File: /General/MLAB-Text/Papers/survival/survival.tex

An MLAB Example: Maximum Likelihood Survival Curve Estimation

Gary D. Knott, Ph.D.
Civilized Software, Inc.
12109 Heritage Park Circle
Silver Spring, MD 20906 USA
Tel. (301) 962-3711
Email: csi@civilized.com

Let $X_1, \ldots, X_k$ be independent positive random variables representing the survival times of k subjects. Each subject has an associated set of n covariate values which serve to categorize that subject. Let $x_i = (x_{i1}, \ldots, x_{in})$ be the vector of covariate values for subject i. The distribution of the survival time X_i is postulated to depend on the covariate values for subject i.

Let $F(t;x) = P[$a subject with covariate values $x = (x_1, \ldots, x_n)$ has a failure time $\leq t]$. The function F maps R_{n+1} to R.

The associated density function is $f(t;x) := dF(t;x)/dt$, and the associated survival function is $S(t;x) := 1 - F(t;x)$. The associated hazard function is $h(t;x) := f(t;x)/S(t;x)$. Note that $log(h(t;x)) = log(f(t;x)) - log(S(t;x))$.

Let $C_1, \ldots, C_k$ be independent identically distributed positive random variables. C_i represents the length of time beyond which subject i is not observed. If $C_i < X_i$ then subject i is lost to follow-up, that is the survival time of subject i is unknown; we know only that subject i survived *at least* C_i years. Let $Y_i =$min(X_i, C_i), and suppose we observe values (*i.e.* samples) $y_1, \ldots, y_n$ of $Y_1, \ldots, Y_k$. When $C_i < X_i$, we say that the value y_i is a *censored* observation. We will define the indicator code $z_i = 0$ when y_i is a censored observation and otherwise z_i will be defined to be 1.

Now suppose we are given the data:

$\underline{subject\#}$	$\underline{covariates}$	$\underline{time}$	$\underline{censor-code}$
1	$x_{11}, \ldots, x_{1n}$	y_1	z_1
$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	$x_{k1}, \ldots, x_{kn}$	y_k	z_k

Thus subject j with the covariate values $x_{j1}, \ldots, x_{jn}$ has the time to failure equal to y_j when $z_j = 1$, or alternately was lost to follow-up after time y_j when $z_j = 0$. Given this data, our goal is to construct a descriptive model, *i.e.* an estimate, for the underlying distribution function F.

We may form the likelihood function for this data as follows:

$$L = \prod_{\{i|z_i=1\}} f(y_i; x_i) \times \prod_{\{i|z_i=0\}} S(y_i; x_i)$$

The corresponding log-likelihood function can be seen to be

$$G = \sum_{i=1}^{k} log(S(y_i; x_i)) + z_i \ log(h(y_i; x_i)).$$

Now suppose that F depends upon some parameters $a = (a_1, a_2, \ldots, a_n)$ and $b = (b_1, b_2, \ldots, b_n)$. Then f, S and h also depend on the vectors a and b, and the log-likelihood function G is a function of a and b alone. Thus

$$G(a, b) = \sum_{i=1}^{k} log(S(y_i; x_i; a, b)) + z_i \ log(h(y_i; x_i; a, b)).$$

If we postulate a specific form for F (and hence for S, f, h and G), we can estimate the parameters $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$ by choosing a and b to maximize $G(a, b)$. This provides a potentially-enlightening descriptive model for our given data.

Piantadosi has proposed the model

$$h(t; x; a, b) = \frac{\beta(x, b)}{\alpha(x, a) + S(t; x; a, b)}$$

where

$$\alpha(x, a) = exp(a_1 x_1 + \ldots + a_n x_n)$$
$$\beta(x, a) = exp(b_1 x_1 + \ldots + b_n x_n).$$

This model defines F implicitly. In particular S satisfies the differential equation

$$\frac{dS}{dt}(t; x; a, b) = \frac{-\beta(x, b) S(t; x; a, b)}{\alpha(x, a) + S(t; x; a, b)}$$

with $S(0; x; a, b) = 1$.

