4

Machine Learning: Fundamentals

Myeongsu Kang[1] *and Noel Jordan Jameson*[2]

[1] *University of Maryland, Center for Advanced Life Cycle Engineering, College Park, MD, USA*
[2] *National Institute of Standards and Technology, Gaithersburg, MD, USA*

To achieve competitive advantages in the global market, prognostics and health management (PHM) has emerged as an essential approach to improving product reliability, maintainability, safety, and affordability [1]. PHM facilitates maintenance decision-making and provides usage feedback for the product design and validation process. Electronic component and product manufacturers need new ways to gain insights from the massive volume of data recently streaming in from their systems and sensors, and this can be accomplished by using machine learning (ML), which is a set of techniques that make it possible to extract useful information from data, to accelerate the development of data-driven anomaly detection, diagnosis, and prognosis methods. Accordingly, this chapter first aims to provide the fundamentals of ML.

4.1 Types of Machine Learning

Samuel [2] defined ML as the field of study that gives computers the ability to learn without being explicitly programmed. Later, a more engineering-oriented definition of ML was provided: a computer problem is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E. For example, a diagnostic system is an ML program that can learn to identify a product's failures from a training dataset involving examples of failures. Each training example is also called a "training instance" (or observation or sample). In this case, the task T is to identify failures of the product, the experience E is the training data, and the performance measure P needs to be defined. This performance measure is called accuracy and is often used in classification tasks.

A diagnostic system based on ML techniques automatically learns which features[1] are good predictors of product failure by detecting failure patterns in the training dataset. Figure 4.1 illustrates a high-level overview of the ML approach to diagnosis.

ML algorithms can be classified in broad categories based on whether they are trained with human supervision (supervised, unsupervised, semi-supervised, and

1 In ML, a feature has several meanings depending on the context, but generally means an attribute plus its value.

Prognostics and Health Management of Electronics: Fundamentals, Machine Learning, and the Internet of Things, First Edition. Edited by Michael G. Pecht and Myeongsu Kang.

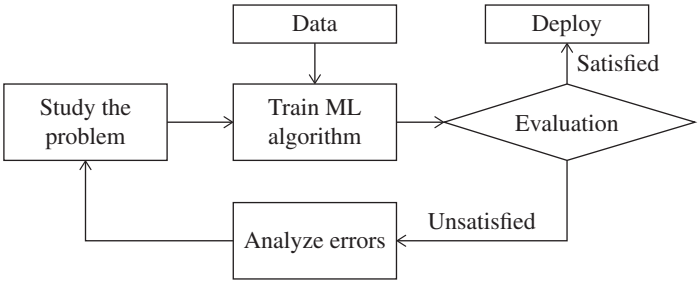


Figure 4.1 A high-level overview of the ML approach to diagnosis.

reinforcement learning); whether they can learn incrementally on the fly (online versus batch learning); and whether they work by simply comparing new data points to known data points, or instead detect patterns in the training data and build a predictive model (instance-based versus model-based learning). The following subsections will explain each category of ML algorithms.

4.1.1 Supervised, Unsupervised, Semi-Supervised, and Reinforcement Learning

ML algorithms can be divided into the following four categories depending on the amount and type of supervision they need while training: supervised, unsupervised, semi-supervised, and reinforcement learning.

In supervised learning, the training data fed to the ML algorithms includes the desired solutions, called labels, as shown in Figure 4.2. Classification is a typical supervised learning task. The diagnostic system is a good example of classification: it is trained with many variables or features along with their class (e.g. faulty or healthy),[2] and it must learn how to classify new variables or features.

Another typical task is to predict a target numeric value, such as a product's remaining useful life (RUL), given a set of features called predictors. This sort of task is called regression (see Figure 4.3). To train the ML algorithms, the training dataset must contain predictors with associated labels. Note that some regression algorithms can be used for classification, and vice versa. For example, logistic regression [3] is commonly used for classification, as it can output a value that corresponds to the probability of belonging to a given class (e.g. 90% chance of being a healthy product). The supervised ML algorithms widely used for both classification and regression in PHM of electronics involve k-nearest neighbor (k-NN), naïve Bayes classifiers, support vector machines (SVMs), neural networks, decision trees, random forests, linear regression, and logistic regression. More details about these supervised ML algorithms will be covered in Chapters 6 and 7.

Unlike supervised learning, the training dataset is unlabeled in unsupervised learning, as illustrated in Figure 4.4. The major tasks using unsupervised learning in PHM

2 The classification task is further divided into a binary classification task and a multi-class classification task based on the number of classes it addresses. For example, if the diagnostic system is to identify whether the product is healthy or not, this would be treated as a binary classification task. On the other hand, if the diagnostic system is to pinpoint multiple failure modes or failure mechanisms of the product, this would be treated as a multi-class classification task.

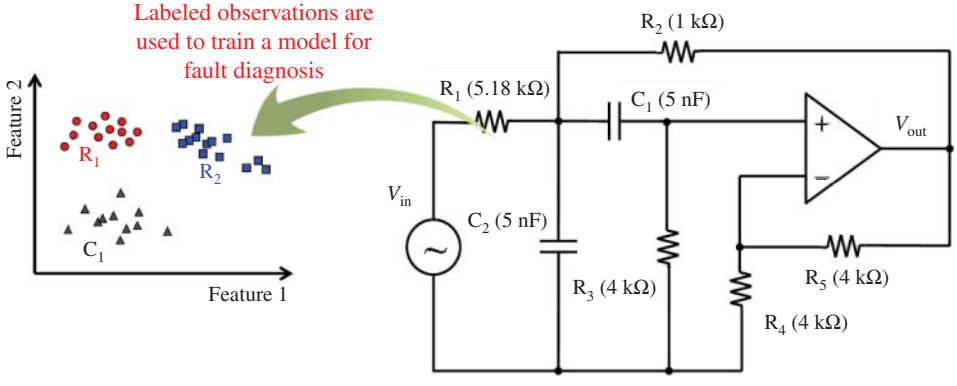


Figure 4.2 A labeled training dataset for supervised learning.

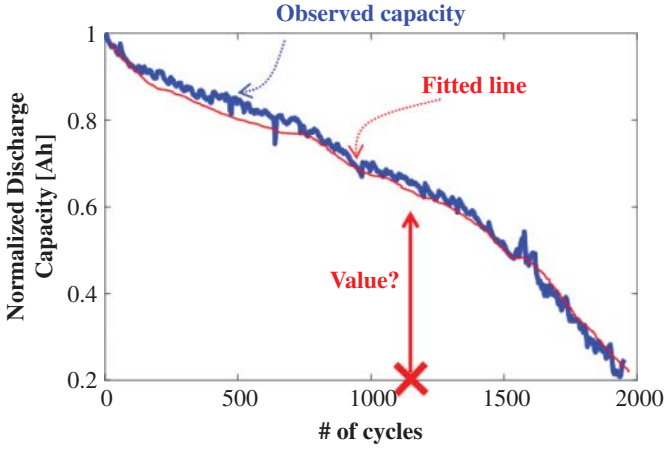
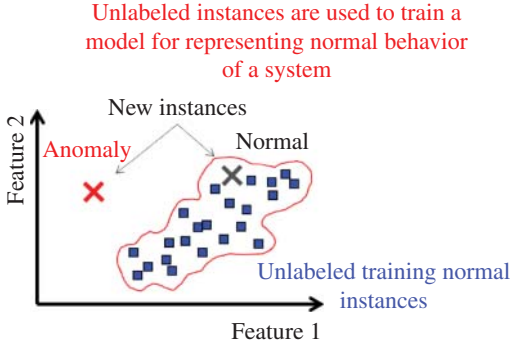


Figure 4.3 Regression concept.

Figure 4.4 An unlabeled training dataset for unsupervised learning.



of electronics are clustering (e.g. k-means, fuzzy c-means, hierarchical cluster analysis, and self-organizing map) and dimensionality reduction (e.g. principal component analysis, locally linear embedding, and t-distributed stochastic neighbor embedding). In fact, clustering has been widely used for anomaly detection (also outlier detection) under the assumption that the majority of the instances in the dataset are normal and facilitate the detection of anomalies in an unlabeled test data by looking for instances that seem to fit least to the remainder of the dataset. In PHM, dimensionality reduction is primarily used for simplifying the data without losing too much information. One way to do this is to merge correlated features into one. For example, a capacitor's capacitance can be highly correlated with its age, so the dimensionality reduction algorithm will merge them into one feature that represents the capacitor's wear, which is called feature extraction. In fact, it is often a good idea to reduce the dimensions of the dataset before it is fed to ML algorithms (e.g. supervised ML algorithms for classification). This is mainly because the dataset will run much faster, the data will take up less disk and memory space, and in some cases, it may also perform better. Likewise, unsupervised learning has been used to output two-dimensional (2D) or three-dimensional (3D) representation of the high-dimensional data that can be easily plotted.

