Chapter 2.
STATISTICAL BACKGROUNDS

In this chapter, we give a brief summary about the classical methodologies on reliability data analysis, accelerated experiments, and three statistical computational methods, which are EM-Algorithm, bootstrap method and Markov Chain Monte Carlo simulations. These three statistical methods are computationally intensive devices, and they are wildly used to deal with complicated statistical problems, which can not be done easily by using classical statistical methods.

2.1 Reliability Data Analysis

In this section, we provide a brief introduction to the analysis methods according to the three types of data set as mentioned in Chapter 1. For general account of reliability data analysis, the reader is referred to Lawless (1982) and Meeker and Escobar (1998) for extensive and comprehensive details.

For lifetime data, we focus only on parametric settings. Consider N products. Assume that the lifetime T_i follows a distribution, with probability density function (pdf) $f_i(T_i;\boldsymbol{\theta})$ and cumulative distribution function (cdf) $F_i(T_i;\boldsymbol{\theta})$. Under the independence assumption, the likelihood can be written as

$$L(\boldsymbol{\theta};\boldsymbol{T}) = \prod_{i=1}^{N} f_i(T_i;\boldsymbol{\theta}).$$

The unknown parameter vector are usually estimated by its MLE, and the asymptotic variance-covariance matrix is estimated by the inverse of the observed information matrix, which is obtained by replacing the unknown parameters with their MLEs in the information matrix. If all the lifetime can be observed, the analysis procedure above is nothing but a regular statistical analysis procedure. The essential feature of reliability analysis is that the censoring is taken into consideration, and the dichotomous data is a special case of the censoring data. When censoring is taking place, the likelihood is rewritten as

$$L(\boldsymbol{\theta}; \boldsymbol{T}) = \prod_{i \in D} f_i(T_i; \boldsymbol{\theta}) \prod_{i \in C^+} [1 - F_i(C_i; \boldsymbol{\theta})] \prod_{i \in C^-} F_i(C_i; \boldsymbol{\theta}),$$

where D is the index set in which the corresponding product's lifetime can be observed; C^+ and C^- are index sets for those which are right and left censoring at time C_i, respectively. We do not consider the use of interval censoring here. When D is empty it is deduced to the likelihood of the dichotomous data. Exponential, Weibull and lognormal distributions are commonly used models for lifetime data analysis.

Another widely-used tool in reliability analysis is the degradation measurement. The essential idea of using degradation measurement to assess the lifetime T is based on the following equation

$$P(T \geq t) = P(D(t) \leq \tau), \tag{2.1}$$

where $D(t)$ is the actual degradation measure at time t, and τ is the threshold (assume that $D(t)$ is increasing in t). In practice, $D(t)$ is unknown and we collect the data at discrete time points, $t_1, t_2, \cdots$ to estimate $D(t)$. Assume that the observed degradation

Y_{ij}, the observation from unit i at time j, can be written as

$$Y_{ij} = D_{ij} + \epsilon_{ij},$$

where $D_{ij} = D(t_{ij};\boldsymbol{\beta})$ and $\epsilon_{ij}'s$ are random errors. The functional form of $D(t;\boldsymbol{\beta})$ is normally based on physical or chemical theories. The unit-to-unit inconsistency is captured by assuming some or all components of $\boldsymbol{\beta}$ to be random, and follows a normal distribution with mean vector $\boldsymbol{\mu}_{\boldsymbol{\beta}}$ and variance-covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\beta}}$. $(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ can be estimated by using a two-stage procedure, proposed by Lu and Meeker (1993). Then using (2.1), the lifetime distribution can be established.

2.2 Accelerated Experiments

Nowadays high manufacture technologies as well as statistical methods such as the experimental designs and quality controls, have been used to improve the quality of products. In order to investigate the lifetime distribution of the high-reliability products, some methodologies are needed for this task. Accelerated experiment is a prototypical one of the widely-used methods.

In an accelerated experiment, we can increase the level of some experimental factors, the so-called accelerated factors, to speed up the occurring of failures. Temperature and humidity are two commonly used accelerated factors. To estimate the lifetime distribution under normal condition, we use, for instance, the proportional hazards model, to investigate the effects of accelerated factor. Once the model is built, we can get the lifetime under normal condition via extrapolation.

The parametric model used to analyze the accelerated experiment data contains two components:

1. *A parametric lifetime distribution at each particular experimental level*: Here we assume that these distributions are from the same family with different parameters.

