基于韦伯分布参数估计对芯片失效率进行预测

Zhizhao Niu, *North & West China Team*

摘要

韦伯分布（Weibull Distribution）作为一种灵活且广泛应用的寿命分布模型，在可靠性工程与产品寿命预测中应用非常广泛。韦伯分布能够有效描述多种失效模式，其形状参数与尺度参数具备明确的物理意义，适用于对芯片、机械部件等在不同应力条件下的失效行为进行预测。尤其在芯片可靠性评估中，韦伯分布凭借对高删失数据良好的拟合能力与成熟的参数估计方法，成为预测产品在特定里程或时间点失效率的常用工具。虽然在各个工程领域有很多基于韦伯分布的软件工具，但是随着对于产品系统复杂度的提高，用户需要对韦伯分布的底层数学模型有更深刻的理解才能更正确的使用工具，确保输出的结果是可信的。本文将以某类车规芯片为例，帮助读者更好地理解韦伯分布的底层数学模型以及运行逻辑。文中以假设的芯片验证数据为数据基础，结合失效与右删失样本（多数器件在观测时未失效），构建似然函数并采用最大似然估计法（MLE）标定韦伯分布参数 β 与 η ，进而实现对芯片在关键里程点（如 50,000 km）的失效数目预测，并为产品质量评估与保修策略制定提供量化依据。

内容

1 引言

在利用统计建模预测产品可靠性时，韦伯分布因其适用范围广、参数可解释，且对删失数据具备成熟的极大似然估计与推断流程，因而成为工程实践中的常用选择。 基于上述优势，本文采用韦伯分布对产品寿命/可靠性进行建模与预测：在失效与删失样本基础上构建对数似然并估计 β ，η ，随后给出联合置信域以及关键里程/时间点的 PPM 预测，以支持产品质量评估与保修政策制定。

2 背景介绍

2.1 问题背景与建模假设

假设某批次车规芯片已装载于量产车辆，客户希望通过该批芯片的早期失效样件数据去预测该批产品在整车行驶 50,000 km 时的可靠性表现和失效率。现场数据来源于在役车辆的里程，存在大量右删失样本（多数器件在观测时未失效），并可能伴随里程分布不均、极小里程记录等数据特征，这要求模型在高删失、机理不完全已知的条件下仍能稳定估计并具备可解释性。

2.2 韦伯分布的概率密度函数（PDF）、累积分布函数（CDF）及其含义

本文中使用的韦伯分布为两参数韦伯分布，其具有两个参数：形状参数 β 与尺度参数 η。两参数韦伯分布的概率密度函数（PDF）为：

$$f(t;\beta,\eta)=\frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1}exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{1}$$

累积分布函数（CDF）为

$$F(t;\beta,\eta)=1-exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{2}$$

在本文的问题背景下, f(t；β，η) 与 F(t；β，η) 中的 t 指的是某车辆的已行驶距离；概率密度函数 f(t；β，η) 表示的是在里程 t 附近失效的发生密度，当 Δt 很小时，失效概率可以近似为 f(t；β，η)Δt；累积分布函数 F(t) 指的是在行驶里程为 t 时，累计已经失效的比例。

2.3 数据集处理

本文以车规芯片为例来解释韦伯分布和最大似然估计的原理。为了对未来失效率和失效里程进行预测，需要将原始数据进行收集分类输入到模型中。

将所有样本分为失效样本和未失效样本并收集每一个样本的实际里程数。但是在实际的工程应用中，失效样本的数据比较容易得到，例如车规芯片的失效里程数。但是对于未失效样品，每一个样品的实际数据散差很大，并且数据量非常的。通常对于未失效的样品可以做适当的假设，例如在本文中对未失效芯片我们假设：

1. 每颗芯片自发货起的有效使用时间占发货后总时间的 5%，且有效使用时间期间的平均行驶速度为 30 km/h。
2. 对于未使用就已损坏的芯片，我们假设其故障发生里程为 1km 而非 0km 以保证使用对数最大似然估计后的数据收敛性。