Semi-supervised learning is a class of supervised learning tasks and techniques that make use of unlabeled data for training; the training dataset involves a lot of unlabeled data and a little bit of labeled data. Anomaly detection in a system can be a good example of semi-supervised learning [4]. For example, a system's anomalies can be detected by comparing in-situ parameters to monitor against a healthy baseline, which must be known (i.e. labeled) in advance. Likewise, the baseline dataset is often composed of a collection of parameters that represent all the possible variations of the healthy operating states of the system. The combination of deep belief networks with unsupervised components called "restricted Boltzmann machines" stacked on top of one another is another example of semi-supervised learning approaches that can be used for health diagnosis [5]. Restricted Boltzmann machines are trained sequentially in an unsupervised manner, and then the whole approach is fine-tuned using supervised learning techniques.

Reinforcement learning is the task of getting an agent that can observe the environment, to select and perform actions, and obtain rewards in return (or penalties in the form of negative rewards). The agent then learns by itself what is the best strategy, called a "policy," to get the most rewards over time; the policy defines what action the agent should choose when it is in a given situation.

4.1.2 Batch and Online Learning

As stated previously, ML algorithms can be classified into two different learning methods based on whether or not the algorithms can learn incrementally from a stream of incoming data: batch and online learning. In batch learning, the ML algorithms lack the ability to learn incrementally; they must be trained using all the available training data. Accordingly, the algorithms are trained, and then they are launched into production and run without learning anymore; they just apply what they have learned. Thus, batch learning is also called offline learning. To learn new incoming data (e.g. new health status of a system), it is necessary to train the algorithm from scratch on the full dataset (not just the new data, but also the old data), then stop the use of the old algorithm and replace it with the new one.

Despite the computing resources (e.g. memory space, disk space, and central processing unit (CPU)) and computational burden required by batch learning when training an ML algorithm on the full set of old and new data, the simplicity of batch learning can be a major reason many anomaly detection, diagnosis, and prognosis approaches are based on it. The challenge faced by batch learning is the substantial expense of maintaining a huge amount of data and automating the ML system to train from scratch every day. It may even be impossible to use a batch learning algorithm depending on the amount of data. Fortunately, a better option in all these cases is to use algorithms that are capable of learning incrementally.

The primary goal of online learning, also called incremental learning, is to train ML algorithms incrementally by feeding them data instances sequentially, either individually or in small groups called mini-batches. The whole process of online learning enables the online learning algorithm to learn about new data on the fly, as it arrives. Hence, incremental learning is more computationally effective for dealing with new data than batch learning, and works well for systems that receive data as a continuous flow and need to adapt to change rapidly or autonomously. Incremental learning is also a good option if computing resources are limited: once the system has learned about new data instances, it does not need them anymore, and they can be discarded (unless the user wants to be able to roll back to a previous state and "replay" the data). This option can save a substantial amount of space. Owing to the advantages of incremental learning, it has been widely used for anomaly detection, diagnosis, and prognosis.

The ever-growing use of sensors and networked things can result in the continuous generation of high-volume, high-velocity, and high-variety data, which is known as "big data" [6]. Today, the use of big data for anomaly detection, diagnosis, and prognosis is now an indispensable part of PHM. For example, General Motors has announced advanced connected vehicle technology that aims to monitor vehicle component health and notify customers if the components need attention. General Electric (GE) uses the OnStar 4G long-term evolution (LTE) connectivity platform to send data collected by the vehicle's sensors to OnStar for analysis. Online learning algorithms can be useful for dealing with big data on the system that cannot fit in its main memory (called out-of-core learning). The online learning algorithms can load part of the data, run a training step on that data, and repeat the process until they have been run on all of the data. However, despite the advantages of online learning, if bad data are fed to the online learning algorithms, their performance will gradually decline.

4.1.3 Instance-Based and Model-Based Learning

Given a number of training observations, the ML algorithms need to be able to generalize observations they have never seen before. Based on how the algorithms generalize, they can be categorized into either instance-based or model-based learning.

The simplest form of learning is probably to memorize. If the diagnostic system for electronic products is designed in this manner, it would diagnose only health states that have already been diagnosed by experts or maintainers. This approach may not be the best solution. Alternatively, the diagnostic system could be programmed to also diagnose health states that are analogous to known health states, which requires a measure of similarity between two health states. This approach is called instance-based learning because the system memorizes the instances and generalizes to new instances

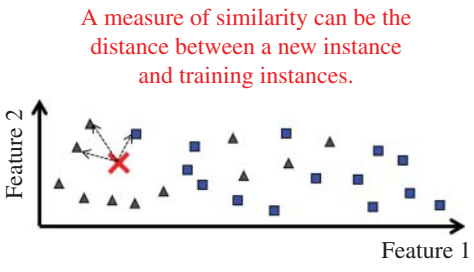


Figure 4.5 Instance-based learning concept.

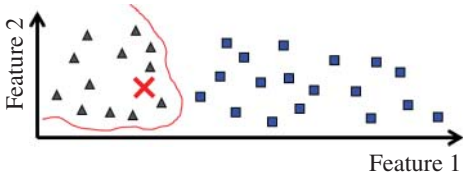


Figure 4.6 Model-based learning concept.

by employing a similarity measure (see Figure 4.5). The k-NN algorithm can be a good example of instance-based learning in PHM; Sutrisno et al. [7] showed the usefulness of the k-NN algorithm to detect faults just before insulated gate bipolar transistors enter a final degradation stage toward failure.

As shown in Figure 4.6, another way to generalize from a set of instances is to build a model of these instances, then use that model to make predictions, which is called model-based learning. The use of Gaussian process regression (GPR) for state of health estimation of lithium-ion batteries is a good example of model-based learning [8].

The following three approaches can be considered in modeling $p(y|\, x)$, where $p(x|\, y)$ is the conditional probability of an output y given an input data point x: generative models, discriminative models, and encoding a function. In generative models, the joint probability of x and y, $p(x, y)$, is directly obtained, and further, the conditional probability $p(y|\, x)$ can be obtained by simply conditioning on x. Then, decision theory will be used to determine which class the data point x belongs to (for classification problems). For example, the naïve Bayes model learns $p(y)$, the prior class probabilities from the training dataset. The naïve Bayes model further learns $p(x|\, y)$ from the training dataset using maximum likelihood estimation (MLE). Once $p(y)$ and $p(x|\, y)$ are obtained, $p(x, y)$ is not difficult to find out. For discriminative models, the conditional probability, $p(x|\, y)$, is directly computed. For example, logistic regression assumes $p(y|\, x)$ to be of the form $\frac{1}{1+e^{-(wx)}}$, where w is a weight vector that would minimize the squared loss. The encoding function approach is to find a function $f(\cdot)$ that directly maps x to a class. A representative example of this approach includes decision trees; Ye et al. [9] investigated an adaptive diagnosis method based on incremental decision trees for complex boards from industry in volume production.

4.2 Probability Theory in Machine Learning: Fundamentals

Probability theory plays a significant role in ML, specifically as the design of learning algorithms often depends on probabilistic assumption of the data. Accordingly, this section covers fundamental probability theory.

4.2.1 Probability Space and Random Variables

The probability of an event is referred to as the measure of the likelihood that the event will occur. Hence, in order to discuss probability theory, a probability space (Ω, F, P) must be properly defined, where Ω is the space of possible outcomes (or outcome space), $F \subseteq 2^{\Omega}$ (the power set of Ω) is the space of measurable events (or event space), and P is the probability measure (or probability distribution) that maps an event $E \in F$ to a real value ranging from 0 to 1. Given an event space F, the probability measure P must satisfy certain axioms: $P(\alpha) \geq 0, \forall\, \alpha \in F$ and $P(\Omega) = 1$. Although further probability axioms are not covered in this book, they are available in a probability and statistics textbook.

Understanding random variables is crucial for studying probability theory. Random variables are functions that map outcomes to real values, but not, in fact, to variables. Consider the process of tossing a coin. Let X be a random variable that relies on the outcome of the toss. A possible choice for X would be to map the event of a "head" to the value of 1. For notations, the probability of a random variable X taking on the value of a will be denoted by either $P(X = a)$ or $P_X(a)$. Note that capital letters, such as X, Y, and Z, are typically used to denote random variables, and lowercase letters, such as x, y, z and a, b, c, are used to denote particular values that the random variables can take on.

