UNIVERSITY OF CALIFORNIA,
IRVINE

Semi- and Non-parametric Methods for Interval Censored Data with Shape Constraints

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Statistics

by

Clifford Anderson-Bergman

Dissertation Committee:
Yaming Yu, Chair
Dan Gillen
Zhaoxia Yu

2014

© 2014 Clifford Anderson-Bergman

The dissertation of Clifford Anderson-Bergman
is approved and is acceptable in quality and form for
publication on microfilm and in digital formats:

_____

_____

_____

Committee Chair

University of California, Irvine
2014

DEDICATION

To my parents, Mary Anderson and George Bergman, for being patient and supportive through this journey.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to thank the Statistics Department at University of California, Irvine and all the professors for giving me the opportunity to study and providing an excellent educational environment. In particular, I would like to thank Yaming Yu, my advisor, for helping guide me through the shadowy forrest of research. I would also like to thank Dan Gillen for heavy early influence, guidance while working at the ADRC and teaching 50% of my graduate courses.

CURRICULUM VITAE

Clifford Anderson-Bergman

EDUCATION

Doctor of Philosophy in Statisics **2014**
University of California, Irvine *Irvine, California*

Bachelor of Science in Mathematics **2007**
University of California, Irvine *Irvine, California*

RESEARCH EXPERIENCE

Research Assistant **2008–2009**
Alzheimer Disease Research Center, UCI

TEACHING EXPERIENCE

Teaching Assistant **2013–2007**
University of California Irvine *Irvine, California*

Math Tutor **2006–2007**
Learning and Academic Resource Center at UCI *Irvine, CA*

SELECTED HONORS AND AWARDS

ICS Fellowship **2007–2010**
Department of Statistics, UCI

NIH Training Grant **2008–2009**
Alzheimer's Disease and Research Center, UCI

SUBMITTED JOURNAL PUBLICATIONS

"Computing the Log-concave NPMLE for Interval Censored Data" **2013**
Journal of Computational and Graphical Statistics

PUBLISHED JOURNAL PUBLICATIONS

Head, E., *et al.*, "A Fibril-Specific, Conformation-Dependent Antibody Recognizes a Subset of Aβ Plaques in Alzheimer Disease, Down Syndrome and Tg2576 Transgenic Mouse Brain" **2009**
Acta Neuropathologica, October 2009, Volume 118, No 4, pp505-517

PRESENTATIONS

"Shape Constrained non-parametric Maximum Likelihood Estimation for Interval Censored Data" **2013**
Joint Statistical Meetings Montreal, Canada

"Multiple Imputations for Sensitivity Analyses" **2012**
Allergan Irvine, CA

"Efficient Computation of the NPMLE for Bivariate Censored Data" **2011**
University of California, Irvine Irvine, CA

"Operating Characteristics of Futility Stopping Rules **2010**
Biogen Idec San Diego, CA

PROFESSIONAL MEMBERSHIPS

American Statistical Association (ASA)

ABSTRACT OF THE DISSERTATION

Semi- and Non-parametric Methods
for Interval Censored Data with Shape Constraints

By

Clifford Anderson-Bergman

Doctor of Philosophy in Statistics

University of California, Irvine, 2014

Yaming Yu, Chair

Interval censoring occurs when event times are known to have occurred within an interval, rather than observing the exact time of event. This includes observations that are right censored, left censored and contained in intervals such that the left side is greater than the origin and the right side is finite (i.e. neither right censored or left censored). For interval censored data, the most common survival estimator used is the non-parametric maximum likelihood estimator (NPMLE), a generalization of the Kaplan-Meier curve which does not require any uncensored event times. The popularity of this estimator is due in part to the fact that assessing model fit for interval censored data can be very difficult. However, the extreme flexibility of the estimator comes at the cost of high variance, often providing an $n^{-1/3}$ convergence rate rather than the more typical $n^{-1/2}$.

In a compromise between a highly constrained parametric estimator and the overly flexible NPMLE, we apply the popular log-concave density constraint to the NPMLE. By constraining a non-parametric estimator to have a log-concave density, an investigator can improve the performance without needing to select a parametric family or smoothing parameter. We describe a fast algorithm we have developed for finding

the log-concave NPMLE for interval censored data. We demonstrate that using the constraint significantly reduces the variance of the survival estimates in comparison to the unconstrained NPMLE via simulations. Next, we present three inference methods for our new estimator. This includes a goodness of fit test, two methods of confidence interval construction and a Cox PH model which incorporates a baseline log-concave distribution. We evaluate the power of the goodness of fit test and compare the other inference methods with the unconstrained counterparts via simulation. We apply these methods to a study on the effects of different environments on the rates of lung cancer among mice and another study investigating age at menopause.

While our work demonstrates that the application of the shape constraints can be very helpful in the context of interval censored data, in some situations the log-concave constraint may not allow for as heavy tailed distributions as the investigator would like. To address this, we propose a new, more flexible "inverse convex" shape constraint, examine its behavior via simulation and show that it provides a better fit than the log-concave estimator when applied to real income data, which is well known to be heavy tailed. We are very optimistic about applying this new estimator to censored data, although we have yet to implement an algorithm to do so.