2. *The relationship between the distribution parameter and the accelerated factors*: The relationship can be established from the estimators of the parameters at each experimental level.

When an experimenter has rich knowledge of the chemical or (and) physical process which is related to failures, a reasonable relationship of accelerated factor and lifetime can be established, and the extrapolation is, of course, less risky. However, the relationship is usually very complicated or unknown so that it may be impossible to build up a model based on chemical or (and) physical knowledge. An empirical model is therefore providing an alternative. The empirical model is built by looking for a model which fits the data well. The empirical model can also provide a statistical relationship which is usually hard to explain in chemical or (and) physical sense. It may be risky to use for extrapolation, especially, when the sample size and levels of experimental stresses are small. Nevertheless, it seems to be the only solution due to the lack of information. The reader is referred to Chapters 18 and 19 of Meeker and Escobar (1998) for details.

2.3 EM-Algorithm

The Expectation-Maximization (EM) algorithm is an iterating procedure to compute the MLE. It as well as its extensions has been broadly used in incomplete-data problems. The "incompleteness" may cause by missing data or introducing some artificial variables so that the model is easy to formulate. The name came from Dempster, Laird and Rubin's seminar paper in 1977 which was from its two main steps in each iteration, E-step and M-step which stands for "Expectation" and "Maximization", respectively. When there are latent or missing parts appearing, the likelihood function is involved with integrals. If the latent parts can not be integrated out, finding the MLE will become difficult. EM-algorithm provides a solution which avoids the difficulty.

Let $\boldsymbol{X}_o$ and $\boldsymbol{X}_c$ be the observed (incomplete) and complete (by assuming the latent parts were given) data, respectively. The complete data can be regarded as an augment of observed data, which means $\boldsymbol{X}_c = [\boldsymbol{X}_o|\boldsymbol{X}_m]$, where $\boldsymbol{X}_m$ is the missing part. Then the likelihood, $L(\boldsymbol{\theta};\mathbf{X_o})$, can be written as an integral of complete likelihood, $L^c(\boldsymbol{\theta};\mathbf{X_c})$, which is

$$L(\boldsymbol{\theta};\mathbf{X_o}) = \int_{\Omega_{\mathbf{X_o}}} L^c(\boldsymbol{\theta};\mathbf{X_c})d\mathbf{X_m},$$

where $\Omega_{\mathbf{X_o}} = \{\mathbf{X_m} : L^c(\boldsymbol{\theta};[\boldsymbol{X}_o|\boldsymbol{X}_m]) > 0\}$. The basic idea is that if the latent parts were given, the maximization for calculating the MLE could be proceeded with respect to the complete log-likelihood which is much easier to the real log-likelihood. Therefore, at the beginning of each iteration, conditional expectation of the complete log-likelihood is taken in order to integrate out the latent parts.

Suppose that the current value of $\boldsymbol{\theta}$ after p iterations of the algorithm is denoted by $\boldsymbol{\theta}^{(p)}$. The next iteration contains the following two steps:

E-Step:

Calculate

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}) = E_{\boldsymbol{\theta}^{(p)}}[\log L^c(\boldsymbol{\theta}; \mathbf{X_c})|\boldsymbol{X}_o].$$

M-Step:

Choose a $\boldsymbol{\theta}^{(p+1)}$ that maximizes $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)})$, that is

$$Q(\boldsymbol{\theta}^{(p+1)}|\boldsymbol{\theta}^{(p)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}), \quad \text{for all } \boldsymbol{\theta}.$$

These two steps are alternated until $\boldsymbol{\theta}^{(p+1)}$ is reasonably close to $\boldsymbol{\theta}$. A simple criterion to justify this phenomenon is stop at k with

$$|L(\boldsymbol{\theta}^{(k+1)}; \mathbf{X_o}) - L(\boldsymbol{\theta}^{(k)}; \mathbf{X_o})| < \varepsilon,$$

where ε is a given positive small number.

The EM-algorithm can also be used in a Bayesian framework. With simple modification, it can be applied to find the posterior mode which is called MAP (Maximum A Posteriori) estimator. Assume that the prior of $\boldsymbol{\theta}$ is given by $\pi(\boldsymbol{\theta})$. Let $f(\boldsymbol{\theta}|\mathbf{X_c})$ and $f(\boldsymbol{\theta}|\mathbf{X_o})$ denote the complete- and observed-data posterior densities, respectively. Then

$$\log f(\boldsymbol{\theta}|\mathbf{X_o}) \propto \log L(\boldsymbol{\theta}; \mathbf{X_o}) + \log \pi(\boldsymbol{\theta}),$$

so the MAP estimator is the maximizer of $\log L(\boldsymbol{\theta}; \mathbf{X_o}) + \log \pi(\boldsymbol{\theta})$. Therefore the EM-algorithm can be implemented as follows:

E-Step:

Calculate

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}) = E_{\boldsymbol{\theta}^{(p)}}[\log L^c(\boldsymbol{\theta}; \mathbf{X_c})|\boldsymbol{X}_o],$$

which is the same as the one above.