4.2.2 Distributions, Joint Distributions, and Marginal Distributions

The distribution of a variable can be formally referred to as the probability of a random variable taking on certain values. Let a random variable X be defined on the outcome space Ω of a dice throw. If the dice is fair, the distribution of X would be $P(X = 1) = P(X = 2) = \cdots = P(X = 6) = \frac{1}{6}$. Likewise, the distribution of more than one random variable refers to a joint distribution. Let X be a random variable defined on the outcome space of a dice throw. Let Y be an indicator variable that takes on a value of 1 if a coin flip turns up heads, and 0 if tails. Then, the joint distribution of $X = 1$ and $Y = 0$, denoted as $P(X = 1, Y = 0)$, will be $\frac{1}{12}$.

The marginal distribution refers to the probability distribution of a random variable on its own. Hence, to find out the marginal distribution of a random variable, all the other random variables from the distribution must be summed out as follows:

$$P(X) = \sum_{b=Val(Y)} P(X, Y = b) \tag{4.1}$$

where $Val(Y)$ is the range of a random variable Y.

4.2.3 Conditional Distributions

One of the key tools in probability theory for reasoning about uncertainty is conditional distributions, which specify the distribution of a random variable when the value of another random variable is known. The conditional probability of $X = a$ given $Y = b$ is defined as follows:

$$P(X = a|Y = b) = \frac{P(X = a, Y = b)}{P(Y = b)} \tag{4.2}$$

Let X be the random variable of a dice throw and Y be an indicator variable that takes on the value of 1 if the dice throw turns up odd. Then, the probability of throwing a "one"

under the assumption that the dice throw was odd can be written as follows:

$$P(X = 1|Y = 1) = \frac{P(X = 1, Y = 1)}{P(Y = 1)} = \frac{1/6}{1/2} = \frac{1}{3} \tag{4.3}$$

Further, the conditional probability extends to the case when the distribution of a random variable is conditioned on multiple variables as follows:

$$P(X = a|Y = b, Z = c) = \frac{P(X = a, Y = b, Z = c)}{P(Y = b, Z = c)} \tag{4.4}$$

4.2.4 Independence

Two events can be said to be independent such that the probability that one event occurs in no way affects the probability of the other event occurring. In ML, the features of the training instance i are often assumed to be independent of the features of the instance j, where $i \neq j$. The distribution of a random variable X, which is independent of a random variable Y, can be written as:

$$P(X) = P(X|Y) \tag{4.5}$$

Further, the joint distribution of random variables X and Y, if and only if they are independent, can be written as follows:

$$P(X, Y) = P(X)P(Y) \tag{4.6}$$

Multiple random variables can be independent of each other given the value of a random variable (or a set of random variables). Mathematically, that means random variables X and Y are conditionally independent given Z if

$$P(X, Y|Z) = P(X|Z)P(Y|Z) \tag{4.7}$$

4.2.5 Chain Rule and Bayes Rule

To evaluate the joint probability of random variables, the chain rule is often used. Further, the chain rule is especially useful when random variables are (conditionally) independent, which is a generalization of Eq. (4.2) to multiple random variables, X_1, X_2, …, X_n:

$$P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2X_1) \ldots P(X_n|X_1, X_2, \ldots, X_{n-1}) \tag{4.8}$$

In ML, the Bayes rule is also widely used to compute the conditional probability $P(X|\,Y)$ from $P(Y|\,X)$. The Bayes rule can be simply derived from Eq. (4.2):

$$P(X|Y) = \frac{P(Y|X)P(X)}{P(Y)} \tag{4.9}$$

Note that it is possible to apply Eq. (4.1) to find $P(Y)$ if it is not given:

$$P(Y) = \sum_{a \in Val(X)} P(X = a, Y) = \sum_{a \in Val(X)} P(Y|X = a)P(X = a) \tag{4.10}$$

The Bayes rule further extends to the case of multiple random variables:

$$P(X, Y|Z) = \frac{P(Z|X, Y)P(X, Y)}{P(Z)} = \frac{P(Y, Z|X)P(X)}{P(Z)} \tag{4.11}$$

4.3 Probability Mass Function and Probability Density Function

To define a distribution, one can broadly consider discrete distributions and continuous distributions. This section will mainly discuss how distributions are specified.

4.3.1 Probability Mass Function

In probability theory, a probability mass function (pmf) gives the probability of a random variable taking on each of finitely different possible values, and it is often the primary means of defining a discrete probability distribution for either scalar or multivariate random variables whose domain is discrete. Figure 4.7 shows the graph of the pmf of a fair dice. All of the values of the pmf must be non-negative and sum up to 1 (see Section 4.2.1).

4.3.2 Probability Density Function

To describe a continuous probability distribution, a probability density function (pdf) is often used. The pdf assigns probabilities not to (discrete) single outcomes, but to (continuous) ranges of outcomes. For a continuous probability distribution, the pdf has the following properties: since the continuous random variable is defined over a continuous range of values, the graph of the pdf will be continuous over that range; and the probability that a random variable assumes a value between a and b is equal to the area under the pdf bounded by a and b. Consider the pdf shown in Figure 4.8. The probability that a random variable X is less than or equal to a is equal to the area under the curve bounded by a and $-\infty$. That is, the shaded area in Figure 4.8 represents the probability that the random variable X is less than or equal to a, which is a cumulative probability. The mathematical statement of this probability is $P(X \leq a) = \int_{-\infty}^{a} f(x)dx$, where f is the pdf. However, the probability of a continuously distributed random variable taking on any given single value is always zero, such as $P(X = a) = 0$.

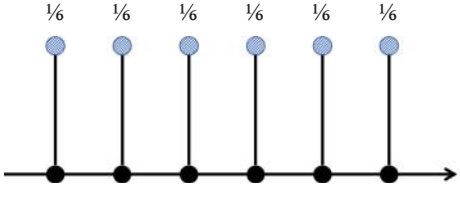


Figure 4.7 Graph of the probability mass function (pmf) of a fair dice.

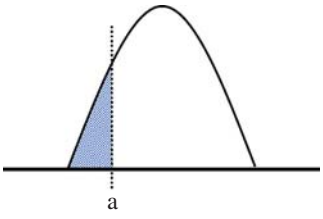


Figure 4.8 Graph of the probability density function (pdf).

4.4 Mean, Variance, and Covariance Estimation

The most common operations one can perform on a random variable are to compute its mean and variance. Likewise, covariance estimation is a significant task in the design of PHM methods. For example, principal component analysis has been widely used for dimensionality reduction (e.g. identification of key principal features for fault detection and diagnosis) [10], which can be done by eigenvalue decomposition of a data covariance (or correlation) matrix [11]. Likewise, the Mahalanobis distance [12], a measure of distance of an observation from a set of observations with mean and covariance matrix, is often used for anomaly detection [13].

4.4.1 Mean

The mean of a random variable X, also known as its expectation, expected value, or first moment, is computed by:

$$E(X) = \sum_{a \in Val(X)} aP(X = a) \text{ or } E(X) = \int_{a \in Val(X)} xf(x)dx \tag{4.12}$$

where $E(X)$ (also denoted as μ) is the mean of the random variable X. Let X be the outcome of rolling a fair dice. The mean of the random variable X is computed as $E(X) = (1)\frac{1}{6} + (2)\frac{1}{6} + \ldots (6)\frac{1}{6} = \frac{21}{6}$.

The linearity of expectations is an important rule when working with the sums of multiple random variables. Likewise, this rule does not depend on whether the random variables are independent or not. Let $X_1, X_2, \ldots, X_n$ be random variables. By the linearity of expectations, the mean of the random variables can be written as:

$$E(X_1, X_2, \ldots, X_n) = E(X_1) + E(X_2) + \ldots E(X_n) \tag{4.13}$$

For independent random variables X and Y, the mean of the product of the variables is calculated as follows:

$$E(XY) = E(X)E(Y) \tag{4.14}$$

4.4.2 Variance

The spread of a distribution can be measured by the variance of a distribution, which is also referred to as the second moment. For a discrete random variable X, the variance is computed as:

$$Var(X) = E[(X - \mu)^2] = \sum_{a \in Val(X)} (a - \mu)^2 P(X = a) \tag{4.15}$$

For a continuous random variable X, the variance is calculated as:

$$Var(X) = E[(X - \mu)^2] = \int_{a \in Val(X)} (x - \mu)^2 f(x) dx \tag{4.16}$$

where $Var(X)$ is the variance of the random variable X, which is often denoted by σ^2. This is because the variance and the standard deviation σ is related by $\sigma = \sqrt{Var(X)}$.