We end this work with an algorithm for finding the (unconstrained) bivariate NPMLE for interval censored data. The bivariate NPMLE is used when each subject has two censored outcomes and the investigator is interested in modeling the relation between the two outcomes. Quickly finding the NPMLE has proven to be a challenging computational problem, as the number of parameters to consider is of order $O(n^2)$. We present an efficient EM algorithm to find the bivariate NPMLE. We note that this is not related to shape constrained estimation.

Chapter 1

Introduction

In the first chapter, we briefly explain key topics in this thesis. In section 1.1, we discuss many of the common forms of interval censored data. In section 1.2, we discuss shape constraints and the motivation for shape constrained density estimation. In section 1.3, we discuss the datasets which will be used to illustrate the application of the new methods presented in this dissertation. In section 1.4, we outline the topics presented in the chapters to come.

1.1 Interval Censored Data

Interval censored data occurs when a response value is known to have occurred within an interval rather than observed exactly. This can happen when a subject is periodically inspected for the occurrence of an event resulting in the event being known to have occurred between inspection times (or potentially at inspection times for a discrete time model). For example, a standard doctor's appointment may include a screening for a disease. If the patient tests positive for a disease for the first time, then

it is known the subject contracted the disease between check ups. Standard notation is as follows: for the i^{th} subject, T_i denotes the unobserved event time, L_i denotes the left side of the censored interval and R_i denotes the right. All that is known about T_i is that $T_i \in [L_i, R_i)$. This notation allows for right censoring ($R_i = \infty$), left censoring ($L_i = 0$) and exact observations ($L_i = R_i$).

Below we briefly describe common forms of interval censoring. For a more in-depth explanation of the topic, we refer to the excellent books of Sun (2006) and Chen *et al.* (2013).

1.1.1 Case I Interval Censoring: Current Status Data

A common form of interval censoring is current status data, or case I interval censored data. Under this censoring scheme, we consider two variables, T_i and C_i, in which T_i is the event time of interest and C_i is an inspection time. If $T_i < C_i$, then we observe $L_i = 0$ and $R_i = C_i$. If $T_i \geq C_i$, then we observe $L_i = C_i$ and $R_i = \infty$. In other words, at time C_i we either observe that the event has already occurred, in which case subject i is left censored, or we observe that the event has not occurred, in which case the subject is right censored. While inference typically uses the assumption of independence of censoring mechanism and event time, it is not necessary for the inspection time to be random, as the inspection times may be chosen by design.

Current status data has the advantage that it is relatively inexpensive to collect. In particular, it is not necessary to follow up on subjects, unlike most survival analysis studies. By merely taking a cross section of the population, an investigator can greatly reduce costs and accelerate a study with such a design. In addition, some studies require a current status design, as inspecting the subjects may have an effect on behavior. A common case of this is when test animals must be sacrificed to

inspect for presence of disease. A disadvantage of current status data is that each observation is fairly uninformative, meaning very large samples may be required for simple inference.

In section 1.3.1, we describe two classic current status datasets which we will use for illustrative examples in chapters 3 and 4.

1.1.2 Case II Interval Censoring

Current status has a very simple censoring mechanism which allows only for right and left censoring. Case II interval censoring is much more general and cannot be represented in such a similar manner. For a dataset to be considered case II interval censored, it will contain at least one observation which is censored, but neither left nor right censored, *i.e* $L_i < R_i$, $L_i > 0$ and $R_i < \infty$. It may also include left and right censored and uncensored data. By its nature, case II interval censoring is typically more informative than current status data, as the intervals are usually shorter than in the current status case. While this is more informative than current status data, it also leads to many complications in computations and derivation of statistical theory. One complication is that for current status data, it is very easy to model the censoring mechanism, as all that is needed to be know is the distribution C_i, which is observed exactly, if not planned in advance. The censoring mechanism for Case II interval censoring is not so easily defined and may not be possible to model without untestable assumptions of the censoring process. Although under standard assumptions, it is not necessary to model the censoring distribution to obtain estimates of parameters, many formulas for the variance depend on the distribution of the censoring mechanism. This can create difficulties in inference for case II interval censoring.

1.1.3 Doubly Censored Data

Another special case of interval censoring is doubly censored data. In this situation, not only is event time censored, but so is the time of origin as well. A classic case of this is studies investigating time from HIV infection to development of AIDS. The event time of interest is development of AIDS, which is only known to have happened between doctor visits. In addition, the origin is time of HIV infection, which again is only known to have happened between visit times. The use of the term "doubly censored" is not consistent in the literature; some authors also refer to data sets that contain both left censoring and right censoring as doubly censored. In this work, when we refer to doubly censored data, we always mean that both the origin and event time are censored.

In section 1.3.4, we describe a classic doubly censored dataset which will be used for an illustrative example in chapter 6.

1.1.4 Bivariate Interval Censoring

Bivariate interval censoring occurs when two variables are collected on each subject and at least one of the variables is interval censored. If the variables are independent, then the marginal fits for each variable are fully informative. If they are not independent and the relationship between the variables is of interest, then they must be jointly modeled. This is typically done with the bivariate NPMLE. This estimator can present computational problems, as the number of parameters considered can be $O(n^2)$ in the worst case scenario. We point out that doubly censored data can be viewed as a special case of bivariate interval censoring.