经以上处理后，该批次 1,000,000 颗芯片的里程数据如下：540,000 颗芯片到 4000 km 时未失效；459,990 颗芯片到 10000 km 时未失效；在 1km 芯片时失效 5 颗芯片；在 10km、100km、200km、400km、500km 时分别失效 1 颗芯片。记未失效的芯片的里程数为 pi ，已失效的芯片的里程数为 qj ，即

$$p_1=\cdots=p_{540000}=4000, p_{540001}=\cdots=p_{999990}=10000 \tag{3}$$

和

$$q_1=\cdots=q_5=1, q_6=10, q_7=100, q_8=200, q_9=400, q_{10}=500 \tag{4}$$

3 极大似然参数估计

3.1 极大似然参数估计的原理

极大似然估计的核心想法是把"模型参数"视为未知量，选择一组参数使得已观测到的事件/数据在该模型下发生的概率最大，也就是说，我们需要找到一组参数 (β , η) 使得下列事件的发生概率最大:

1. 540,000 颗芯片到 4000 km 时未失效；
2. 459,990 颗芯片到 10000 km 时未失效；
3. 5 颗在 1km 芯片时失效；
4. 在 10km、100km、200km、400km、500km 时分别失效 1 颗芯片；

根据在累积分布函数的定义，*t* km 时仍未失效的芯片对这一事件发生的概率的贡献为：

$$1 - F(t;\beta,\eta) = exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{5}$$

在 *t* km 时已失效的芯片对这一事件发生的概率的贡献为:

$$f(t;\beta,\eta) = \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta - 1} exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] \tag{6}$$

也就是说，我们需要找到一组参数 (β , η) 使得函数:

$$L(\beta,\eta) = \{\prod_{i=1}^{999990}(1 - F(p_i;\beta,\eta))\} \cdot \{\prod_{j=1}^{10} f\left(q_j;\beta,\eta\right)\} \tag{7}$$

的值最大。

在求解这一问题的过程中，1-F(p_i; β , η) 与 f(q_j; β , η) 都是(0,1)之间的数，若直接求解较易因为大量小于 1 的数相乘而迅速趋近 0，产生下溢，使得目标函数被数值截断为 0，从而使计算不收敛。故我们选择最大化函数 log L(β , η)，其中 *log* 表示以自然对数 *e* 为底的对数函数以保证计算收敛和简化运算。

3.2 引入联合置信域

通过最大化似然函数 L(β , η)，我们找到了若干组参数 (β ˆ, η ˆ) 使得 L(β , η) 最大，但仍有很多组参数 (β , η) 使得似然函数 L(β , η) 很大，甚至接近 L(β ˆ, η ˆ)，因此我们需要引入联合置信域以解决估计中可能出现的误差。

3.3 计算结果

通过运行 3.4 节中的代码，我们得到这批芯片服从的参数 β ˆ ≈0.1585, η ˆ ≈2.137·10^35。利用点估计得到在 50,000 km 时约有 13.92 个芯片失效，利用联合置信域计算出在 50,000 km 时芯片失效数量有 90% 可能性落在 [6.56,25.23] 这一区间中。

