then g can be any function that interpolates the training data points, making the loss function zero, while when $\gamma = \infty$, it forces $g''(t) = 0 \forall t$, resulting a simple least squares line fit.

The solution of the above optimization formulation is that the smoothing spline is a natural cubic spline with knots at the unique values of every x_i in the training dataset. At the face value, a natural cubic spline with knots at each and every x_i implies that the spline may have as many as n knots and may cause over-parametrization and overfitting. In actuality, when $\gamma > 0$, the number of *effective* knot positions, or the effective number of parameters in the resulting model, can be much smaller than n . Which one of the x_i 's to be

selected as a knot, or which other to stay away, depends on the outcome of the optimization in Eq. 5.22.

In Exercise 3.1, we mention that for a linear smoother with a smoother matrix \mathbf{S} , the effective number of parameters is $\text{tr}(\mathbf{S})$. The spline regression can in fact be expressed as a linear smoother, so that the effective number of knots or the effective number of parameters can be decided by the trace of the smoother matrix (see Exercise 5.4). The resulting effective number of parameters apparently depends on the choice of γ , the cost coefficient trading off between the loss function and the penalty function.

One may have noticed that we use a scalar x_i in Eq. 5.22, implying a univariate regression and smoothing. This is because the spline-based regression, in its most general form, is not particularly scalable. When it comes to handling multivariate covariates like in the circumstance of building a multi-dimensional power curve, the plain version of smoothing splines is difficult to use. There are two popular multivariate extensions of the spline methods. One is the smoothing spline ANOVA (SSANOVA) [79] and the other one is the multivariate adaptive regression splines (MARS) [68]. MARS is used in Chapter 10 for extreme load analysis. The smoothing spline ANOVA is used here to be one of the alternatives for multi-dimensional power curve modeling.

SSANOVA employs a functional ANOVA (analysis of variance) decomposition that limits the interactions to the two-way interactions and ignores the higher-order terms, so as to help rein in the curse of dimensionality. A higher-order interaction can be included but through a recursive two-way interaction mechanism. For instance, a four-way interaction is modeled as adding one extra factor to an already existing three-way interaction term. This strategy is in fact similar to what is used in MARS, as MARS accomplishes the scalability also through a hierarchical inclusion of interaction terms.

Analysts can use the `ssanova` function from the `gss` package in R for implementing the smoothing spline ANOVA method.

5.4 CASE STUDY

Two datasets are used in this case study, the `Inland Wind Farm Dataset1` and `Offshore Wind Farm Dataset1`. Table 5.1 summarizes the specifications of the datasets; for certain entries an approximation rather than the accurate value is given for the protection of the identities of the turbine manufacturers and wind farms. Fig. 5.6 presents the turbines/masts layout and turbine-to-mast distances.

5.4.1 Model Parameter Estimation

The quality of point estimation is to be evaluated using RMSE, while that of the density estimation is to be evaluated using CRPS. An out-of-sample test is to be conducted based on a five-fold cross validation. In each iteration of the five-fold cross validation, a dataset is divided into a partition of 80%

TABLE 5.1 Specifications of turbines in the two wind farms.

Wind farm	Inland	Offshore
Number of met masts	Multiple	Single
Number of wind turbines	200+	30+
Hub height (m)	80	70
Rotor diameter (m)	about 80	about 90
Cut-in wind speed (m/s)	3.5	3.5
Cut-out wind speed (m/s)	20	25
Rated wind speed (m/s)	around 13	around 15
Rated power (MW)	1.5–2.0	around 3
Location	Inland, U.S.	Offshore, Europe

Source: Lee et al. [132]. With permission.

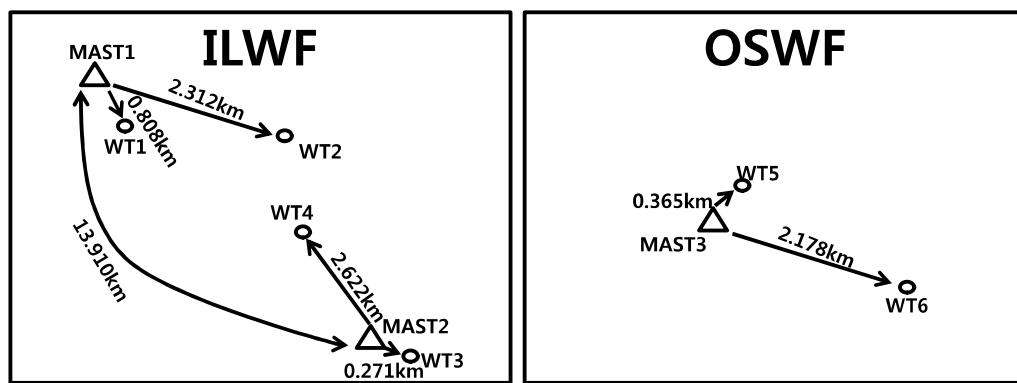


FIGURE 5.6 Layout of the turbines and masts and turbine-to-mast distances. ILWF: inland wind farm; OSWF: offshore wind farm. (Reprinted with permission from Lee et al. [132].)

for training and 20% for testing. Then, the average of the five error values is reported to represent a method's performance. Note that in Part II of the book, the performance metrics are computed in terms of wind power, requiring a change in notation to be made to the formulas of the performance metrics presented in Section 2.6. Specifically, we use

$$\text{RMSE} = \sqrt{\frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} (\hat{y}(\mathbf{x}_i) - y_i)^2}, \quad \text{and} \quad (5.23)$$

$$\text{CRPS} = \frac{1}{n_{\text{test}}} \sum_{i=1}^{n_{\text{test}}} \int \left(\hat{F}(y|\mathbf{x}_i) - \mathbb{1}(y > y_i) \right)^2 dy.$$

Algorithm 5.1 works well for estimating the bandwidth parameters. For point estimation, Lee et al. [132] are able to use all the training data for bandwidth selection and the computational time is of no concern at all. But for density estimation, even with the greedy algorithm, the last step (Step 4 in Algorithm 5.1) that finds the bandwidth for y would still take a long running time, had all the training data been used. In the end, Lee et al. decide to randomly select 25% of the training data for estimating the bandwidth used in density estimation.

For the out-of-sample testing, Lee et al. are able to use all the testing data points for computing the out-of-sample RMSEs. But for computing the CRPS values, using all the testing data again requires a long time. Lee et al. find that using 1,000 randomly sampled data points to calculate the CRPS values remains reasonably stable over different random sampling. As such, the CRPS values reported here are based on 1,000 test data points.

Note that the RMSE values of the same method in this section may differ slightly in different studies because of the random split of the training and test datasets. The RMSE values here see a more noticeable difference from those presented in [132] because of two reasons: (a) the RMSE values associated with AMK are computed using the `kernplus` package. Although based on the algorithm originally developed in [132], the implementation in the `kernplus` package made small changes to deal with certain application complexities. (b) What is reported here is the average of a five-fold cross validation, whereas what was reported in [132] was based on a one-time 80-20 random split. Despite the numerical differences, the main message from the numerical studies stays the same as in [132].

5.4.2 Important Environmental Factors Affecting Power Output

Based on the physical understanding hinted by the power generation law in Eq. 5.2, it is apparent that wind speed, direction, and air density are important factors to be included in a power curve model. The question is what else may also need to be included.

To address that question, the first set of results is to show the RMSEs when

AMK includes a single additive term from $\mathbf{x} = (x_1, x_2, x_3)$ to $\mathbf{x} = (x_1, x_2, x_q)$, where $q = 5$ in the Inland Wind Farm Dataset1 and $q = 7$ in the Offshore Wind Farm Dataset1. Recall that each additive term is a trivariate kernel with the first two of the variables always being the wind speed, V , and wind direction, D .

The baseline model for comparison is chosen to be the kernel model that has only the wind speed and wind direction (V, D) in a product kernel, which is in fact the same as the one used in [109]. This bivariate kernel model is referred to as BVK.

The results are shown in Table 5.2, in which the notation of (\cdot, \cdot, ρ) means that the additive term included in the model has the wind speed, V , and wind direction, D , and air density, ρ , as its inputs, where the wind speed and wind direction are shorthanded as two dots. Other notations follow the same convention. These results lead to the following observations:

1. In both the inland wind farm and offshore wind farm, air density, ρ , is indeed, after the wind speed and wind direction, the most significant factor in wind power generation. Including ρ in the model delivers reductions in RMSE from 9% to 17% across the board. This outcome is consistent with the physical understanding expressed earlier.
2. For the offshore wind turbines, humidity, H , appears to be another important factor in explaining variations in power outputs. Because humidity measurements are not available in the Inland Wind Farm Dataset1, it is unknown whether humidity is also a significant factor there.
3. The remaining three factors, namely turbulence intensity and the two wind shears, which each represents some other aspects of wind dynamics, show also positive impact, except for the case of WT5. The impact of turbulence intensity is rather pronounced for the inland turbines, nearly as significant as humidity on the offshore turbines. The impact of the below-hub wind shear is noticeably positive, although not as much as turbulence intensity. Both factors have shown more obvious effects for the inland turbines than for the offshore ones, but the significance of their impact is definitely after that of ρ .

The next step undertaken is to determine which other factors may impact the power output when AMK includes more than one additive term, conditional on the factors that have already been included. Based on the observations expressed above, for both inland and offshore turbines, the first additive term included is always (V, D, ρ) . For the inland turbines, in addition to this first term, there are two more terms that have either turbulence intensity, I , or the below-hub wind shear, S_b . For the offshore turbines, a second additive term, (V, D, H) , is also always included. In addition to the first two terms, there are three more terms that have either the two wind shears, S_a , S_b , or turbulence intensity, I . The two wind shears are always included or excluded together in the numerical analysis to keep the total number of model

TABLE 5.2 Impact on RMSE when including different environmental factors. The percentages in the parentheses are the reduction in terms of RMSE when the corresponding model's point estimation is compared with that of BVK.

WT	BVK	(\cdot, \cdot, ρ)	(\cdot, \cdot, I)	(\cdot, \cdot, S_b)	(\cdot, \cdot, S_a)	(\cdot, \cdot, H)
WT1	0.0884	0.0748 (15.3%)	0.0856 (3.1%)	0.0869 (1.7%)	.	.
WT2	0.0921	0.0814 (11.6%)	0.0887 (3.8%)	0.0894 (3.0%)	.	.
WT3	0.0817	0.0681 (16.7%)	0.0755 (7.6%)	0.0747 (8.6%)	.	.
WT4	0.1163	0.1030 (11.4%)	0.1093 (6.0%)	0.1109 (4.6%)	.	.
WT5	0.0907	0.0824 (9.1%)	0.0928 (-2.2%)	0.0917 (-1.1%)	0.0922 (-1.6%)	0.0858 (5.4%)
WT6	0.0944	0.0815 (13.6%)	0.0939 (0.5%)	0.0918 (2.7%)	0.0927 (1.7%)	0.0873 (7.5%)

TABLE 5.3 Model comparison using data in the **Inland Wind Farm Dataset1**. RMSE values are reported in the table. Boldface values are the smallest RSME in the row.

WT	(\cdot, \cdot, ρ)	(\cdot, \cdot, ρ, I)	$(\cdot, \cdot, \rho, S_b)$	$(\cdot, \cdot, \rho, I, S_b)$
WT1	0.0747	0.0743	0.0742	0.0751
WT2	0.0816	0.0800	0.0802	0.0802
WT3	0.0680	0.0651	0.0645	0.0646
WT4	0.1028	0.1001	0.1010	0.1004

comparisons manageable. Tables 5.3 and 5.4 present the model comparison results.

For some of the inland turbines, the best AMK explaining their power output includes the input factors of the wind speed and wind direction (V and D), air density (ρ), and turbulence intensity (I), while some others include the wind speed and wind direction (V and D), air density (ρ), and wind shear (S_b). These versions differ marginally. For the offshore turbines, it is rather clear that the model with the wind speed (V), wind direction (D), air density (ρ), and humidity (H) produces the lowest RMSE. Including other environmental factors in the model could instead increase the RMSE. The increase in RMSE is consistent and can be as much as 5.1% for WT6. If the above analysis is repeated using CRPS, the insights remain the same, but the presentation of the CRPS results is omitted.

TABLE 5.4 Model comparison using data in the Offshore Wind Farm
Dataset1. RMSE values are reported in the table. Boldface values are the smallest RSME in the row.

WT	(\cdot, \cdot, ρ, H)	$(\cdot, \cdot, \rho, H, I)$	$(\cdot, \cdot, \rho, H, S_a, S_b)$	$(\cdot, \cdot, \rho, H, I, S_a, S_b)$
WT5	0.0790	0.0801	0.0810	0.0818
WT6	0.0800	0.0816	0.0822	0.0832

5.4.3 Estimation Accuracy of Different Models

In this subsection, we compare various power curve methods. In the comparisons, AMK is selected based on the best subset of variables revealed in the previous section. Other methods use the same subset of variables to level the playground.

Let us first take a look at the prediction errors of IEC binning method and AMK. The binning method used here is the air density corrected version. Only the RMSE values are presented, because the IEC binning does not produce a density estimation.

Table 5.5 presents the comparison. The reduction in terms of RMSE made by AMK over IEC is astonishing, but it may not be that surprising. Recall that we mention earlier in this chapter that the IEC method's intention is to provide a benchmark for verification purpose, rather than providing a method for real-life performance assessment or wind power prediction. Consider the following analogy in the context of vehicle fuel economy. At the time of sale, a new car is displayed with a published fuel economy, in the unit of miles per gallon. The published fuel economy value is obtained under a standardized, ideal testing condition, which cannot be replicated in real-life driving. A car's real-life fuel economy based on someone's actual driving is almost always worse than the published one. In the wind power production, engineers observe something similar—using the IEC binning power curve often leads to a conclusion of under performance, which is to say that the actual power production falls short of the prediction.

Still, car buyers and car manufacturers feel that the fuel economy obtained under the ideal condition provides a reasonable benchmark, offering some ballpark ideas of how fast a car consumes its fuel. However, for consumers who care very much about the real-life fuel economy, such as in commercial driving, they are not advised to use the published fuel economy value, as using the published value will surely lead to biased estimations. The same conclusion should have been extended to the IEC method, but in actuality, in the vacuum of robust, reliable, and capable power curve models, the IEC binning method is routinely used in the tasks or for the missions it is not designed for.

We would also like to articulate one important limitation of the IEC binning method. From the wind power production law in Eq. 5.2, we understand that the inclusion of air density is important. That is the reason why the IEC binning uses the air density correction. The same piece of information (air

TABLE 5.5 Compare the RMSE between the IEC binning method and AMK.

Wind Farm	Turbine	IEC	AMK	Error reduction rate over IEC
Inland	WT1	0.1305	0.0742	43%
	WT2	0.1158	0.0800	31%
	WT3	0.1217	0.0645	47%
	WT4	0.1567	0.1001	36%
Offshore	WT5	0.0970	0.0790	19%
	WT6	0.1089	0.0800	27%

density) can be included in AMK as well. We wonder—by making use of the same covariate, which method will benefit more? The benefit of including air density can be discerned by comparing the same method with and without using air density while making wind power prediction. For the IEC binning, this is between the plain version of binning and the air density adjusted binning. For AMK, this is between the AMK with only wind speed and wind direction (which is in fact BVK) and the AMK with wind speed, wind direction, and air density.

Table 5.6 presents the comparison using the four inland turbines, but the same conclusion is extendable to the offshore turbine data as well. The percentage values reported in parentheses are the reductions in terms of RMSE between the two versions of the same method, rather than the reduction between the two different methods. Take WT1 as an example. The -0.1% in the fourth column means a very slight increase in RMSE when using the air density adjusted binning method, as compared to the plain version of the binning method, whereas the 15.3% in the sixth column is the reduction in RMSE when using the AMK with inputs (V, D, ρ) , as compared to AMK with inputs (V, D) .

The comparison makes it clear that while air density is an important factor to be included in a power curve model, the air density adjustment used in the IEC binning is not optimal. It does help reduce $1 - 2\%$ in RMSE in fitting the wind power. But on the other hand, the potential benefit of having air density in a power curve model is much greater. When used in AMK, it can help reduce RMSE as much as 17% . This example demonstrates the power of data science methods over a pure engineering heuristics.

Table 5.7 further compares the point estimation, in terms of RMSE, among the four data science-based multi-dimensional power curve methods: kNN, SSANOVA, BART, and AMK. In this comparative study, kNN uses the normalized covariates, i.e., each covariate is normalized by its standard deviation, while the other methods use the raw measurements. The action of normalization has a profound impact on kNN but not so much on other methods. When

TABLE 5.6 Impact of air density on IEC binning and on AMK. Reported below are the RMSE values.

Wind Farm	Turbine	IEC Binning		AMK	
		Plain	Adjusted	BVK	BVK + air density
Inland	WT1	0.1303	0.1305 (−0.1%)	0.0884	0.0748 (15.3%)
	WT2	0.1180	0.1158 (1.9%)	0.0921	0.0814 (11.6%)
	WT3	0.1237	0.1217 (1.6%)	0.0817	0.0681 (16.7%)
	WT4	0.1592	0.1567 (1.6%)	0.1163	0.1030 (11.4%)

using SSANOVA, a full model considering all possible interactions takes too long to run. Instead, the main effects and selected interactions are included. For WT5 and WT6, V , D , ρ , H , $V \times D$, $V \times \rho$, $V \times H$, $V \times D \times \rho$, and $V \times D \times H$ are included in the SSANOVA model. For WT1 and WT3, H in the aforementioned terms is replaced by S_b , whereas for WT2 and WT4, H in the aforementioned terms is replaced by I .