Unlike expectation, variance is not a linear function of a random variable. Accordingly, the variance of $aX + b$ is calculated as:

$$Var(aX + b) = a^2 Var(X) \tag{4.17}$$

Likewise, the variance of independent random variables X and Y is given by:

$$Var(X + Y) = Var(X)Var(Y) \tag{4.18}$$

4.4.3 Robust Covariance Estimation

The covariance of two random variables is a measure of how closely related two random variables are, defined as:

$$Cov(X, Y) = E((X - \mu)(Y - \mu)) \tag{4.19}$$

As stated previously, the estimation of a population's covariance matrix is required to develop PHM methods. The covariance matrix of a dataset has been approximated with the classical maximum likelihood estimator (or empirical covariance) [14] and shrinkage estimator [15]. However, because the aforementioned empirical covariance estimator and the shrinkage covariance estimator are sensitive to the presence of outlying observations in the dataset, robust covariance estimators are alternatively employed for covariance estimation. Hence, this section will primarily explain robust covariance estimators involving minimum covariance determinant, minimum volume ellipsoid, and successive differences.

Robust covariance estimation is a tool that is used to estimate the covariance of a dataset while attempting to minimize the influence of outliers. The simplest method is called successive differences [16, 17]. The successive difference covariance matrix is robust against shifts or drifts in the mean of the data. This robust covariance matrix is computed by first forming the difference vectors $\mathrm{v_i} = \mathrm{x_{i+1}} - \mathrm{x_i}$, for $i = 1, \ldots, n-1$, where each $\mathrm{x_i}$ is an observation in d-dimensional space. Then each of these difference vectors is stacked into a matrix V, and the successive differences covariance matrix estimate is given by:

$$S_D = \frac{V^{\mathrm{T}} V}{2(m-1)} \tag{4.20}$$

The minimum covariance determinant covariance matrix estimate builds a covariance matrix based upon the h observations for which the determinant of the sample covariance matrix is minimal [18, 19]. Let $\lfloor \cdot \rfloor$ denote the floor function, n be the number of observations, and d be the number of dimensions/features, then h is bounded according to:

$$\left\lfloor \frac{(n + d + 1)}{2} \right\rfloor \leq h \leq n \tag{4.21}$$

Hence, the algorithm consists of randomly selecting subsets of the data of size h, and computing the sample covariance matrix, and then selecting the sample mean and covariance estimates as the robust mean and covariance estimates.

The minimum volume ellipsoid covariance matrix builds a covariance matrix based upon the subset h of n observations within the data that yield an ellipsoid of minimal volume [20, 21]. For a single iteration in this algorithm, h (the same h as described in Eq. (4.21)) random samples of observations are drawn from the dataset, then the sample mean and covariance are computed; if the covariance is singular, then a single randomly selected data point is added until the sample covariance matrix is not singular. These steps are repeated for t iterations, and the sample mean and covariance for which the volume of the covariance ellipsoid is minimum is chosen.

4.5 Probability Distributions

This section reviews some of the probability distributions: Bernoulli, normal, and, uniform. These distributions are widely used for probability problems; in many methods using particle filters (e.g. standard particle filter and unscented particle filter) for RUL estimation of lithium-ion batteries, degradation model parameters were assumed to be uniformly or normally distributed [22, 23].

4.5.1 Bernoulli Distribution

In probability theory, the Bernoulli distribution is one of the most basic probability distributions of a random variable which takes two possible values: the value 1 with probability p and the value 0 with probability $q = 1 - p$. Accordingly, the Bernoulli distribution can be used to represent the presence of a fault in a system (or component) where 1 and 0 would represent "healthy" and "faulty" (or vice versa), respectively.

If a random variable X is distributed along with the Bernoulli distribution, the probability of the variable can be written as:

$$P(X = 1) = p \text{ and } P(X = 0) = q = 1 - p \tag{4.22}$$

Thus, the probability mass function f of the Bernoulli distribution can be expressed as follows:

$$f(k; p) = p^k (1 - p)^{1-k} \tag{4.23}$$

where k is one of the possible outcomes (i.e. 1 and 0). The Bernoulli distribution is used in logistic regression to predict machinery conditions (i.e. healthy and faulty) [24].

4.5.2 Normal Distribution

The normal (or Gaussian) distribution is the probability distribution of a continuous random variable. In fact, it is one of the most versatile distributions used for a wide variety of applications such as uncertainty management in RUL prediction [25] and modeling measurement noise [26].

The usefulness of the normal distribution in many applications is because of the central limit theorem, which establishes that when independent random variables are added, their sum tends toward a normal distribution regardless of the underlying distribution of the variables. For example, if one flips a coin many times, the probability of getting a given number of heads in a series of flips should follow a normal distribution, with the mean equal to half the total number of flips in each series.

The mean μ and the variance σ^2 are two parameters specifying the normal distribution, and the probability density function f of a continuous random variable X is expressed as:

$$f(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma_2}} \tag{4.24}$$

4.5.3 Uniform Distribution

The uniform distribution is the probability distribution of a continuous random variable that has a constant probability between two bounding parameters, and its probability density function $f(x)$ is expressed as:

$$f(x;a,b) = \begin{cases} \frac{1}{b-a} & \text{for} \quad a \le x \le b \\ 0 & \text{for} \quad x < a \quad \text{or} \quad x > b \end{cases} \tag{4.25}$$

where a and b are the two bounding parameters. In fact, the values of the probability density function $f(x)$ at a and b are commonly not important.

4.6 Maximum Likelihood and Maximum A Posteriori Estimation

Let $D = \{d_1, d_2, \ldots, d_n\}$ be a set of data generated from a probability distribution by a vector of parameters θ, where each instance in D can be mathematically expressed as:

$$d_i \sim P(d_i|\theta), \quad i = 1, \ 2, \ldots, n \tag{4.26}$$

where n is the total number of instances in the dataset. Note that all the instances in D are independent and identically distributed; each instance in D is independent of all other instances given θ; and all instances in D are drawn from the same distribution. To estimate the fixed but unknown parameters given D, that is, $\arg\max_{\theta} P(\theta|D)$, the following classic methods are widely used: maximum likelihood estimation (MLE) and maximum a posteriori estimation (MAP).

4.6.1 Maximum Likelihood Estimation

The MLE is a method of estimating the parameters θ of a statistical model given observations, given by Bayes' theorem:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)} \tag{4.27}$$

where $P(\theta)$ is the prior distribution for the parameters θ and $P(D)$ is the probability of the data averaged over all parameters. Then, the estimated parameters $\widehat{\theta}_{\text{MLE}}$ are obtained by maximizing $P(D|\,\theta)P(\theta)$ with respect to θ because the denominator is independent of θ in Eq. (4.27). If the prior $P(\theta)$ is further assumed to be a uniform distribution, the estimated parameters $\widehat{\theta}_{\text{MLE}}$ are finally obtained by maximizing $P(D|\,\theta)$, defined as:

$$\widehat{\theta}_{\text{MLE}} = \arg\max_{\theta}\ P(D|\theta) = \arg\max_{\theta}\ P(d_1, d_2, \ldots, d_n|\theta) \tag{4.28}$$

Due to the underlying assumption that the instances in D are independent and identically distributed, the joint density function for all instances can be written as follows:

$$P(d_1, d_2, \ldots, d_n|\theta) = P(d_1|\theta) \times P(d_2|\theta) \times \ldots \times P(d_n|\theta) = \prod_{i=1}^{n} P(d_i|\theta) \tag{4.29}$$

To simplify the computation, MLE often maximizes the log-likelihood by taking the logarithm of the likelihood because the logarithm is monotonically increasing:

$$\log P(D|\theta) = \sum_{i=1}^{n} \log P(d_i|\theta) \tag{4.30}$$

Now, the mathematical expression of the estimated parameters $\widehat{\theta}_{\text{MLE}}$ is represented by:

$$\widehat{\theta}_{\text{MLE}} = \arg\max_{\theta} \sum_{i=1}^{n} \log P(d_i|\theta) \tag{4.31}$$

If the distribution P is known, the estimated parameters $\widehat{\theta}_{\text{MLE}}$ will be obtained by solving the following:

$$\frac{\partial \sum_{i=1}^{n} \log P(d_i|\theta)}{\partial \theta} = 0 \tag{4.32}$$

4.6.2 Maximum A Posteriori Estimation

Unlike MLE, the estimated parameters $\widehat{\theta}_{\text{MAP}}$ in MAP are obtained by directly maximizing a posteriori $P(\theta|\,D)$:

$$\widehat{\theta}_{\text{MAP}} = \arg\max_{\theta}\ P(\theta|D) = \arg\max_{\theta} \frac{P(D|\theta)P(\theta)}{P(D)} = \arg\max_{\theta}\ P(D|\theta)P(\theta) \tag{4.33}$$

Note that the last step in Eq. (4.33) is because $P(D)$ is independent of θ, that is, $P(D)$ is treated as a normalized term that is not necessarily considered in estimating the parameters θ. Likewise, one can say that MAP is more general than MLE because it is possible to remove $P(\theta)$ from Eq. (4.33) under the assumption that the possible θ are equally probable a priori, that is, θ is uniformly distributed. Analogous to MLE, MAP will obtain the estimated parameters $\widehat{\theta}_{\text{MAP}}$ by taking a logarithm on Eq. (4.33) for the sake of computational simplification:

$$\widehat{\theta}_{\text{MAP}} = \arg\max_{\theta} \left(\sum_{i=1}^{n} \log P(d_i|\theta) + \log P(\theta) \right) \tag{4.34}$$

The term $\log P(\theta)$ in Eq. (4.34) has the effect that one can essentially pull the θ distribution toward prior value. This makes sense if one can put their domain knowledge as prior.