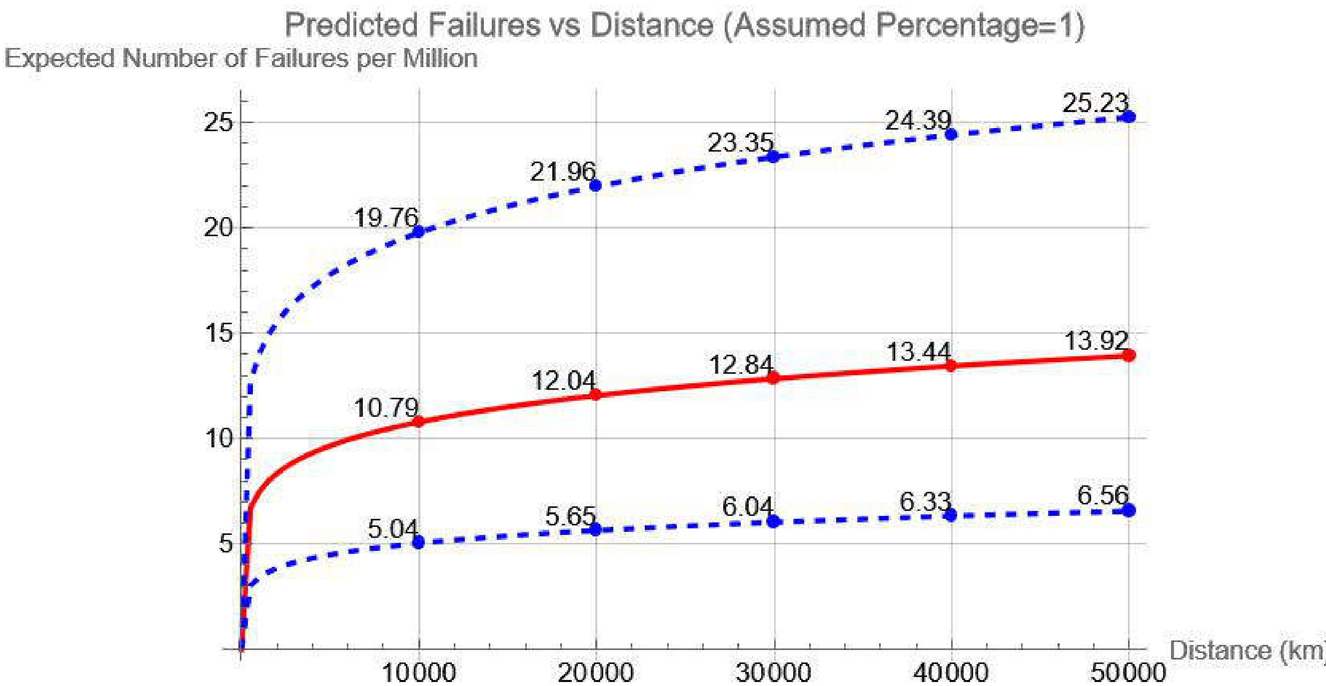

图 **3-1. 90%**联合置信域芯片失效率韦伯分布结果

3.4 代码实现与结果

在本节中，我们将使用 Mathematica，通过定义数值稳定的韦伯分布 CDF 以构造对数似然，随后用 Nelder–Mead 方法极大化得到 (β^, η^)。在点估计确定后，在 (β, η) 网格上筛选满足条件的参数对，以得到 90% 联合置信域；最后计算出所需要的 50000km 时的产品可靠性。代码和注释如下所示：

```
Clear[f]; (*初始化*)
rawFailures =
  Join[ConstantArray[1, 5], {10, 100, 200, 400, 500}]; (*定义损坏数据集*)
nTotalObserved = 1000000;(*定义总数据数*)
```

```
(*防止溢出版韦伯分布累积分布函数(CDF),beta, eta 为服从的参数, d 为行驶距离*)
safeProb[beta_?NumericQ, eta_?NumericQ, d_?NumericQ] :=
  Module[{ex}, If[d <= 0, Return[0.]];
   ex = beta*(Log[d] - Log[eta]); (*避免溢出*)
   If[ex < -800, 0.,
    If[ex > 800, 1.,
     1. - Exp[-Exp[
         ex]]]]]];
```

