THE MAXIMUM LIKELIHOOD ESTIMATOR IN COMPETING RISKS WITH INTERVAL CENSORING

BRAM KUIJVENHOVEN

Abstract. We discuss the maximum likelihood estimator (MLE) for competing risk problems under various forms of interval censoring. The number of intervals is allowed to be random and to be different for different observations. It is shown that the MLE is uniquely determined. Moreover, a general algorithm is developed to compute the MLE. This is an extension of work reported in Maathuis (2005).

1. Model

Let $X \in [0, \infty)$ denote the failure time and $Y \in \{1, \ldots, K\}$ denote the failure cause in the competing risks problem. $K \geq 1$ is the number of possible causes of failure, and is not stochastic.

We can't observe X and Y directly. Instead we observe the failure state at stochastic times $T_1, \ldots, T_C$, with the stochastic variable $C \geq 1$ the number of observation times and $0 \leq T_1 < \ldots < T_C$. We assume the pair (X, Y) to be independent from $(C, T_1, \ldots, T_C)$. For convenience we define $T_0 = -\infty$ and $T_{C+1} = \infty$. The failure state S_i at each T_i, for $i = 1, \ldots, C$, tells whether failure has already occurred ($X \leq T_i$), and only in that case also the failure cause (Y).

Note that if $C = 2$ is a constant, we are dealing with what is called Interval Censoring, case 2 in Groeneboom and Wellner (1992). If $C = 1$ is a constant, we are dealing with the Current Status Data problem discussed in Maathuis (2005).

Now we define I, U and V by

$$U = T_{I-1} < X \leq T_I = V.$$

Note that I is uniquely defined in this way and can be directly observed from the failure states at $T_1, \ldots, T_C$. It takes values in the range $1 \ldots C + 1$ and indicates the index of the first observation time at which a failure was observed. We know the failure cause if and only if $I \leq C$. Next we define the $K + 1$ element vector Δ by

$$\begin{aligned} \Delta_k &= 1_{\{I \leq C, Y = k\}} \quad \text{for } k = 1, \ldots, K \\ \Delta_{K+1} &= 1_{\{I > C\}}. \end{aligned}$$

Note that Δ can also be directly observed from the failure states.

Let us now define one observation by $Z = (C, T_1, \ldots, T_C, S_1, \ldots, S_C)$ and let $Z_1, \ldots, Z_n$ be a sample of observations, where we write $Z_i = (C_i, T_{i,1}, \ldots, T_{i,C_i}, S_{i,1}, \ldots, S_{i,C_i})$, $i = 1, \ldots, n$.

Date: September 11, 2005.

2000 *Mathematics Subject Classification.* Primary: 60C05, 60K35; Secondary: 60F05.

Key words and phrases. Competing risk, nonparametric maximum likelihood estimator, interval censoring, Fenchel conditions, convex minorants .

Next, let $T'_{(1)}, \ldots, T'_{(p)}$, $1 \leq p \leq 2n$, denote the order statistics of the (non-empty) set

$$\bigcup_{i=1}^{n} \{U_i, V_i\} \setminus \{-\infty, \infty\},$$

so we have

$$-\infty < T'_{(1)} < \ldots < T'_{(p)} < \infty.$$

These will be our times of interest.

Now define

$$\begin{aligned} N_{k,i} &= \sum_{l=1}^{n} \Delta_{k,l} 1_{\left\{U_l = -\infty, V_l = T'_{(i)}\right\}} \quad \text{for } k = 1, \ldots, K,\ i = 1, \ldots, p \\ N_{k,i,j} &= \sum_{l=1}^{n} \Delta_{k,l} 1_{\left\{U_l = T'_{(i)}, V_l = T'_{(j)}\right\}} \quad \text{for } k = 1, \ldots, K,\ 1 \leq i < j \leq p \\ N_i &= \sum_{l=1}^{n} \Delta_{K+1,l} 1_{\left\{U_l = T'_{(i)}, V_l = \infty\right\}} \quad \text{for } i = 1, \ldots, p. \end{aligned}$$

Note that

- $N_{k,i}$ counts observations l where $X_l \leq T_{l,1} = T'_{(i)}$ and $Y_l = k$,
- $N_{k,i,j}$ counts observations l where $T'_{(i)} = T_{l,I_l - 1} < X_l \leq T_{l,I_l} = T'_{(j)}$ and $Y_l = k$ and
- N_i counts observations l where $T'_{(i)} = T_{l,C_l} < X_l$.

In this way, we count every observation once, so we have

$$\sum_{i=1}^{p} \left[\sum_{k=1}^{K} \left[N_{k,i} + \sum_{j=i+1}^{p} N_{k,i,j} \right] + N_i \right] = n. \tag{1.1}$$

2. The Maximum Likelihood Estimator

2.1. **Derivation of the MLE.** For this problem, we are interested in the sub-distribution functions

$$F_k(t) = P(X \leq t, Y = k) \quad \text{for } k = 1, \ldots, K.$$

Let us denote

$$F_{k,i} = F_k\left(T'_{(i)}\right) \quad \text{for } k = 1, \ldots, K,\ i = 1, \ldots, p$$

and

$$F_{+,i} = \sum_{k=1}^{K} F_{k,i} \quad \text{for } i = 1, \ldots, p.$$

The likelihood[1] for the sub-distribution functions $F = (F_1, \ldots, F_K)$ now is given by

$$\prod_{i=1}^{p} \left[\prod_{k=1}^{K} \left[F_{k,i}^{N_{k,i}} \prod_{j=i+1}^{p} \left(F_{k,j} - F_{k,i}\right)^{N_{k,i,j}} \right] \left(1 - F_{+,i}\right)^{N_i} \right] \tag{2.1}$$

[1]Actually this is a *conditional* likelihood, conditioned on the observation times $\{(C_i, T_{i,1}, \ldots, T_{i,C_i})\}_{i=1}^{n}$. We use the assumption that the failure times $X_1, \ldots, X_n$ are independent of these observation times, so that the conditional maximum likelihood estimator still is a good estimator.

and the corresponding log-likelihood is

$$l(F) = \sum_{i=1}^{p} \left[\sum_{k=1}^{K} \left[N_{k,i} \ln F_{k,i} + \sum_{j=i+1}^{p} N_{k,i,j} \ln \left(F_{k,j} - F_{k,i} \right) \right] + N_i \ln \left(1 - F_{+,i} \right) \right]. \tag{2.2}$$

The maximum likelihood estimator (MLE) $\hat{F} = \left(\hat{F}_1, \ldots, \hat{F}_K \right)$ for the sub-distributions is given by the maximisation problem

$$l\left(\hat{F} \right) = \max_{F \in \mathcal{F}_K} l\left(F \right) \tag{2.3}$$

with

$$\mathcal{F}_K = \left\{ F \in \mathbb{R}^{pK} : 0 \leq F_{k,1} \leq \ldots \leq F_{k,p}, k = 1, \ldots, K, F_{+,p} \leq 1 \right\}. \tag{2.4}$$

2.2. **Uniqueness of the MLE.** We are interested to know for which pairs k and i the maximum likelihood estimator $\hat{F}_{k,i}$ is uniquely determined. It is easy to see that such $\hat{F}_{k,i}$ must appear somewhere in the log-likelihood. This motivates the definition for $k = 1, \ldots, K$ of the set

$$\mathcal{I}_k = \left\{ i \in \{1, \ldots, p\} : N_i + N_{k,i} + \sum_{j=i+1}^{p} N_{k,i,j} + \sum_{j=1}^{i-1} N_{k,j,i} > 0 \right\}. \tag{2.5}$$

and the set of index pairs

$$\mathcal{I} = \left\{ (k, i) : i \in \mathcal{I}_k, k \in \{1, \ldots, K\} \right\}. \tag{2.6}$$

This is, for given k, the set of indices i for which$F_{k,i}$ actually appears in the log-likelihood $l(F)$. Later, we will also use the following notation

$$m_k = \left| \mathcal{I}_k \right|, k = 1, \ldots, K, \quad \text{and} \quad m = \sum_{k=1}^{K} m_k \tag{2.7}$$

and for $k = 1, \ldots, K$ the ordered indices

$$i_{k,1} < \cdots < i_{k,m_k} \text{ such that } \mathcal{I}_k = \left\{ i_{k,j} \right\}_{j=1}^{m_k}. \tag{2.8}$$

Theorem 2.1. *The maximum likelihood estimator $\hat{F}_{k,i}$ is uniquely determined for all $(k, i) \in \mathcal{I}$.*

Proof. First note that the natural logarithm is a strictly concave function[2]. Also note that the log-likelihood l is the sum of terms that are all concave[3].