1.2 Shape Constraints

Under shape constrained estimation, restraints on various estimates are made to improve operating characteristics. In some cases, the restraints may be made in the parametric framework. For example, we may want to constrain $\mu_1 \leq \mu_2 \leq ... \leq \mu_k$ if we have strong reason to believe the parameters should behave in an ordered manner and the variance of the individual $\hat{\mu}_i$ is relatively high. This strictly monotonic relation of parameters is referred to as *isotonic regression* (van Eeden 1958). A classic application of isotonic regression is estimating the effect of a drug over many different dose levels with relatively small sample sizes at each dose. We note this allows for more flexibility than linearly modeling the effect of dose size but still makes estimation more efficient than a completely unspecified model which does not take advantage of the assumed relation between the parameters. This is the general theme for shape constraints: more flexible than a model with fewer parameters, but still taking advantage of the assumed structure of the estimation space. An excellent book on parametric shape constrains is Silvapulle and Sen (2004).

In this work, we will focus on non-parametric shape constraints. Specifically, we will consider restraints put on the density function. In this case, the investigator restrains the basic shape of the density function of the data and finds the function which maximizes the likelihood while still respecting the shape constraints. It should be noted that the motivation for this is different than typical non-parametric estimators. Most parametric estimators estimate over an infinite parameter space, with the goal of being as flexible as possible. While non-parametric shape constrained estimators still estimate over an infinite parameters space, the shape constraints can be used to provide more efficient estimation than unconstrained non-parametric estimation, while still being more flexible than a finite dimensional parametric estimator. In a sense, it can be seen as a compromise between the very flexible but inefficient unconstrained

non-parametric estimator and the efficient but overly restrictive parametric estimator.

The earliest such restraint studied is the Grenander estimator, or strictly decreasing density estimator (Grenander 1956). The set of all possible decreasing density functions cannot be described by a finite set of parameters, making it a non-parametric estimator. However, the constraints change the behavior the estimator considerably compared with the more flexible empirical distribution function (EDF) estimator. For example, while the EDF is a consistent estimator of the cumulative distribution function, it results in degenerate density estimates. By enforcing a strictly decreasing density, we will have a consistent density estimator if the assumption of decreasing density is correct. This necessary assumption is a big drawback with the Grenander estimator; a strictly decreasing density is not commonly a robust assumption.

Another classic density constraint considered is that of a unimodal density. This can also be thought of as a Grenander estimator on either side of a mode. This is attractive, as empirically many types of data appear unimodal. By enforcing the flexible unimodal density constraint, it would seem that smoothness of the density would be implemented in a more reasonable manner than the Grenander estimator. However, if the mode of the distribution is not known in advance the estimator is degenerate, as it would place infinite density at the mode (Wegman 1969). Even if the mode were to be known in advance, it has been shown that heavy spiking will occur at the mode.

In this work, we will focus on the popular log-concave constraint. This restrains the density function to be described by $e^{\phi(x)}$, in which $\phi(x)$ is a concave function. This is considered a fairly flexible unimodal constraint, as many traditional parametric families are log-concave. The normal, logistic, beta with shape parameters ≥ 1, gamma with shape ≥ 1, weibull with exponent ≥ 1 are all contained within the class of log-concave distributions. However, distributions with heavier tails than an

exponential (i.e. log linear), such as t-distributions, and multimodal distributions, such as mixture models, are not log-concave. Additionally, the log-concave constraint ensures an increasing hazard function, although an increasing hazard function does not insure log concavity. By enforcing the slightly more restrictive constraint of log-concave, the degeneracy problems faced by the uni-modal estimator are avoided.

Two recent applications of the log-concave constraint are that of density estimation (Rufibach 2007) and clustering with mixture modeling (Chang and Walther 2007). For mixture cluster modeling, it has been shown that incorrect specification of component densities can lead to poor classification rates. The flexible nature of the log-concave estimator allows for consistent and efficient estimates of mixing probabilities without selection of a parametric family, and Chang and Walther (2007) showed it can improve classification rates compared with the Gaussian mixture estimator. The flexibility of the log-concave estimator is very helpful in this problem, as assessing model fit in a mixture model problem is very difficult.

In this work, we will concentrate on application of the log-concave estimator for interval censored data. In particular, we are interested in survival estimates. While the unconstrained NPMLE does lead to consistent survival estimates under some basic assumptions, it is notoriously inefficient, displaying a $n^{-1/3}$ convergence rate for current status data. In addition, model fit is very difficult to assess for heavily interval censored data such as current status data, implying that parametric models may not be trustworthy. We present the log-concave NPMLE as a method for significantly reducing the variance of the estimates, without needing to make strict parametric assumptions. We present methods for computation of the log-concave NPMLE in chapter 3 and inference methods based on the log-concave NPMLE in chapter 4. In chapter 5, we introduce a new shape constraint we call "inverse convex", which we believe will be more robust for survival analysis (currently it is only implemented for

uncensored data).