4.7 Correlation and Causation

Correlation analysis is a method of statistical evaluation used to study the strength of a relationship between two, numerically measured, continuous variables (e.g. height and weight). This analysis is useful when a researcher wants to establish whether there are possible connections between variables. It is often misunderstood that correlation analysis determines cause and effect.

As aforementioned, that correlation does not imply causation is one of the most famous axioms in an elementary study of statistics. However, in this case, Sherlock Holmes may never proclaim, "Elementary, my dear Watson" because this axiom is not unarguable. A more pertinent statement could be "Correlation is not causation, but it surely is a hint" [27]. However, it is unquestionable that controlled but randomized experiments are needed to discern the difference between the two. The Australian Bureau of Statistics (http://www.abs.gov.au/websitedbs/a3121120.nsf/home/statistical+language+-+correlation+and+causation) defines correlation as a statistical measure that describes the size and direction of a relationship between two or more variables, whereas causality indicates that there is a causal relationship between the two events, that is, one event results from the occurrence of the other event. This is also referred to as a cause and effect relationship. To understand any process, it is extremely important to distinguish between causation and correlation.

From the definition of the two concepts, it may seem that the distinction is trivial to identify, especially in outlandish examples like the observation that homicide rates rise with rising ice cream sales. However, this distinction can be really tricky to make in obscure scenarios like in the case of numerous epidemiological studies which showed that women taking combined hormone replacement therapy (HRT) were also observed to have a comparatively lower incidence of coronary heart disease (CHD). However, when randomized controlled trials were conducted for the aforementioned scenario, researchers actually found that HRT leads to a small but statistically significant increase in CHD risk. Had the trials not been conducted, researchers would have made the logical fallacy of "post hoc, ergo procter hoc"; that is, since event Y followed event X, event Y must have been caused by event X. As a result, most statisticians suggest conducting randomized experiments to assess whether the relationship between variables is a causal one or merely due to incidental correlation. The most effective way of establishing causality between variables is a controlled study. In such a study, the sample or population is split in two and an effort is made to make both groups comparable in almost every comprehensible way. The two groups are then administered different treatments, after which the outcomes of each group are assessed. In medical research, one group is given a new type of medication while the other group is given a placebo. If the two groups have noticeably different responses, then a case can be made for the causality of the medicine and its effect on the group. However, due to ethical considerations, conducting controlled studies are not always possible. As a result, observational studies are often used to investigate correlation and causation for the population of interest. These studies monitor the groups' behaviors and outcomes over time.

There has been a lot of debate in the statistical community regarding the adage that correlation does not imply causation, because the definition of "pure causality" is fraught with philosophical arguments. There are numerous cases of researchers using correlation as scientific evidence, but in such cases the burden of proof falls on the researcher

to show why the correlations are logical. In other words, the correlation has to be proven to be transcendental as opposed to being just incidental. There are also scenarios where conducting experimental trials is difficult, and hence correlation from several angles is used to build up the strongest possible causal evidence. For example, the Granger Causality Test [28] is a statistical hypothesis test for causality used for determining whether a time series is useful in forecasting another using correlations between the different lags of the two time series. A cautionary tale is the rejection of the correlation evidence between smoking and lung cancer by the tobacco industry. Limited experimental trials combined with the correlation fallacy has been used to counter a scientific finding.

Another way to look at it is that whenever we conduct randomized experiments, all we are looking for are explanatory variables for a process. These variables explain variability in a process because they are correlated with the process. Hence, experiments relate correlation to causality, but in a controlled environment where confounding variables can be blocked. This is what sets experiments apart from merely observing correlated variables and inferring causality.

In conclusion, correlation can be used as an evidence for a cause–effect relationship by ensuring that correlations are logical in the context they are used. At the same time, one has to be mindful of the fact that correlational evidences tend to come under extreme scrutiny because of their tendency to be abused by arriving at premature or even a favorable conclusions.

4.8 Kernel Trick

An $m \times n$ data matrix D can be formed for fault diagnosis in analog circuits, involving wavelet features [29]:

$$D = \begin{bmatrix} d_1^{(1)} & d_2^{(1)} & \cdots & d_n^{(1)} \\ d_1^{(2)} & d_2^{(2)} & \cdots & d_n^{(2)} \\ \cdots & \cdots & \cdots & \cdots \\ d_1^{(m)} & d_2^{(m)} & \cdots & d_n^{(m)} \end{bmatrix} \tag{4.35}$$

where m is the total number of instances (or observations) used for fault diagnosis and n is the number of features (e.g. wavelet features calculated from an impulse response of the analog circuit). That is, $d_i = [d_i^{(1)}, d_i^{(2)}, \ldots, d_i^{(n)}]$ represents an n-dimensional feature vector of the ith instance, which will be further used to train or test ML algorithms.

Suppose that the primary goal of fault diagnosis in analog circuits is to determine whether they are healthy or faulty with the help of ML algorithms using a 2-class data matrix D (see Figure 4.9). The first step is to pick and train a classifier to predict the class labels of future instances. Since the given problem is a binary classification problem, one can possibly come up with a linear support vector machine (SVM), a simple and well-known binary classifier, to solve the problem. In fact, the objective of the linear SVM is to find a hyperplane $\vec{\omega}$, also known as the decision boundary, that maximally separates the training instances by class label. The hyperplane $\vec{\omega}$ can probably cut the space into two halves: one half for class 0 (or healthy) and the other half for class 1 (or faulty), as illustrated in Figure 4.9b. Then, one can observe which side of $\vec{\omega}$ that an unseen future instance lies, to determine the circuit's health status.

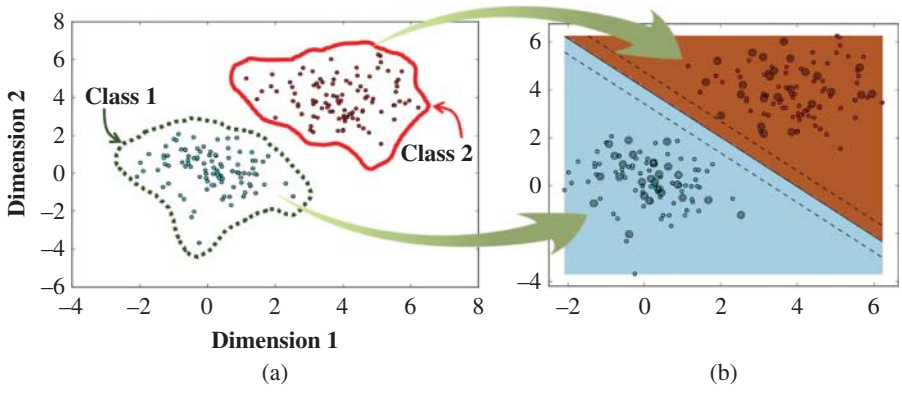


Figure 4.9 (a) A two-class, linearly separable dataset and (b) the decision boundary $\vec{\omega}$ of a linear SVM on the dataset, where the solid line is the boundary.

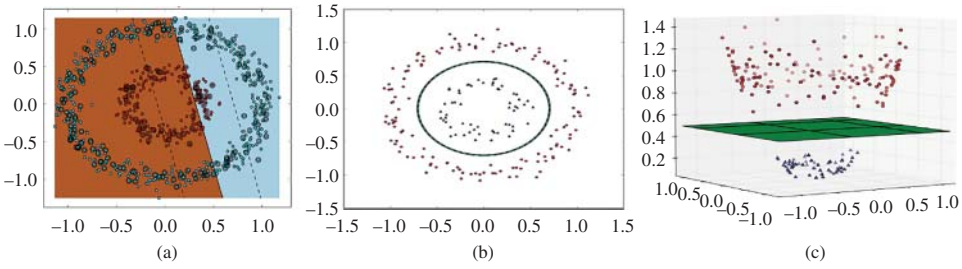


Figure 4.10 (a) A dataset in $\mathbb{R}^2$, not linearly separable; (b) a circular decision boundary that can separate the outer ring from the inner ring; and (c) a dataset transformed by the transformation $T([d_1, d_2]) = [d_1, d_2, d_1^2 + d_2^2]$.