(*定义主函数 f, 输入 AssumedPercent, 返回在"未损坏器件里程数为 2.3 节中假设里程数乘以 AssumedPercent"这个前提下的相关数据*)

```
f[AssumedPercent_?NumericQ] :=
 Module[{failures, r, nCensored, logLik, resMLE, betaHat, etaHat,
   logLmax, chi2p90, threshold, betaVals, etaVals, jointCIpoints,
   distances, pMLE, pLower, pUpper, plotObj, p50, expectedPerMillion,
   predCI},(*定义参变量*)
  failures = rawFailures;
  r = Length[failures];
  nCensored = nTotalObserved - r;
  logLik[beta_?NumericQ, eta_?NumericQ] := Module[{llF, llC},
    llF =
     Total[Log[beta/eta] + (beta - 1) Log[failures/eta] - (failures/eta)^ beta]; (*失效器件的对数似然和
*)
    llC = -459990*(10000*AssumedPercent/eta)^beta -
      540000*(4000*AssumedPercent/eta)^beta; (*未失效器件的对数似然和*)
    llF + llC];(*总对数似然*)
```

```
(*最大似然估计，优化 logLik*)
  resMLE = NMaximize[
    {logLik[beta, eta], beta > 0 && eta > 0},(*给定参数合理范围 beta>0,eta>0*)
    {beta, eta},
    Method -> "NelderMead",
    MaxIterations -> 20000,
    WorkingPrecision -> 16
    ];
  {betaHat, etaHat} = {beta, eta} /. Last[resMLE]; (*提取估计值 betaHat, etaHat *)
  logLmax = First[resMLE];(*对应的最大 log-likelihood 值*)
```

```
 (*90% 联合置信区间阈值计算*)
  chi2p90 = Quantile[ChiSquareDistribution[2], 0.9];(*卡方分布估计两个参数, 分位点 90%*)
  threshold = logLmax - 0.5*chi2p90;(*似然比检验阈值*)

  (*定义搜索网格*)
  betaVals = Subdivide[0.01, Max[2, 2*betaHat], 800];
  etaVals = Subdivide[Max[1, 0.1*etaHat], 10*etaHat, 800];

  (*找到满足联合 CI 条件的参数点 beta, eta*)
  jointCIpoints = Select[
    Flatten[Table[{b, e}, {b, betaVals}, {e, etaVals}], 1],
    logLik @@ # >= threshold &(*保留 logLik 大于等于阈值的点*)
    ];
```

```
 (*距离采样点*)
  distances = Range[0, 50000, 500];

  (*用 MLE 参数计算预测曲线*)
  pMLE = Table[safeProb[betaHat, etaHat, d]*10^6, {d, distances}];

  (*用联合 CI 的参数点求上下界包络*)
  If[jointCIpoints === {},
   (*如果没有找到联合 CI 点, 返回 Indeterminate*)
   pLower = ConstantArray[Indeterminate, Length[distances]];
   pUpper = ConstantArray[Indeterminate, Length[distances]],

   (*否则计算联合 CI 下的下限和上限*)
   pLower =
    Table[Min[
      safeProb[#[[1]], #[[2]], d]*10^6 & /@ jointCIpoints], {d,
      distances}];
   pUpper =
    Table[Max[
      safeProb[#[[1]], #[[2]], d]*10^6 & /@ jointCIpoints], {d,
      distances}];];
```

```
   (*计算 10000,20000,30000,40000,50000km 时三条预测曲线上的失效数*)
  highlightDistances = {10000, 20000, 30000, 40000, 50000};
  highlightMLE =
   Table[{d, safeProb[betaHat, etaHat, d]*10^6}, {d,
     highlightDistances}];
```

```
highlightLower =
  Table[{d,
    Min[safeProb[#[[1]], #[[2]], d]*10^6 & /@ jointCIpoints]}, {d,
    highlightDistances}];
highlightUpper =
  Table[{d,