Now let $\hat{F}$ and $\hat{G}$ be such that

$$l\left(\hat{F} \right) = l\left(\hat{G} \right) = \max_{F \in \mathcal{F}_K} l\left(F \right).$$

So we need to prove that $\hat{F}_{k,i} = \hat{G}_{k,i}$ for all $(k, i) \in \mathcal{I}$. Let

$$\hat{H} = \frac{1}{2}\hat{F} + \frac{1}{2}\hat{G}$$

[2]A function f is STRICTLY CONCAVE when $f\left(\lambda x + (1 - \lambda) y \right) > \lambda f\left(x \right) + (1 - \lambda) f\left(y \right)$ for all x, y in the domain of f and all $\lambda \in (0, 1)$.

[3]A function f is CONCAVE when $f\left(\lambda x + (1 - \lambda) y \right) \geq \lambda f\left(x \right) + (1 - \lambda) f\left(y \right)$ for all x, y in the domain of f and all $\lambda \in (0, 1)$ (or, equivalently, for all $\lambda \in [0, 1]$).

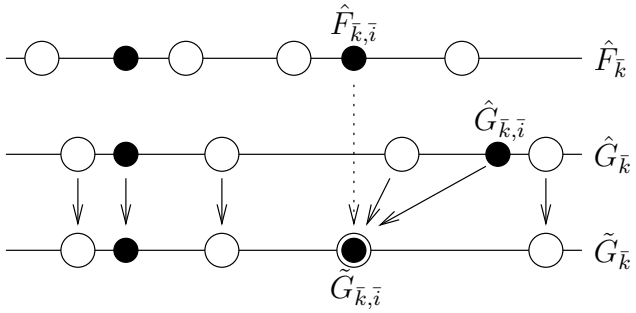

FIGURE 2.1. Graphical representation of $\hat{G}_{\bar{k}}$, $\hat{F}_{\bar{k}}$ and $\tilde{G}_{\bar{k}}$. Each horizontal line represents the interval $[0,1]$ and each dot indicates a value $F_{\bar{k},i}$, $i = 1,\ldots,p$. Black dots have $i \in \mathcal{I}_{\bar{k}}$ and white dots $i \notin \mathcal{I}_{\bar{k}}$. The arrows indicate how $\tilde{G}_{\bar{k}}$ is constructed.

be a convex combination of $\hat{F}$ and $\hat{G}$. Then also

$$\hat{H}_{k,i} - \hat{H}_{k,j} = \frac{1}{2}\left(\hat{F}_{k,i} - \hat{F}_{k,j}\right) + \frac{1}{2}\left(\hat{G}_{k,i} - \hat{G}_{k,j}\right)$$

is a convex combination of $\hat{F}_{k,i} - \hat{F}_{k,j}$ and $\hat{G}_{k,i} - \hat{G}_{k,j}$, and

$$1 - \hat{H}_{+,i} = \frac{1}{2}\left(1 - \hat{F}_{+,i}\right) + \frac{1}{2}\left(1 - \hat{G}_{+,i}\right)$$

is a convex combination of $1 - \hat{F}_{+,i}$ and $1 - \hat{G}_{+,i}$.

This shows that

- (I) for k and i with $N_{k,i} > 0$ we must have $\hat{F}_{k,i} = \hat{G}_{k,i}$ and
- (II) for k, i and j with $N_{k,i,j} > 0$ we must have $\hat{F}_{k,j} - \hat{F}_{k,i} = \hat{G}_{k,j} - \hat{G}_{k,i}$ and
- (III) for i with $N_i > 0$ we must have $1 - \hat{F}_{+,i} = 1 - \hat{G}_{+,i}$,

because otherwise

- the concavity of each of the terms l consists of, combined with
- the *strict* concavity of the logarithm

would give

$$l\left(\hat{H}\right) > \frac{1}{2}l\left(\hat{F}\right) + \frac{1}{2}l\left(\hat{G}\right) = l\left(\hat{F}\right) = l\left(\hat{G}\right)$$

which would contradict the assumption that l has a maximum in $\hat{F}$ (and $\hat{G}$).

Now assume there exists a k and an$i \in \mathcal{I}_k$ with $\hat{F}_{k,i} \neq \hat{G}_{k,i}$. Take one such k fixed and name it $\bar{k}$. Denote the *smallesti* $\in \mathcal{I}_{\bar{k}}$ for which $\hat{F}_{\bar{k},i} \neq \hat{G}_{\bar{k},i}$ with $\bar{i}$. Now assume that $\hat{F}_{\bar{k},\bar{i}} < \hat{G}_{\bar{k},\bar{i}}$. We can now make a $\tilde{G}$ from $\hat{G}$ by defining for $k = 1,\ldots,K$ and $i = 1,\ldots,p$

$$\tilde{G}_{k,i} = \begin{cases} \min\left\{\hat{G}_{k,i}, \hat{F}_{\bar{k},\bar{i}}\right\} & \text{if } k = \bar{k} \text{ and } i \leq \bar{i} \\ \hat{G}_{k,i} & \text{if } k \neq \bar{k} \text{ or } i > \bar{i} \end{cases}.$$

See figure2.1 for a graphical representation.

Note that the only pair of indices $(k,i) \in \mathcal{I}$ where $\hat{G}_{k,i}$ and $\tilde{G}_{k,i}$ differ is $\left(\bar{k},\bar{i}\right)$, because

- if $k \neq \bar{k}$ or $i > \bar{i}$, $\tilde{G}_{k,i} = \hat{G}_{k,i}$ by the definition of $\tilde{G}$,
- if $k = \bar{k}$ and $i = \bar{i}$ then indeed $\tilde{G}_{k,i} = \hat{F}_{\bar{k},\bar{i}} < \hat{G}_{k,i}$ and

- if $k = \bar{k}$ and $i < \bar{i}$ with $i \in \mathcal{I}_k$ then we have$\hat{G}_{k,i} = \hat{F}_{k,i}$ by the definition of $\bar{k}$ and $\bar{i}$, and since$\hat{F}_{k,i} < \hat{F}_{\bar{k},\bar{i}}$, we have $\tilde{G}_{k,i} = \min\left\{\hat{G}_{k,i}, \hat{F}_{\bar{k},\bar{i}}\right\} = \hat{G}_{k,i}$.

Also note that $\tilde{G} \in \mathcal{F}_K$, because

(1) the monotonicity of the $\hat{G}_{k,i}$ for a certain k is preserved in the order of the $\tilde{G}_{k,i}$ and
(2) $\tilde{G}_{k,i} \leq \hat{G}_{k,i}$ for all k and i, so $\tilde{G}_{+,i} \leq \hat{G}_{+,i} \leq 1$ for all k.

Next note that we must have $N_{\bar{k},\bar{i}} = 0$, because of (I). Also for all$j < \bar{i}$ we must have $N_{\bar{k},j,\bar{i}} = 0$ because otherwise property (II) would give $\hat{F}_{\bar{k},j} - \hat{G}_{\bar{k},j} = \hat{F}_{\bar{k},\bar{i}} - \hat{G}_{\bar{k},\bar{i}} \neq 0$ and j would be a smaller index i in $\mathcal{I}_{\bar{k}}$ than $\bar{i}$ with $\hat{F}_{\bar{k},i} \neq \hat{G}_{\bar{k},i}$, contradicting our definition of $\bar{i}$ to be the smallest index in $\mathcal{I}_{\bar{k}}$ with this property.

But then $F_{\bar{k},\bar{i}}$ can only appear in the log-likelihood $l(F)$ in the terms $N_{\bar{k},\bar{i},j} \ln\left(F_{\bar{k},j} - F_{\bar{k},\bar{i}}\right)$ with $j > \bar{i}$ and the term $N_{\bar{i}} \ln\left(1 - F_{+,\bar{i}}\right)$. Also, at least one of these terms must have a non-zero coefficient because of the definition of $\mathcal{I}_k$. If we look carefully at these terms, we see that l is strictly decreasing in the parameter $F_{\bar{k},\bar{i}}$. Combining this with the facts that the only pair of indices (k,i) with $i \in \mathcal{I}_k$ where $\hat{G}_{k,i}$ and $\tilde{G}_{k,i}$ differ is $\left(\bar{k},\bar{i}\right)$ and that $\tilde{G}_{\bar{k},\bar{i}} < \hat{G}_{\bar{k},\bar{i}}$, we get

$$l\left(\tilde{G}\right) > l\left(\hat{G}\right),$$

a contradiction. So our assumption that $\hat{F}_{\bar{k},\bar{i}} < \hat{G}_{\bar{k},\bar{i}}$ is wrong.

A similar reasoning could be held for the assumption that $\hat{F}_{\bar{k},\bar{i}} > \hat{G}_{\bar{k},\bar{i}}$ by exchanging the roles of $\hat{F}$ and $\hat{G}$. Then we would find a $\tilde{F}$ with

$$l\left(\tilde{F}\right) > l\left(\hat{F}\right),$$

another contradiction.