Unfortunately, in practice one will not always encounter such well-behaved datasets. Consider a dataset in Figure 4.10a. One can expect that the linear SVM will perform poorly on this dataset because the decision boundary fails to coherently separate the circuit's health status (see the decision boundary in Figure 4.10a). As illustrated in Figure 4.10b, a better decision boundary would be a circular decision boundary that separates the outer ring from the inner ring. However, the problem faced by the linear SVM is that the decision boundary is linear in the original feature space. Figure 4.10c is a 2D version of the true dataset that lives in a three-dimensional feature space. In fact, the dataset in the 3D space can easily be separable by a hyperplane $\vec{\omega}$. Namely, the linear SVM will likely perform well for classification. However, the challenge is to find a transformation, that is T: $\mathbb{R}^2 \rightarrow \mathbb{R}^3$, such that the transformed dataset is linearly separable in $\mathbb{R}^3$. In Figure 4.10c, the transformation $T([d_1, d_2]) = [d_1, d_2, d_1^2 + d_2^2]$ was used, which, after being applied to every data point (or instance) in Figure 4.10b, yields the linearly separable dataset in Figure 4.10c. This approach is called the "kernel trick" and efficiently avoids the explicit mapping to get linear learning algorithms to learn a nonlinear decision boundary. Note that the transformation to map an original dataset into a higher-dimensional feature space is often referred to as a kernel or a kernel function.

ML algorithms capable of operating with the kernel trick for PHM of electronics include SVMs, Gaussian process, principal component analysis, canonical correlation analysis, ridge regression, and spectral clustering. Likewise, popular kernels, including the polynomial (Gaussian), radial basis function, and sigmoid kernel, are defined as:

$$\text{Polynomial kernel} : K(d_i, d_j) = (\alpha d_i \cdot d_j + \beta)^p, \quad i = j = 1, 2, \ldots, m \tag{4.36}$$

where $K()$ is a kernel function, d_i and d_j are the ith and jth n-dimensional feature vectors in a $m \times n$ dataset, respectively, $d_i \cdot d_j$ is the inner product to multiply the two feature vectors d_i and d_j, with the result of this multiplication being a scalar, α is the slope of the polynomial function, β is the intercept constant, and p is the order of the polynomial kernel. The (Gaussian) radial basis function is expressed as:

$$\text{(Gaussian) radial basis function kernel} : K(d_i, d_j) = \exp(-\gamma \|d_i - d_j\|^2), \quad i = j = 1, 2, \ldots, m \tag{4.37}$$

where exp() is the exponential function, $\gamma = \frac{1}{2\sigma^2}$, and σ is an adjustable parameter. If σ is small, the exponential is linear, and the higher-dimensional projection loses its nonlinear power. In contrast, if σ is too large, the decision boundary is very sensitive to noise due to the lack of regularization. Likewise, $\|d_i - d_j\|^2$ is the squared Euclidean distance between the two feature vectors d_i and d_j in Eq. (4.37). The sigmoid kernel, also called the hyperbolic tangent kernel, is defined as:

$$\text{Sigmoid kernel} : K(d_i, d_j) = \tanh(\alpha d_i \cdot d_j + \beta), \quad i = j = 1, 2, \ldots, m \tag{4.38}$$

where tanh() is the hyperbolic tangent function.

4.9 Performance Metrics

This section primarily reviews performance metrics used in data-driven diagnostics and prognostics in PHM.

4.9.1 Diagnostic Metrics

From a ML point of view, diagnostics, defined as the action of determining the presence, location, and severity of a fault (or faults), can possibly be a binary or multi-class classification task. Accordingly, performance metrics used in ML classification tasks can also be useful for assessing the diagnostic performance in PHM.

To assess the performance of a classification model (or classifier) on a test dataset for which the true values (or classes) are known, a confusion matrix as presented in Table 4.1, constituting true positive (TP), true negative (TN), false positive (FP), and false negative (FN), respectively, is widely used.

In Table 4.1, TP is the case that a test instance in the positive class is correctly recognized as the positive class, TN is the case that a test instance in the negative class is correctly identified as the negative class, FP is the case that a test instance belonging to the negative class is incorrectly recognized as the positive class, and FN is the case that a test instance belonging to the positive class is incorrectly assigned to the negative class, respectively.

Table 4.1 A confusion matrix.

		Predicted	
		Positive	Negative
Actual	Positive	True positive (TP)	False negative (FN)
	Negative	False positive (FP)	True negative (TN)

The common performance measures for diagnostics include accuracy, sensitivity (or recall), and specificity. These measures are computed based on the number of TPs, TNs, FPs, and FNs, defined as:

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4.39}$$

$$\text{Sensitivity (or recall, or true positive rate)} = \frac{TP}{(TP + FN)} \tag{4.40}$$

$$\text{Specificity} = \frac{TN}{(TN + FP)} \tag{4.41}$$

Matthews correlation voefficient (MCC)

$$= \frac{(TP \cdot TN + FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \tag{4.42}$$

$$F_\beta = \frac{(\beta^2 + 1) \cdot TP}{((\beta^2 + 1) \cdot TP + FP + \beta \cdot FN)} \tag{4.43}$$

Accuracy is the proportion of true assessments, either TP or TN, in a population; that is, it measures the degree of diagnostic veracity. However, the problem that can be faced by the accuracy measure is the "accuracy paradox" [30]. This states that a classification model with a given level of accuracy may have greater predictive power than models with higher accuracy. Accordingly, it may be better to avoid the accuracy metric in favor of other metrics such as precision and recall. For example, a well-trained classifier was tested on 100 unseen instances – a total of 80 instances were labeled "healthy" and the remaining 20 instances were labeled "faulty" – and yielded classification accuracy of 80%. At first glance, it seems that the classifier performs well. However, 80% accuracy can be a frustrating result because the classifier may not be able to predict "faulty" instances at all.

Sensitivity measures the proportion of TPs (i.e. the percentage of "healthy" instances that are correctly identified as "healthy"). Accordingly, a classifier with high sensitivity is especially good at detecting a system's health status (not TNs, i.e. a system's faulty status). In Eq. (4.41), specificity measures the proportion of TNs (i.e. the percentage of "faulty" instances that are correctly identified as not "healthy"). More specifically, a classifier with high specificity is good at avoiding false alarms. In summary, both sensitivity and specificity are widely used with accuracy as diagnostic metrics.

To assess classification performance (especially for a binary classification problem), a well-known method is receiver operating characteristics (ROCs) analysis using the true positive rate (TPR), also called sensitivity (or recall), against the false positive rate (FPR),

where FPR can be measured as:

$$\text{FPR}\,(1-\text{specificity}) = \frac{FP}{FP+TN} \tag{4.44}$$

In Eq. (4.44), FPR is equivalent to (1 – specificity). All possible combinations of TPR and FPR consist of an ROC space; that is, a location of a point in the ROC space can show the trade-off between sensitivity and specificity (i.e. the increase in sensitivity is accompanied by a decrease in specificity). Accordingly, the location of the point in the space can represent whether the (binary) classifier performs accurately or not. As illustrated in Figure 4.11, if a classifier works perfectly, a point determined by both TPR and FPR would be a coordinate (0, 1), indicating that the classifier achieves a sensitivity of 100% and a specificity of 100%, respectively. If the classifier yields a sensitivity of 50% and a specificity of 50%, a data point can lie on the diagonal line (see Figure 4.11) determined by coordinates (0, 0) and (1, 0), respectively. Theoretically, a random guess would give a point along the diagonal line. In Figure 4.11, an ROC curve can be plotted by employing TPR against FPR for different cut-points, starting from a coordinate (0, 0) and ending at a coordinate (1, 1). More specifically, the x-axis represents FPR, 1 – specificity, and the y-axis represents TPR, sensitivity. In the ROC curve, the closer the point on the ROC curve to the ideal coordinate (1, 0), the less accurate is the classifier. In ROC analysis, the area under the receiver operating characteristic curve, also known as AUC, can be calculated to provide a way to measure the accuracy of a classifier (i.e. a binary classifier):

$$AUC = \int_0^1 ROC(t)dt \tag{4.45}$$

where t equals FPR, and ROC(t) is TPR (see Figure 4.11). Likewise, the larger the area, the more accurate is the classifier. In practice, if the classifier yields $0.8 \leq \text{AUC} \leq 1$, its classification performance can be said to be good or excellent.