Since $S(t; x; a, b) \cdot h(t; x; a, b) = dF(t; x; a, b)/dt$, this differential equation corresponds to the algebraic relationship:

$$\alpha(x, a) log(S(t; x; a, b)) + S(t; x; a, b) = 1 - \beta(x, b)t,$$

and since $\alpha(x, a) > 0$, $\beta(x, b) > 0$, $t \geq 0$, and $S(0; x; a, b) = 1$, there is always a solution for $S(t; x; a, b)$ in $[0, 1]$.

Now our log-likelihood function becomes:

$$G(a, b) = \sum_{i=1}^{k} log(S(y_i; x_i; a, b)) + z_i [log(\beta(x_i, b)) - log(\alpha(x_i, a) + S(y_i; x_i; a, b))].$$

In order to estimate a and b for this model via maximizing the log-likelihood, we may use the *maximize* functional in *MLAB*. An example of this for $n = 3$, where the parameters become $a_1, a_2, a_3, b_1, b_2, b_3$, is given below for the following data. Here x_1 is fixed equal to 1 for all subjects in order to introduce constant terms in $log(\alpha(x, a))$ and $log(\beta(x, b))$; and $x_{i2} = 1$ when subject i had treatment 1 and $x_{i3} = 1$ when subject i had treatment 2. The observed survival time for subject i was truncated (censored) if $z_i = 0$.

subject#	$\underline{x_1}$	$\underline{x_2}$	$\underline{x_3}$	$\underline{y}$	$\underline{z}$
1	1	1	0	2.3	0
2	1	0	1	4	1
3	1	0	1	5	1
4	1	0	1	2.2	1
5	1	1	0	6	0
6	1	0	0	3.6	1
7	1	1	0	4.1	1
8	1	1	0	2	1
9	1	1	0	1.5	1
10	1	0	1	3	0
11	1	0	1	2.5	1
12	1	0	1	0.8	1
13	1	1	0	0.9	1
14	1	0	1	1.1	1
15	1	1	0	1.4	0
16	1	0	1	1.9	1
17	1	1	0	2.3	0

Here is the *MLAB* dialog that uses the survival model in Piantadosi and Crowley Biometrics (in press).

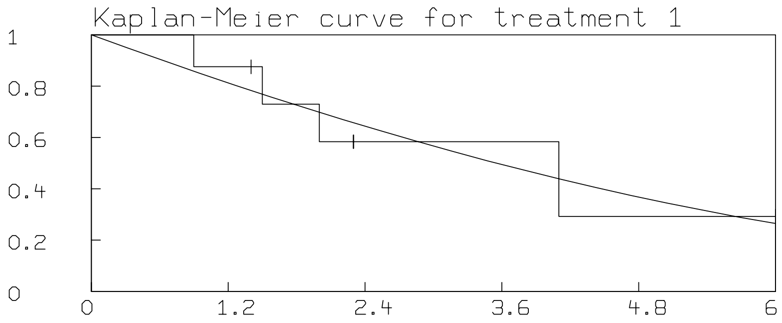
```
"log-likelihood function"
fct g() = sum(i,1,k,gt(alpha(i),betav(i), i))
fct gt(av,bv,i) = gs(logs(av,bv*t[i]/av), z[i],av,bv)
fct gs(ls,z,av,bv) = ls + z * (log(bv) - log(av+exp(ls)))
fct logs(av,ub) = root(w,0,ub,av*w+exp(w-ub)-1)-ub
fct alpha(i) = exp(a1*x[i,1]+a2*x[i,2]+a3*x[i,3])
fct betav(i) = exp(b1*x[i,1]+b2*x[i,2]+b3*x[i,3])

/* surv1, surv2 = survival function for treatment 1 and treatment 2 groups */
fct av1(s) = max(1e-30,exp(a1 + s*a2 + (1-s)*a3))
fct bv1(s) = max(1e-30,exp(b1 + s*b2 + (1-s)*b3))
fct surv(t,s) = exp(logs(av1(s),bv1(s)*t/av1(s)))
fct surv1(t) = surv(t,1);
fct surv2(t) = surv(t,0);
```