But then it would follow that $\hat{F}_{\bar{k},\bar{i}} = \hat{G}_{\bar{k},\bar{i}}$, which in turn contradicts our definition of $\bar{k}$ and $\bar{i}$. We conclude that our assumption that there exists a $(k,i) \in \mathcal{I}$ with $\hat{F}_{k,i} \neq \hat{G}_{k,i}$ is wrong. This proves the theorem. □

3. Some Theory

In this section, we will introduce some optimization theory that we will use to compute the MLE introduced in the previous section.

3.1. **Convex optimization.**

We start with the definition of a convex optimization problem. See for example De Klerk, Roos and Terlaky (2003) for more information on convex optimization.

Definition 3.1. A convex optimization problem is of the form

$$\begin{array}{lll} \text{(CO)} & \min & f(x) \\ & \text{s.t.} & g_j(x) \leq 0,\ j = 1,\ldots,m \\ & & x \in \mathcal{C} \end{array}$$

where f and g_j for $j = 1,\ldots,m$ are convex functions and $\mathcal{C} \subset \mathbb{R}^n$ is a convex set. The function f is called the objective function and the requirements $g_j(x) \leq 0$ are called constraints.

We define the feasible set and a feasible solution as follows.

Definition 3.2. The FEASIBLE SET of (CO) is defined by

$$\mathcal{F} := \{x \in \mathcal{C} : g_j(x) \leq 0, j = 1, \ldots, m\}$$

and every $x \in \mathcal{F}$ is called a FEASIBLE SOLUTION of (CO).

The following definition of an optimal solution is quite trivial

Definition 3.3. A vector $\bar{x} \in \mathcal{F}$ is called an OPTIMAL SOLUTION of (CO) if

$$f(\bar{x}) \leq f(x) \text{ for all } x \in \mathcal{F}.$$

Our goal is to get rid of the constraints $g_j(x) \leq 0$ in (CO). We start by defining the so called Lagrangian.

Definition 3.4. The LAGRANGIAN associated with (CO) is defined by

$$L(x, y) := f(x) + \sum_{j=1}^{m} y_j g_j(x)$$

where $y \in \mathbb{R}^m$. The variable y_j is called a LAGRANGE MULTIPLIER for the constraint $g_j(x) \leq 0$.

In order to derive the most powerful theorems, we need to introduce a so called regularity condition on our optimization problem. Before introducing the so called Slater regularity condition, we will first introduce the relative interior.

Definition 3.5. The RELATIVE INTERIOR $\mathcal{C}^0$ of a convex set $\mathcal{C}$ consists of all points $x \in \mathcal{C}$ such that for all $\bar{x} \in \mathcal{C}$ there exists an $\tilde{x} \in \mathcal{C}$ and a $\lambda \in (0, 1)$ such that $x = \lambda\bar{x} + (1 - \lambda)\tilde{x}$.

Note that the relative interior of $\mathbb{R}^n$ is $\mathbb{R}^n$ itself. The relative interior differs from the interior for example when the affine hull of the set is not of the same dimension as the space the set is in. An example of this is $\{(x, y) \in \mathbb{R}^2 : y = 0\}$, whose interior is empty, but whose relative interior is the set itself.

Definition 3.6. The problem (CO) satisfies the SLATER CONDITION if there exists an $x^0 \in \mathcal{C}^0$ such that

$$g_j(x^0) < 0 \quad \text{for all } j \text{ where } g_j \text{ is nonlinear}$$
$$g_j(x^0) \leq 0 \quad \text{for all } j \text{ where } g_j \text{ is linear}$$

Note that this Slater condition is easily satisfied for problem with only linear constraints. In that case, it simply reduces to finding a feasible solution in the relative interior of $\mathcal{C}$. This can always be done if $\mathcal{C}$ is non-empty, according to the following theorem:

Theorem 3.7. *If a convex set $\mathcal{C}$ is non-empty, then its relative interior $\mathcal{C}^0$ is non-empty as well.*

We now present the following important result from optimization theory.

Theorem 3.8. *Given a problem (CO) that satisfies the Slater condition, let $\bar{x} \in \mathcal{F}$ be a feasible vector and L the Lagrangian. Then $\bar{x}$ is an optimal solution of (CO) if and only if there exists a vector $\bar{y} \geq 0$ such that*

$$(i) \quad \bar{x} = \arg\min_{x \in \mathcal{C}} L(x, \bar{y}) \text{ and}$$
$$(ii) \quad \sum_{j=1}^{m} \bar{y}_j g_j(\bar{x}) = 0.$$

Here we use the notation $y \geq 0$ with $y \in \mathbb{R}^m$, which means that $y_j \geq 0$ for $j = 1, \ldots, m$. Note that the minimization problem in the theorem is over the set $\mathcal{C}$ and not over $\mathcal{F}$.

Note that we can also handle equality constraints in this manner, because

$$\begin{aligned} g(x) = 0 &\Leftrightarrow g(x) \leq 0 \text{ and } g(x) \geq 0 \\ &\Leftrightarrow g(x) \leq 0 \text{ and } -g(x) \leq 0. \end{aligned}$$

So, we replace an equality constraint $g(x) = 0$ by the two inequality constraints derived above. Let μ denote the Lagrange multiplier for the constraint $g(x) \leq 0$ and ν the Lagrange multiplier for the constraint $-g(x) \leq 0$, then we have

$$\mu g(x) + \nu \cdot -g(x) = (\mu - \nu) g(x).$$

We can thus substitute $\mu - \nu$ by one Lagrange multiplier, λ, that we will multiply $g(x)$ by. In theorem 3.8 above, we then let $\lambda \in \mathbb{R}$, because $\mu - \nu$ can become arbitrarily positive or negative.

3.2. **Fenchel optimality.** We will use the so called Fenchel optimality conditions later on, so we define them here. See Robertson, Wright and Dykstra (1998) for more information on Fenchel duality and isotonic regression, which is introduced in the next section. First, we introduce the concept of a convex cone.

Definition 3.9. A convex set $\mathcal{K}$ is called a convex cone if for all $x \in \mathcal{K}$ and all $\lambda \geq 0$ also $\lambda x \in \mathcal{K}$.

Now, the following theorem gives the Fenchel optimality conditions.

Theorem 3.10. *Let $\mathcal{K}$ be a convex cone and $\phi : \mathbb{R}^n \to (-\infty, \infty]$ a convex, continuous function with a unique minimum over $\mathcal{K}$ and that is continuously differentiable on the set where ϕ is finite: $\{x \in \mathbb{R}^n : \phi(x) < \infty\}$. Then*

$$\hat{x} = \arg\min_{x \in \mathcal{K}} \phi(x)$$

if and only if it satisfies the Fenchel optimality conditions

$$\langle \hat{x}, \nabla\phi(\hat{x}) \rangle = 0 \text{ and } \langle x, \nabla\phi(\hat{x}) \rangle \geq 0 \text{ for all } x \in \mathcal{K}.$$

3.3. **Isotonic regression.** A special kind of optimization problem that we will discuss here, is isotonic regression. Here the convex minorants will come into play. We start by defining what isotonic regression is.

Definition 3.11. The isotonic regression problem is defined as

$$\min_{x \in \mathcal{C}} \sum_{i=1}^{n} (x_i - y_i)^2 w_i$$

with $y, w \in \mathbb{R}^n$ given vectors, $w_i > 0$, $i = 1, \ldots, n$, and

$$\mathcal{C} = \{x \in \mathbb{R}^n : 0 \leq x_1 \leq \ldots \leq x_n\} \tag{3.1}$$

the set of increasing finite sequences, bounded below by 0.

The solution of an isotonic regression problem can be calculated by means of a greatest convex minorant. We start by defining the latter.

Definition 3.12. Let $\mathcal{S} \subset \mathbb{R}^{n+1}$ be a set of points and define the set of convex minorants of $\mathcal{S}$ by

$$\mathcal{M} := \{f : \mathbb{R}^n \to \mathbb{R} : f \text{ is convex and } f(x) \leq y \text{ for all } (x, y) \in \mathcal{S}\},$$

then the GREATEST CONVEX MINORANT (GCM) of S is defined by

$$g(x) = \sup_{f \in \mathcal{M}} f(x) \text{ for all } x \in \mathbb{R}^n.$$

Note that the greatest convex minorant is again a convex function, and in case S has a finite number of elements and $n = 2$, the greatest convex minorant is a piecewise linear function where the boundaries of the pieces are at points in $\mathcal{S}$.

We now arrive at the theorem that describes how to solve the isotonic regression problem using greatest convex minorants.