Besides the above-mentioned diagnostic metrics, such as accuracy, sensitivity (or recall and TPR), specificity, and AUC, both Matthews correlation coefficient (MCC) and F_β are also useful for evaluating classification performance of a binary classifier, where MCC is a correlation coefficient calculated from all values in the confusion matrix (i.e. TPs, TNs, FPs, and FNs). Additionally, F_β is a harmonic mean of recall and precision. Precision is the ratio of TPs to all positives (i.e. TPs and FPs), defined as $\frac{TP}{TP+FP}$. The F-score reaches its best value at 1 and worst at 0. In fact, two commonly used F-scores are the F_2 measure (i.e. $\beta = 2$ in Eq. (4.43)), which weights recall higher than

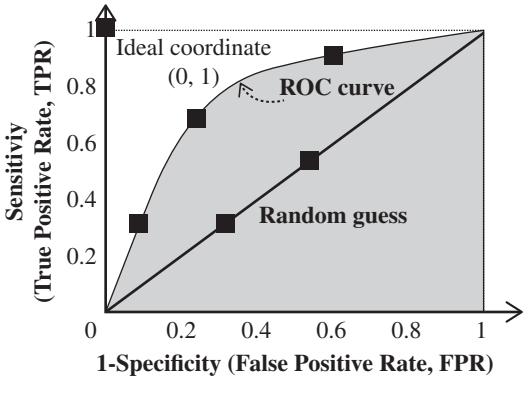


Figure 4.11 Example of an ROC space.

precision (by placing more emphasis on FNs), and the $F_{0.5}$ measure (i.e. $\beta = 0.5$ in Eq. (4.43)), which weights recall lower than precision (by attenuating the influence of FNs).

In PHM, one can often meet multi-class classification problems. For example, the identification of failure modes can be a multi-class classification task because the number of classes (i.e. failure modes) to be classified is greater than 2. The above-mentioned diagnostic metrics can extend to the metrics for multi-class classification, defined as:

$$\text{Average accuracy} = \frac{1}{N_{\text{class}}} \sum_{i=1}^{N_{\text{class}}} \frac{(TP_i + TN_i)}{(TP_i + TN_i + FP_i + FN_i)} \tag{4.46}$$

$$\mu - \text{averaging of sensitivity} = \frac{\sum_{i=1}^{N_{\text{class}}} TP_i}{\sum_{i=1}^{N_{\text{class}}} (TP_i + FN_i)} \tag{4.47}$$

$$M - \text{averaging of sensitivity} = \frac{1}{N_{\text{class}}} \sum_{i=1}^{N_{\text{class}}} \frac{TP_i}{(TP_i + FN_i)} \tag{4.48}$$

$$\mu - \text{averaging of specificity} = \frac{\sum_{i=1}^{N_{\text{class}}} TN_i}{\sum_{i=1}^{N_{\text{class}}} (TN_i + FP_i)} \tag{4.49}$$

$$M - \text{averaging of specificity} = \frac{1}{N_{\text{class}}} \sum_{i=1}^{N_{\text{class}}} \frac{TN_i}{(TN_i + FP_i)} \tag{4.50}$$

$$\mu - \text{averaging of MCC} = \frac{\sum_{i=1}^{N_{\text{class}}} (TP_i \cdot TN_i + FP_i \cdot FN_i)}{\sum_{i=1}^{N_{\text{class}}} \sqrt{(TP_i + FP_i) \cdot (TP_i + FN_i) \cdot (TN_i + FP_i) \cdot (TN_i + FN_i)}} \tag{4.51}$$

$$\text{M} - \text{averaging of MCC} = \frac{1}{N_{\text{class}}} \sum_{i=1}^{N_{\text{class}}} \frac{(TP_i \cdot TN_i + FP_i \cdot FN_i)}{\sqrt{(TP_i + FP_i) \cdot (TP_i + FN_i) \cdot (TN_i + FP_i) \cdot (TN_i + FN_i)}} \tag{4.52}$$

$$\mu - \text{averaging of } F_\beta = \frac{\sum_{i=1}^{N_{\text{class}}} (\beta^2 + 1) \cdot TP_i}{\sum_{i=1}^{N_{\text{class}}} ((\beta^2 + 1) \cdot TP_i + FP_i + \beta \cdot FN_i)} \tag{4.53}$$

$$M - \text{averaging of } F_\beta = \frac{1}{N_{\text{class}}} \sum_{i=1}^{N_{\text{class}}} \frac{(\beta^2 + 1) \cdot TP_i}{((\beta^2 + 1) \cdot TP_i + FP_i + \beta \cdot FN_i)} \tag{4.54}$$

where TP_i, TN_i, FP_i, and FN_i are true positive, true negative, false positive, and false negative obtained for the ith class, respectively. Likewise, N_{class} is the total number of classes that can be specified by the given classification problem. Additionally, the terms "μ-averaging" and "M-averaging" are used to indicate micro- and macro-averaging methods, respectively. That is, in the μ-averaging method, one can get statistics by summing up the individual TPs, TNs, FPs, and FNs, whereas the M-averaging method simply takes the average of sensitivity, specificity, MCC, and F-score for different classes.

4.9.2 Prognostic Metrics

Prognostics is defined as the process of estimating an object system's RUL (mostly with a confidence bound) by predicting the progression of a fault given the current degree of degradation, the load history, and the anticipated future operational and environ-

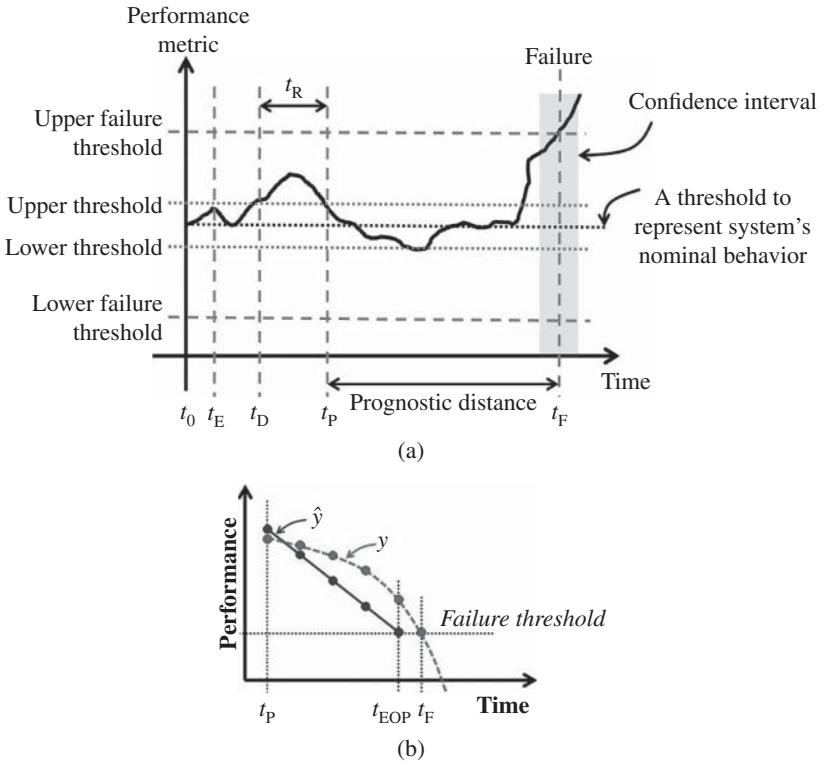


Figure 4.12 (a) Milestones on the path to object system failure and (b) the end-of-prediction (EOP) time t_{EOP} to measure the goodness-of-fit between the actual performance degradation trend y and estimated degradation trend $\hat{y}$.

mental conditions. In other words, prognostics predicts when an object system will no longer perform its intended function within the desired specifications. RUL is specified by the length of time from the present time to the estimated time at which the system is expected to no longer perform its intended function. This section reviews a variety of prognostic metrics rather than provide details about prognostic methods.

Figure 4.12a pictorially illustrates the times related to a prediction event in the operational life of an object system. First of all, the PHM designer specifies the upper and lower failure thresholds,[3] and the upper and lower off-nominal thresholds for the PHM sensor in the system. In Figure 4.12a, t_0 can be assumed to start at any time (e.g. when the system is turned on), and t_E is the occurrence of the off-nominal event. Off-nominal events occur when the PHM sensor measures an exceedance of the threshold limits specified by the PHM designer. The PHM metrics are initiated when such an event is detected at time t_D by a PHM system. The PHM system then computes a predicted failure time of the part or subsystem with its associated confidence interval. The response time t_R is the amount of time the PHM system uses to produce a predicted time of failure and make a usable prediction at time t_P. In Figure 4.12a, t_F is the actual time that the system fails and the RUL is the time difference between t_P and t_F.