The comparison shows that AMK overall performs the best, with kNN as a close second. BART performs slightly better than kNN on WT1–WT3 cases and slightly worse than AMK, but it does slightly worse than kNN for WT4 and noticeably worse for WT5–WT6. SSANOVA is the worst performer among the four. In fact, SSANOVA performs closer to what BVK does, as shown in Table 5.6 (the fifth column). One closer look reveals that the data associated with WT4 have the largest variation among the four inland turbine datasets, as evident by its large RMSE values. This large variation could be due to the fact that WT4 is located the farthest away from its companion mast so that the wind measurements taken at the mast are less representative of the wind condition at the turbine site. For WT5 and WT6, the RMSE in terms of the normalized power is slightly higher than that of WT1 to WT3, but because WT5 and WT6 have a higher rated power, almost double that of the inland turbines, the absolute value of the noises are greater. This observation appears to suggest that BART is sensitive to the noise level in a dataset and its performance suffers when the data noise level is elevated.

Next, let us take a look at the density estimation based on CRPS. Note that the IEC binning method and kNN can produce only point estimation, while BVK, BART and AMK produce both point and density estimations. SSANOVA is supposed to produce density estimation as well, but doing so takes way too long. Therefore, in the CRPS-based comparison, only BVK, AMK and BART are included, and the baseline model is BVK.

The CRPS-based comparison is presented in Table 5.8, in which the per-

TABLE 5.7 Comparing various data science-based power curve methods. Reported below are the RMSE values. The boldface font indicates the best performance.

	kNN	SSANOVA	BART	AMK	AMK improvement over		
					kNN	SSANOVA	BART
WT1	0.0766	0.0870	0.0762	0.0741	3%	15%	3%
WT2	0.0828	0.0907	0.0817	0.0799	4%	12%	2%
WT3	0.0669	0.0766	0.0667	0.0645	4%	16%	3%
WT4	0.1035	0.1118	0.1039	0.1000	3%	11%	4%
WT5	0.0810	0.0947	0.0876	0.0792	2%	16%	10%
WT6	0.0830	0.1039	0.0906	0.0804	2%	23%	11%

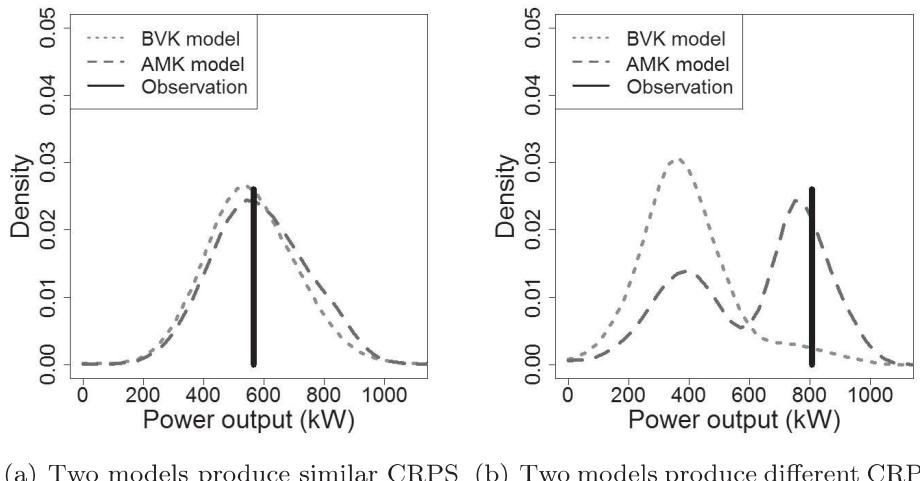
TABLE 5.8 Comparing CRPS among BVK, AMK, and BART. Boldface font indicates the best performance.

Turbine	BVK	AMK		BART	
		three inputs	four inputs	three inputs	four inputs
WT1	0.0432	0.0377 (12.7%)	0.0370 (14.3%)	0.0487 (-12.7%)	0.0475 (-9.9%)
WT2	0.0456	0.0413 (9.4%)	0.0400 (14.0%)	0.0539 (-18.2%)	0.0518 (-13.6%)
WT3	0.0378	0.0337 (10.8%)	0.0311 (17.7%)	0.0419 (-10.8%)	0.0385 (-1.8%)
WT4	0.0571	0.0498 (12.8%)	0.0473 (17.2%)	0.0693 (-21.4%)	0.0631 (-10.5%)
WT5	0.0461	0.0408 (11.5%)	0.0388 (15.8%)	0.0565 (-22.5%)	0.0553 (-19.9%)
WT6	0.0462	0.0378 (18.2%)	0.0375 (18.8%)	0.0561 (-21.4%)	0.0550 (-19.0%)

centage values in the parentheses are the reductions in CRPS a method makes relative to BVK. A negative value suggests an increase, rather than a reduction, in the respective CRPS. There are two versions of AMK and BART that are included: the three-input version uses (V, D, ρ) for both inland and offshore turbines, and the four-input version uses (V, D, ρ, I) for inland turbines but (V, D, ρ, H) for offshore turbines.

BART turns out to be the worst performer for predicting the conditional density among the three models and AMK the best. AMK is 14%–18% better than BVK, which is in turn 2%–20% better than BART. AMK appears to exhibit competitiveness and robustness, thanks in part to its model structure being advised by the physical understanding of wind power production.

To facilitate an intuitive understanding how AMK improves the density estimation, we present in Fig. 5.7 an illustration of density estimations using BVK and AMK. To produce the result in Fig. 5.7, WT5 data are used. A



(a) Two models produce similar CRPS (b) Two models produce different CRPS

FIGURE 5.7 Comparison of the predictive distributions of power output when BVK and AMK produce similar CRPS values versus when they produce different CRPS values.

four-input AMK is employed, with input variables as (V, D, ρ, H) . The left panel of Fig. 5.7 shows the predictive distributions of the power output from the two models, when their CRPS values are not much different. The two distributions are similar and either model produces a good estimate. The right panel of Fig. 5.7 presents the predictive distributions of the two models, when their CRPS values differ considerably. One can see that the distribution from the BVK model is centered incorrectly.

GLOSSARY

AMK: Additive multiplicative kernel method

ANOVA: Analysis of variance

BART: Bayesian additive regression trees

BVK: Bivariate kernel method

CART: Classification and regression tree

CKD: Conditional kernel density

CRPS: Continuous ranked probability score

DPI: Direct plug-in estimator

GAM: Generalized additive model

IEC: International Electrotechnical Commission

ILWF: Inland wind farm

ISE: Integrated squared error criterion

kNN: k nearest neighborhood

MARS: Multivariate adaptive regression splines

MART: Multiple additive regression trees

OSWF: Offshore wind farm

pdf: Probability density function

RF: Random forest

RMSE: Root mean squared error

SSANOVA: Smoothing spline ANOVA

SVM: Support vector machine

WT: Wind turbine

EXERCISES

5.1 Use the 10-min data in the **Wind Time Series Dataset** and treat the wind power as y and the wind speed as x . Conduct the following exercise.

- Random split the data into 80% for training and 20% for testing. Use the training data to build a power curve model, following the IEC binning method. Select the bin width as 0.5 m/s.
- Use the 20% test data to calculate the RMSE.
- Change the bin width to 1 m/s, 1.5 m/s, and 2 m/s, respectively, and for each one of them, build a respective power curve model and calculate its corresponding RMSE. Observe how the bin width affects the quality of the power curve method.

5.2 Suppose that the number of bins used in Exercise 5.1 under different bin widths are denoted as $B_{(0.5)}$, $B_{(1)}$, $B_{(1.5)}$, and $B_{(2)}$, respectively. Still use the 10-min data in the **Wind Time Series Dataset**.

- Build a CART model with the number of terminal nodes set to be $B_{(0.5)}$, $B_{(1)}$, $B_{(1.5)}$, and $B_{(2)}$, respectively.
- Visualize the partition on wind speed by the CART model for each of the terminal node choices, and compare the partition outcome with the respective partition used in the IEC binning.

- c. Conduct an out-of-sample test through, again, the 80-20 random split. Compute the RMSE for each of the choices and compare with the respective binning outcome.
- 5.3 Again use the 10-min data in the *Wind Time Series Dataset*.
- Build a one-dimensional kernel regression using the Gaussian kernel function. Set the bandwidth parameter λ to 0.5 m/s, 1 m/s, 1.5 m/s, and 2 m/s, respectively. Use the 80-20 random split, conduct the out-of-sample test, and report the corresponding RMSE.
 - Use five-fold cross validations to search for the optimal λ . Is the optimal λ different from the above prescribed choices?
 - Compare the RMSE of the kernel regression under the optimal λ , the best binning outcome in Exercise 5.1, and the best CART outcome in Exercise 5.2. How much are they different? If they do not differ a lot, does that surprise you? Why or why not? If they do differ a lot, can you explain why?
- 5.4 Because the smoothing spline is a natural cubic spline with knots at every data point x_j , $j = 1, \dots, n$, we can write the smoothing spline function as
- $$g(x) = \sum_{j=1}^n h_j(x)\beta_j,$$
- where $h_j(x)$'s are the basis functions used in the natural cubic spline and n is the number of data points in the training set. For the natural cubic splines, the first two basis functions are $h_1(x) = 1$ and $h_2(x) = x$. The other basis functions take the form of a third-order polynomial function but the detailed expressions are omitted here. Please derive the smoother matrix \mathbf{S} in terms of $h_j(\cdot)$ and β_j . Show that the degree of freedom of the smoothing splines, or the effective number of knots, equals n when $\gamma = 0$ and equals two when $\gamma \rightarrow \infty$. Do the two extreme values make intuitive sense? What this means is that the degree of freedom of the smoothing splines, or its effective number of knots, is between two and n , as γ goes from 0 to infinity.
- 5.5 Rasmussen and Williams [173, pages 138-141] state that if one chooses a particular type of covariance function (i.e., a kernel function), the smoothing spline and the one-dimensional Gaussian process regression (namely kriging) can be made equivalent. To appreciate this understanding, please generate a set of one-dimensional data and do a simple numerical test.
- Let $y = e^{-1.4x} \cos(7\pi x/2) + \varepsilon$ and $\varepsilon \sim \mathcal{N}(0, 0.5)$. Use this function to simulate 200 data pairs, i.e., $\{(x_1, y_1), (x_2, y_2), \dots, (x_{200}, y_{200})\}$.

- b. First fit an ordinary kriging model to the simulated one-dimensional data, and then, fit a smoothing spline using the R function `smoothing.spline` in the `stat` package. Adjust the penalty coefficient used in the smoothing spline fit and see if you could produce a spline fit close to the kriging fit.
- 5.6 Use the 10-min data in the `Wind Time Series Dataset`, and build a kNN-based power curve model. Test different choices of k , the neighborhood size in kNN. Use the 80-20 random split and conduct an out-of-sample test. Observe what choice of k produces a model whose RMSE is close to, respectively, that of the best binning outcome in Exercise 5.1, that of the best CART outcome in Exercise 5.2, and that of the best kernel regression in Exercise 5.3.
- 5.7 Use the WT5 data in the `Offshore Wind Farm Dataset1`, and build a CART and a BART, respectively, using all seven covariates. Conduct an out-of-sample test based on a 80-20 random split and compare their RMSEs. Does BART outperform CART? Is that what you anticipated?
- 5.8 Use the WT1 data in the `Inland Wind Farm Dataset1`, and build an SVM, an ANN, and an AMK, respectively, using all five covariates. Conduct an out-of-sample test based on a 80-20 random split and compare their RMSEs. How do their performances compare to each other?
- 5.9 To select the best subset of variables to be included in the final model, two versions of a greedy strategy are used and referred to, respectively, as the *forward stepwise selection* and *backward stepwise selection* [86, Section 3.3.2].
- The forward stepwise selection is to screen through all the candidate variables, one at a time, and select the one whose addition to the model reduces the out-of-sample RMSE the greatest. Once chosen, remove the variable from the candidate set and select the next variable from the remaining candidates, until the addition of a new variable no longer reduces the out-of-sample RMSE.
 - The backward stepwise selection starts off with the whole set of candidate variables in the model. Remove one at a time, and select the one that reduces the out-of-sample RMSE the greatest and remove it. Screen the remaining variables in the model following the same fashion and stop when the deletion of an existing variable no longer reduces the out-of-sample RMSE.

Use the AMK as the power curve model and the WT6 data in the `Offshore Wind Farm Dataset2`. Test both the forward stepwise selection strategy and the backward stepwise selection strategy and see what subset of variables they select.

- 5.10 Take the BVK model, which is the same as AMK with two inputs (V, D) , and the WT1 data in the `Inland Wind Farm Dataset1`. Build a BVK model using the original wind speed and then build another BVK model using the air-density-corrected wind speed, while all other things are kept the same. Denote the latter BVK model by BVK_a , with the subscript indicating air density correction. Conduct an out-of-sample test on BVK and BVK_a and observe what type of effect the air density correction has on the kernel regression. Also compare the RMSE of BVK_a with that of $\text{AMK}(V, D, \rho)$ in Table 5.6, i.e., the column under “BVK+air density,” and see which one performs better. If $\text{AMK}(V, D, \rho)$ performs better, what does that tell us?

Production Efficiency Analysis and Power Curve

The use of efficiency metrics for wind turbines is important for evaluating their productivity and quantifying the effectiveness of actions that are meant to improve their energy production. The IEC [102] recommends using (1) annual energy production (AEP), (2) the power curve, or (3) the power coefficient, for the purpose of performance evaluation of wind turbines. The drawback of using power output directly, as in the case of AEP, is obvious, because wind power output is affected by wind input conditions, which are variable and not controllable. While the total output does matter in an owner/operator's decisions, a wind turbine's efficiency should be evaluated while the input conditions are controlled for or set to comparable levels. Generally speaking, productive efficiency metrics used in the wind industry take the form of a ratio, which is often the observed wind power production normalized by a benchmark denominator. Different metrics apparently use different denominators. Power curves as we discuss in Chapter 5 are useful in producing some of the denominators.

6.1 THREE EFFICIENCY METRICS

We describe three efficiency metrics commonly used for wind power production—availability, power generation ratio (PGR), and power coefficient. Please be reminded that the wind speed used is adjusted through the air density correction in Eq. 5.3, unless otherwise noted.

The efficiency of a wind turbine is usually measured for a specific time duration, be it a week, a month, or a year, in which the turbine's efficiency is assumed constant. Consider the weekly resolution as an example. Analysts

calculate a single value for the chosen efficiency metric for every single week and evaluate the time series of that metric. The same calculation can be easily extended to other time resolutions. Denote by (V_t, ρ_t, y_t) , $t = 1, \dots, n$, the data pairs of wind speed, air density, and wind power, observed during a unit period under the given resolution.

6.1.1 Availability

One of the efficiency metrics used broadly in the wind industry is *availability* [39, 209] described in the industry standard IEC Technical Specifications (TS) 61400-26-1 [103]. The availability tracks the amount of time in which power is produced by a turbine and then compares it to the amount of time when the turbine could have produced power. A wind turbine is supposed to produce power when the wind speed is between the cut-in and cut-out wind speeds, the two design characteristics of a turbine as described in connection with Fig. 1.2. Turbines are expected to produce power at all times when the recorded wind speed is within these two limits. If a turbine does not produce power when the wind conditions allow, the turbine is then deemed unavailable. The availability is thus defined as

$$\text{Availability} = \frac{\#\{(V_t, \rho_t, y_t) \text{ s.t. } y_t > 0, V_{ci} \leq V_t \leq V_{co}\}}{\#\{(V_t, \rho_t, y_t) \text{ s.t. } V_{ci} \leq V_t \leq V_{co}\}}, \quad (6.1)$$

where s.t. means *such that* and $\#\{\cdot\}$ counts the number of elements in the set defined by the brackets. The denominator in Eq. 6.1 approximates the total time, in terms of the number of 10-min intervals, that a turbine is expected to produce power, whereas the numerator approximates the total time that a turbine does produce power.

6.1.2 Power Generation Ratio

While the availability calculates a ratio in terms of the amount of up running time, the power generation ratio defines a ratio relevant to the amount of power output. The idea is similar to that of *production-based availability*, recently advocated by the industry standard IEC TS 61400-26-2 [105]. By contrast, the availability discussed in Section 6.1.1 is referred to as the *time-based availability*. The production-based availability calculates the ratio of actual energy production relative to the potential energy production, where the potential energy production is the sum of the actual energy production and the lost production caused by an abnormal operational status of a turbine (e.g., downtime, curtailment). The lost production needs to be estimated and its estimation requires detailed information about a turbine's operating status, not something easily accessible to anyone outside the immediate operator of a wind turbine or a wind farm.

Instead of estimating the lost production, let us make a revision so that the assessment is easier to carry out. The revision is to use a power curve to

provide the value of potential energy production under a given wind or weather condition. The power curve used could be the turbine manufacturer's nominal power curve for its simplicity, or the advanced multi-dimensional power curves as described in Chapter 5 for better accuracy. The resulting ratio is referred to as PGR, which is in spirit similar to the production-based availability.

Let $\hat{y}(\mathbf{x})$ denote the potential energy production under input condition \mathbf{x} and $y(\mathbf{x})$ denote the actual energy production under the same condition. In fact, $y(\mathbf{x}_t) = y_t$. The PGR of a given time duration (including n observations) can then be computed as

$$\text{PGR} = \frac{\sum_{t=1}^n y(\mathbf{x}_t)}{\sum_{t=1}^n \hat{y}(\mathbf{x}_t)} = \frac{\sum_{t=1}^n y_t}{\sum_{t=1}^n \hat{y}(\mathbf{x}_t)}. \quad (6.2)$$

If only the wind speed is considered, then $\mathbf{x} = (V)$ and the potential and actual energy production are, respectively, $\hat{y}(V_t)$ and $y(V_t)$.