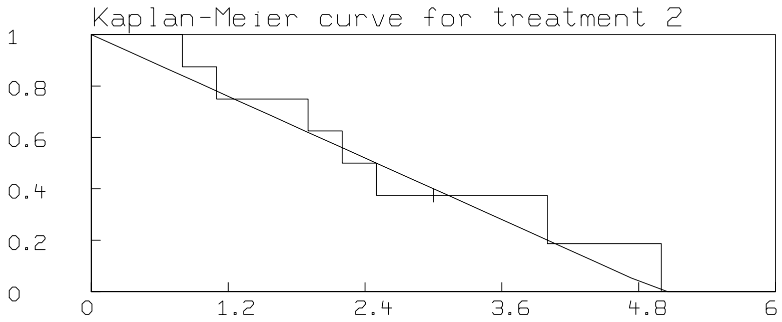
```
n = 3; "# of covariates "
k = 17; "sample size"

data = read(datafile,k,n+2)

x = data col 1:n
t = data col (n+1)
z = data col (n+2)

/* establish initial guesses */
a1 = 0; a2 = 0; a3 = 0; b1 = -2; b2 = -2; b3 = -2;

Hessmsw = 0; /* starting with identity Hessian */
maximize(g,b3,b2,b1,a3,a2,a1)

The function value is: -2.389855e+01
Argument(s): (-3.284358e-01 -7.916285e-02 -1.281012e+00
              -2.723759e+00 1.824017e+01 -1.877267e+01 )
Gradient: (3.302230e-01 4.906215e-03 1.718221e-01
          -4.880071e-05 -1.584679e-03 -1.723857e-03 )
# of function evaluations: 242
# of gradient evaluations: 99
# of Quasi-Newton iterations: 94
    = -23.8985504
```

We will use the *MLAB* function *KMSURV* to compute and plot the two *Kaplan-Meier* survival curves and compare them with the estimated survival curves for treatment 1 and treadment 2.

```
/* draw the Kaplan-Meier curve and surv1 for treatment 1 in w1. */
d1 = compress(data,2); /* data for first treatment */
d = (d1 col 4) &' (d1 col 5)

d = sort(sort(d,2,-1),1)
"double sorting makes censored events occur in front of failure events if
 they happen at the same time."
h1 = kmsurv(d); "Column (1,2) of h1 is the Kaplan-Meier survival curve for d"

h = stepgraph(h1 col (1,2))
r = (0 &' 1) & h & (h[nrows(h),1] &' 0); "the graph starts at (0,1)"
draw r, color red; "Draw the step-graph of the Kaplan-Meier curve"

"draw censored data tic-marks"
y1 = compress(d,2,1) col 1; "y1 = the list of censoring times"
fct f(x) = lookup(h,x)
```

```
draw points(f, y1) lt none, pt vbar, ptsize .015, color green
top title "Kaplan-Meier curve for treatment 1"
frame 0 to 1, .5 to 1
"draw the estimated survival curve for treatment 1"
draw points(surv1,0:6!20)
w1 = w

/* draw the Kaplan-Meier curve for treatment 2 in w2. */
d2 = compress(data,3); /* data for second treatment */
d = (d2 col 4) &' (d2 col 5)

d = sort(sort(d,2,-1),1)
h1 = kmsurv(d); "Column (1,2) of h1 is the Kaplan-Meier survival curve for d"

h = stepgraph(h1 col (1,2))
r = (0 &' 1) & h & (h[nrows(h),1] &' 0); "the graph starts at (0,1)"
draw r, color red; "Draw the step-graph of the Kaplan-Meier curve"

"draw censored data tic-marks"
y1 = compress(d,2,1) col 1; "y1 = the list of censoring times"
fct f(x) = lookup(h,x)

draw points(f, y1), lt none, pt vbar, ptsize .015, color green
top title "Kaplan-Meier curve for treatment 2"
frame 0 to 1, 0 to .5
draw points(surv2,0:6!20)
w2 = w
view
```





Here is the entire surface plot for the function surv.

```
M = cross(0:6!17, 0:1!13)
M col 3 = surv on M
del w1, w2
draw M lt hidden
view
```