Theorem 3.13. *Consider the isotonic regression problem from definition 3.11. Define the cloud of points $\mathcal{S} = \{P_i : i = 0, \ldots, n\}$ in $\mathbb{R}^2$ by $P_0 = (0, 0)$ and*

$$P_j = \left(\sum_{i=1}^{j} w_i, \sum_{i=1}^{j} y_i w_i \right) \text{ for } j = 1, \ldots, n.$$

Now, $\hat{x}$ is the optimal solution of the isotonic regression problem if and only if for $i = 1, \ldots, n$ $\hat{x}_i$ equals the maximum of 0 and the left derivate of the greatest convex minorant of the cloud $\mathcal{S}$, evaluated at the point P_i.

Finding the greatest convex minorant of a finite cloud of points is related to finding the so called convex hull. In fact, the lower half of the convex hull of the cloud is the greatest convex minorant. Finding a convex hull can be done in $O(n \log n)$ time with for example a divide and conquer algorithm, or in $O(n^2)$ time using a simple wrapping algorithm.

4. Computing The Maximum Likelihood Estimator

In this section we will address the computation of the MLE using Iterative Convex Minorant (ICM) algorithms.

4.1. Adding a Lagrange Multiplier.

The definition of the MLE is given by equation 2.3. This is a maximization problem over the space $\mathcal{F}_K$, as defined in equation 2.4. Note that $\mathcal{F}_K$ is not a convex cone, because of the constraint $F_{+,p} \leq 1$. We want to rewrite our optimization problem in such a way that the space over which we optimize looks more like the one in isotonic regression; as defined in equation 3.1.

We will use the following short hand notations for summations:

$$\sum_k \text{ for } \sum_{k=1}^{K}, \quad \sum_i \text{ for } \sum_{i=1}^{p}, \quad \sum_{j>i} \text{ for } \sum_{j=i+1}^{p}, \quad \sum_{j<i} \text{ for } \sum_{j=1}^{i-1}.$$

Let's first rewrite the problem as a convex optimization problem in a form similar to (CO):

$$\begin{array}{lll} (A) & \min & \phi(F) \\ & \text{s.t.} & F_{+,p} - 1 \leq 0 \\ & & F \in \mathcal{A}_K \end{array}$$

with

$$\begin{aligned}\phi(F) &= -l(F) \\ &= -\sum_k \sum_i N_{k,i} \ln F_{k,i} - \sum_k \sum_i \sum_{j>i} N_{k,i,j} \ln(F_{k,j} - F_{k,i}) \\ &\quad - \sum_i N_i \ln(1 - F_{+,i}), \end{aligned} \tag{4.1}$$

minus the log-likelihood, and

$$\mathcal{A}_K = \left\{ F \in \mathbb{R}^{pK} : 0 \leq F_{k,1} \leq \ldots \leq F_{k,p}, k = 1, \ldots, K \right\} \tag{4.2}$$

the space we are optimizing over.

Note that we don't always need a Lagrange multiplier to get rid of the constraint $F_{+,p} \leq 1$. If $N_p > 0$, the term $-N_p \ln(1 - F_{+,p})$ in ϕ makes sure $F_{+,p} \leq 1$.

We can add a Lagrange multiplier and obtain, for each $\lambda \geq 0$, the problem

$$\begin{array}{lll} (A_\lambda) & \min & \phi_\lambda(F) \\ & \text{s.t.} & F \in \mathcal{A}_K \end{array}$$

with

$$\phi_\lambda(F) = \phi(F) + \lambda(F_{+,p} - 1). \tag{4.3}$$

We could now proceed and apply the Fenchel optimality conditions to this problem. However, it will be more convenient later if we first adapt the problem (A) as follows:

$$\begin{array}{lll} (B) & \min & \psi(F, s) \\ & \text{s.t.} & F_{+,p} + s - 1 = 0 \\ & & (F, s) \in \mathcal{B}_K \end{array}$$

with

$$\begin{aligned}\psi(F, s) &= -\sum_i \sum_k N_{k,i} \ln F_{k,i} - \sum_k \sum_i \sum_{j>i} N_{k,i,j} \ln(F_{k,j} - F_{k,i}) \\ &\quad - \sum_i N_i \ln(F_{+,p} + s - F_{+,i}) \end{aligned} \tag{4.4}$$

and

$$\begin{aligned}\mathcal{B}_K &= \left\{ (F, s) \in \mathbb{R}^{pK+1} : F \in \mathcal{A}_K, s \geq 0 \right\} \\ &= \left\{ (F, s) \in \mathbb{R}^{pK+1} : 0 \leq F_{k,1} \leq \ldots \leq F_{k,p}, k = 1, \ldots, K, s \geq 0 \right\}. \end{aligned} \tag{4.5}$$

Note that we added a variable s in this problem, added the constraints $s \geq 0$ and $F_{+,p} + s = 1$ and replaced the 1 in ϕ by $F_{+,p} + s$. The following theorem defines exactly how the problems (A) and (B) are related.

Theorem 4.1. *Let F_A be an optimal solution of problem (A) and (F_B, s_B) an optimal solution of problem (B). Then F_B is also an optimal solution of problem (A) and $(F_A, 1 - (F_A)_{+,p})$ is also an optimal solution of problem (B).*

Proof. Note that F_B is a feasible solution of (A) and F_A is an optimal solution of (A), so

$$\phi(F_B) \leq \phi(F_A).$$

Similarly, note that $(F_A, 1-(F_A)_{+,p})$ is a feasible solution of (B) and (F_B, s_B) is an optimal solution of (B), so

$$\psi(F_A, 1-(F_A)_{+,p}) \leq \psi\left(F_{B,} s_B\right).$$

Next, note that since F_B is an optimal solution of (B), it is also a feasible solution of (B) and thus satisfies $(F_B)_{+,p} + s_B = 1$, yielding

$$\psi\left(F_B, s_B\right) = \phi\left(F_B\right).$$

Similarly, F_A is a feasible solution of (A) and satisfies

$$\phi\left(F_A\right) = \psi(F_A, 1-(F_A)_{+,p}).$$

Combining these four equations yields

$$\phi\left(F_B\right) \leq \phi\left(F_A\right) = \psi(F_A, 1-(F_A)_{+,p}) \leq \psi\left(F_{B,} s_B\right) = \phi\left(F_B\right),$$

which can only be true if

$$\phi\left(F_B\right) = \phi\left(F_A\right) = \psi(F_A, 1-(F_A)_{+,p}) = \psi\left(F_{B,} s_B\right) = \phi\left(F_B\right),$$

proving the theorem. □

Now, we can add a Lagrange multiplier to (B) and obtain for each $\lambda \in \mathbb{R}$ the related problem

$$\begin{array}{lll} (B_\lambda) & \min & \psi_\lambda\left(F, s\right) \\ & \text{s.t.} & (F, s) \in \mathcal{B}_K \end{array}$$

with the Lagrangian

$$\psi_\lambda\left(F, s\right) = \psi\left(F, s\right) + \lambda\left(F_{+,p} + s - 1\right). \tag{4.6}$$

4.2. **Applying the Fenchel optimality conditions.** We now want to apply the Fenchel optimality conditions from theorem 3.10 to problem (B_λ). Note that this can be done because $\mathcal{B}_K$ is a convex cone. The Fenchel optimality conditions for the optimum $(\hat{F}, \hat{s})$ are:

$$\begin{array}{rcll} \left\langle (\hat{F}, \hat{s}), \nabla\psi_\lambda(\hat{F}, \hat{s}) \right\rangle & = & 0 & \text{and} \\ \left\langle (F, s), \nabla\psi_\lambda(\hat{F}, \hat{s}) \right\rangle & \geq & 0 & \text{for all } (F, s) \in \mathcal{B}_K. \end{array}$$

We need to compute the first order derivatives of ψ_λ. We start with computing the derivatives of ψ with respect to $F_{\bar{k},\bar{i}}$. Note that we need to be careful by considering the two separate cases $\bar{i} < p$ and $\bar{i} = p$, and by noting that

$$\sum_i -N_i \ln\left(F_{+,p} + s - F_{+,i}\right) = \sum_{i<p} -N_i \ln\left(F_{+,p} + s - F_{+,i}\right) + N_p \ln s.$$

Keeping this in our minds, we continue:

$$\begin{aligned}\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\psi(F,s) &= \frac{\partial}{\partial F_{\bar{k},\bar{i}}}\sum_i -N_i\ln(F_{+,p}+s-F_{+,i})\\ &\quad+\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\sum_k\sum_i -N_{k,i}\ln(F_{k,i})\\ &\quad+\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\sum_k\sum_i\sum_{j>i} -N_{k,i,j}\ln(F_{k,j}-F_{k,i})\\ &= 1_{\{\bar{i}<p\}}\frac{N_{\bar{i}}}{F_{+,p}+s-F_{+,\bar{i}}}-1_{\{\bar{i}=p\}}\sum_{i<p}\frac{N_i}{F_{+,p}+s-F_{+,i}}\\ &\quad-\frac{N_{\bar{k},\bar{i}}}{F_{\bar{k},\bar{i}}}+\sum_{j>\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{F_{\bar{k},j}-F_{\bar{k},\bar{i}}}-\sum_{j<\bar{i}}\frac{N_{\bar{k},j,\bar{i}}}{F_{\bar{k},\bar{i}}-F_{\bar{k},j}}.\end{aligned}\tag{4.7}$$