1.3 Illustrative Datasets

In this work, we provide a variety of model techniques which can be applied to a variety of different situations. Because of this, we will consider a variety of illustrative datasets.

1.3.1 Menopause Study

Our first example will borrow data from MacMahon and Worcestor (1966). Questionnaires were collected from $n = 2,423$ participants regarding age at menopause. Because several of the subjects had not experienced menopause yet, this data set contained right censored data. However, MacMahon and Worcestor (1966) found that there was a marked terminal digit clustering in the response of reported time of menopause. In particular, they found heavy clustering around the digits 5 and 0 for reported age at menopause. Given that menopause typically occurs within a 10 year interval, this could lead to heavy bias. Krailo and Pike (1983) recommended only using the menopausal status of women at the time of the questionnaire, thus resulting in current status data. The reasoning was that although subject may not be able to accurately recall the age at menopause, they should know whether it occurred. This would reduce bias at the cost of increasing the variance from throwing out informative data. Because the sample size was so large, this bias-variance trade off seemed quite reasonable. The data also contains two types of menopause; operative menopause and natural menopause.

Earlier analyses of this data used a competing risks model (Jewell *et al.* 2003,

Maathuis and Hudgens 2008). For demonstrative purposes, we will only examine the time to menopause, regardless of the type of menopause. Although a competing risks model would be considerably more informative about the nature of this dataset, we still choice to use this data set for our simpler analysis methods, as it is largest interval censored dataset publicly available. In chapter 3, we apply our newly developed algorithm for finding the log-concave NPMLE to this data set and would like to show that it finds the solution sufficiently quickly, despite the large size of the dataset. In chapter 4, we apply a goodness of fit to evaluate whether a log-concave assumption is appropriate, which is natural concern given the two types of menopause in the dataset.

1.3.2 Lung Tumor Study

For our third example, we will use data found in Hoel and Walberg (1972). This study involved 144 male RFM mice, a line which is bred to have higher rates of lung tumors than standard mice. In this study, 96 mice were placed in a conventional environment (CE) and and 48 mice were placed in a germ free environment (GE). The mice were sacrificed at different times and examined for lung tumors. The tumors were predominately non-lethal, so the authors argued that the assumption of independence of inspection process and event time should be approximately valid. This led to current status data. The focus of this study will be to determine whether GE mice suffered different rates of tumors compared to the CE mice.

In chapter 4, we apply two methods to evaluate the difference in cancer rates between the two groups. In the first analysis, we fit log-concave estimates to each group individually and compare medians via confidence intervals. In the second analysis, we combine the two groups and use a Cox-PH model with a baseline log-concave

distribution to compare the two groups.

This data set was analyzed in Sun (2006). Sun applied the unconstrained NPMLE to each group separately to estimate survival probabilities and applied a Cox PH model with an unconstrained baseline to compare the tumor rates between the two groups. Using the Cox PH model, Sun found that there was a statistically significant difference between hazard rates between the groups, with the germ free environment suffering from a hazard rate that was 1.99 times higher than the conventional environment. While Sun is able to make inference on the regression parameter, currently there are no inference methods for comparing estimates based on the baseline survival curve in a Cox PH model, such as comparing medians for the two groups.

In chapters 3 and 4, we apply log-concave NPMLE to each group separately and apply a Cox PH model in which the baseline is log-concave. Our estimates of the regression parameter is nearly identical (estimated hazard ratio = 2.05). By using the log-concave baseline, we allow for efficient inference when comparing survival curves of the two groups in the Cox PH model.

1.3.3 Income Datasets

In chapter 5 of this work, we consider two datasets. The first contains income data collected by the Philippines National Statistics office. A total of 632 subjects were surveyed in this study and their reported income was recorded. In addition, we have recorded the region in which they live. The goal of this study is to compare the disparity between individuals in residing in the La Union district ($n = 116$) and the Pangasinan district ($n = 381$). Two other regions are reported, but we only compare these two groups as they are the only groups with more than 100 subjects in their sample. The disparity will be summarized by the Gini coefficient, which will

be described in more detail in chapter 5. This dataset is titled "Ilocos" and can be found in the publicly available CRAN package "ineq".

In addition, we will examine a second dataset collected by the US Census. This dataset contains reported wages from the March 1988 Current Population Survey. A total of $n = 28,155$ subjects are included in the data set. The data also includes an indicator of region, broken up into Northeast ($n = 6,441$), Midwest ($n = 6,863$), South ($n = 8,760$) and West ($n = 6,091$). In our work, we inspect this data much more causally than the Ilocos dataset, merely assessing the fit of various models to the data by comparing estimated cumulative distribution functions. This dataset is title "CPS1988" and can found in the CRAN package "AER".

Unlike the other datasets, both of these datasets contain no censored data.