M-Step:

Choose a $\boldsymbol{\theta}^{(p+1)}$ that maximizes $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}) + \log \pi(\boldsymbol{\theta})$, that is

$$Q(\boldsymbol{\theta}^{(p+1)}|\boldsymbol{\theta}^{(p)}) + \log \pi(\boldsymbol{\theta}^{(p+1)}) \geq Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(p)}) + \log \pi(\boldsymbol{\theta}), \quad \text{for all } \boldsymbol{\theta}.$$

The reader is referred to McLachlan and Krishnan (1997) for a comprehensive summary of EM-algorithm and its extensions.

2.4 Bootstrap Method

Since Efron published the seminar paper on bootstrap methods in 1979, this simulation-based statistical method has been wildly used in finding the variance of an estimator as well as the confidence interval of an unknown parameter. In a simple setting, the bootstrap method usually provides a more accurate result than the classical normal approximation. cf. Hall (1992), and Efron and Tibshirani (1993).

In a typical estimation problem, it is nature to ask how accurate of the estimator is? To answer this question, we need to figure out the sampling distribution of the estimator (or some moments of the sampling distribution). Usually the sampling distribution can not be obtained exactly. The classical way is to approximate the sampling distribution by its asymptotic distribution, usually based on normal approximation. However, when the sample size is small to moderate, the asymptotic normal approximation does not work well. Furthermore, when the estimator is a complicated function of samples, such as sample median, it is a tedious job to find an asymptotic sampling distribution. The bootstrap method provides an alternative solution to these problems.

The basic concept of bootstrap methods is the *plug in principle.* That is, we draw samples from the estimated population, and they are called the bootstrap samples. Then we use the outcomes to estimate the sampling distribution of the estimator of interest. Assume that we have sample $\boldsymbol{X} = (X_1, \cdots, X_n)$ from the population $F(\theta)$, and we are interested in estimating a function of $F(\theta)$, which may be the mean, median, or variance of the distribution. The estimator is denoted by $t(\boldsymbol{X})$. There are typically two different ways to generate the bootstrap samples, parametric bootstrap and nonparametric bootstrap.

In the parametric bootstrap, we estimate the unknown parameter θ first, mostly by the MLE. Then, we approximate the sampling distribution by plug in the MLE $\widehat{\theta}$ into the assumed population, which is $F(\widehat{\theta})$. The approximation then can be done by using Monte Carlo simulation. That is, B bootstrap samples with size n, $\boldsymbol{X}^{*b} = (X_1^{*b}, \cdots, X_n^{*b})$, $b = 1, \cdots, B$, are drawn from $F(\widehat{\theta})$ afterwards. For each $\boldsymbol{X}^{*b}$, the corresponding estimator $t(\boldsymbol{X}^{*b})$ is calculated. Then, we apply these B outcomes, $t(\boldsymbol{X}_1^*), \cdots, t(\boldsymbol{X}_B^*)$, to approximate the sampling distribution of $t(\boldsymbol{X})$. The variance of the estimator, confidence interval, and other related accuracy judgment criteria can easily be obtained from the approximated sampling distribution of $t(\boldsymbol{X})$.

For the nonparametric bootstrap, instead of estimating the parameter θ and plug in $F(\theta)$, we estimate it by the empirical cumulative distribution function (cdf) based on

$\boldsymbol{X} = (X_1, \cdots, X_n)$. That is, we use

$$\widehat{F}(X) = \frac{\sum_{i=1}^{n} I(X_i < X)}{n}$$

to approximate the population distribution. Then we draw random samples with size n from the empirical cdf via the mechanism of simple random sampling without replacement from $\boldsymbol{X} = (X_1, \cdots, X_n)$. Since there are C_n^{2n+1} different samples for given n, it is not efficient to get all the samples comprehensively when n is large. Instead, the Monte Carlo methods can be used to do the simulation. Typically, we choose B bootstrap samples with size n, $\boldsymbol{X}^{*b} = (X_1^{*b}, \cdots, X_n^{*b})$, and then follow the same pattern as that in the parametric bootstrap.