6.1.3 Power Coefficient

Different from the availability and PGR, the power coefficient explicitly reflects the law of wind energy production, as described in Eq. 5.2, and measures the aerodynamic efficiency of a wind turbine. Power coefficient, C_p , refers to the ratio of actual energy production to the energy available in the ambient wind flowing into the turbine blades [229]. Based on Eq. 5.2, C_p can be expressed as

$$C_p(t) = \frac{2y_t}{\rho_t \cdot \pi R^2 \cdot V_t^3}, \quad (6.3)$$

for any given observation t . Note here that the C_p calculation uses the wind speed without the air density correction since the calculation itself involves air density.

Power coefficient, C_p , is typically modeled as a function of the tip speed ratio (i.e., the ratio of the tangential speed of the tip of a blade and the hub height wind speed), attack angle (related to wind direction), and air density. This dependency of C_p on weather-related inputs makes the power coefficient a *functional curve*, often plotted against the tip speed ratio. Like in the binning-based estimation of power curves, analysts bin individual C_p values by groups of one meter per second according to their respective wind speed and average the power coefficients in each bin to produce a C_p curve. In practice, the largest power coefficient on the curve, as the representative of the whole curve, is used for quantification of the aerodynamic efficiency [123, 126]. The peak power coefficient is a popular efficiency measure used to evaluate wind turbine designs and various control schemes including pitch and torque controls. The theoretical upper limit for the power coefficient is known as the Betz limit ($=0.593$) [18].

Fig. 6.1 presents a plot of two power curves and a power coefficient curve. In the left panel, the relative position of a power curve suggests relative pro-

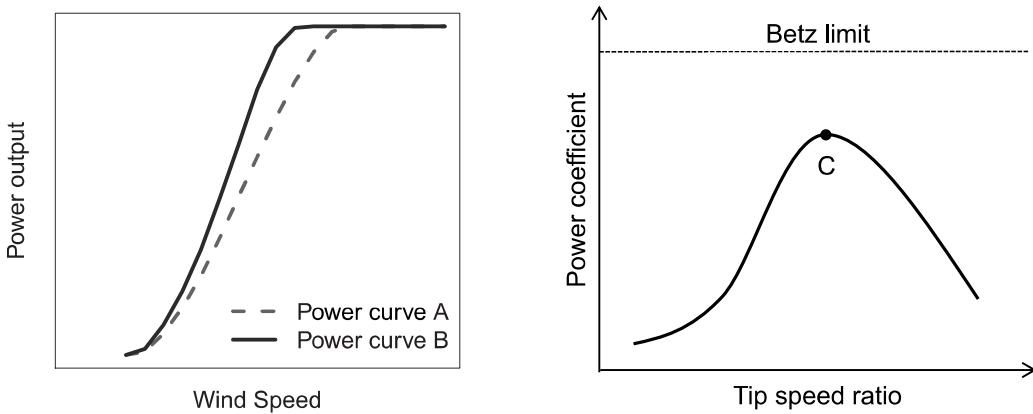


FIGURE 6.1 Left panel: two power curves indicating relative efficiencies of wind turbines, in which curve B suggests a higher productive efficiency; right panel: power coefficient curve and the Betz limit. (Reprinted with permission from Hwangbo. et al. [96].)

ductive efficiency, whereas in the right panel, point C corresponds to the peak power coefficient, used to represent a turbine's efficiency.

6.2 COMPARISON OF EFFICIENCY METRICS

When evaluating the efficiency based on multiple metrics, an immediate question to be addressed is whether or not the evaluation from each metric draws the same conclusion. If the metrics do not always agree with one another (they indeed do not), then subsequent questions are how consistent the results are based on the different metrics and which metric provides a better insight into turbine efficiency.

Niu et al. [155] compare the metrics described in the previous section by using the Offshore Wind Farm Dataset2. The layout of the offshore wind farm is sketched in Fig. 6.2.

The wind power data in all datasets associated with the book are normalized by a turbine's rated power. But to compute the power coefficient, the actual power output is needed. For the offshore wind turbines in the Offshore Wind Farm Dataset2, their characteristics follow what is presented in Table 5.1, meaning that the rated power of these offshore turbines is around 3 MW. So, we use 3 MW as the rated power to recover the actual power output in the subsequent calculation.

Temporal resolutions that are examined include weekly, monthly, quarterly, and yearly resolutions, with a primary focus on weekly and monthly as they provide greater numbers of data points and details.

For each temporal resolution, Niu et al. [155] calculate the three metrics of availability, PGR, and power coefficient as described in Section 6.1; hereafter

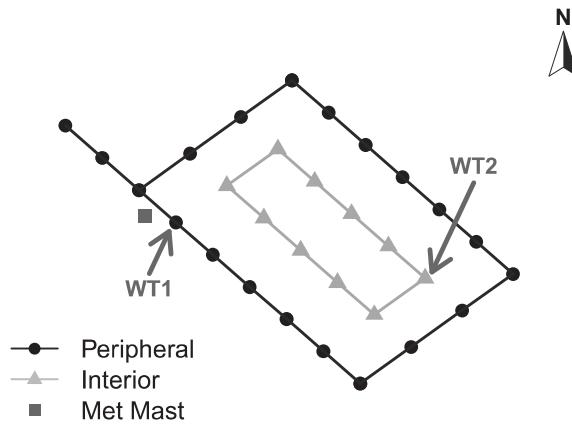


FIGURE 6.2 Basic layout of the offshore wind farm used in the study. Peripheral turbines are located along the black lines and interior turbines along the gray lines. A meteorological mast is indicated by a point along the left edge of the farm. (Reprinted with permission from Niu et al. [155].)

denoted as M1, M2, and M3, respectively. While the averages of M1 and those of M2 calculated for each turbine are within a similar range (0.75–1), the averages of M3 are noticeably lower, at the 0.35–0.5 range, about half the values of M1 and M2. This is understandable as the power coefficient (M3) is limited by the Betz limit to a theoretical maximum of 0.593, and commercial turbines realistically operate around 0.45 [19, page 16]. To make all three metrics comparable in magnitude, Niu et al. multiply M3 by two and use the rescaled metric ($2 \times M3$) for the subsequent analysis.

Fig. 6.3, left panel, presents the time-series plots of the three metrics at the monthly resolution over a four-year span. The figure demonstrates that the metrics follow similar overall trends, with peaks and troughs at similar periods of time. The level of variation associated with the three metrics looks similar. In fact, all the three metrics have similar coefficients of variation, though the one for M2 tends to be slightly higher—on average, 0.264 for M2 compared to 0.254 and 0.252 for M1 and $2 \times M3$, respectively. These patterns and characteristics are consistently observed in the other turbines on the wind farm. Similar insights can be drawn for the weekly resolution.

Table 6.1 presents the correlation coefficients between the metrics for the peripheral turbine. As shown in the first two rows, the correlation coefficients are above 0.9, indicating strong correlations between the metrics. By considering the well-aligned time-series and the high correlation coefficients, one may impulsively conclude that the three metrics are consistent with each other and they can substitute for one another when evaluating turbine efficiency. However, if we eliminate some periods of nearly zero power production (for

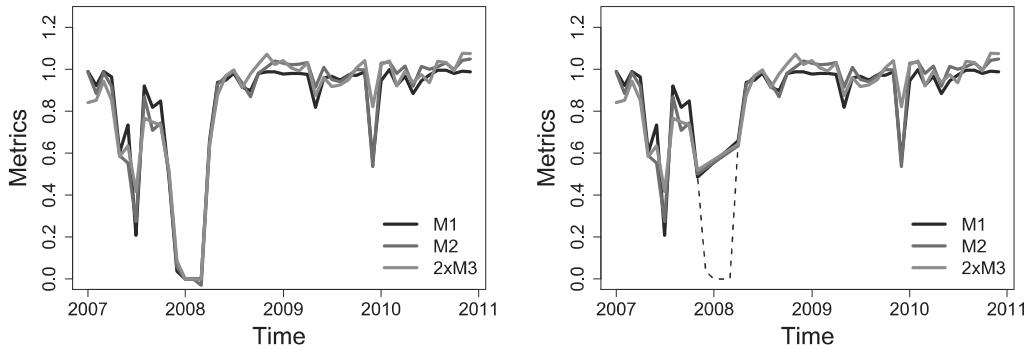


FIGURE 6.3 All three metrics plotted at monthly time resolution for one of the peripheral turbines closest to the meteorological mast. Left panel: for the full period; right panel: after eliminating the periods in which the turbine does not operate for most of the time (dashed line). (Reprinted with permission from Niu et al. [155].)

TABLE 6.1 Correlation between metrics for a peripheral turbine. Weekly and monthly temporal resolutions are shown below.

	M1 vs. M2	M1 vs. 2×M3	M2 vs. 2×M3
Weekly resolution (full)	0.975	0.946	0.959
Monthly resolution (full)	0.986	0.966	0.978
Weekly resolution (reduced)	0.843	0.661	0.785
Monthly resolution (reduced)	0.956	0.876	0.929

Source: Niu et al. [155]. With permission.

example, a period for which any metric is below 0.2; see Fig. 6.3, right panel), the metrics based on such a reduced period produce significantly lower correlation coefficients—for this particular turbine, as low as 0.661 between M1 and 2×M3 at the weekly time resolution. This implies that the original high correlation derived from the full period data could be contributed to substantially by the non-operating periods of the turbine. The lower correlation based on the reduced period further suggests possible disparity between the metrics under typical operating conditions. In the following subsections, the metric values presented are calculated for the reduced period only, for better differentiating the metrics’ capability in quantifying turbine efficiency.

6.2.1 Distributions

Fig. 6.4 demonstrates the distributions of the calculated metrics for a single turbine, but it is representative of the other turbines as they all show similar distribution spreads. While M2 and 2×M3 both have relatively broad spreads

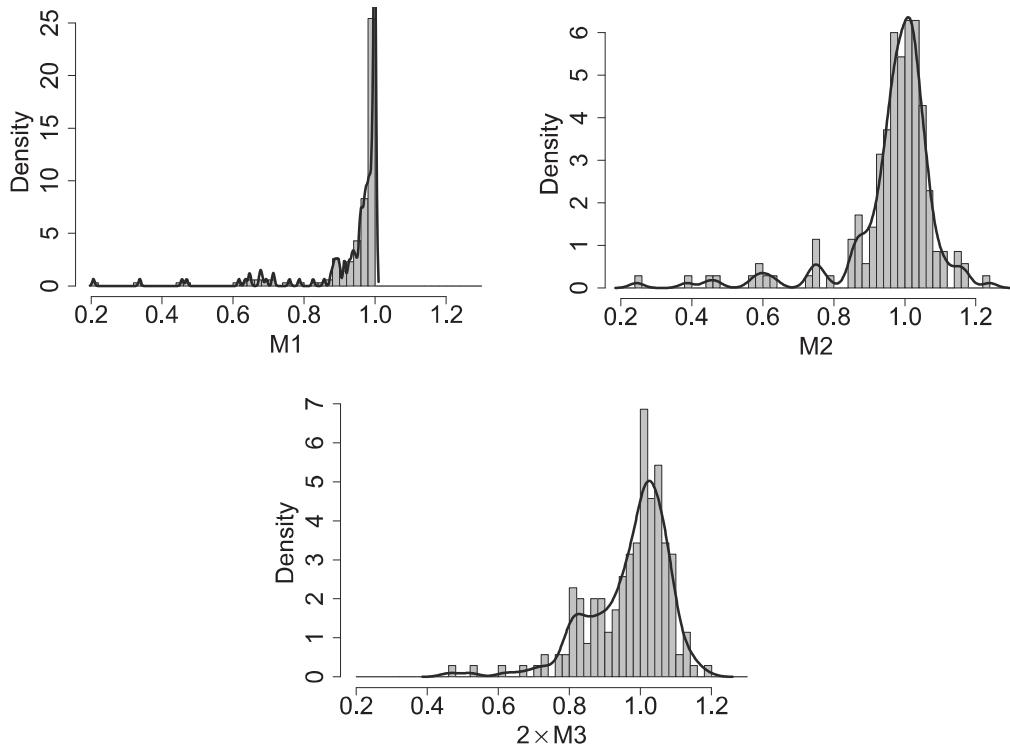


FIGURE 6.4 Histograms and probability density curves of the metric values at the weekly time resolution for the peripheral wind turbine. Top-left panel: availability (M1), top-right panel: power generation ratio (M2), bottom-middle panel: power coefficient ($2 \times M3$). (Reprinted with permission from Niu et al. [155].)

of data, M1 has a much narrower range. A significant portion of its density is concentrated near one at which the distribution is truncated with a steep taper to lower values. In contrast, M2 and $2 \times M3$ both take a shape similar to the bell-shaped curve with smoother tapers in both directions. M1's concentration of values makes it difficult to differentiate between the turbine efficiency at different time periods, because as more values are within the same range, the variations in turbine performance are concealed. This can potentially mislead turbine operators into a conclusion that the turbines operate at a similar efficiency level, even though the underlying turbine efficiencies differ.

Such a unique distributional characteristic of M1 can be inferred by its calculation procedure. As expressed in Eq. 6.1, the numerator of M1 counts the number of elements in a set that is a subset of the one associated with the denominator, so it has a maximum value of one at all points in time. This is a desired property for an efficiency metric, which is not observed in M2 or $2 \times M3$. M2 can exceed one because a power curve displays the expected power values as an averaged measure but particular instances of wind power

production could exceed the expected productions. The power coefficient itself is smaller than one, but doubling the power coefficient value, namely $2 \times M_3$, is bounded from above by twice the Betz limit at 1.186, which itself is greater than one. It is interesting to observe that M_2 appears to be bounded by a value similar to 1.186.

The unique property of M_1 when combined with its binary quantification of whether or not power was generated, however, adversely affects its quantification capability. As long as a turbine is generating power at a point in time, that point would be counted as a one. Even the cases when the power production is significantly lower than expected would still be counted as ones. Averaging over these counts produces the metric weighted heavily towards one. Using the availability metric, M_1 , periods with high actual efficiency, in terms of the amount of actual power production, look the same as low efficiency periods as long as the power produced exceeds a low threshold.

The methods calculating M_2 and M_3 , on the other hand, allow for a sliding scale measure of power production so that they account for how much power is produced. Values of M_2 and $2 \times M_3$ thus have greater spread and do not concentrate around any particular value as narrowly as M_1 does. This ability to better distinguish between time periods of differing performance as well as the distributional features render M_2 and $2 \times M_3$ stronger metrics than M_1 . They allow for a fuller portrayal of a turbine's efficiency over time as opposed to M_1 's more general overview of whether or not the turbine is in operation.

6.2.2 Pairwise Differences

Fig. 6.5 illustrates the absolute difference between the calculated metrics on a weekly basis. Darker bars indicate the periods of significantly large differences, whereas lighter bars are the periods of smaller differences.

Fig. 6.5, bottom panel, shows that the large differences between M_2 and $2 \times M_3$ are sparsely distributed through the four years. In contrast, as shown in Fig. 6.5, top and middle panels, there are significantly more instances of large value differences between M_1 and either of the other metrics, especially between M_1 and $2 \times M_3$. This implies that both M_1 and $2 \times M_3$ are more similar to M_2 than to each other. M_1 and M_2 calculate a ratio of the actual performance over the expected performance, although M_1 focuses on the amount of time and M_2 examines the amount of power. This sets $2 \times M_3$ apart from M_1 and M_2 . On the other hand, M_2 and $2 \times M_3$ quantify turbine efficiency with respect to the amount of power production, whereas M_1 concerns the amount of operational time, which makes M_1 distinct from the other two.

In Fig. 6.5, the large or medium differences tend to be heavily concentrated within some specific periods, notably in the second half of 2007 and the first half of 2010. In fact, these periods represent those in which turbines' true efficiencies are relatively low. There are two different aspects describing this phenomenon.