3 The upper and lower failure thresholds can also be specified by standards, historical data, and so forth.

Figure 4.12b further shows the end-of-prediction time t_{EOP} to measure prognostic metrics. The common prognostic metrics include mean absolute error (MAE), mean squared error (MSE), and root-mean-squared error (RMSE). The MAE is a quantity used to measure how close the estimated performance degradation trend $\hat{y}$ (or estimates) is to the actual performance degradation trend y (or actual responses), defined by:

$$\text{MAE} = \frac{1}{(t_{\text{EOP}} - t_P + 1)} \sum_{t=t_p}^{t_{\text{EOP}}} |\hat{y}(t) - y(t)| \tag{4.55}$$

The MAE is also known as a scale-dependent accuracy measure and therefore cannot be used to make comparisons between series using different scales. Likewise, the MSE, also known as the mean squared deviation, is a measure of the average of the squares of the errors or deviations – that is, the difference between $\hat{y}$ and y is expressed as:

$$\text{MSE} = \frac{1}{(t_{\text{EOP}} - t_P + 1)} \sum_{t=t_p}^{t_{\text{EOP}}} (\hat{y}(t) - y(t))^2 \tag{4.56}$$

In practice, the MSE is a risk function, corresponding to the expected value of the squared error loss [31]. Although the MSE is widely used in the field, it has the disadvantage of heavily weighting any outliers. This is a result of the squaring of each term, which effectively weights large errors more heavily than small ones. This property sometimes has led to the use of alternatives such as the MAE.

The RMSE, also called the root-mean-squared deviation, is a measure of the differences between the values predicted by a prediction model and the values actually observed, defined as:

$$\text{RMSE} = \sqrt{\text{MSE}} \tag{4.57}$$

The RMSE is a good measure of accuracy, but only to compare different prediction errors for a particular variable and not between variables, because it is scale-dependent.

Four more prognostic metrics include prediction horizon, $\alpha - \gamma$ performance, relative accuracy, and convergence [32]. The prediction horizon identifies whether a prediction model can predict within a specified error margin, which can be specified by the parameter α around the actual end of life (EOL) of an object system. Then the $\alpha - \gamma$ performance further identifies whether the prediction model performs within desired error margins of the actual RUL at any given time instant, where the margins and time instant are specified by the parameters α and γ, respectively. The relative accuracy is obtained by quantifying the accuracy levels relative to the actual RUL, whereas the convergence quantifies how fast the prediction model converges, provided that it meets all the aforementioned prognostic metrics.

References

1 Tsui, K.L., Chen, N., Zhou, Q. et al. (2015). Prognostics and health management: a review on data driven approaches. *Mathematical Problems in Engineering* 2015: 1–17.

2 Samuel, A.L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal* 3 (3): 210–229.

3 Harrell, F.E. (2001). Ordinal logistic regression. *Regression Modeling Strategies* 331–343.

4 Pecht, M. and Jaai, R. (2010). A prognostics and health management roadmap for information and electronics-rich systems. *Microelectronics Reliability* 50: 317–323.

5 Tamilselvan, P. and Wang, P. (2013). Failure diagnosis using deep belief learning based health state classification. *Reliability Engineering & System Safety* 115: 124–135.

6 McAfee, A. and Brynjolfsson, E. (2012). Big data: the management revolution. *Harvard Business Review* 90 (10): 61–68.

7 Sutrisno, E., Fan, Q., Das, D., and Pecht, M. (2012). Anomaly detection for insulated gate bipolar transistor (IGBT) under power cycling using principal component analysis and k-nearest neighbor algorithm. *Journal of the Washington Academy of Sciences* 98 (1): 1–8.

8 Liu, D., Pang, J., Zhou, J. et al. (2013). Prognostics for state of health estimation of lithium-ion batteries based on combination Gaussian process functional regression. *Microelectronics Reliability* 53 (6): 832–839.

9 Ye, F., Zhang, Z., Chakrabarty, K., and Gu, X. (2016). Adaptive board-level functional fault diagnosis using incremental decision trees. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35 (2): 323–336.

10 Kumar, S., Dolev, E., and Pecht, M. (2010). Parameter selection for health monitoring of electronic products. *Microelectronics Reliability* 50: 61–168.

11 Abdi, H. and Williams, L.J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (4): 433–459.

12 De Maesschalck, R., Jouan-Rimbaud, D., and Massart, D.L. (2000). The Mahalanobis distance. *Chemometrics and Intelligent Laboratory Systems* 50 (1): 1–18.

13 Wang, Y., Miao, Q., Ma, E.W.M. et al. (2013). Online anomaly detection for hard disk drives based on Mahalanobis distance. *IEEE Transactions on Reliability* 62 (1): 136–145.

14 Kleinbaum, D.G. and Klein, M. (2010). Maximum likelihood techniques: an overview. *Statistics for Biology and Health* 103–127.

15 Chen, Y., Wiesel, A., Eldar, Y.C., and Hero, A.O. (2010). Shrinkage algorithms for MMSE covariance estimation. *IEEE Transactions on Signal Processing* 58 (10): 5016–5029.

16 Williams, J., Woodall, W., Birch, J., and Sullivan, J. (2006). Distribution of Hotelling's T2 statistic based on the successive differences estimator. *Journal of Quality Technology* 38 (3): 217–229.

17 Williams, J., Sullivan, J., and Birch, J. (2009). Maximum value of Hotelling's T2 statistics based on the successive differences covariance matrix estimator. *Communications in Statistics – Theory and Methods* 38 (4): 471–483.

18 Rpisseeiw, P.J. and van Driessen, K. (1999). A fast algorithm for the minimum covariance determinant estimator. *Technometrics* 41 (3): 212–223.

19 Hubert, M. and Debruyne, M. (2010). Minimum covariance determinant. *Wiley Interdisciplinary Reviews: Computational Statistics* 2 (1): 36–43.

20 Rousseeuw, P.J. and van Zomeren, B.C. (1990). Unmasking multivariate outliers and leverage points. *Journal of the American Statistical Association* 85 (411): 633–639.

21 Van Aelst, S. and Rousseeuw, P.J. (2009). Minimum volume ellipsoid. *Wiley Interdisciplinary Reviews: Computational Statistics* 1 (1): 71–82.

22 An, D., Choi, J.-H., and Kim, N.H. (2013). Prognostics 101: a tutorial for particle filter-based prognostics algorithm using Matlab. *Reliability Engineering & System Safety* 115: 161–169.

23 Miao, Q., Xie, L., Cui, H. et al. (2013). Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. *Microelectronics Reliability* 53: 805–810.

24 Phillips, J., Cripps, E., Lau, J.W., and Hodkiewicz, M.R. (2015). Classifying machinery condition using oil samples and binary logistic regression. *Mechanical Systems and Signal Processing* 60–61: 316–325.

25 Sankararaman, S. (2015). Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction. *Mechanical Systems and Signal Processing* 52–53: 228–247.

26 Fan, J., Yung, K.-C., and Pecht, M. (2015). Predicting long-term lumen maintenance life of LED light sources using a particle filter-based prognostic approach. *Expert Systems with Applications* 42 (5): 2411–2420.

27 Tufte, E.R. (2006). *The Cognitive Style of PowerPoint: Pitching out Corrupts within*, 2e. Cheshire, CT: Graphics Press.

28 Granger, C.W.J. (1969). Investigating causal relations by econometric models and cross-spectral methods. *Econometrica* 37 (3): 424–438.

29 Vasan, A.S.S., Long, B., and Pecht, M. (2013). Diagnostics and prognostics method for analog electronic circuits. *IEEE Transactions on Industrial Electronics* 60 (11): 5277–5291.

30 Reiner, M., Lev, D.D., and Rosen, A. (2017). Theta neurofeedback effects on motor memory consolidation and performance accuracy: an apparent paradox? *Neuroscience* doi: 10.1016/j.neuroscience.2017.07.022.

31 Lehmann, E.L. and Casella, G. (1998). *Theory of Point Estimation*, 2e. Springer.

32 Saxena, A., Celaya, J., Balaban, E., et al. (2008). Metrics for evaluating performance of prognostic techniques. Proceedings of the International Conference on Prognostics and Health Management, Denver, CO, USA (October 6–9).