1.3.4 Hemophiliac AIDS Cohort Study

A classic data found in the literature presented by Kim *et al.* (1993), this study involved 257 hemophiliac patients treated with HIV contaminated blood at two French hospitals beginning in 1978. Of these 257 patients, 188 contracted HIV by August 1988. Of these 188, 41 of them progressed to develop AIDS. Interval censoring occurs because the patients are not continually monitored, but rather only observed during doctor visits. Before HIV was detected, the censoring was very heavy, as patients had their blood checked only during standard check ups. Once they had been diagnosed with HIV, they were still subject to interval censoring, although the check ups were more frequent so the censoring of the development of AIDS is lighter. In particular, the average length of intervals in which HIV infection were known to occur was 2.1 years, while the average length of intervals in which AIDS was known to have developed was 0.56 years. We note that the time of interest is doubly censored, as

both time of infection and time of seroconversion are interval censored.

We will use this dataset to estimate the survival curve for time of seroconversion to development of AIDS. The doubly censored data will be dealt with modeled by first fitting the bivariate NPMLE to both events and then using this estimated joint distribution to calculate the estimated survival curve for time from seroconversion to development of HIV. We apply the algorithm for finding the bivariate NPMLE presented in the appendix to this dataset.

1.4 Summary of Chapters

In chapter 2, we will review methods used to analyze interval censored data and applications of shape constrained density estimation. In chapter 3, we present an efficient algorithm for finding the log-concave NPMLE for univariate interval censored data and use this algorithm to compare the behavior of the log-concave NPMLE to the unconstrained NPMLE. In chapter 4, we present three methods of inference for the log-concave estimator for interval censored data, including a goodness-of-fit test, construction of confidence intervals and a Cox PH model with a log-concave baseline. In chapter 5, we present a new shape constraint we call inverse convex, which is more appropriate for the heavier tailed distributions seen in survival analysis problems. In chapter 6, we present an efficient EM algorithm for finding the (unconstrained) bivariate NPMLE for interval censored data. In chapter 7, we present several future research topics we are interested. In chapter 8, we discuss the general findings presented in this work.

Chapter 2

Review of Methodology

In this chapter we briefly present current methodology related to our work. In section 2.1, we discuss current methods for interval censored data. In section 2.2, we discuss current applications of the log-concave shape constraint.

2.1 Interval Censoring

2.1.1 Parametric and Imputation Models

The simplest way to deal with interval censoring is using standard fully parametric models. The standard parametric models used for interval censored data are identical to those found in standard right censored survival analysis; exponential, weibull, gamma, lognormal and log logistic families are typically used. In addition, typical survival regression models are used, such as the Cox PH, proportional odds model and accelerated failure time (AFT) model. Standard MLE procedures lead to valid inference. One small complication with interval censored data is that the log likelihood

function is not necessarily concave, meaning theoretically optimization procedures such as Newton's method may not work. However, as the data becomes more informative, this is less likely to be an issue, so standard optimization procedures work in almost all situations. If Newton's method fails, more robust algorithms such as conjugate gradient methods can be used. A basic tutorial method can be found in Lindsey and Ryan (1998).

A very serious drawback to fully parametric models is that it is very difficult to assess model fit for interval censored data. The limited information provided by each censored observation means that the data can appear to fit most parametric families fairly well, making results highly dependent on model choice and difficult determine the validity of such choices. Because of this, there is very limited work in the literature on fully parametric models for interval censored data.

Another fairly easy set of methods for dealing with interval censoring are imputation models. While the basic imputation procedures presented in Rubin (1978) readily apply, interval censoring has two unique complications. Typically, imputation models condition only on the other co-variates to predict the missing values. However, interval censoring not only conditions on all other variables, but also must condition on the fact that the missing data must be contained within the censored interval. Thus, it is more accurate it call it incomplete data rather than missing data. In addition, because interval censored data allows for the possibility that none of the exact response times will be observed, methods such as hot deck imputation which directly sample the observed values cannot be employed. Bebchuk and Betensky (2000) use multiple imputation for estimation of the hazard function in interval censored data and Satten *et al.* (1998) and Pan (2000a) used multiple imputation to create a Cox PH model for interval censored data.

However, multiple imputation methods still require modeling of the censored data for

imputation, so an investigator still requires modeling the distribution of the censored data before the data can be imputed. This means imputation must be paired with an appropriate modeling scheme and is not a stand alone solution.

2.1.2 Univariate NPMLE

In this work, we focus on non-parametric and semi-parametric estimation methods for modeling interval censored data. The first works regarding the non-parametric maximum likelihood estimator (NPMLE) appeared in Turnbull (1976). The Kaplan Meier estimator is a special case of the univariate NPMLE in which the data is only right censored. The NPMLE is sometimes represented as a step function, similar to the Kaplan Meier estimator. However, NPMLE for interval censored data typically displays non-uniqueness and is more accurately represented as two step functions. Any survival curve that lies between the two step functions will have the same likelihood. Figure 2.1 presents an example of the univariate NPMLE generated from a simulated current status dataset with $n = 100$.