First, recall from Fig. 6.4 that M_1 tends to be heavily weighted towards

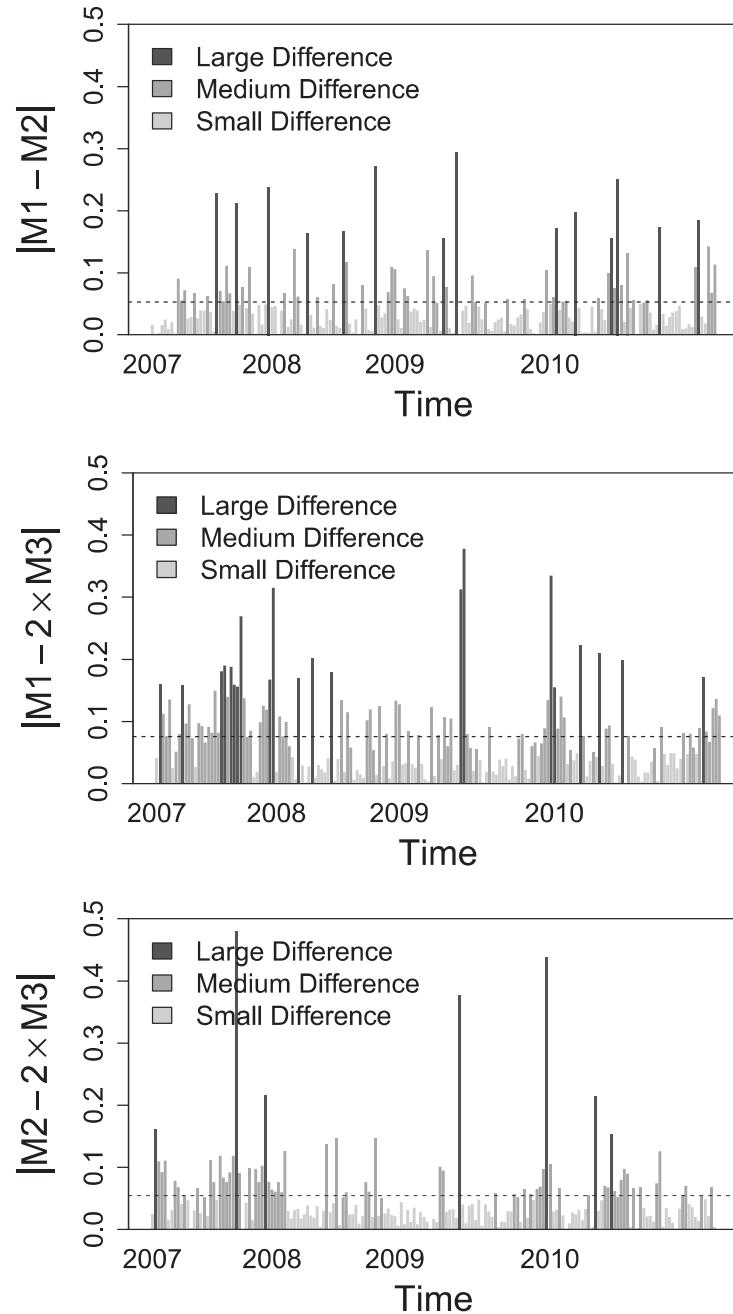


FIGURE 6.5 Magnitudes of absolute pairwise difference between metric values at weekly resolution for the peripheral wind turbine. Top panel: availability (M1) versus PGR (M2), middle panel: availability (M1) versus power coefficient ($2 \times M3$), bottom panel: PGR (M2) versus power coefficient ($2 \times M3$). The dashed line in each plot is the average of the absolute differences in that plot. An absolute difference is considered as a small difference, if its value is smaller than 0.05, as a large difference, if its value is greater than 0.15, and as a medium difference, if its value is in between. (Reprinted with permission from Niu et al. [155].)

its maximum, overestimating the turbine's efficiency in the relative scale. If a turbine produces some power for most time instances within a given period, its availability should be close to one. The large differences between M1 and the other two metrics then imply that the turbine is producing some power for most of the times but the amount of the power production is considerably lower relative to the expectation. If one refers to Fig. 6.3, one may notice that M1 is higher than the other two in the later part of 2007.

Secondly, recall that M3 represents a *maximum effect* (on the C_p curve), whereas M2 is an *integration effect*. For a functional response, the two effects can be understandably different. The large differences between M2 and $2 \times M3$ suggest that a turbine produces a sufficient amount of power only for a small portion of the given time period. In this case, the turbine's maximum efficiency measured by $2 \times M3$ is relatively high, but M2 is relatively low because the turbine does not produce much power on average during the same period (see the middle of 2007 and the beginning of 2010 in Fig. 6.3). M1 also measures an *integration effect*, but in terms of the operational time, so that the same argument is applicable to explaining the difference between M1 and $2 \times M3$. Most of the time when there is a large difference between M2 and $2 \times M3$, a large difference between M1 and $2 \times M3$ is also observed (see Fig. 6.5, middle and bottom panels).

All of these observations can be found in the cases of other turbines as well. Although the concentration periods of large and medium differences vary, all turbines display the clustering pattern, and such clusters are closely related to different characteristics of the metrics.

When comparing the mean of the absolute differences between the metrics, indicated by the dashed horizontal lines in Fig. 6.5, the disparity between the metrics becomes less pronounced. While a metric pair with the smallest mean difference varies by different turbines, the metric pair of the largest mean difference is consistently observed as between M1 and $2 \times M3$. This suggests that M2 has comparably closer values to M1 and $2 \times M3$. As such, M2 is more consistent in value with either of M1 and $2 \times M3$ and its values are a better reflection of all three metrics.

6.2.3 Correlations and Linear Relationships

Table 6.1 shows that the correlation calculated using the reduced period is the highest between M1 and M2 for most turbines. The correlations between M2 and $2 \times M3$ (or equivalently, between M2 and M3) are also relatively high. For most turbines, the correlation coefficients between M1 and M2 remain within the 0.8 range at weekly resolution, whereas those between M2 and M3 are generally in the 0.7 range.

The lowest correlations are found between M1 and M3 for all turbines and time resolutions, with the correlation coefficient values usually around 0.5–0.6 but dipping sometimes into the 0.4 range. The values displayed in Table 6.1 are among the higher values of M1–M3 correlation of turbines. Another turbine

has an M1–M3 correlation of just 0.417 for the reduced weekly data. This indicates that the relationship between these two metrics is much weaker, highlighting the strength of M2 for its much stronger relationship with either of the other metrics.

Weekly time resolution is best for highlighting differences in correlation between metrics. Correlations rise as the time resolution becomes coarse; monthly, quarterly, and yearly resolutions in general return a correlation in the range of 0.9. Niu et al. [155] state that the averaging effect when using a coarse time resolution irons out a certain degree of details, making the metrics based on the coarse time resolutions less differentiating.

To analyze the consistency of the metrics, Niu et al. [155] also evaluate the linearity between any pair of the metrics around the $y = x$ line. Let us generate data points (x, y) paired by the values of two selected metrics. If the data points perfectly fit to the $y = x$ line, an increase in one metric implies the same amount of increase in the other metric. As such, their ability to capture change in efficiency is identical, or equivalently, they are consistent.

However, as noted earlier, the scales of the metrics are not the same, e.g., M1 and M2 are about twice the unscaled M3. To assess the extent of linearity around the $y = x$ line requires us to estimate the exact scale between the metrics.

To align the scales, Niu et al. [155] perform a linear regression upon the different metric pairs. For example, for the M1–M2 pair, fit a linear model of $M1 = \beta \cdot M2 + \varepsilon$ to estimate β , where ε is the random noise. Let $\hat{\beta}$ denote the coefficient estimate. Then, the estimate, $\hat{\beta}$, is used to rescale the values of M2, generating the scale-adjusted data points $(M1, \hat{\beta} \cdot M2)$. With the scale adjustment, the data points should be centered about the $y = x$ line. If they show strong linearity around the $y = x$ line, one can conclude that the metrics for the corresponding pair are consistent with each other. To determine the extent of linearity, the average magnitude of the data points' vertical distance from the $y = x$ line (in an absolute value) is computed.

Fig. 6.6 presents the scatter plots of the scale-adjusted metrics and the $y = x$ line. For illustration purposes, two scatter plots are presented, one for the peripheral turbine used previously and the other is an interior turbine. For the metrics calculated for the peripheral turbine, the scale adjustment coefficients ($\hat{\beta}$) are 0.97, 1.93, and 1.99 for M1–M2, M1–M3, and M2–M3 pairs, respectively. The coefficient of 0.97 for the M1–M2 pair, for instance, implies that M2 will have the same scale with M1 after multiplying it by 0.97. For the interior turbine, the scale adjustment coefficients are 0.98, 2.01, and 2.06, for the three pairs of metrics in the same order, respectively.

In the figure, points are more concentrated near where x and y equal one. Whenever x refers to M1, there is a very apparent clustering of points at $x = 1$ due to the truncation of the distribution of M1 at one. On the other hand, the data points for the M2–M3 pair are well spread around the region, a characteristic reminiscent of the metrics distributions examined earlier.

After the scale-adjustment, whenever the y -axis represents a rescaled M3

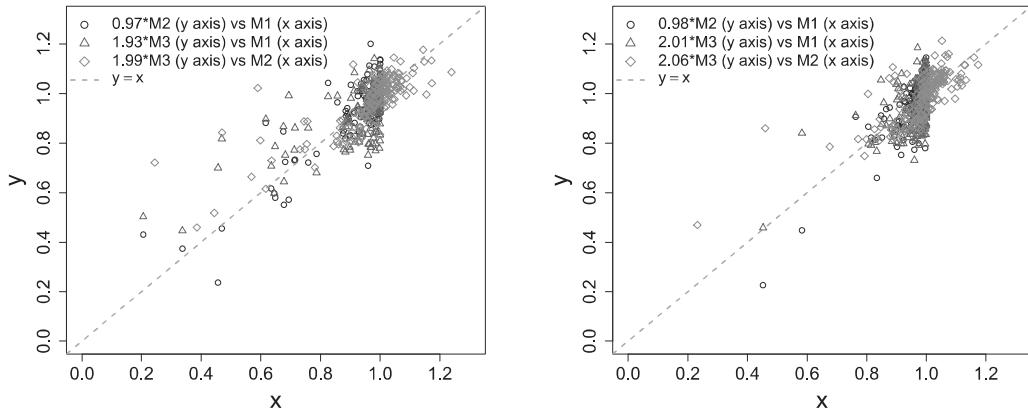


FIGURE 6.6 Linear relationships between metrics at the weekly time resolution. Left panel: for a peripheral turbine, right panel: for an interior turbine. Plots generated from scaling values by x -to- y ratio. The dashed line illustrates the $y = x$ line. (Reprinted with permission from Niu et al. [155].)

TABLE 6.2 Linearity between a pair of performance metrics measured by the average absolute vertical distances from the $y = x$ line.

	M1 vs $\hat{\beta} \cdot M2$	M1 vs $\hat{\beta} \cdot M3$	M2 vs $\hat{\beta} \cdot M3$
A peripheral turbine	0.050	0.068	0.055
An interior turbine	0.046	0.068	0.052

Source: Niu et al. [155]. With permission.

(triangles and diamonds), the data points tend to be placed above the $y = x$ line for relatively low x values, e.g., less than 0.8. This confirms the difference between the maximum effect (for M3) and the integration effect (for M1 and M2) discussed earlier.

As shown in Table 6.2, the average distances between the points and the $y = x$ line is the greatest for the M1–M3 pair for both turbines, suggesting that the M1–M3 pair has the weakest extent of linearity. This reinforces the understanding from the analysis of absolute differences that M1 and M3 are the least consistent metrics, while M2 has a stronger relationship with both other metrics.

6.2.4 Overall Insight

According to the previous analyses, while all metrics display some level of consistency, PGR (M2) is the most consistent with the other metrics. The absolute differences in metric values demonstrate that PGR produces values that are more representative of the three metrics. Correlations between the metrics

also suggest that changes in turbine performance mapped by PGR are illustrative of such trends displayed by other metrics. Moreover, evaluation of the linearity between the metrics shows that availability (M1) or power coefficient (M3) has a stronger relation with PGR (M2) than with each other. It is not too far fetched to reach the conclusion that PGR better represents all three metrics. Various aspects of the analysis have shown availability's deficiency in discriminating changes in turbine performance. Practitioners are well aware of availability's deficiency, which becomes the chief reason for adopting the production-based availability measure recently.

6.3 A SHAPE-CONSTRAINED POWER CURVE MODEL

As said earlier in this chapter, efficiency metrics used in the wind industry take the form of a ratio, which is often the observed wind power production normalized by a benchmark denominator. Availability, albeit a ratio, does not use power output in either numerator or denominator. The metric that resembles availability and does use power output in both numerator and denominator is the capacity factor (CF). We did not include the capacity factor in the discussion of the previous section, because it is typically used for wind farms and much less often used for individual wind turbines. Its concept, however, is indeed applicable to wind turbines.

The capacity factor of a turbine is the ratio of the observed power output over the turbine's maximum production capacity at the rated power. When calculating the capacity factor, one assumes that the turbine is operating at its full capacity all the time. The use of the capacity factor entirely ignores the wind condition, so much so that its denominator calculates the absolute maximally possible wind power that can be produced for a given period for the specific design of the said wind turbine. In this sense, the capacity factor's denominator is even more aggressive than that used in availability, as availability only counts the time when the wind speed is between the cut-in and cut-out speeds.

Nevertheless, if analysts put all the metrics that use powers in their numerator and denominator side by side, as shown in Fig. 6.7, one can notice that they indeed have the same numerator, which is the observed power output. But the denominators are different, meaning that different benchmarks are used in computing the respective ratio. This in fact raises a question—what should be used as a performance benchmark? Hwangbo et al. [96] argue that to quantify a turbine's productive efficiency, one would need to estimate the best achievable performance as a benchmark, so that the ratio of the current performance to the best achievable performance quantifies the degree to which the turbine has performed relative to its full potential. In order to estimate the best achievable performance of a wind turbine, Hwangbo et al. look into the field of production economics [81], which refers to the “best achievable performance” as an *efficient frontier*. To facilitate this line of discussion, we start with some background on production economics.

Capacity Factor	Power Coefficient	Production-based Availability
$\frac{\text{Observed power output}}{\text{Maximum power output}}$	$\frac{\text{Observed power output}}{\text{Energy available in the wind}}$	$\frac{\text{Observed power output}}{\text{Expected power output}}$

FIGURE 6.7 Capacity factor, power coefficient, and the production-based availability or power production ratio.

6.3.1 Background of Production Economics

Efficiency analysis is a primary focus in production economics. Efficiency quantification is based on the estimation of a production function and the explicit modeling of systematic inefficiency, using input and output data for a set of production units, be it firms, factories, hospitals or power plants. Consider a set of production units (e.g., a wind farm) using x input (e.g., investment in a wind energy project) and producing y output (e.g., revenue from power generation). Analysts can create a scatter plot of many x - y data pairs coming from different production units or the same production unit but over different periods; see Fig. 6.8. Assuming no measurement errors associated with x and y , a common estimator in production economics—data envelopment analysis (DEA) [11]—estimates the efficient frontier enveloping all the observations.

The concept of an efficient frontier is understood as follows—a production unit whose input-output is on the frontier is more efficient than the production units whose input-output is being enveloped by the frontier. Consider observation D. Using the same input, the production unit associated with D produces less output than the production unit associated with point E; while to produce the same output, the production unit associated with D needs more input than the production unit associated with point F. For this reason, the production unit associated with D must be inefficient.

The efficient frontier is also called the *production function*, denoted by $Q(x)$. The production function characterizes producible output given input x in the absence of inefficiency. Using the production function, the output of the inefficient production unit D can be expressed as

$$y_D = Q(x_D) - u_D, \quad (6.4)$$

where $u_D \geq 0$ denotes the systematic inefficiency.

To estimate the production function $Q(x)$, certain assumptions are made restricting the shape of the frontier. The most common assumption is that the frontier forms a monotone increasing concave function consistent with basic stylized characteristics of production [222]. When the data are assumed noise free, the tightest boundary enveloping all observations and maintaining monotonicity and concavity is a piecewise linear function.

Let us consider the context of power production of a wind turbine, in which a wind turbine is a power production unit, wind speed is the dominating

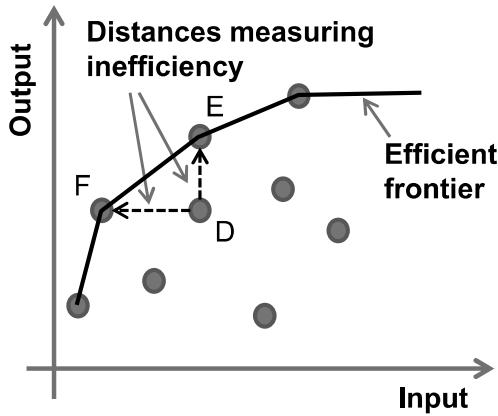


FIGURE 6.8 Production data and efficient frontier. (Reprinted with permission from Hwangbo et al. [96].)

input driving power production, and the generated power is the output. When applied to the wind turbine data, the use of convex or concave piecewise linear methods assuming noise-free data encounter some problems.

The first is the noise-free assumption. The frontier analysis approaches assuming noise-free observations are referred to as *deterministic*. But the wind turbine data, like all other physical measurements, are inevitably contaminated by noises. The problem with applying a deterministic approach to noisy wind production data is that it tends to overestimate the best performance benchmark because every observation is assumed to be achievable.

The second difference is that the shape of the wind-power scatter plot is not concave. When discussing Fig. 1.2, we show that the wind-power data appears to follow an S-shape, comprising a convex region, followed by a concave region, and the two segments of curves are connected at an inflection point. Fig. 6.9 makes this point clearer with its right panel illustrating the meaning of convexity and concavity.

In production economics, the need to model noise is established, promoting the *stochastic* frontier analysis (SFA) [5], which adds a random noise term ε to Eq. 6.4. When applying the SFA modeling idea to wind turbine data and replacing the generic input variable x with wind speed variable V , Hwangbo et al. [96] define their production function as

$$y = Q(V) - u(V) + \varepsilon, \quad (6.5)$$

where ε is assumed having a zero mean, while the systematic inefficiency term $u(V)$ is a non-negative random variable with positive mean, i.e., $\mu(u(V)) := \mathbb{E}[u(V)] > 0$. Note that $u(V)$ is a function of V , meaning that the amount of inefficiency varies as the input changes, known as a *heteroskedastic* inefficiency term.