The derivative of ψ_λ with respect to $F_{\bar{k},\bar{i}}$ is now:

$$\begin{aligned}\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\psi_\lambda(F,s) &= \frac{\partial}{\partial F_{\bar{k},\bar{i}}}\left[\psi(F,s)+\lambda(F_{+,p}+s-1)\right]\\ &= 1_{\{\bar{i}<p\}}\frac{N_{\bar{i}}}{F_{+,p}+s-F_{+,\bar{i}}}+1_{\{\bar{i}=p\}}\left(\lambda-\sum_{i<p}\frac{N_i}{F_{+,p}+s-F_{+,i}}\right)\\ &\quad-\frac{N_{\bar{k},\bar{i}}}{F_{\bar{k},\bar{i}}}+\sum_{j>\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{F_{\bar{k},j}-F_{\bar{k},\bar{i}}}-\sum_{j<\bar{i}}\frac{N_{\bar{k},j,\bar{i}}}{F_{\bar{k},\bar{i}}-F_{\bar{k},j}}\end{aligned}\tag{4.8}$$

The derivative of ψ with respect to s is:

$$\begin{aligned}\frac{\partial}{\partial s}\psi(F,s) &= \frac{\partial}{\partial s}\sum_i -N_i\ln(F_{+,p}+s-F_{+,i})\\ &= -\sum_i\frac{N_i}{F_{+,p}+s-F_{+,i}}\end{aligned}\tag{4.9}$$

The derivative of ψ_λ with respect to s is:

$$\begin{aligned}\frac{\partial}{\partial s}\psi_\lambda(F,s) &= \frac{\partial}{\partial s}\left[\psi(F,s)+\lambda(F_{+,p}+s-1)\right]\\ &= -\sum_i\frac{N_i}{F_{+,p}+s-F_{+,i}}+\lambda\end{aligned}\tag{4.10}$$

We will now apply the first Fenchel optimality condition:

$$\sum_{\bar{k}}\sum_{\bar{i}}\hat{F}_{\bar{k},\bar{i}}\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\psi_\lambda\left(\hat{F},\hat{s}\right)+\hat{s}\frac{\partial}{\partial s}\psi_\lambda\left(\hat{F},\hat{s}\right)=0.$$

The terms with $\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\psi_\lambda\left(\hat{F},\hat{s}\right)$ expand to quite a number of new terms, so will handle them one by one. We start with

$$\sum_{\bar{k}}\sum_{\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}\cdot-\frac{N_{\bar{k},\bar{i}}}{\hat{F}_{\bar{k},\bar{i}}}\right]=-\sum_{\bar{k}}\sum_{\bar{i}}N_{\bar{k},\bar{i}}.\tag{4.11}$$

The following two terms can be rewritten as follows, by changing the order of the summations properly and then renaming the summation index variables in a suitable way:

$$\begin{aligned}
&\sum_{\bar{k}}\sum_{\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}\sum_{j>\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{\hat{F}_{\bar{k},j}-\hat{F}_{\bar{k},\bar{i}}}\right]-\sum_{\bar{k}}\sum_{\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}\sum_{j<\bar{i}}\frac{N_{\bar{k},j,\bar{i}}}{\hat{F}_{\bar{k},\bar{i}}-\hat{F}_{\bar{k},j}}\right]\\
=&\sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{\hat{F}_{\bar{k},j}-\hat{F}_{\bar{k},\bar{i}}}\right]-\sum_{\bar{k}}\sum_{j}\sum_{\bar{i}>j}\left[\hat{F}_{\bar{k},\bar{i}}\frac{N_{\bar{k},j,\bar{i}}}{\hat{F}_{\bar{k},\bar{i}}-\hat{F}_{\bar{k},j}}\right]\\
=&\sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{\hat{F}_{\bar{k},j}-\hat{F}_{\bar{k},\bar{i}}}-\hat{F}_{\bar{k},j}\frac{N_{\bar{k},\bar{i},j}}{\hat{F}_{\bar{k},j}-\hat{F}_{\bar{k},\bar{i}}}\right]\\
=&-\sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}}\left[\left(\hat{F}_{\bar{k},j}-\hat{F}_{\bar{k},\bar{i}}\right)\frac{N_{\bar{k},\bar{i},j}}{\hat{F}_{\bar{k},j}-\hat{F}_{\bar{k},\bar{i}}}\right]\\
=&-\sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}}N_{\bar{k},\bar{i},j}. \qquad (4.12)
\end{aligned}$$

Next, we use in the fact that $\sum_k F_{k,i}=F_{+,i}$ by the definition of $F_{+,i}$ for these terms:

$$\begin{aligned}
&\sum_{\bar{k}}\sum_{\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}1_{\{\bar{i}<p\}}\frac{N_{\bar{i}}}{F_{+,p}+s-F_{+,\bar{i}}}\right]+\sum_{\bar{k}}\sum_{\bar{i}}\left[\hat{F}_{\bar{k},\bar{i}}1_{\{\bar{i}=p\}}\left(\lambda-\sum_{i<p}\frac{N_i}{F_{+,p}+s-F_{+,i}}\right)\right]\\
=&\sum_{\bar{k}}\sum_{\bar{i}<p}\left[\hat{F}_{\bar{k},\bar{i}}\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right]+\sum_{\bar{k}}\left[\hat{F}_{\bar{k},p}\left(\lambda-\sum_{i<p}\frac{N_i}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,i}}\right)\right]\\
=&\sum_{\bar{i}<p}\left[\left(\sum_{\bar{k}}\hat{F}_{\bar{k},\bar{i}}\right)\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right]+\left(\sum_{\bar{k}}\hat{F}_{\bar{k},p}\right)\left(\lambda-\sum_{i<p}\frac{N_i}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,i}}\right)\\
=&-\sum_{\bar{i}<p}\left[\left(\hat{F}_{+,p}-\hat{F}_{+,\bar{i}}\right)\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right]+\hat{F}_{+,p}\lambda. \qquad (4.13)
\end{aligned}$$

The last term is the one for $\hat{s}$:

$$\begin{aligned}
&\hat{s}\cdot\left[-\sum_i\frac{N_i}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,i}}+\lambda\right]\\
=&-\sum_{\bar{i}}\left[\hat{s}\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right]+\hat{s}\lambda\\
=&-\sum_{\bar{i}<p}\left[\hat{s}\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right]+N_p+\hat{s}\lambda. \qquad (4.14)
\end{aligned}$$

We can now sum up equations 4.11 to 4.14, and use equation 1.1 to get rid of a lot of terms:

$$\begin{aligned}
0 &= -\sum_{\bar{k}}\sum_{\bar{i}} N_{\bar{k},\bar{i}} - \sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}} N_{\bar{k},\bar{i},j} - \sum_{\bar{i}<p}\left[\left(\hat{F}_{+,p} - \hat{F}_{+,\bar{i}}\right)\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right] + \hat{F}_{+,p}\lambda \\
&\quad - \sum_{\bar{i}<p}\left[\hat{s}\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right] + N_p + \hat{s}\lambda \\
&= -\sum_{\bar{k}}\sum_{\bar{i}} N_{\bar{k},\bar{i}} - \sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}} N_{\bar{k},\bar{i},j} - \sum_{\bar{i}<p}\left[\left(\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}\right)\frac{N_{\bar{i}}}{\hat{F}_{+,p}+\hat{s}-\hat{F}_{+,\bar{i}}}\right] + N_p \\
&\quad + \left(\hat{F}_{+,p}+\hat{s}\right)\lambda \\
&= -\sum_{\bar{k}}\sum_{\bar{i}} N_{\bar{k},\bar{i}} - \sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}} N_{\bar{k},\bar{i},j} - \sum_{\bar{i}<p} N_{\bar{i}} + N_p + \left(\hat{F}_{+,p}+\hat{s}\right)\lambda \\
&= -n + \left(\hat{F}_{+,p}+\hat{s}\right)\lambda
\end{aligned}$$

So we find this very nice equation for an optimal solution $(\hat{F},\hat{s})$ of problem (B_λ):

$$n = \left(\hat{F}_{+,p}+\hat{s}\right)\lambda. \tag{4.15}$$

So choosing $\lambda = n$ yields an optimal $(\hat{F},\hat{s}) \in \mathcal{B}_K$ for problem (B_λ) with $\hat{F}_{+,p}+\hat{s} = 1$. Then $(\hat{F},\hat{s})$ is also a feasible solution for problem (B). Note that the Slater condition from definition 3.6 is satisfied, because the constraint $F_{+,p}+s-1=0$ is linear. Now we can apply theorem 3.8, because we have found a feasible $(\hat{F},\hat{s})$ and a $\hat{\lambda}$ such that

(1) $(\hat{F},\hat{s})$ minimizes the Lagrangian $\psi_{\hat{\lambda}}(F,s)$ and
(2) $\hat{\lambda}(\hat{F}_{+,p}+\hat{s}-1)=0$,

so $(\hat{F},\hat{s})$ is an optimal solution of problem(B).