The first step in finding the NPMLE was determining a finite set of parameters which can fully describe the NPMLE. Turnbull characterized the support set consisting of disjoint "Turnbull intervals" which would receive positive mass at the NPMLE. The set of Turnbull intervals for a data set is the set of all intervals such that the left side of the interval was the left side of an observation interval and the right side was the right side of any observation interval, and no observation interval end points were contained inbetween. The probability mass assigned to each Turnbull interval at NPMLE is unique. This unique assignment to each interval is called *mixture uniqueness*. However, how probability is assigned within each interval does not affect the likelihood function and leads to the non-uniqueness seen in figure 2.1. This is

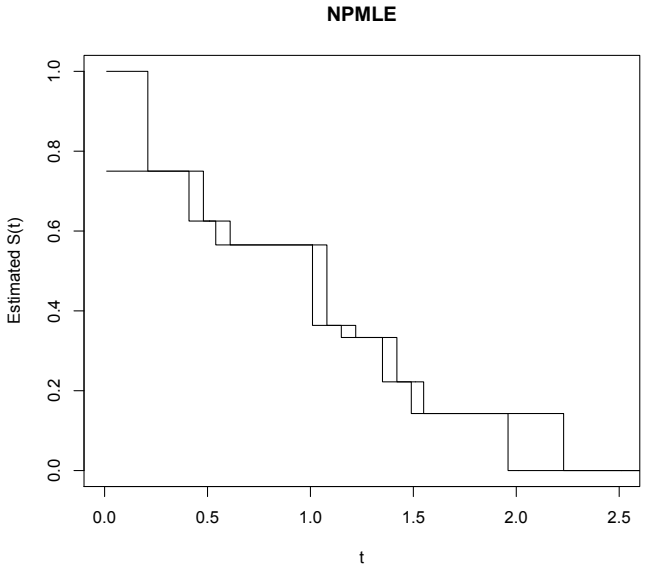

Figure 2.1: Example of univariate NPMLE for simulated current status data

referred to as *representational non-uniquness*. The solution is fully defined by the set of Turnbull intervals and the probability mass assigned to each interval, which allows the problem to be treated as a discrete probability problem.

Turnbull presented a simple EM algorithm to compute the NPMLE with interval censored data which also allowed for truncation. While this algorithm is sufficient for finding the NPMLE, it is notoriously slow. Later, Groeneboom (1991) developed the ICM algorithm, which approximates the log likelihood as a quadratic function of the estimated survival curve and maximizes this approximation via quadratic programming. This is very similar to Newton's method, although quadratic programming is necessary to insure the estimate is a proper survival function (*i.e.* sum of probability masses equals 1, probability mass in each Turnbull interval is non-negative). Jongbloed (1998) insured the algorithm would always converge by forcing the likelihood to be strictly increasing in the algorithm via half stepping if the proposed step reduced the likelihood. Wellner and Zhan (1997) showed that combining the ICM algorithm and the EM algorithm greatly accelerated the algorithm.

Quite a bit of literature exists on the distribution of the NPMLE for interval censored data. In the case of current status data, Groeneboom (1987) described the asymptotic behavior of the estimator. In particular, if G is the cdf of the inspection process, with g the pdf, F_0 is the cdf of the event time, with f as its pdf and $\hat{F}_n$ is the NPMLE of F_0, then as n $\to \infty$,

$$n^{1/3}\{\hat{F}_n(t_0) - F_0(t_0)\} \to_d 2c_1(t_0)Z$$

where

$$c_1(t_0) = \left(\frac{f(t_0)F_0(t_0)(1 - F_0(t_0)}{2g(t_0)}\right)^{1/3}$$

and Z follows a Chernoff distribution, or

$$Z = \arg\max_t(W(t) - t^2)$$

where $W(t)$ is a two sided Brownian motion with $W(0) = 0$.

For case II interval censored data, the asymptotic distribution of $\hat{F}_n$ is much more tricky. Groeneboom and Wellner (1992) proved that it is a consistent estimator. Groeneboom (1991) conjectured that with a different rate of convergence ($(n \log n)^{1/3}$ instead of $n^{1/3}$) and a different scaling constant, the estimator should have the same limiting distribution under some assumptions.

This creates problems with inference, as using the observed information matrix to create confidence intervals and hypothesis tests does not have theoretical justification. In addition, the use of bootstrap estimates for cube root estimators has come into question recently (Leger and MacGibbon 2005). However, in the case of current status data, the above distribution formula can lead to valid confidence intervals. One complication is that it requires the density estimates $f(t_0)$ and $g(t_0)$. Because the inspection times are known exactly, estimating $g(t_0)$ is considered trivial. Because the NPMLE does not provide density estimates, estimating $f(t_0)$ is considered a tricky problem. Banerjee and Wellner (2005) create these types of confidence intervals using the kernel density estimator for both $f(t_0)$ and $g(t_0)$. In addition, they compare confidence intervals based on these distribution results to profile confidence intervals and the subsampling methods of Politis *et al.* (1999), which are valid for cube root estimators. They found that the profile confidence intervals appeared to be the best choice, having the minimal mean length while this maintaining approximately correct coverage. While the subsampling methods also had approximately correct coverage probability, the mean width was typically much higher. And while confidence intervals constructed via the limiting theory had only slightly wider intervals, the coverage probabilities were very poor.

2.1.3 Bivariate NPMLE

Computation of the bivariate NPMLE has received quite a bit of attention in the literature being a classically difficult problem. Even defining the parameter space is computationally intensive. In the univariate NPMLE, Turnbull (1976) shows that all probability mass must be assigned to Turnbull intervals, which are very easy to find based on their definition. For bivariate data, this generalizes to *maximal intersections* (Wong and Yu 1999). Perhaps the easiest definition of the maximal

intersection is resented in Maathuis (2005), which is if we consider a "height map" of the 2 dimensional space, where "height" is the number of overlapping observation rectangles in a given region, then maximal intersections are local maxima of the height map.