While the SFA research considers the noise effect in observational data,

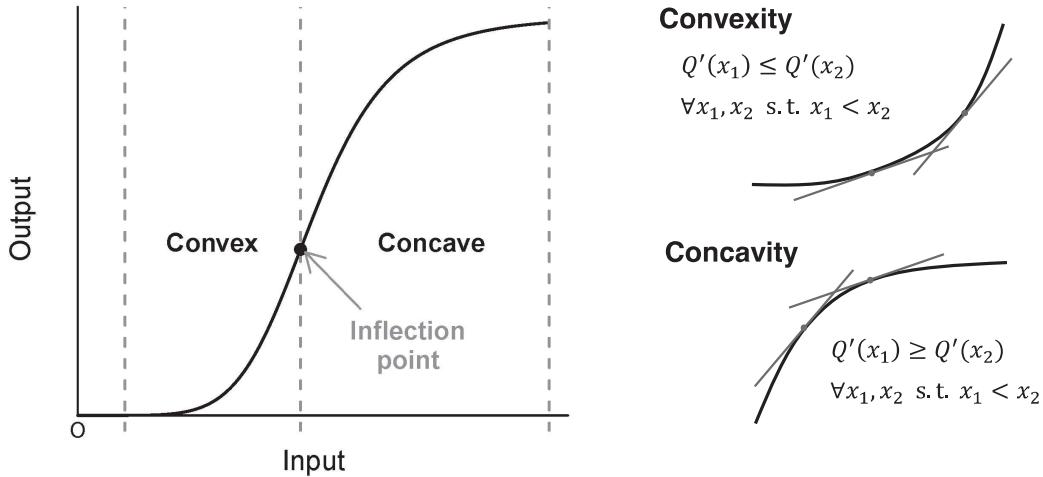


FIGURE 6.9 S-shaped curve, convexity and concavity. $Q'(x)$ is the first derivative of $Q(x)$ with respect to x . (Left panel reprinted with permission from Hwangbo et al. [96].)

analysts still need to address the second challenge mentioned above, namely, the S-shaped curve exhibited in the wind turbine data. In the vernacular of economists, the S-shape constraint is known as the regular ultra passum (RUP) law [81], which is motivated by production units having an increasing marginal rate of productivity followed by a decreasing rate of marginal productivity. Olesen and Ruggiero [156] develop a RUP law satisfying frontier analysis but their production function estimator is a deterministic DEA-type estimator, enveloping all observations from above, and consequently suffers from the overestimation problem that all other deterministic production function estimators suffer.

We want to note that methods from production economics have recently been used in wind energy applications [33, 96, 97, 162]. Two of them are deterministic, with one being a DEA type estimator [33] and the other called a free disposal hall estimator [162]. The production function estimator proposed by Hwangbo et al. [96, 97] is stochastic and attempting to be RUP law satisfying, subject to a rigorous proof in terms of consistency.

6.3.2 Average Performance Curve

The basic production function model in Eq. 6.5 can be re-written as,

$$\begin{aligned} y_i &= [Q(V_i) - \mu(V_i)] + [\mu(V_i) - u(V_i) + \varepsilon_i] \\ &= g(V_i) + e_i, \quad i = 1, \dots, n, \end{aligned} \tag{6.6}$$

by letting $g(V_i) := Q(V_i) - \mu(V_i)$ and $e_i := \mu(V_i) - u(V_i) + \varepsilon_i$. The error term e is a redefinition of the error term with expectation zero. The above expres-

sion connects the power curve with the production function because $g(V)$ is effectively the power curve. As the power curve passes through the middle of the wind-power data, it is a curve representing the average performance of a turbine, also known as the average-practice function in the production economics literature. Understandably, the production frontier function, $Q(V)$, differs from the power curve, $g(V)$, by the mean of the inefficiency varying by V .

This connection helps lay out the intuition behind the procedure of estimating the production frontier function, $Q(V)$. One would start with a power curve from the wind turbine data. Then estimate the mean function of the inefficiency and use it to rotate the average performance curve to the new position to be the production frontier function.

Hwangbo et al. [96] note that because the final $Q(V)$ needs to satisfy the RUP law, i.e., the S-shape constraint, the average performance power curve, $g(V)$, that comes before the production function must, too, satisfy the same shape constraint. This requirement makes this specific power curve estimation procedure different from those currently used in practice, including all discussed in Chapter 5 because none of them imposes the S-shape constraint explicitly.

Estimating the shape-constrained power curve, $g(V)$, requires imposing convexity and concavity in the low and high wind speed regions, respectively. The convex segment should connect to the concave segment at the inflection point, which itself needs to be estimated from the data. Hwangbo et al. [97] formulate the estimation of the average performance curve as a constrained least squares estimation problem, which is to minimize the residual sum of squares, $\sum_{i=1}^n (y_i - g(x_i))^2$, subject to the constraints imposing monotonicity and S-shape on $g(\cdot)$.

In the absence of prior knowledge of the location of the inflection point, x^* , Hwangbo et al. [97] make use of a set of grid points and treat each of the grid points as a potential inflection point location. Provided a grid point, Hwangbo et al. [97] partition the functional domain left of the grid point as the convex region and the domain right of the grid point as the concave region and then construct a function g , minimizing the aforementioned residual sum of squares, while satisfying the sets of constraints applicable to the partitioned regions. The estimation of the convex segment or the concave segment can be done individually by using the method of convex nonparametric least squares (CNLS) [127]. Then, Hwangbo et al. [97] choose the grid point resulting in the smallest residual sum of squares as the estimate of the inflection point and the corresponding g as the estimator of the average performance power curve.

Suppose that m grid points, t_1, \dots, t_m , are given. Also, assume that x_i 's are distinct (no duplicated values) and arranged in a non-decreasing order for a given n , i.e., $x_i \leq x_j$ whenever $i < j$. For each t_k for $k = 1, \dots, m$, let $\mathcal{V}^{(t_k)}$ and $\mathcal{C}^{(t_k)}$ be the sets of input points that belong to the (imposed) convex and

concave regions, respectively, i.e.,

$$\mathcal{V}^{(t_k)} = \{x_i \mid x_i < t_k, i = 1, \dots, n\},$$

and

$$\mathcal{C}^{(t_k)} = \{x_i \mid x_i \geq t_k, i = 1, \dots, n\}.$$

Then, for each t_k , solve the following quadratic programming with respect to $\mathbf{g} = (g(x_1), \dots, g(x_n))$ and $\boldsymbol{\beta} = (\beta_1, \dots, \beta_n)$:

$$\min_{\mathbf{g}, \boldsymbol{\beta}} z^{(t_k)} = \sum_{i=1}^n (y_i - g(x_i))^2 \quad (6.7a)$$

$$\text{s.t. } \beta_i = \frac{g(x_{i+1}) - g(x_i)}{x_{i+1} - x_i}, \forall i \text{ such that } x_i \in \mathcal{V}^{(t_k)}, \quad (6.7b)$$

$$\beta_i \leq \beta_{i+1}, \forall i \text{ such that } x_{i+1} \in \mathcal{V}^{(t_k)}, \quad (6.7c)$$

$$\beta_i = \frac{g(x_i) - g(x_{i-1})}{x_i - x_{i-1}}, \forall i \text{ such that } x_i \in \mathcal{C}^{(t_k)}, \quad (6.7d)$$

$$\beta_{i-1} \geq \beta_i, \forall i \text{ such that } x_{i-1} \in \mathcal{C}^{(t_k)}, \quad (6.7e)$$

$$\beta_i \geq 0, \forall i = 1, \dots, n. \quad (6.7f)$$

The constraints in Eq. 6.7d–6.7e, together with the objective function in Eq. 6.7a, are equivalent to the Hildreth type estimator [92] of a function that is concave over $[\max_{x_i \in \mathcal{V}^{(t_k)}} x_i, x_n]$, the constraints in Eq. 6.7b–6.7c, together with the objective function in Eq. 6.7a, describe the estimator for a convex function defined over $[x_1, \min_{x_i \in \mathcal{C}^{(t_k)}} x_i]$, and the inequalities in Eq. 6.7f ensure the monotonicity of g .

Let the minimizer of Eq. 6.7a for a given t_k be $\mathbf{g}^{(t_k)} = (g^{(t_k)}(x_1), \dots, g^{(t_k)}(x_n))$ and $\boldsymbol{\beta}^{(t_k)} = (\beta_1^{(t_k)}, \dots, \beta_n^{(t_k)})$, and let the corresponding optimal objective function value be $z^{(t_k)}$. The vector, $\mathbf{g}^{(t_k)}$, provides estimates only at the given locations, i.e., the g -values at x_i 's. For the functional estimator over the interval between two observational data points, x_i and x_j , Hwangbo et al. [97] use a hyperplane to interpolate between the two locations, i.e., they define $\hat{g}^{(t_k)}(x)$ as

$$\hat{g}^{(t_k)}(x) = \begin{cases} \max\{\alpha_i^{(t_k)} + \beta_i^{(t_k)}x \mid \forall i \text{ such that } x_i \in \mathcal{V}^{(t_k)}\}, & \text{if } x < t_k \\ \min\{\alpha_i^{(t_k)} + \beta_i^{(t_k)}x \mid \forall i \text{ such that } x_i \in \mathcal{C}^{(t_k)}\}, & \text{if } x \geq t_k, \end{cases} \quad (6.8)$$

where $\alpha_i^{(t_k)} := g^{(t_k)}(x_i) - \beta_i^{(t_k)}x_i$ for $i = 1, \dots, n$. Apparently, $\hat{g}^{(t_k)}$ is a piecewise linear function, connecting two adjacent points in the set of $\{(x_i, g^{(t_k)}(x_i)), \forall i = 1, \dots, n\}$ and extending the hyperplanes, $\alpha_1^{(t_k)} + \beta_1^{(t_k)}x$ and $\alpha_n^{(t_k)} + \beta_n^{(t_k)}x$, each toward the adjacent boundary of the input domain. As such, $\hat{g}^{(t_k)}$ is convex on $[\min x, \max_{x_i \in \mathcal{V}^{(t_k)}} x_i]$, concave on

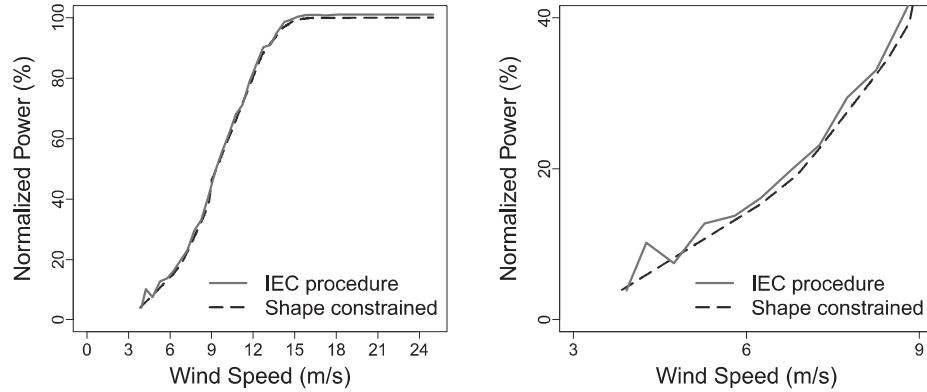


FIGURE 6.10 Illustration of shape-constrained and unconstrained power curves. Left panel: shaped-constrained power curve compared to the IEC binning power curve, right panel: a zoomed view at the wind speed from 3 m/s to 9 m/s. (Reprinted with permission from Hwangbo et al. [96].)

$[\min_{x_i \in C^{(t_k)}} x_i, \max x]$, and linear within $[\max_{x_i \in V^{(t_k)}} x_i, \min_{x_i \in C^{(t_k)}} x_i]$. By letting $t^* = \operatorname{argmin}_{t_k} z^{(t_k)}$, the final average performance curve is simply $\hat{g}^{(t^*)}(x)$. Fig. 6.10 presents a comparison between the shape-constrained power curve versus its unconstrained counterpart using the IEC binning method.

6.3.3 Production Frontier Function and Efficiency Metric

After the average performance power curve, $g(V)$, is estimated, one can take differences between the fitted power curve and the output y . According to the relationship in Eq. 6.6, the resulting residuals are the summation of two random components: $\mu - u$ and ε . The modeling assumption used here is that u is non-negative and ε is symmetrically distributed with respect to a zero mean. As such, one can expect to see a significant decrease at the value of μ in the density curve of the residuals. This understanding is used to estimate μ —if one can locate where the greatest decrease in the residual distribution occurs, it provides an estimate of μ . Hwangbo et al. [96] use the technique in [84] for this estimation. An illustration is given in Fig. 6.11, but we skip the procedure and refer interested readers to [97] for technical details.

The following summarizes the steps in estimating the shape-constrained stochastic production function, $Q(V)$:

1. Use the wind turbine data (wind speed and power) to estimate $g(V)$ while imposing the shape constraints and the continuity requirement at the inflection point. Denote the estimated power curve by $\hat{g}(V)$.
2. Estimate $\mu(V)$, the mean function of the inefficiency term.

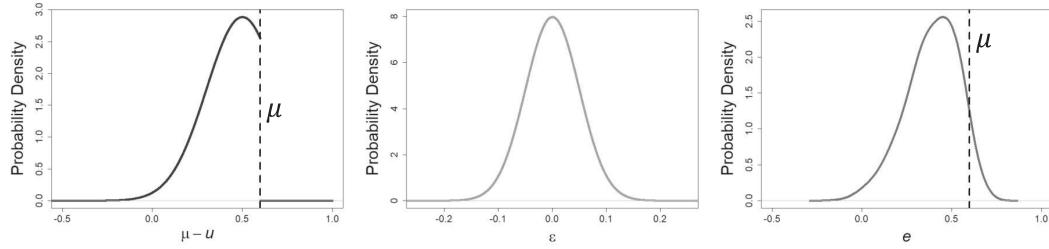


FIGURE 6.11 Estimation of the mean function of the inefficiency term μ . Left panel: density curve of $\mu - u$, middle panel: density curve of ε , right panel: density curve of the residual e .

3. Estimate the production function, $Q(V)$, based on the relationship of $Q(V) = g(V) + \mu(V)$. Denote the estimated frontier curve by $\hat{Q}(V)$.

The estimated production frontier function, $\hat{Q}(V)$, provides a performance benchmark for wind turbines. Hwangbo et al. [96] propose to quantify the productive efficiency of a wind turbine by using the estimated performance benchmark curve and the average performance power curve. Specifically, they propose the following production efficiency measure, PE, which is the ratio of the energy produced under the average performance curve over that under the performance benchmark curve, integrated over the whole wind speed spectrum,

$$\text{PE} = \frac{\int_{V_{ci}}^{V_{co}} \hat{g}(V)dV}{\int_{V_{ci}}^{V_{co}} \hat{Q}(V)dV}. \quad (6.9)$$

Apparently, PE takes a value between zero and one; the closer a PE is to one, the closer the wind turbine performs to its full potential.

One may have noticed that the discussion above treats the production functions, both the frontier function and the average performance function, as univariate. This is because a multivariate production function satisfying the RUP law is still not fully developed. On the other hand, however, besides wind speed, air density and several other environmental variables, including wind direction, humidity, turbulence intensity and wind shear, all potentially affect wind power production, as seen in the analysis in Chapter 5. These environmental influences are not controllable but their existence does play a role affecting the inefficiency estimated from the power output data. Consequently, when comparing the productive efficiency of different turbines or the same turbine over different operational periods, analysts may need to control for the influence of these environmental factors; otherwise, one may wonder what part of inefficiency is due to a turbine's intrinsic differences and what part of inefficiency comes from differences in environmental characteristics such as air density.

This sort of ambiguity can be alleviated if the comparison periods have comparable environmental profiles. The environmental variables are referred

to as *covariates* in the statistical literature. Hwangbo et al. [96] use a covariate matching procedure to select a subset of the data, in order to make the environmental profiles across different time periods as similar as possible, thus removing the effect of environmental influences from the efficiency analysis. The detail of the matching process is described in Section 7.2. One thing worth noting is that the covariate matching does not produce an exact match but a good match instead, subject to the degree of dissimilarity allowed by a prescribed threshold, ϖ . To confirm the quality of the matches, Hwangbo et al. suggest plotting the pdfs of each environmental variable, empirically estimated from the data and visually inspected for assessing how well the pdfs match across the comparison periods.

6.4 CASE STUDY

In this case study, data from two wind turbines in the **Inland Wind Farm Dataset2** and two turbines from the **Offshore Wind Farm Dataset2** are used. The two inland turbines are referred to as WT1 and WT2, respectively, and the two offshore ones as WT3 and WT4, respectively. These turbine names do not imply relationship with the turbines of the same names in Chapter 5. But they do come from the same wind farms, so that the characteristics in Table 5.1 can be referenced for respective turbines. Table 5.1 states that the rated power for the offshore turbines is around 3 MW and that for the inland turbines is between 1.5 MW and 2 MW. In the following numerical analysis, in order to compute the power coefficient, we use 3 MW as the rated power for the offshore turbines and 1.65 MW as the rated power for the inland turbines.

Hwangbo et al. [96] analyze the wind turbine data on an annual basis, which means that they divide the four-year data into four consecutive annual periods. The number of periods is denoted by $T = 4$ and the period index is $t = 1, 2, 3, 4$. Hwangbo et al. evaluate turbine efficiency for each year.

The first step of data processing is to control for the influence of environmental factors, which is to select the subset of data with comparable environmental profiles through the covariate matching method described in Section 7.2. For inland wind turbines, the covariates to be matched include $\boldsymbol{x} = (V, D, \rho, I, S)^T$, whereas for offshore wind turbines, $\boldsymbol{x} = (V, D, \rho, H, I)^T$. The wind shear, S , is left out in the offshore cases because the study in Chapter 5 suggest that conditioned on the inclusion of (V, D, ρ, H, I) , the effect of the two-height vertical wind shear on the offshore turbine's power output appears weak. For all turbine cases, Hwangbo et al. [96] set the similarity threshold as $\varpi = 0.25$. Before the covariate matching, the number of observations in each annual dataset ranges from 14,000 to 37,000, and these numbers reduce to 1,400–2,300 after the matching. The matched dataset includes thousands of observations which is still large enough for estimating the performance benchmark function curve as well as the average performance curve.

Figs. 6.12 and 6.13 present, for inland turbine WT1 and offshore turbine

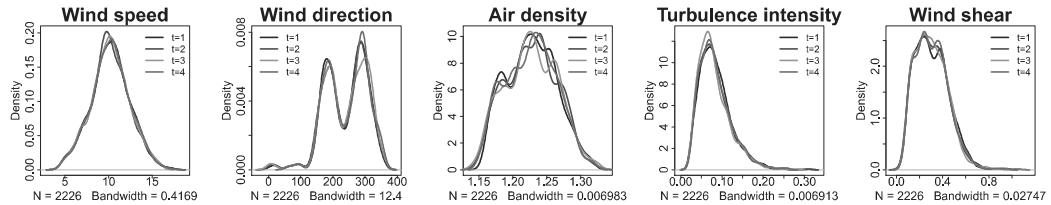


FIGURE 6.12 Probability density function plots of the matched covariates over the four comparison periods for inland turbine WT1. (Reprinted with permission from Hwangbo et al. [96].)