    Max[safeProb[#[[1]], #[[2]], d]*10^6 & /@ jointCIpoints]}, {d,
    highlightDistances}];
(*绘图*)
plotObj =
  ListLinePlot[{Transpose[{distances, pMLE}],
    Transpose[{distances, pLower}], Transpose[{distances, pUpper}]},
   PlotStyle -> {Directive[Red, Thick], Directive[Blue, Dashed],
     Directive[Blue, Dashed]}, PlotRange -> All,
   AxesLabel -> {"Distance (km)",
     "Expected Number of Failures per Million"},
   PlotLabel ->
    StringForm[
     "Predicted Failures vs Distance (Assumed Percent=`1`%)",
     100  AssumedPercent], GridLines -> Automatic,
   ImageSize -> {500, 400},
   Epilog -> (Flatten[{(*MLE 曲线标注*)
       Table[{Red, PointSize[Medium], Point[pt], Black,
         Text[NumberForm[pt[[2]], {6, 2}], pt, {1, -1}]}, {pt,
         highlightMLE}],(*联合置信域下限标注*)
       Table[{Blue, PointSize[Medium], Point[pt], Black,
         Text[NumberForm[pt[[2]], {6, 2}], pt, {1, -1}]}, {pt,
         highlightLower}],(*联合置信域上限标注*)
       Table[{Blue, PointSize[Medium], Point[pt], Black,
         Text[NumberForm[pt[[2]], {6, 2}], pt, {1, -1}]}, {pt,
         highlightUpper}]}])];
```

```
p50 = safeProb[betaHat, etaHat, 50000]; (*50000km 的预测概率*)
expectedPerMillion = 10^6*p50;(*转换为每百万台预期失效数*)

(*50,000 km 的联合置信域范围*)
predCI =
  If[jointCIpoints === {},
   "NotFound", {Min[
     safeProb[#[[1]], #[[2]], 50000]*10^6 & /@ jointCIpoints],
    Max[safeProb[#[[1]], #[[2]], 50000]*10^6 & /@ jointCIpoints]}];
(*返回结果，其中 Tc 表示 AssumedPercent(即给定的输入)，即假定没坏的车都已经开了已交货小时数
*0.05*30*AssumedPercent 千米;
betaHat 表示计算出的服从的韦伯分布的 beta 值;
etaHat 表示计算出的服从的韦伯分布的 eta 值;
jointCIpointsCount 为搜索网格中在联合置信域中的点对数量;
expectedPerMillion_at50000_MLE 为使用点预测计算的 50000km 时的失效总数;
expectedPerMillion_at50000_JointCI 为在 90%联合置信区间下的 50000km 时的失效数.
*)
<|"Tc" -> AssumedPercent, "betaHat" -> N[betaHat, 10],
 "etaHat" -> N[etaHat, 10],
 "jointCIpointsCount" -> Length[jointCIpoints],
 "expectedPerMillion_at50000_MLE" -> N[expectedPerMillion, 8],
 "expectedPerMillion_at50000_JointCI" -> predCI,
 "PredictionPlot" -> plotObj|>]
```

4 结论

本文基于芯片的假设验证数据，采用韦伯分布对车规芯片的可靠性进行建模，通过极大似然估计方法求得形状参数与尺度参数，并构建 90%联合置信域以量化估计不确定性。结果表明，该方法能够有效处理高删失、小样本失效的工程数据，实现对特定里程点（如 50,000 km）失效数量的稳定预测。本文虽然是以芯片的历史数据为例，对韦伯分布和最大似然估计的原理进行讲解分析，但是原理也同样适用于其他的电子系统或产品，为芯片可靠性评估与保修策略制定提供了实用工具。

5 参考文献

1. Scholz, F. W. (2015). Inference for the Weibull Distribution: A Tutorial. Quantitative Methods for Psychology, 11(3), 148 – 165.
2. Almalki, S. J., & Nadarajah, S. (2014). Modifications of the Weibull Distribution: A Review. Reliability Engineering & System Safety, 124, 32 – 55.

重要通知和免责声明