We could also have applied the Fenchel optimality conditions to problem (A_λ). The space $\mathcal{A}_K$ is also a convex cone. The derivatives of ϕ_λ are quite similar to those of ψ_λ, but in fact even simpler. We find for the derivatives of ϕ (compare this to equation 4.7):

$$\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\phi(F) = \frac{N_{\bar{i}}}{1-F_{+,\bar{i}}} - \frac{N_{\bar{k},\bar{i}}}{F_{\bar{k},\bar{i}}} + \sum_{j>\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{F_{\bar{k},j}-F_{\bar{k},\bar{i}}} - \sum_{j<\bar{i}}\frac{N_{\bar{k},j,\bar{i}}}{F_{\bar{k},\bar{i}}-F_{\bar{k},j}}. \tag{4.16}$$

For the derivatives of ϕ_λ we find (compare this to equation 4.8)

$$\frac{\partial}{\partial F_{\bar{k},\bar{i}}}\phi_\lambda(F) = \frac{N_{\bar{i}}}{1-F_{+,\bar{i}}} - \frac{N_{\bar{k},\bar{i}}}{F_{\bar{k},\bar{i}}} + \sum_{j>\bar{i}}\frac{N_{\bar{k},\bar{i},j}}{F_{\bar{k},j}-F_{\bar{k},\bar{i}}} - \sum_{j<\bar{i}}\frac{N_{\bar{k},j,\bar{i}}}{F_{\bar{k},\bar{i}}-F_{\bar{k},j}} + 1_{\{\bar{i}=p\}}\lambda. \tag{4.17}$$

The Fenchel optimality conditions for an optimal solution $\hat{F}$ of problem (A_λ) now are

$$\begin{aligned}
\left\langle \hat{F}, \nabla\phi_\lambda(\hat{F})\right\rangle &= 0 \quad \text{and} \\
\left\langle F, \nabla\phi_\lambda(\hat{F})\right\rangle &\geq 0 \quad \text{for all } F \in \mathcal{A}_K.
\end{aligned}$$

The first condition can be calculated in a similar way as we did with problem (B_λ). Using equations 4.11 and 4.12, together with the equations

$$\sum_{\bar{k}}\sum_{\bar{i}}\left[F_{\bar{k},\bar{i}}\cdot\frac{N_{\bar{i}}}{1-F_{+,\bar{i}}}\right] = \sum_{\bar{i}}\left[F_{+,\bar{i}}\cdot\frac{N_{\bar{i}}}{1-F_{+,\bar{i}}}\right] \tag{4.18}$$

$$\sum_{\bar{k}}\sum_{\bar{i}}\left[F_{\bar{k},\bar{i}}\cdot 1_{\{\bar{i}=p\}}\lambda\right] = \lambda F_{+,p}, \tag{4.19}$$

we get

$$\begin{aligned}
0 &= -\sum_{\bar{k}}\sum_{\bar{i}} N_{\bar{k},\bar{i}} - \sum_{\bar{k}}\sum_{\bar{i}}\sum_{j>\bar{i}} N_{\bar{k},\bar{i},j} + \sum_{\bar{i}}\left[\hat{F}_{+,\bar{i}}\cdot\frac{N_{\bar{i}}}{1-\hat{F}_{+,\bar{i}}}\right] + \hat{F}_{+,p}\lambda \\
&= -n + \sum_{\bar{i}} N_{\bar{i}} + \sum_{\bar{i}}\left[\hat{F}_{+,\bar{i}}\cdot\frac{N_{\bar{i}}}{1-\hat{F}_{+,\bar{i}}}\right] + \hat{F}_{+,p}\lambda \\
&= -n + \sum_{\bar{i}}\frac{N_{\bar{i}}}{1-\hat{F}_{+,\bar{i}}} + \hat{F}_{+,p}\lambda.
\end{aligned}$$

So we find

$$n - \sum_i \frac{N_i}{1-\hat{F}_{+,i}} = \hat{F}_{+,p}\lambda \tag{4.20}$$

for an optimal solution $\hat{F}$ of (A_λ). If $N_p = 0$, then we know that $\hat{F}_{+,p} = 1$ for an optimal solution of problem (A). If we would know $\hat{F}_{+,i}$ for all $i < p$, then we could choose

$$\hat{\lambda} = n - \sum_i \frac{N_i}{1-\hat{F}_{+,i}}$$

such that an optimal solution of $\left(A_{\hat{\lambda}}\right)$ is also an optimal solution of (A) by theorem 3.8.

4.3. **Using the ICM algorithm.** We will describe here how to use the Iterative Convex Minorant (ICM) algorithm for computing the MLE. See for example Jongbloed (1998) for a clear description of the ICM algorithm, and Maathuis (2005) for its application in competing risks subject to current status data censoring.

4.3.1. *A suitable optimization problem.* If $N_p > 0$, we know that the term $-N_p \ln(1-F_{+,p})$ in ϕ makes sure the constraint $F_{+,p} \leq 1$ is satisfied in problem (A). If $N_p = 0$ however, we need a Lagrange multiplier as in problem (B_λ). We know we have to take $\lambda = n$ in that case. Also, in an optimal solution we will have $\hat{F}_{+,p} = 1$, and thus $\hat{s} = 0$, because ϕ must be strictly decreasing in $F_{k,p}$ for at least one k and is decreasing in all $F_{k,p}$. So we may also minimize $\psi_n(F,0)$ over $\mathcal{A}_K$ in that case in order to compute the MLE.

Hence, we can define our minimization problem by

$$\begin{aligned}
(C)\quad &\min \quad \varphi(F) \\
&\text{s.t.} \quad F \in \mathcal{C}_K
\end{aligned}$$

with

$$\varphi(F) = \begin{cases} \phi(F) & \text{if } N_p > 0, \\ \psi_n(F,0) & \text{if } N_p = 0, \end{cases}$$

and

$$\mathcal{C}_K = \left\{ F \in \mathbb{R}^m : 0 \leq F_{k,i_{k,1}} \leq \cdots \leq F_{k,i_{k,m_k}}, k = 1, \ldots, K \right\}.$$

(Recall the definitions of m and $i_{k,j}$ in equations 2.7 and 2.8 here.) The MLE is given by the unique solution of this problem. The notation $F \in \mathbb{R}^m$ simply indicates that F only contains the elements $F_{k,i}$ with $(k,i) \in \mathcal{I}$.

4.3.2. *Overview.* The ICM algorithm is an iterative algorithm. It can be seen as an Sequential Quadratic Programming algorithm that only uses the diagonal elements of the Hessian matrix, i.e. only the second order derivatives with respect to the same variable and no mixed second order derivatives.

We start with a fixed estimate $F^{(0)} \in \mathcal{C}_K$, and for each iteration $l = 0, 1, 2, \ldots$ we construct $F^{(l+1)}$ as follows:

- We approximate φ around $F^{(l)}$ by $\varphi^{(l)}$ (using the gradient and Hessian diagonal elements)
- We find the minimum F^{new} of $\varphi^{(l)}$ (using greatest convex minorants)
- We find the best $\alpha > 0$ to set $F^{(l+1)} = F^{(l)} + \alpha \left(F^{\text{new}} - F^{(l)} \right)$ (the so called line search)

A suitable stop condition for the algorithm can be determined by using the Fenchel optimality conditions.