2.5 Markov Chain Monte Carlo Simulation

One important factor in Bayesian analysis is to compute (or approximate) the posterior distribution

$$f(\boldsymbol{\theta}|\boldsymbol{X}) = \frac{f(\boldsymbol{\theta})f(\boldsymbol{X}|\boldsymbol{\theta})}{\int f(\boldsymbol{\theta})f(\boldsymbol{X}|\boldsymbol{\theta})d\boldsymbol{\theta}},$$

where $f(\boldsymbol{\theta})$ is the prior distribution of $\boldsymbol{\theta}$ and $f(\boldsymbol{X}|\boldsymbol{\theta})$ is the likelihood function. Although the kernel $f(\boldsymbol{\theta})f(\boldsymbol{X}|\boldsymbol{\theta})$ can be easily derived, the computation of the normalizing constant $\int f(\boldsymbol{\theta})f(\boldsymbol{X}|\boldsymbol{\theta})d\boldsymbol{\theta}$ involving high-dimensional integrals which is always a tedious task. Markov Chain Monte Carlo (MCMC) simulation provides an approximation to the posterior distribution.

Suppose we are interested in sampling from a certain distribution, say $\pi(.)$. Sometimes it is not easy to sample from $\pi(.)$ directly if the distribution is not standard, like normal or exponential. Some independent sampling methods, such as importance sampling, can be used to overcome this problem. Instead of using independent sampling, Markov Chain Monte Carlo (MCMC) goes from another direction by using dependent samples which are from a Markov Chain.

If we have a Markov chain whose stationary distribution is the distribution of interest $\pi(.)$, after a sufficiently long burn-in runs, we should get a distribution which is approximately the same as $\pi(.)$. It is well-known that if the initial state X_0 of a Markov chain is from the stationary distribution, then all the following samples will follow the stationary distribution as well. Hence, after a sufficient long sampling, we may assume that the Markov chain reaches the stationary distribution, and the following samples also come from the stationary distribution.

Instead of sampling from the given transition probability of a given Markov chain, MCMC is a simulation method sampling from a given dynamic. Two important ways which are used to construct the Markov chain are the Metropolis-Hastings algorithm (cf. Metropolis et al., 1953, and Hastings, 1970) and the Gibbs sampler (cf. Geman and Geman, 1984).

The Metropolis-Hastings algorithm starts with X_0, which can be a vector. Assume the current state is X_t, then the next point is generated from the following procedure:

1. Sample a candidate, Y, from a proposal distribution, $q(.|X_t)$.

2. Generate a random number, U, from $U(0, 1)$.

3. Set X_{t+1} to be Y if $U < \alpha(X_t, Y)$; otherwise, keep the chain at X_t, where

$$\alpha(X_t, Y) = min(1, \frac{\pi(Y)q(X_t|Y)}{\pi(X_t)q(Y|X_t)}).$$

The Gibbs sampler is used when the full conditional distributions (the conditional distributions given all the other arguments) can be derived easily and are easy to sample from. Assume that $\pi(.)$ is with n arguments, $X_1, \cdots, X_n$ (the arguments can contain more than one component). The Gibbs sampler divides each iteration into n sub-iterations. In each sub-iteration, only one argument changes from the current state to the one which is sampled from the full conditional distribution given the other arguments in the latest state. That is starting the Gibbs chain at $(X_{1,0}, \cdots, X_{n,0})$. At $(X_{1,t}, \cdots, X_{n,t})$, the state moves into the next state according to the following procedure:

1. Sample the $X_{1,t+1}$ from $\pi(X_1|X_{2,t}, \cdots, X_{n,t})$.

$$\vdots$$

i. Sample the $X_{i,t+1}$ from $\pi(X_i|X_{1,t+1}, \cdots, X_{i-1,t+1}, X_{i+1,t}, \cdots, X_{n,t})$.

$$\vdots$$

n. Sample the $X_{n,t+1}$ from $\pi(X_n|X_{1,t+1}, \cdots, X_{n-1,t})$.

MCMC can be used to approximate the posterior distribution (cf. Gelfand and Smith, 1990, and Tierney, 1994) in Bayesian analysis. Further quantities such as posterior mean and posterior median, can be obtained easily from the approximated posterior distribution. The reader is referred to Gilk et al. (1996) for a compressive summary.