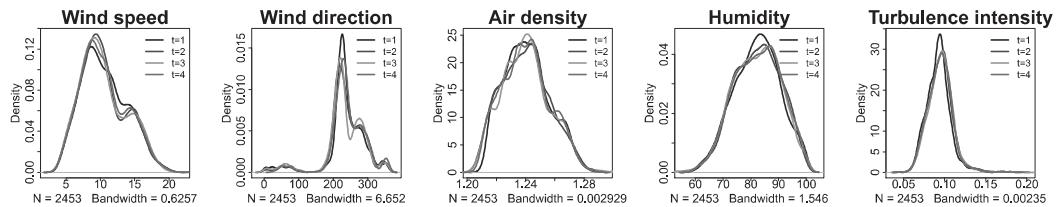


FIGURE 6.13 Probability density function plots of the matched covariates over the four comparison periods for offshore turbine WT3. (Reprinted with permission from Hwangbo et al. [96].)

WT3, respectively, the pdfs of each environmental variable across the four comparison periods after the covariate matching. The same plots for WT2 and WT4 are omitted because they convey similar messages. One can notice that the choice of $\varpi = 0.25$ leads to sufficiently good matching as demonstrated in the pdf plots.

Subsequently, Hwangbo et al. [96] use the matched subset of data to estimate the productive efficiency measure for each comparison period, as defined in Eq. 6.9. To add a confidence interval, 100 bootstrapping samples are randomly drawn from a respective original dataset, and for each resampled dataset, the efficiency metric is computed once. Doing this 100 times allows the calculation of the 90% confidence intervals for the productive efficiency metric. The bootstrap procedure can be performed on any other performance metrics as well; for more details about the bootstrap technique, please refer to [55].

Fig. 6.14 shows the PEs and its confidence intervals for the four comparison periods, which happen to be the first four years of a turbine's operation. Interestingly, one can notice that for all four turbines, their productive efficiency appears to have increased slightly, rather than deteriorated, during the early stage of operation. This pattern is more obvious for offshore turbines. This initial increase in efficiency was also recognized by Staffell and Green (2014) [203, Figure 9b]. Staffell and Green plot the fleet-level performance degradation of wind turbines over a twenty-year period using the fleet's load

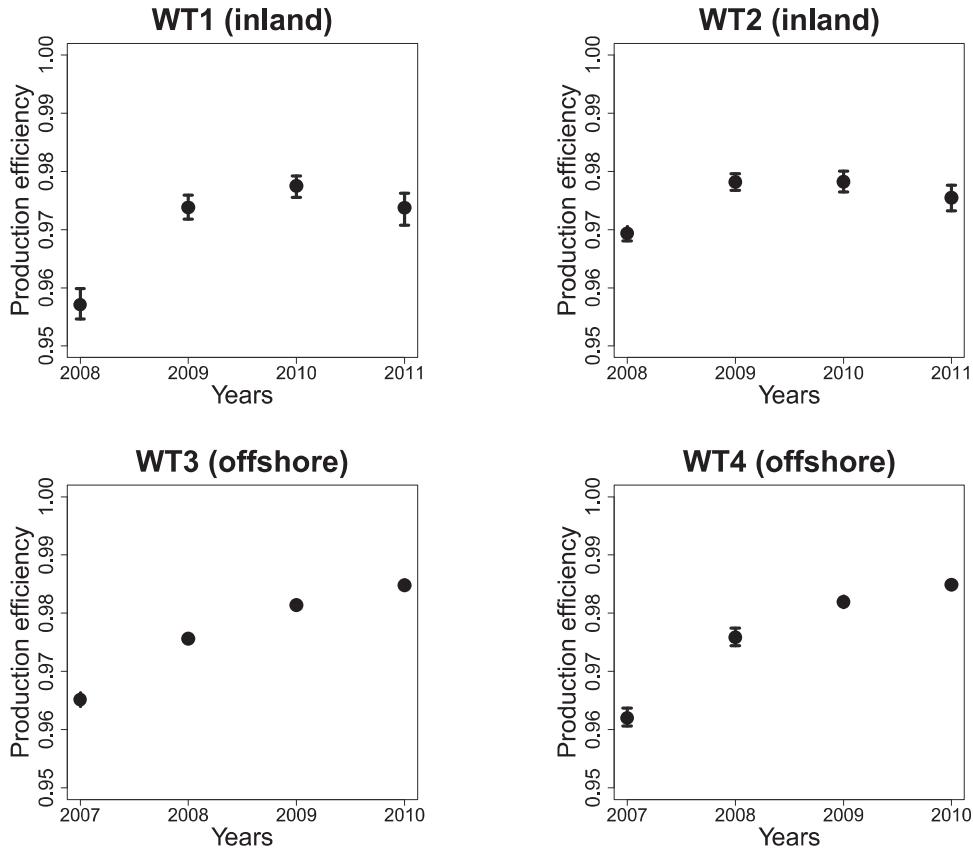


FIGURE 6.14 Productive efficiency, PE_t , $t = 1, 2, 3, 4$. The bars represent 90% confidence intervals and the dots denote the mean values of the efficiency. For offshore wind turbines, some of the confidence intervals are very narrow and are thus not visible. (Adapted with permission from Hwangbo et al. [96].)

factor as the performance measure. Staffell and Green's study appears to suggest an initial period of four to five years before any noticeable degradation is witnessed, as well as an increase in turbine performance for the first one and one-half years, which is rather consistent with what is observed here.

This increasing efficiency phenomenon, however slight, is perhaps counter-intuitive. Hwangbo et al. [96] theorize that this could be due to the rational behavior of the operator when faced with initial start-up risk. Recall the typical bathtub curve used in reliability engineering [90], in which there is a short "infant mortality" period at the beginning of a system's operation. In this period, the failure rate of a system appears to be higher than that in the subsequent stable operation period but the failure rate declines rapidly as the components in the system break in with each other. Flipping the bathtub curve upside down shows the reverse effect of failures, or rather, the effective

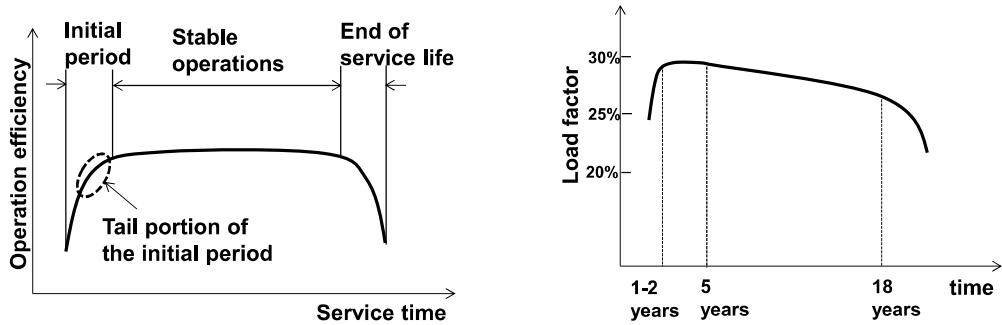


FIGURE 6.15 Flipped bathtub curve and its relevance to turbine reliability. Left panel: a flipped bathtub curve, right panel: load factor average curve implied by Figure 9b in [203].

production of a system. On the flipped curve, one expects to see an increase in production efficiency in the initial period; see Fig. 6.15.

Aware of the existence of this period of increased risk of failure, engineers and operators take proactive actions to reduce risk in their operational policies. One typical action is called “burn-in,” in which the manufacturers break in key components before the final system is put together and shipped off to the end users, so that a system installed at a user’s site can skip the rapid ascending phase of the initial period. But a system after burn-in can still experience the tail portion of the initial period, seeing a slight increase in efficiency.

Another common recommendation is for operators to ramp up the production of a system slowly in its initial operation period. Wind turbine operators may have operated the turbines following a similar ramping-up strategy. Consequently, the initial increase in production efficiency could be the combined effect of components breaking in and a strategic ramping up.

An interesting observation of Staffell and Green’s Figure 9b is that for a wind turbine fleet, there is a noticeable decline even during its stable operation period (until around 18 years in service). This differs apparently from the typical bathtub curve which has a flat stable operation period. Analysts indeed have been arguing [27] that wind turbines, subject to non-steady loads as a result of the uncontrollable nature of wind, degrade faster than a turbine machinery operating under relatively long periods of steady loads (such as the gas turbines in fossil fuel power plants). This tilted stable operation line may be a testimonial to this faster degradation argument.

Hwangbo et al. [96] compare their productive efficiency metric, PE, and power coefficient, C_p . Table 6.3 presents the power coefficient values and the PE values for the four turbines in the four periods. The values in the parentheses are the respective confidence intervals.

As mentioned before, the theoretical maximum of power coefficient is the Betz limit. One could divide a power coefficient by the Betz limit to get an

TABLE 6.3 Comparison between the productive efficiency and the (peak) power coefficient.

	Year 1	Year 2	Year 3	Year 4
Power coefficient C_p				
WT1	0.374 (0.369, 0.380)	0.390 (0.387, 0.394)	0.394 (0.390, 0.399)	0.393 (0.390, 0.398)
WT2	0.442 (0.436, 0.449)	0.463 (0.459, 0.468)	0.463 (0.458, 0.473)	0.461 (0.456, 0.468)
WT3	0.419 (0.417, 0.422)	0.464 (0.460, 0.470)	0.482 (0.479, 0.489)	0.506 (0.498, 0.515)
WT4	0.418 (0.415, 0.422)	0.467 (0.461, 0.474)	0.483 (0.477, 0.491)	0.504 (0.497, 0.511)
Productive efficiency PE				
WT1	0.957 (0.955, 0.960)	0.974 (0.972, 0.976)	0.978 (0.976, 0.979)	0.974 (0.971, 0.976)
WT2	0.969 (0.968, 0.971)	0.978 (0.977, 0.980)	0.978 (0.976, 0.980)	0.976 (0.973, 0.978)
WT3	0.965 (0.964, 0.966)	0.976 (0.975, 0.976)	0.981 (0.981, 0.982)	0.985 (0.984, 0.986)
WT4	0.962 (0.961, 0.964)	0.976 (0.974, 0.977)	0.982 (0.981, 0.983)	0.985 (0.984, 0.986)

interpretation of how much the turbine performs relative to its potential, similar to what the productive efficiency attempts to do. But the Betz limit is not practically achievable, so that power coefficients are generally below 0.45. When normalizing by the Betz limit, the resulting power coefficient-based efficiency measure never approaches one. Considering the yearly power coefficient in Table 6.3, ranging from 0.371 to 0.506, the relative power coefficient efficiency is therefore between 63% and 85%. These low percentages should not be interpreted as saying that power production of the wind turbines is inefficient. Looking at the PE values, the wind turbine operations are actually reasonably efficient, relative to their full potentials.

Using the power coefficient values, one also notices a general upward trend and a leveling off. This message appears to reinforce what is observed using the PE measure. In fact, there appears a fairly obvious positive correlation between the two metrics. Using all the average values in Table 6.3 yields a correlation of 0.75 between C_p and PE values. This positive correlation suggests that the productive efficiency metric measures a turbine's performance on a broad common ground with the power coefficient.

GLOSSARY

AEP: Annual energy production

CF: Capacity factor

CNLS: Convex nonparametric least squares

DEA: Data envelope analysis

IEC: International Electrotechnical Commission

pdf: Probability density distribution

PE: Productive efficiency

PGR: Power generation ratio

RUP: Regular ultra passum

SFA: Stochastic frontier analysis

s.t.: Such that

TS: Technical specifications

EXERCISES

- 6.1 Use the `Offshore Wind Farm Dataset2`, select one of the turbines, and conduct the following exercise.
 - a. Calculate the capacity factor for this wind turbine on the weekly time resolution, and let us call this capacity factor M4.
 - b. Plot the histogram and the empirically estimated density curve of M4. Compare M4 with the other three metrics in Fig. 6.4.
 - c. Replicate the results in Table 6.1 but now it is M4 versus M1, M2, and M3, respectively.
- 6.2 Use the `Inland Wind Farm Dataset2`, select one of the turbines, and replicate the analysis in Section 6.2 but for this inland turbine. Are the overall insights concerning the three performance metrics still valid?
- 6.3 Generate a plot like Fig. 6.14 but using the data of the (peak) power coefficient in Table 6.3.
- 6.4 The power coefficient computed in Section 6.2 is the peak power coefficient, i.e., that the largest value on a power coefficient curve is used to represent the whole curve. Let us compute the average power coefficient instead, i.e., the average of all values on the power coefficient curve. Then, use the average power coefficient to re-do the analysis in Tables 6.1 and 6.2 and Figs. 6.4–6.6. Is the average power coefficient closer to the other two performance metrics than the peak power coefficient?

- 6.5 Khalfallah and Koliub [121] study the effect of dust accumulation on turbine blades on power production performance of the affected turbine. They reckon that the power production, in the presence of dust accumulation, deteriorates more significantly for wind speed higher than 9 m/s than the lower wind speeds. They estimate the annual loss is around 3%.
- Take the WT1's 2008 data and modify it by decreasing the power output value by 3% for those power outputs corresponding to wind speed of 9 m/s or higher. Treat this as the 2009 data. Then, reduce the 2008 power data by 6% and 9% and use them as the substitute of 2010 and 2011 power data, respectively. This gives us a set of simulated wind-power data, mimicking the dust accumulation effect over four years. This data simulation is first suggested by Hwangbo et al. [96].
 - Compute the (peak) power coefficient for the four years on an annual basis.
 - Use 100 bootstrap samples to compute the 90% confidence interval for each of the point estimates of power coefficient obtained in (b).
 - Plot the point estimates and the associated 90% confidence intervals in the fashion similar to Fig. 6.14. What do you observe? Does it tell you certain limitation of using the power coefficient as a performance metric?
- 6.6 Find some other real-life examples, if possible, supporting or illustrating the flipped bathtub curve. Do any of your examples have an accelerated deterioration even during its supposedly stable operation period, like the curve in the right panel of Fig. 6.15?
- 6.7 The details of the covariate matching procedure can be found in [96, Section 3] or in Chapter 7 of this book. Please read the material and understand how it works. Choose different ϖ values and see how it affects the matching quality and the resulting data amount. Choices of ϖ can be 0.1, 0.15, 0.2, 0.3 or 0.5.
- 6.8 Use the 10-min resolution data in the **Wind Time Series Dataset** and split the data into 80% for training and 20% for testing. Conduct the following exercise.
- Use the training data to construct a V -versus- y power curve using the IEC binning method.
 - Use the same training data to construct a shape-constrained power curve using the method outlined in Section 6.3.2.

- c. Use the test data to perform an out-of-sample test on the two power curves. Please compare the two power curve estimates in terms of both RMSE and MAE.

Quantification of Turbine Upgrade

Turbine performance assessment, as discussed in Chapter 6, plays an important role in wind turbine maintenance, equipment procurement and wind energy planning. Over time, a wind turbine naturally degrades, losing efficiency in power generation. To maintain the production efficiency of a wind turbine, the owners or operators sometimes perform a retrofit to an existing wind turbine, also known as an upgrade, in the hope to restore or enhance the production efficiency of the existing asset. But upgrading can be costly. Owners or operators of wind farms understandably wonder whether the performance of a wind turbine is improved enough to justify the cost of upgrading. This chapter presents several data science methods aiming at addressing this issue.

7.1 PASSIVE DEVICE INSTALLATION UPGRADE

Power output from a wind turbine is driven by wind input. It therefore makes little sense to compare, without controlling for the input conditions, the difference in wind power production before and after an upgrade. The output-only difference, even if present, would not reveal whether the difference comes from upgrading the turbine or from the occurrence of a strong wind after the upgrade.

The output-only comparison could be effective for some of the upgrades that change the control logic without necessarily installing or adjusting physical devices on a turbine. For these cases, analysts suggest switching back and forth between two operational options in 30-second intervals and recording the power production under each option, respectively. Conducting such test for a long enough duration and under a broad variety of environmental conditions, and comparing the power outputs under respective operational options sheds light on which option leads to better power production. The assumption here

is that the environmental conditions as well as the turbine's own conditions, besides the operational option under test, are unlikely different in a duration as short as 30 seconds apart. The difference between the power output, if existing, must thus be attributed to the difference in the operational options.

Not all the upgrades can be tested in the aforementioned manner. Many upgrades involve installing a passive device to the existing turbine or adjusting existing turbine components physically. One such upgrade is the vortex generator (VG) installation. The wind industry has long been aware of the VG technology and the potential benefit that VG installation may bring to wind power production, as past studies [151, 157, 224] claim that having VGs could improve the lift characteristics of the blades. Installing vortex generators requires retrofitting turbine blades, incurs material and labor costs, and halts energy production during installation. Once installed, owners and operators would rather not take them off, as doing so incurs even more costs. Turbine upgrades like VG installation are definitely not candidates for conducting the aforementioned 30-second operational switching test.

Although the precise magnitude of the benefit from VG installation is unknown, the general feeling in the industry is that it would be moderate, likely resulting in 1–5% extra wind energy production under the same wind and environmental conditions. Detecting this moderate improvement in the turbine operational data, with the presence of large amounts of noise, is not a trivial task. The IEC binning method for power curve modeling, as explained in Section 5.1, is probably the most widely used approach in the wind industry for estimating and quantifying a turbine's performance before and after VG installation. The IEC standard method is, however, ineffective in this endeavor, which has been noticed by industrial practitioners and documented in previous studies [50, 133]. The IEC admits that "*Depending on site conditions and climate, the uncertainty may amount to several percent*" [102].