4.3.3. *Approximation.* We need the second order derivatives with respect to the same variable of ϕ and ψ_λ for the ICM algorithm. We find

$$\frac{\partial^2}{\partial F_{\bar{k},\bar{i}}^2} \phi(F) = \frac{N_{\bar{i}}}{\left(1 - F_{+,\bar{i}}\right)^2} + \frac{N_{\bar{k},\bar{i}}}{F_{\bar{k},\bar{i}}^2} + \sum_{j<\bar{i}} \frac{N_{\bar{k},j,\bar{i}}}{\left(F_{\bar{k},\bar{i}} - F_{\bar{k},j}\right)^2} + \sum_{j>\bar{i}} \frac{N_{\bar{k},\bar{i},j}}{\left(F_{\bar{k},j} - F_{\bar{k},\bar{i}}\right)^2}$$

and

$$\begin{aligned} \frac{\partial^2}{\partial F_{\bar{k},\bar{i}}^2} \psi_\lambda(F,s) &= 1_{\{\bar{i}<p\}} \frac{N_{\bar{i}}}{\left(F_{+,p} + s - F_{+,\bar{i}}\right)^2} + 1_{\{\bar{i}=p\}} \sum_{i<p} \frac{N_i}{\left(F_{+,p} + s - F_{+,i}\right)^2} \\ &\quad + \frac{N_{\bar{k},\bar{i}}}{F_{\bar{k},\bar{i}}^2} + \sum_{j<\bar{i}} \frac{N_{\bar{k},j,\bar{i}}}{\left(F_{\bar{k},\bar{i}} - F_{\bar{k},j}\right)^2} + \sum_{j>\bar{i}} \frac{N_{\bar{k},\bar{i},j}}{\left(F_{\bar{k},j} - F_{\bar{k},\bar{i}}\right)^2}. \end{aligned}$$

The approximation of φ around $F^{(l)}$ is given by

$$\begin{aligned} \varphi^{(l)}(F) &= \varphi\left(F^{(l)}\right) + \sum_{k,i} \left(F_{k,i} - F_{k,i}^{(l)}\right) \frac{\partial \varphi}{\partial F_{k,i}}\left(F^{(l)}\right) \\ &\quad + \frac{1}{2} \sum_{k,i} \left(F_{k,i} - F_{k,i}^{(l)}\right)^2 \frac{\partial^2 \varphi}{\partial F_{k,i}^2}\left(F^{(l)}\right) \\ &= \varphi\left(F^{(l)}\right) - \frac{1}{2} \sum_{k,i} \frac{\left(\frac{\partial \varphi}{\partial F_{k,i}}\left(F^{(l)}\right)\right)^2}{\frac{\partial^2 \varphi}{\partial F_{k,i}^2}\left(F^{(l)}\right)} \\ &\quad + \frac{1}{2} \sum_{k,i} \left(F_{k,i} - \left(F_{k,i}^{(l)} - \frac{\frac{\partial \varphi}{\partial F_{k,i}}\left(F^{(l)}\right)}{\frac{\partial^2 \varphi}{\partial F_{k,i}^2}\left(F^{(l)}\right)} \right) \right)^2 \frac{\partial^2 \varphi}{\partial F_{k,i}^2}\left(F^{(l)}\right). \end{aligned}$$

4.3.4. *Minimization.* The next step is the minimization of the approximation $\varphi^{(l)}$. Note that the first two terms in the previous equation are constant terms, so minimizing $\varphi^{(l)}$ over $\mathcal{C}_K$ is equivalent to

$$\min_{F \in \mathcal{C}_K} \frac{1}{2} \sum_{k,i} \left(F_{k,i} - y_{k,i}\right)^2 w_{k,i}$$

with

$$\begin{aligned} y_{k,i} &= F_{k,i}^{(l)} - \frac{\frac{\partial \varphi}{\partial F_{k,i}}\left(F^{(l)}\right)}{\frac{\partial^2 \varphi}{\partial F_{k,i}^2}\left(F^{(l)}\right)} \\ w_{k,i} &= \frac{\partial^2 \varphi}{\partial F_{k,i}^2}\left(F^{(l)}\right). \end{aligned}$$

We can split this problem in K smaller problems because there are no constraints between $F_{k,i}$s with different k in the definition of $\mathcal{C}_K$. So F^{new} should be composed of the solutions to the problems

$$\min_{F_k \in \mathcal{C}_K} \frac{1}{2} \sum_{i} \left(F_{k,i} - y_{k,i}\right) w_{k,i}$$

for $k = 1, \ldots, K$. These are isotonic regression problems, as defined in 3.11. Their solutions are hence given by theorem 3.13. This involves calculating the greatest convex minorant of a cloud op points, hence the name Iterative Convex Minorant algorithm.

4.3.5. *Line search.* The last procedure in each step is the line search. We search for the $\alpha > 0$ such that setting $F^{(l+1)} = F^{(l)} + \alpha\left(F^{\text{new}} - F^{(l)}\right)$ minimizes $\varphi\left(F^{(l+1)}\right)$. There are several algorithms to do this line search.

Since F^{new} and $F^{(l)}$ are both in $\mathcal{C}_K$, we can choose α anywhere in the interval $[0, 1]$. It is also possible to determine the maximum value for α such that $F^{(l+1)} \in \mathcal{C}_K$. When $\varphi = \phi$ because $N_p > 0$, the maximum value of α can also be determined such that $F_{+,p} \leq 1$.

The simplest line search algorithm tries several values for α and picks the one which yields the smallest value for the objective function. We know that $\alpha \mapsto \varphi\left(F^{(l)} + \alpha\left(F^{\text{new}} - F^{(l)}\right)\right)$ is a convex mapping, so we can use a bisection algorithm that tries to find the point where its derivative with respect to α is zero.

4.3.6. *Optimality check.* The Fenchel optimality condition can be used to formulate a suitable stop condition. The basic idea is that we can replace the 0's by ε's. Also, we don't have to

try all $F \in \mathcal{C}_K$, but only a set of vectors that form a kind of 'base' for $\mathcal{C}_K$. These vectors are

$$\begin{aligned}
e_{1,i_{1,1}} &= (\underbrace{1,1,\ldots,1,1}_{m_1 \text{ elements}}, \underbrace{0,0,\ldots,0,0}_{m_2 \text{ elements}}, \ldots, \underbrace{0,0,\ldots,0,0}_{m_K \text{ elements}}),\\
e_{1,i_{1,2}} &= (\underbrace{0,1,\ldots,1,1}_{m_1 \text{ elements}}, \underbrace{0,0,\ldots,0,0}_{m_2 \text{ elements}}, \ldots, \underbrace{0,0,\ldots,0,0}_{m_K \text{ elements}}),\\
\vdots \quad &\quad \vdots \qquad\qquad\qquad\qquad \vdots\\
e_{1,i_{1,m_1}} &= (\underbrace{0,0,\ldots,0,1}_{m_1 \text{ elements}}, \underbrace{0,0,\ldots,0,0}_{m_2 \text{ elements}}, \ldots, \underbrace{0,0,\ldots,0,0}_{m_K \text{ elements}}),\\
e_{2,i_{2,1}} &= (\underbrace{0,0,\ldots,0,0}_{m_1 \text{ elements}}, \underbrace{1,1,\ldots,1,1}_{m_2 \text{ elements}}, \ldots, \underbrace{0,0,\ldots,0,0}_{m_K \text{ elements}}),\\
\vdots \quad &\quad \vdots \qquad\qquad\qquad\qquad \vdots\\
e_{K,i_{K,m_K}} &= (\underbrace{0,0,\ldots,0,0}_{m_1 \text{ elements}}, \underbrace{0,0,\ldots,0,0}_{m_2 \text{ elements}}, \ldots, \underbrace{0,0,\ldots,0,1}_{m_K \text{ elements}}).
\end{aligned}$$

Any vector $F \in \mathcal{C}_K$ is a linear combination of these vectors with non-negative weights. Furthermore, the sum of these weights equals $\sum_k F_{k,i_{k,m_k}}$. So when

$$\left\langle e_{k,i}, \nabla\varphi(\hat{F}) \right\rangle \geq -\varepsilon \text{ for all } (k,i) \in \mathcal{I}, \tag{4.21}$$

we know that

$$\left\langle F, \nabla\varphi(\hat{F}) \right\rangle \geq -\varepsilon F_{+,p} \text{ for all } F \in \mathcal{C}_K.$$

The first Fenchel optimality condition is changed into

$$\left| \left\langle \hat{F}, \nabla\varphi(\hat{F}) \right\rangle \right| \leq \varepsilon. \tag{4.22}$$

Equations 4.21 and 4.22 intuitively provide suitable conditions to check whether a solution $\hat{F}$ is near an optimal solution.

Note that since the values that φ attains are in the neighbourhood of n, $\nabla\varphi$ also takes values in the neighbourhood of n. Therefore, we would choose ε to be n times an accuracy parameter for the algorithm, such that the same value for the accuracyparameter can be used for inputs with different n.

Acknowledgements

I would like to thank Marloes Maathuis for her help and assistance. She also gave me hints about improving this article. My supervisors for the project in the context of which I wrote this article are Piet Groeneboom and Rik Lopuhaä from the Delft University of Technology.

Appendix A. Implementation Example

I've written an implementation of the ICM algorithm, applied to the Competing Risks Interval Censoring (CRIC) problem, in C++. Also, I have written some MatLab routines for creating random samples and displaying the results of the C++ program graphically.

In this appendix, I will shortly describe the code and how to use it. Many details about the implementation can be found in comments in the code itself, so I will only give an overview here to help you getting started.