Several algorithms have been proposed to find the set of maximal intersections. These algorithms are called *reduction algorithms*, as they reduce the support space considered. In contrast, algorithms that assign probability mass to maximal intersections are called *optimization algorithms*. The first reduction algorithm found in the literature was presented in Betensky and Finkelstein (1999), although it was fairly inefficient (we were not able to find a report of its complexity). Gentleman and Vandal (2001) presented a reduction algorithm of time complexity $O(n^5)$ and Bogaerts and Lesaffre (2004) presented an algorithm of $O(n^3)$. Most recently, Maathius (2005) presented the HeightMap algorithm, which finds all maximal intersections within $O(n^2)$. The HeightMap algorithm is highly efficient, being able to handle quite large data sets in very little time. For example, in the worst case scenario of a continuous censoring mechanism, with $n = 5000$, HeightMap found all maximal intersections in 0.289 seconds. By comparison, even the current leading optimization algorithm would be expected to take days to converge in that scenario.

Because of the speed of this new reduction algorithm, there is in turn demand for fast optimization algorithms. The EM algorithm presented for the univariate NPMLE found in Turnbull 1976 easily generalizes to the bivariate NPMLE. However, this algorithm is even slower in the bivariate case, meaning it will not converge within a reasonable amount of time even in some very small datasets (*i.e.* $n = 50$). Böhning *et al.* (1996) pointed out that the NPMLE can viewed as a mixture model and suggested use of the Vertex Exchange Method (VEM) algorithm. Most recently, Maathius implemented the Support Reduction algorithm found in Groeneboom, Jongbloed and

Wellner (2007) in CRAN package "MLEcens".

The theoretic characteristics of the bivariate NPMLE are very interesting as well. As noted earlier, the univariate NPMLE suffers from representational non-uniqueness but has mixture uniqueness. However, for the bivariate NPMLE (or higher dimensional), the estimator suffers from both representational and mixture non-uniqueness (Gentleman and Vandal 2002). Although the effect of representational non-uniqueness is fairly straightforward, the effect of mixture non-uniqueness is unclear. In addition, the bivariate NPMLE has the odd problem that is inconsistent in the special case of singly censored observations (Laan 1996), that is when one of the bivariate events is interval censored but the other is not. Maathius (2006) show the estimator has cubed root convergence under some regularity conditions.

2.1.4 Semi-parametric Regression Models

For interval censored data, the most popular regression model is the Cox PH model (Cox 1972). Finkelstein (1986) presented a Newton Raphson for finding the MLE, applied to a reparameterization of the likelihood function. However, this was found to be highly unstable and typically required inverse of a very large Hessian matrix. Huang (1996) presented a two step algorithm which alternated between optimizing the baseline NPMLE and the regression coefficients. This algorithm used a similar parameterization to Finkelstein (1986) and also was found to be unstable. Pan (1999) present an ICM algorithm for computing the semi-parametric MLE, which avoided the use of matrix inversion. While this algorithm was found to be stable, we found that the current implementation of this algorithm in the CRAN "intcox" package was quite slow (see chapter 4).

One very attractive feature of these regression models presented in Huang (1996)

showed that

$$n^{1/2}(\beta - \hat{\beta}) \rightarrow_d N(0, \Sigma^{-1})$$

where Σ is the information matrix of β. Thus while comparing two groups by fitting two NPMLE fits can be inefficient due to cube root convergence, regression models are still fairly efficient, as they display square root convergence.

A difficulty in obtaining standard errors is that Σ is actually infinite dimensional, as semi-parametric modeling allows for infinite dimensional parameter space. In the right censored case, the likelihood can be factored so the Hessian of the covariates can be considered separately. However, with interval censored data, the likelihood is not factorable, so the baseline covariates cannot be ignored for estimation of the regression parameters. Huang suggested "sieve estimation", in which the infinite dimension parameter space is approximated with a finite dimensional parameter space. In addition, because the regression parameter has square root convergence rather than cube root, there is no reason to question the validity of bootstrap estimation. Finally, profile likelihoods can be used for inference as well, although currently there is no publicly available software for such methods.

2.2 Shape Constraints

Although shape constrained estimation has existed for a long time, advancements in non-linear programming allow for more efficient computation, which in turn has allowed for application of more useful constraints. Thus, despite being a mature field,

it has experienced a resurgence of attention in the literature recently.

As mentioned in the introduction, shape constraints can be either for parametric models, where the parameters have constraints on them, or non-parametric, in which the shape of a particular function of the estimate is constrained. In this work, we will be examining the case of non-parametric constraints. The earliest shape constraint to appear in the literature is the Grenander estimator (Grenander 1956), or a strictly decreasing density estimator. One of the key advantages of this constraint is that it allows for non-parametric density estimates, while the unconstrained non-parametric estimator (better known as the Empirical Distribution Function) has degenerate density estimates. However, in practice the constraint of strictly decreasing density is not widely acceptable for most problems.