A second difficulty in quantifying the benefit of a turbine upgrade lies in the lack of a good method to validate the estimated effect. In order to validate the estimated VG effect, one ought to know the ground truth of the actual effect. For that purpose, one would ideally conduct a controlled experiment, in which all environmental conditions are set the same before and after VGs are installed, so that the difference in power outputs before and after the installation signifies the VG effect. The problem is that such a controlled experiment is impractical and will probably never be feasible, considering the sheer physical size of commercial wind turbine generators. Analysts could conduct small-scale experiments in a wind tunnel, but the amount of uncertainty encountered in the extrapolation of the small-scale wind tunnel test to commercial operations makes such results much less credible to use.

We want to caution readers that our purpose here is not to advocate a specific type of turbine upgrade or retrofit option but to present some options that may better serve the purpose of upgrade quantification. When a quantification method, for example, the IEC binning method, indicates that there

is no difference in a turbine's power output before and after an upgrade, a question remains: Is there really no benefit to have that type of upgrade for this particular turbine, or is it possible that the method used is incapable of detecting small to moderate changes due to the method's inherent limitations?

7.2 COVARIATE MATCHING-BASED APPROACH

The IEC method's ineffectiveness is rooted in its lack of control of the influence of multiple environmental factors other than wind speed. Shin et al. [198] present a covariate matching approach to select a subset of data from datasets before and after an action of upgrade and to ensure the environmental covariates of the selected subset to have comparable distribution profiles. Recall from Chapter 5 that \mathbf{x} denotes the vector of environmental variables, including wind speed, and the variables in \mathbf{x} are called *covariates*. Once the covariates are matched, Shin et al. then quantify the benefit of the upgrade by taking the difference of power outputs under the matched environmental condition and apply a paired *t*-test for testing the significance of the upgrade effect.

Covariate matching methods are rooted in the statistical literature [206]. In stabilizing the non-experimental discrepancy between non-treated and treated subjects of observational data, Rubin [185] adjusts covariate distributions by selecting non-treated subjects that have a similar covariate condition as that of treated ones. Through the process of matching, *non-treated* and *treated* groups become only randomly different on all background covariates, as if these covariates were designed by experimenters. As a result, the outcomes of the matched non-treated and treated groups, which keep the originally observed values, are comparable under the matched covariate conditions.

Fig. 7.1 demonstrates the discrepancy of the covariate distributions of the un-matched or non-treated data in the Turbine Upgrade Dataset. It presents for each covariate the difference between the pre-upgrade and post-upgrade periods using the empirically estimated density functions. The last subplot in both the upper and lower panels is the density function of the power output of the respective control turbine. For the control turbine, as it is not modified, the distribution of its power output is supposed to be comparable, should the environmental conditions be maintained the same. But the data show otherwise, signifying the impact of the confounding environmental influence.

7.2.1 Hierarchical Subgrouping

In the context of wind turbine upgrade, the data records collected before the upgrade form the non-treated data group, whereas those collected after the upgrade form the treated group. Let Q_{bef} and Q_{aft} be the index set of the data records in the non-treated and treated group, respectively. Let x_Q denote the values of a covariate x for data indices in Q . For example, $V_{Q_{\text{bef}}}$ is the vector of all wind speed values that are observed before the upgrade.

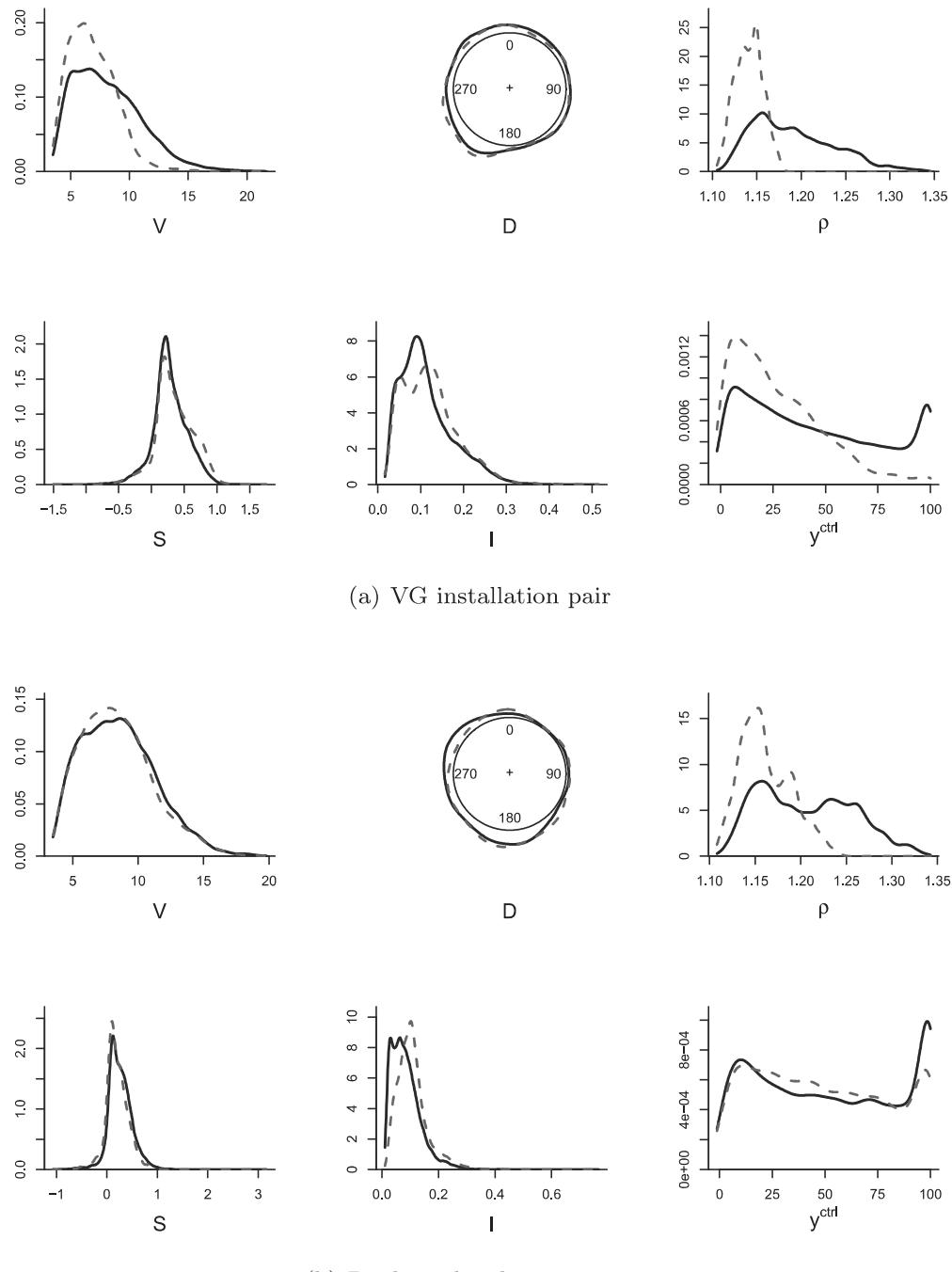


FIGURE 7.1 Overlaid density functions of the unmatched covariates and the density functions of control turbine power output. Solid line = before upgrade (non-treated) and dashed line = after upgrade (treated). (Reprinted with permission from Shin et al. [198].)

The first action of preprocessing is to narrow down the dataset and create a subset on which one subsequently performs the data records matching. The reason for this preprocessing is to alleviate a computational demand arising from too many pairwise combinations when comparing two large size datasets. This goal is fulfilled through a procedure labeled as *hierarchical subgrouping*. The idea is summarized in Algorithm 7.1.

Algorithm 7.1 Hierarchical subgrouping procedure to match covariates.

1. Locate a data record in the treated group, Q_{aft} , and label it by the index j .
 2. Select one of the covariates, for instance, wind speed, V , and designate it as the variable of which the similarity between two data records is computed.
 3. Go through the data records in the non-treated group, Q_{bef} , by selecting the subset of data records such that the difference, in terms of the designated covariate, between the data record j in Q_{aft} and any one of the records in Q_{bef} is smaller than a pre-specified threshold, ϖ . When V is in fact the one designated in Step 2, the resulting subset is then labeled by placing V as the subscript to Q , namely Q_V .
 4. Next, designate another covariate and use it to prune Q_V in the same way as one prunes Q_{bef} into Q_V in Step 3. Doing so produces a smaller subset nested within Q_V . Then continue with another covariate until all covariates are used.
-

The order of the covariates in the above hierarchical subgrouping procedure is based on the importance of them in affecting wind power outputs. According to the analysis in Chapter 5, V , D , and ρ are more important, whereas S and I are less so. Shin et al. [198] use the order, V , D , ρ , S , and I . While it is generally a good idea to follow the physically meaningful order when conducting the hierarchical subgrouping, Shin et al. present a numerical analysis based on the reserve order, i.e., I , S , ρ , D , and V , and find that using the reserve order hardly affects the downstream analysis outcome, suggesting a certain degree of robustness in the overall analysis procedure. The robustness appears to be a result of the checks and diagnostics included in the later stage of procedure and to be explained in the sequel. If a priority order of the covariates is unknown, it is recommended to perform some statistical analysis using, for example, random forests [21], which can measure the importance of covariates before applying the matching method.

Note that wind direction, D , is a circular variable, so that an absolute difference between two angular degrees is between 0 and 180° . The circular variable formula, $180^\circ - |180^\circ - (|D_i - D_j|)|$, is used to calculate the difference between two D values.

The above process can also be written in set representation. For a data record j in Q_{aft} , define subsets of data records in Q_{bef} , hierarchically chosen, as

$$\begin{aligned} Q_V &:= \{i \in Q_{\text{bef}} : |V_i - V_j| < \varpi \cdot \sigma(V_{Q_{\text{bef}}})\}, \\ Q_D &:= \{i \in Q_V : 180^\circ - |180^\circ - (D_i - D_j)| < \varpi \cdot \sigma(D_{Q_V})\}, \\ Q_\rho &:= \{i \in Q_D : |\rho_i - \rho_j| < \varpi \cdot \sigma(\rho_{Q_D})\}, \\ Q_S &:= \{i \in Q_\rho : |S_i - S_j| < \varpi \cdot \sigma(S_{Q_\rho})\}, \\ Q_I &:= \{i \in Q_S : |I_i - I_j| < \varpi \cdot \sigma(I_{Q_S})\}, \end{aligned} \quad (7.1)$$

where $\sigma(x)$ is the standard deviation of x in the specified dataset. The thresholding coefficient, ϖ , can be different at each layer but to make thing simple, analysts usually select a single constant threshold for the whole procedure. This hierarchical subgrouping establishes the subsets nested as such: $Q_I \subset Q_S \subset Q_\rho \subset Q_D \subset Q_V \subset Q_{\text{bef}}$. Consequently, the data records in the last hierarchical set, Q_I , have the closest environmental conditions as compared with the data record j in Q_{aft} .

There could be other conditions, in addition to the five variables mentioned above, which may affect wind power production while not measured. The possible existence of unmeasured factors presents the risk of causing a distortion in comparison, even when the aforementioned measured environmental factors are matched between the treated and non-treated groups. In order to alleviate this risk, Shin et al. [198] make use of the power output, y^{ctrl} , of the control turbine in each turbine pair. The idea is to further narrow down from the most nested subset, produced above, by taking the following action—select records from Q_I whose y^{ctrl} values are comparable to the y^{ctrl} value of a data record j in Q_{aft} . Doing this actually amounts to continuing the hierarchical subgrouping action to produce a Q_y , a subset of Q_I , based on y^{ctrl} , such that

$$Q_y := \{i \in Q_I : |y_i^{\text{ctrl}} - y_j^{\text{ctrl}}| < \varpi \sigma(y_{Q_I}^{\text{ctrl}})\}. \quad (7.2)$$

Shin et al. [198] perform this procedure for all data records in the treated group so that each record j in Q_{aft} has its matched set, which is denoted by $Q_{y,j}$. The set of $Q_{y,j}$ is in fact the set of data records in the non-treated group matched to the data record j in the treated group. In the case that $Q_{y,j}$ is an empty set, one should then discard the respective index j from Q_{aft} . Because of this, Q_{aft} may shrink after the subgrouping steps.

7.2.2 One-to-One Matching

The next action is to choose a data record in $Q_{y,j}$ that is the closest to the data record j . For this purpose, analysts need to define a dissimilarity measure to quantify the closeness between two data records. Shin et al. [198] decided to use the Mahalanobis distance [140] as the dissimilarity measure,

which is popularly used in the context of multivariate analysis. A Mahalanobis distance re-weights the Euclidean distance between two covariate vectors with the reciprocal of a variance-covariance matrix. Before presenting the definition of the Mahalanobis distance between two wind turbine data records, Shin et al. first transform \mathbf{x} into \mathbf{x}^* , such that

$$\mathbf{x}^* := (V \cos D, V \sin D, \rho, S, I)^T.$$

Using \mathbf{x}^* makes it easier to deal with the circular wind direction variable, D . The Mahalanobis distance, MD_{ij} , between a data record j in Q_{aft} and a data record i in $Q_{y,j}$, is defined as

$$\text{MD}_{ij} := \sqrt{(\mathbf{x}_i^* - \mathbf{x}_j^*)^T \boldsymbol{\Sigma}^{-1} (\mathbf{x}_i^* - \mathbf{x}_j^*)}, \quad (7.3)$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of \mathbf{x}^* . Obviously, the larger an MD value, the less similar the two data records.

With the Mahalanobis distance defined, one can simply select the data record i_j in $Q_{y,j}$ that has the smallest Mahalanobis distance as the best match to data record j in Q_{aft} . In other words, i_j is found such that

$$i_j = \arg \min_{i \in Q_{y,j}} \text{MD}_{ij},$$

for each j in Q_{aft} . In case two or more are tied for the smallest value, Shin et al. [198] choose one of them randomly.

After this step, each data record j in the treated group is matched to a non-treated counterpart i_j , with the exception of those already discarded during the subgrouping step. Let us define the index set of the matched data records from the non-treated group as

$$Q_{\text{bef}}^* := \{i_j \in Q_{\text{bef}} \mid j \in Q_{\text{aft}}\}.$$

It should be noted that Shin et al. [198] allow replacement in the matching procedure. In other words, i_j is not eliminated from the candidate set Q_{bef} , even though it has matched to j once. When the next data record $j+1$ is selected from Q_{aft} , the same non-treated data i is possible to be matched again.

7.2.3 Diagnostics

After performing the matching procedure, it is important to diagnose how much of the discrepancy in the covariate distributions has been removed. Only after the diagnostics signifies a sufficient improvement, can an outcome analysis be performed in the next step.

Shin et al. [198] measure the discrepancy of distributions in two ways, numerically and graphically. For the numerical diagnostics, the standardized

TABLE 7.1 SDM values serve as the numerical diagnostics for covariate matching.

VG installation pair						
	V	D	ρ	S	I	y^{ctrl}
Unmatched	0.6685	0.0803	3.2715	0.2312	0.1382	0.8132
Matched	0.0142	0.0026	0.0589	0.0721	0.0003	0.0083
Pitch angle adjustment pair						
	V	D	ρ	S	I	y^{ctrl}
Unmatched	0.0605	0.1647	1.6060	0.2759	0.4141	0.0798
Matched	0.0077	0.0029	0.0263	0.0158	0.0111	0.0036

Source: Shin et al. [198]. With permission.

difference of means (SDM) is used as a measure of dissimilarity of a covariate between the treated and non-treated groups, i.e.,

$$\text{SDM} := \frac{\bar{x}_{Q_{\text{aft}}} - \bar{x}_{Q_{\text{bef}}}}{\sigma(x_{Q_{\text{aft}}})}, \quad (7.4)$$

where x is one of the covariates and \bar{x}_Q denotes the average of x in the set of Q . The SDM decreases if the matching procedure indeed reduces the discrepancy between the two groups. As shown in Table 7.1, SDM decreases significantly for all covariates after matching. A previous study [186] suggests that SDM should be less than 0.25 to render the two distributions in question comparable.

The graphical diagnostics uses the pdf plots just like in Figs. 6.12 and 6.13, in which the empirical density functions before and after the upgrade are overlaid on top of each other. Then, a visual inspection is conducted to check and verify that the two respective density functions are similar enough. Fig. 7.2 presents the well-matched distributions of covariates after the matching process. The improvements in terms of distribution similarity are apparent when compared to Fig. 7.1.

It should also be noted that, if the size of Q_{aft} after the matching loses too many data records, and this can happen when too small ϖ 's are applied, Shin et al. [198] suggest to enlarge the size of the original Q_{aft} prior to the matching process, in order to secure a sufficient amount of representative weather conditions in the matched Q_{aft} . Enlarging Q_{aft} can be done by extending the post-upgrade data collection period, for instance.