The program sources can be downloaded from `http://www.kuijvenhoven.net/comprisk.zip`. If you have any questions or comments, please email me at bram at kuijvenhoven dot net.

A.1. **The `comprisk` program.** In the zip file you'll find a subdirectory named `CompetingRisk`. It contains the C++ source files (`.cpp`) and header files (`.h`) of the `comprisk` program, accompanied by a simple `Makefile` file.

For our convenience I have included an executable called `comprisk.exe` for the Windows platform. For the other platforms, you'll have to compile the sources yourself, but I assume that won't be a problem. Note that you might want to change the line `EXE=comprisk.exe` into `EXE=comprisk` for Unix-based platforms.

The ICM algorithm is implemented in the `CICMSolver` class, defined in the `icm.h` header file. Its most important method is called `Solve`. Other methods include `FenchelOptimality` (implements the procedure explained in section 4.3.6), `GreatestConvexMinorant` (which finds the GCM of a given set of points) and `LineSearch` (which implements a bisection line search algorithm, see section 4.3.5).

Abstract methods from `CICMSolver` are overriden in the `CCRICSolver` descendant, defined in the `cric.h` header file. These abstract methods include `Phi`, `GradPhi` and `HessianDiagPhi` for the calculation of the object function φ and its derivatives, and `InitialEstimate`, which calculates an initial estimate $F^{(0)}$. In this way, the logic for the general ICM algorithm and the specifics for the CRIC problem are kept in separate places.

`CCRICSolver` also has the very important public method `SetSample`, which initialises the internal tables of the object using a given CRIC sample. These internal tables have an optimised format such that cases with large K or many duplicate observation times are handled efficient as well. The code also takes care it only calculates the $\hat{F}_{k,i}$ for (k,i) in the uniqueness set $\mathcal{I}$ (the `fDistSize` and `fDistIndex` members come into play here).

The `comprisk.cpp` file provides a command line interface to the solver classes. Please run `comprisk.exe -h` for help on usage. (An example of calling and using comprisk can be found in the MatLab file `example.m`.) The program uses getopt.h, which is a very simple GNU getopt replacement that deals with command line parameters. It can only deal with so called 'short options' (of the form `-a [arg]`), but that is sufficient in this case.

A.2. **The MatLab routines.** There is also a bunch of `.m` files distributed in the zip file. These are MatLab files. Most of them deal with the generation of random CRIC samples. Some can be used to process the output of the `comprisk` program. Most files should have some comments at the start of the file explaining its purpose and use, so you can use the `help` command from the MatLab console (e.g. `help randcricsample`).

The `example.m` script gives a nice example of how to use the MatLab routines. This script will generate some random samples, call `comprisk` on it and finally display the results. When calling it the second time, it won't re-generate the samples, because this takes quite some time. You can remove the `examplen.sample.txt` files, so it will generate them again. The resulting graph displays the actual distribution in black, and several estimates, derived from several kinds of observations, in other colors.

Note that you need to call `comprisk` with the `-f` option to get full output, with information about the tables at each iteration etc. The `example.m` script does do this. I won't give a complete reference here, as you probably need to play around a bit with the code anyway to

get the graphs you want or to work with data sets you have in your favorite format. To get started, I recommend the following steps:

- Make sure the Current Directory of MatLab is at the path where you extracted the .m files
- Make sure you have a compiled `comprisk` executable in the same directory
- Run from the MatLab Command Window the command `example`. (Note: this might take some time.) A graph will appear as described above.
- You can run `showicm(s(1))` to `showicm(s(4))` to see the convergence process. (Click in the graph or press a key while the figure window has focus to see the next frame of the animation each time. Hint: you can hold the space bar pressed for example.)

A.3. **File format.** Here I will give a short description of the file formats the `comprisk` program uses, so you can plug in your own data and read the data back.

A.3.1. *`comprisk` input.* The input of the `comprisk` program should be in the following format:

$$
\begin{array}{l}
n\ K \\
t_{1,1}\ t_{1,2}\ k_{1,1}\ k_{1,2} \\
t_{2,1}\ t_{2,2}\ k_{2,1}\ k_{2,2} \\
\dots \\
t_{n,1}\ t_{n,2}\ k_{n,1}\ k_{n,2}
\end{array}
$$

Here, n and K denote the same things as in this article. The meaning of $t_{i,1}$, $t_{i,2}$, $k_{i,1}$ and $k_{i,2}$ is as described in the following table (recall the definitions from section 1)

I_i	$t_{i,1}$	$t_{i,2}$	$k_{i,1}$	$k_{i,2}$
1	(any)	V_i	-1	Y_i
$2 \dots C_i$	U_i	V_i	0	Y_i
$C_i + 1$	U_i	(any)	0	-1

The table should be read as follows: for each value of I_i, the interpretation of $t_{i,1}$, $t_{i,2}$, $k_{i,1}$ and $k_{i,2}$ is given. The entries with '(any)' denote that any value could be given. $t_{i,1}$ and $t_{i,2}$ can be floating point numbers; $k_{i,1}$ and $k_{i,2}$should be integers.

A.3.2. *`comprisk` output.* The output from `comprisk` can be read by the `readoutput.m` file. This m-file returns a MatLab struct which holds all the information in the file. The output of comprisk is such that MatLab can easily read it. The output consist of a number of tables. There are two types of tables in the output.

Simple tables are of the format: $NAME : V_1\ V_2\ \dots\ V_n$. Whitespace does not matter here. The table is terminated by the first token that is not a valid number (the number n is thus not given explicitly in the input). This table type can be used for a single number, or for a vector. The *NAME* can contain any non whitespace character, except a colon, and should be followed by a colon.

Headed tables are of the format:

NAME	Headers	Needs `-f`?
K	-	No
T	-	No
distIndex$\{k\}$	-	No
needLagrangian	-	Yes
Ni	i n	Yes
Nki(k)	i n	Yes
Nkij(k)	i j n	Yes
iteration(i).GCM(k)	G V inHull	Yes
iteration(i).dists(k)	start new optimal grad hdiag	Yes
conv	phi alpha stepSize	Yes
solution$\{k\}$	-	No

TABLE 1. Tables that can appear in the output of `comprisk`. *NAME*s with a k will appear for each $k = 1, \ldots, K$; the index i will appear for every iteration. The middle column list the headers for header tables; simple tables are indicates by a dash. The last column tells whether you need to run `comprisk` with the `-f` option in order to get these tables.

$$
\begin{array}{l}
NAME :: \\
H_1\ H_2\ \ldots\ H_m \\
V_{1,1}\ V_{2,1}\ \ldots\ V_{m,1} \\
V_{1,2}\ V_{2,2}\ \ldots\ V_{m,2} \\
\ldots \\
V_{1,n}\ V_{2,n}\ \ldots\ V_{m,n}
\end{array}
$$

These tables consist of three parts: the *NAME* section, the header section and the value section. The *NAME* section should end with two colons. The header section should be on a separate line. The value section is terminated by the first token that is not a number. This table type is suitable for displaying several vectors of the same length next to each other. This allows for some tables to be in a format that can be read easily. The *NAME* should follow the same rules as for the simple tables. H_1 to H_m are header names for each column of the value data. These names should only consist of letters from the alphabet. The vector for header H_i has values $V_{i,1}$ to $V_{i,n}$.

The *NAME*s the `comprisk` program outputs, are such that MatLab can easily interpret them. In fact, I use parenthesis and curly braces with a numeric index, and dots followed by field names to make the output structured. This will probably all become clear to you when you take a look at some output file of `comprisk`, and at the structure of the variable the `readoutput.m` file returns.

`readoutput.m` deals with the tables as follows: it returns a struct called `s`, that has all *NAME*s as fields. (Of course MatLab will interpret the array indices, cell array indices and struct field names that can be in *NAME*.) For headed tables, the variable indicated by *NAME* are structs again, with fields H_1 to H_m.

Table 1 list details about the tables that can appear in the `comprisk` output.

References

Maathuis, M. (2005). *Competing risk data subject to current status censoring: Nonparametric estimation of the sub-distribution functions*. Ph.D. Thesis in preparation.

Groeneboom, P. and Wellner, J.A. (1992). *Information Bounds and Nonparametric Maximum Likelihood Estimation*. Birkhäuser Verlag, Basel.?

De Klerk, E., Roos, C. and Terlaky, T. (2003). *Nonlinear Optimization*. Faculty of Information Technology and Systems, Delft University of Technology, Delft. Available at `http://www.isa.ewi.tudelft.nl/~roos/courses/wi387/wi387dic.ps`.

Robertson, T., Wright, F.T. and Dykstra, R.L. (1998). *Order restricted statistical inference*. Wiley, New York.

Jongbloed, G. (1998). "The Iterative Convex Minorant Algorithm for Nonparametric Estimation". *Journal of Computational and Graphical Statistics*, Volume 7, Number 3, Pages 310-321.