7.2.4 Paired t -tests and Upgrade Quantification

The matching procedure produces a set of paired data records of the two groups, each pair denoted by (i_j, j) , where $i_j \in Q_{\text{bef}}^*$ and $j \in Q_{\text{aft}}$. Using these paired indices, Shin et al. [198] retrieve the paired power outputs for the test turbine, i.e., $(y_{i_j}^{\text{test}}, y_j^{\text{test}})$. The power output pair can be interpreted

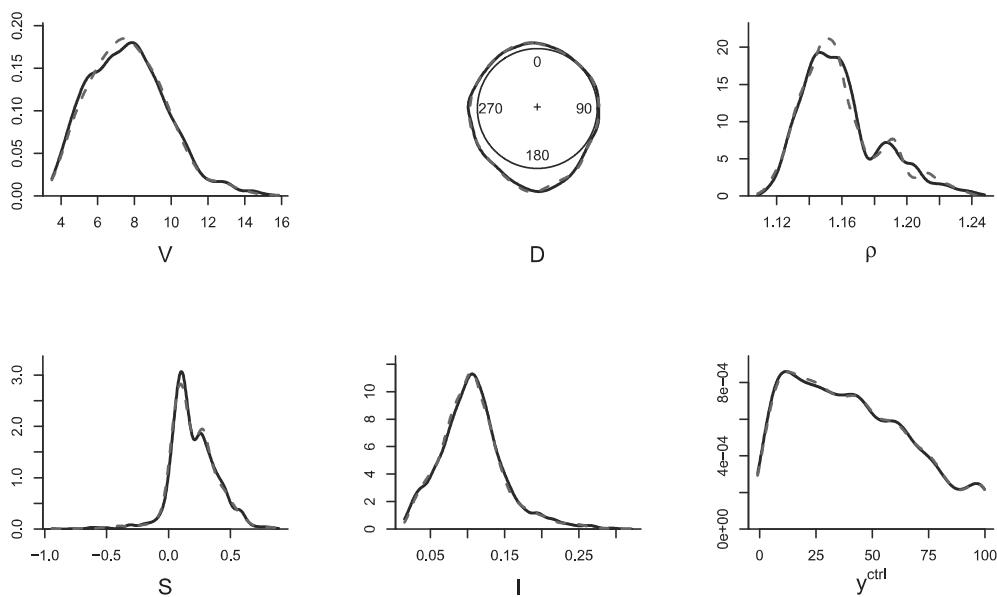
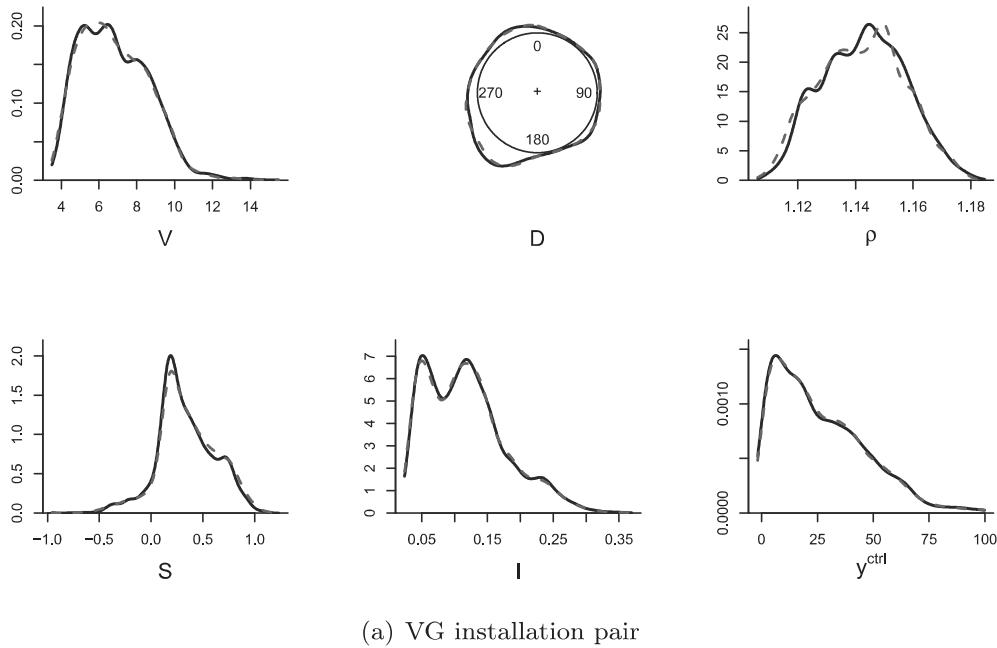


FIGURE 7.2 Overlaid density functions of the matched covariates and the density functions of the matched control turbine power output. Solid line = before upgrade (non-treated) and dashed line = after upgrade (treated). (Reprinted with permission from Shin et al. [198].)

TABLE 7.2 The results of paired t -tests and upgrade quantification.

VG installation pair			Pitch angle adjustment pair		
t -stat	p-value	UPG	t -stat	p-value	UPG
3.015	0.003	1.13%	7.447	< 0.0001	3.16%

Source: Shin et al. [198]. With permission.

as repeated measurements under comparable environmental conditions, thus making the power outputs also comparable. One can apply a t -test to analyze the difference of the paired power outputs, computed as $\delta_j = y_j^{\text{test}} - y_{ij}^{\text{test}}$. The null hypothesis is that the expected mean of the difference is zero, that is, $H_0 : \mathbb{E}(\delta) = 0$, where $\bar{\delta}$ is the sample mean of $\{\delta_j : j \in Q_{\text{aft}}\}$. Accordingly, the test statistic is

$$t\text{-stat} := \frac{\bar{\delta}}{s_\delta / \sqrt{n_\delta}}, \quad (7.5)$$

where s_δ and n_δ are the sample standard deviation and the sample size of $\{\delta_j : j \in Q_{\text{aft}}\}$, respectively. If the test concludes a significant positive mean difference, the upgrade on the test turbine is deemed effective. Table 7.2 presents the results from the paired t -test of both datasets, which show a significant upgrade effect at the 0.05 level.

Shin et al. [198] quantify the upgrade effect, denoted by UPG, by computing

$$\text{UPG} := \frac{\sum_{j \in Q_{\text{aft}}} (y_j^{\text{test}} - y_{ij}^{\text{test}})}{\sum_{j \in Q_{\text{aft}}} y_{ij}^{\text{test}}} \times 100. \quad (7.6)$$

The quantification results are shown in Table 7.2 as well. Recall that in the pitch angle adjustment pair, the test turbine's power is increased by 5% for wind speeds of 9 m/s and above, which translates to a 3.11% increase for the whole wind spectrum. The quantification outcome shows an improvement of 3.16% overall, which appears to present a fair agreement with the simulated amount.

7.2.5 Sensitivity Analysis

The pitch angle adjustment pair is analyzed for the purpose of getting a sense of how well a proposed method can estimate a power production change. Recall that the upgrade in the pitch angle adjustment pair is simulated, so we know the true upgrade amount and can use that as a reference for comparison. In Section 7.2.4, however, only a single simulated improvement value (5%) is used. To have a fuller sense, this section conducts the matching-based quantification on various degrees of the simulated improvement.

Denote by r the nominal power increase rate. Because the nominal power increase rate is applied only to the partial range of wind power corresponding to wind speed higher than 9 m/s, the effective power increase rate for the

TABLE 7.3 Sensitivity analysis of covariate matching-based turbine upgrade quantification.

r	2%	3%	4%	5%	6%	7%	8%	9%
r'	1.25%	1.87%	2.49%	3.11%	3.74%	4.36%	4.98%	5.60%
UPG	1.74%	2.21%	2.68%	3.16%	3.63%	4.11%	4.58%	5.05%
UPG/ r'	1.39	1.18	1.08	1.02	0.97	0.94	0.92	0.90

Source: Shin et al. [198]. With permission.

whole wind spectrum, denoted by r' , is different. When it comes to verifying the upgrade effect for the simulated case, the effective power increase rate r' is computed through

$$r' := \frac{\sum_{j \in Q_{\text{aft}}} y_j^{\text{test}} \{1 + r \cdot \mathbb{1}(V_j^{\text{test}} > 9)\} - \sum_{j \in Q_{\text{aft}}} y_j^{\text{test}}}{\sum_{j \in Q_{\text{aft}}} y_j^{\text{test}}}. \quad (7.7)$$

Shown in Table 7.3, as r changes from 2% to 9%, r' changes from 1.25% to 5.6%. This range of power increases is practical for the detection purpose. If an improvement is smaller than 1%, it is going to be considerably hard to detect. On the other hand, when an improvement is greater than 6%, it is possible that even the IEC binning method can detect that level of change.

Table 7.3 presents the UPGs corresponding to the respective r' . One can observe that UPG noticeably overestimates r' when r' is small (smaller than 2%)—the overestimation is as much as 40% for the smallest change in the table. But the quantification quality using UPG gets stabilized as r' increases. For the last six cases in Table 7.3, the differences between UPG and r' are within 10%.

7.3 POWER CURVE-BASED APPROACH

The multi-dimensional power curve methods, explained in Chapter 5, can account for the influence of environmental variables on power output. It is thus not surprising that upgrade quantification approaches are developed based on power curve models.

The basic idea is as follows. Once a power curve model is established using the pre-upgrade data, it captures the power production characteristics of the old turbine before the upgrade. Feeding the post-upgrade wind and environmental data to the power curve model is analogous to running the old, unmodified turbine under the new conditions. Comparing the model outputs with the actual physical outputs under the same input conditions is supposed to reveal the difference that an upgrade makes.

Using the IEC binning method to quantify the benefit of an upgrade is in fact a power curve-based approach. The drawback of that specific approach lies in the fact that IEC binning controls for practically only the wind speed effect, which accounts for roughly 85% of the variation in the power data.

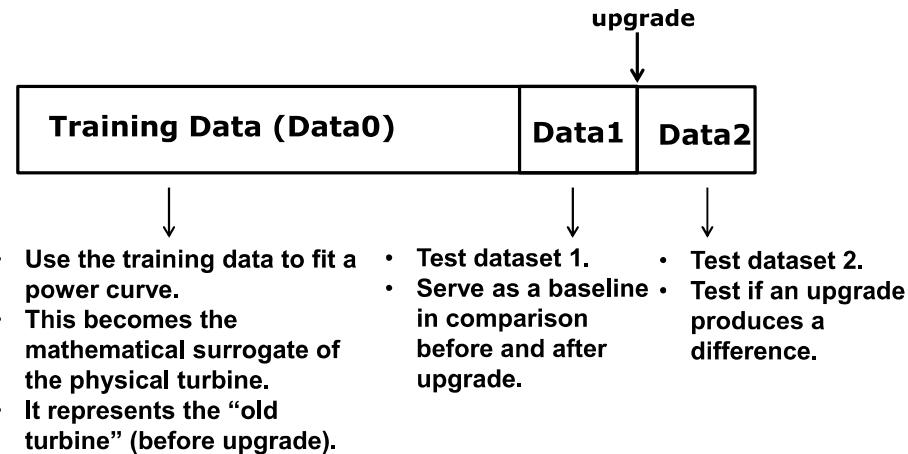


FIGURE 7.3 Dataset partition used in Kernel Plus for quantifying a turbine upgrade. (Reprinted with permission from Lee et al. [133].)

The remaining unaccounted variation is still too much relative to the typical upgrade effect, and without accounting for that, the resulting method is rendered ineffective, which is what happens to the IEC binning-based quantification method [50].

Lee et al. [133] present a turbine upgrade quantification method based on the additive-multiplicative kernel power curve model, introduced in Section 5.2.3. The resulting method is nicknamed *Kernel Plus*.

7.3.1 The Kernel Plus Method

The central element in Kernel Plus is the AMK model in Eq. 5.16. In the context of upgrade quantification, only the mean prediction equation, $\hat{y}(\mathbf{x})$, is used. To make it explicit that the power curve model is from AMK, let us denote it with an AMK superscript, namely $\hat{y}^{\text{AMK}}(\mathbf{x})$.

In its procedure to quantify a turbine upgrade, Kernel Plus involves three datasets, as illustrated in Fig. 7.3. The three datasets include a training dataset of historical observations of (\mathbf{x}, y) pairs and two test datasets before and after an upgrade. The training dataset is referred to as “Data0” and is used to fit the power curve model. Data0 should be collected from a reasonable duration of a turbine’s operation, for instance, one year, such that the seasonal weather effects are well represented in the data. The two test sets, referred to as “Data1” and “Data2,” respectively, are collected for the same length of duration before and after the upgrade. They are used to detect and quantify the upgrade. Their corresponding data duration can be much shorter than that of Data0; a few weeks to a few months may be sufficient.

In the Kernel Plus method, Lee et al. [133] introduce a self-calibration procedure to alleviate the bias associated with a Nadaraya-Watson kernel es-

timator when it is used on a new dataset. The existence of bias based on finite samples is a common problem in statistical prediction. Hastie et al. present a full illustration of biases and variances involved in statistical prediction [86, Figure 7.2].

The self-calibration procedure is done by using subsets of the training data in Data0. To select a calibration set of data that has similar weather conditions to those in Data1 and Data2, Lee et al. [133] define a distance measure, which is in spirit similar to the Mahalanobis distance—recall that the Mahalanobis distance is used in the covariate matching procedure in Section 7.2.2. Like a Mahalanobis distance, the distance measure in the self-calibration procedure is a weighted distance but unlike a Mahalanobis distance, it is not weighted by the reciprocal of the corresponding variance-covariance matrix. Instead, the weighting matrix is a diagonal matrix whose diagonal elements are from the bandwidth vector $\boldsymbol{\lambda}$. Let us denote this diagonal matrix by $\boldsymbol{\Lambda}$, such that $\boldsymbol{\Lambda}_{i,i} = \lambda_i$ and $\boldsymbol{\Lambda}_{i,j} = 0 \quad \forall i \neq j$. The resulting distance measure between a training data point, $\mathbf{x}_i \in \text{Data0}$, and a test data point, \mathbf{x}_j , in either Data1 or Data2, is

$$D(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \boldsymbol{\Lambda}^{-1} (\mathbf{x}_i - \mathbf{x}_j)}. \quad (7.8)$$

Lee et al. [133] elaborate that the reason to choose this distance measure is because a simple Euclidean distance does not reflect well the similarity between the \mathbf{x} 's, as different elements in \mathbf{x} have different physical units, leading to different value ranges. To define a sensible similarity measure, a key issue is to weigh different elements in \mathbf{x} consistently with their relative importance pertinent to the power output. The original Mahalanobis distance does not serve this purpose because the squared distance associated with an input variable is weighted by the inverse of its variance. In a power curve model, wind speed is arguably the most important variable, yet it has a large variance. Because of this large variance, using the Mahalanobis distance will in fact diminish the importance of wind speed relative to other variables that have a smaller variance. The choice in Eq. 7.8 that uses the kernel bandwidth parameters as the weighting coefficients in $\boldsymbol{\Lambda}$ is consistent with the goal of weighting each element according to its relative importance, because the bandwidth parameters are selected based on how sensitive the power output is to a unit change in the corresponding input variable. If an input variable has a small bandwidth, it means that the power output could produce an appreciable difference with a small change in the corresponding input, suggesting that this variable is relatively important. On the other hand, a large bandwidth indicates a less important input variable.

For any test data point \mathbf{x}_j , one can choose a calibration data point, $\mathbf{x}_i^{\text{cal}}$, from Data0, which has the minimum $D(\mathbf{x}_i^{\text{cal}}, \mathbf{x}_j)$. The calibration procedure proceeds as described in Algorithm 7.2.

Algorithm 7.2 Self-calibration procedure in Kernel Plus.

1. For $\mathbf{x}_i^{\text{cal}} \in \text{Data0}$, compute $\hat{y}^{\text{AMK}}(\mathbf{x}_i^{\text{cal}})$.
2. Compute the calibration value $\mathfrak{R}^{\text{cal}}(\mathbf{x}_j) = y(\mathbf{x}_j) - \hat{y}^{\text{AMK}}(\mathbf{x}_j)$, where \mathbf{x}_j is paired to the calibration data point $\mathbf{x}_i^{\text{cal}}$.
3. For any test data point \mathbf{x}_j , the final, calibrated power estimate from the Kernel Plus method is $\hat{y}^{\text{KP}}(\mathbf{x}_j) = \hat{y}^{\text{AMK}}(\mathbf{x}_j) + \mathfrak{R}^{\text{cal}}(\mathbf{x}_j)$.

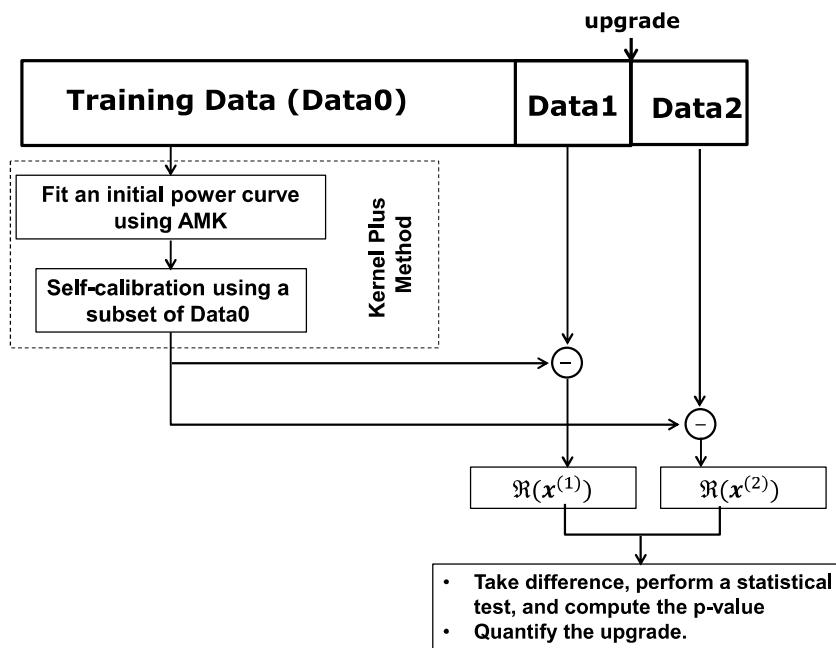


FIGURE 7.4 Diagram of quantifying a turbine upgrade using the Kernel Plus method. (Reprinted with permission from Lee et al. [133].)