Another constraint considered was the unimodal estimator. The limitations on the distributions this sets is considered ideal: by being unimodal, it seems we would prevent erratic behavior of the probability density function and insure a smooth cumulative density function. Unfortunately, it was found that the unimodal NPMLE would always place infinite density at the mode, setting the likelihood function to ∞ (Wegman 1969). Even if the mode were known in advance (which in practice is quite rare), the estimator experienced heavy spiking at the mode.

A constraint that has gathered a lot of attention recently, and is the one we will consider in this work, is that of log concavity. A function $f(x)$ is said to be log-concave if $f(x) = e^{\phi(x)}$, where $\phi(x)$ is a concave function. Many of the basic properties of the log-concave density function are discussed in Dümbgen and Rufibach (2009). They show that by assuming log concavity, one can insure continuous density estimation without having to specify any sort of smoothing parameters. In addition, this ensures the estimated density is unimodal with an increasing hazard function. The family of log-concave densities includes several classic parametric families, including normal,

gamma with shape parameter ≥ 1, weibull with shape parameter ≥ 1, beta with both parameters ≥ 1 and the logistic distribution. On the other hand, in addition to the parametric models explicitly excluded from those listed above (*e.g.* gamma with shape parameter < 1, etc.), all t-distributions, log-logistic and log-normal distributions are not log-concave due to heavy tails. Likewise, multimodal distributions, such as mixture models with components sufficiently separated, are not log-concave. Recent work by Chang and Walther (2007) showed that using log-concave components in mixture modeling can improve estimates when the component distributions are skewed compared to a normal component, so log-concave estimation can still be useful in multimodal problems.

Recently, Dümbgen *et al.* (2011) presented an efficient active set algorithm for finding the log-concave NPMLE for exact data and presented an EM algorithm finding the log-concave NPMLE for interval censored data. The EM algorithm required discretizing the support space and thus provides an approximation to the NPMLE. Implemented algorithms for the both cases can be found in the CRAN package "logcondens". The active set algorithm for datasets without censored data is very fast and should be sufficient for moderately sized data sets. The EM algorithm for censored data is considerably slower, especially for interval censored data. In addition, the algorithm failed to converge quite often. This is discussed in more detail and average computation times are presented in chapter 3.

Chapter 3

Computation of the Log-concave NPMLE

3.1 Introduction

The NPMLE is a very popular estimator for interval censored data, the main strength being the flexibility of the model. This is especially important for interval censored data, as assessing model fit can be very difficult. However, the flexibility of the estimator comes at the cost of high variability of survival estimates. In particular, the convergence rate of the NPMLE has been shown to be $n^{1/3}$ for current status data (Groeneboom 1987) and conjectured to be $n^{1/3}\log(n)$ in the case of case II interval censored data (Groeneboom 1991). Heuristically, the poor behavior of the NPMLE is due to the fact that the estimated survival curves are similar to step functions with a relatively small number of steps, even for larger datasets (see figure 3.1).

Parametric models allow for much more efficient estimation, providing $n^{1/2}$ convergence rates. This depends on parametric assumptions which are hard to assess for

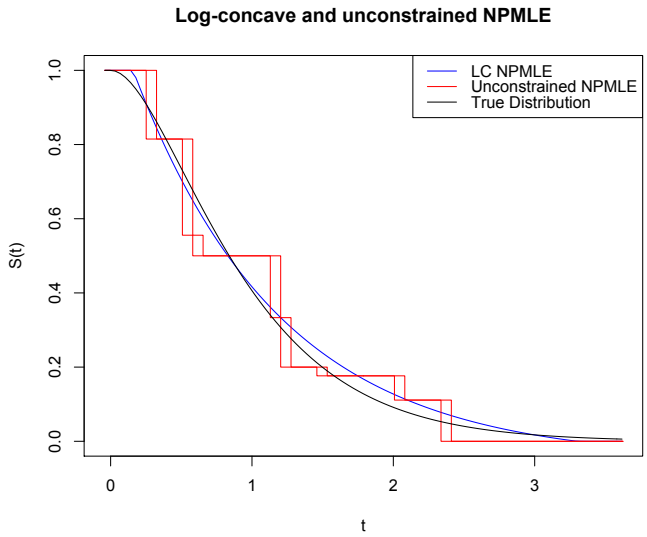

Figure 3.1: Log-concave NPMLE and unconstrained NPMLE. Data are current status data simulated from a Gamma(2,2) distribution with $n = 200$

interval censored data and can lead to heavy bias in survival estimates if the assumptions are inappropriate. Because of this, parametric models are fairly unpopular for use in interval censored data.

Alternatively, issues of the erratic nature of the NPMLE can be remedied by making an assumption about the smoothness of the underlying distribution, rather than making a more restrictive assumption of belonging to a parametric family. Doing so will reduce the variance of survival estimates, as it prevents the erratic step behavior of the NPMLE.

One approach to smoothing the estimator is by use of a smoothness parameter. One method for this is smoothing over the unconstrained NPMLE with a kernel estimator (Betensky *et al.* 1999), which requires selection of a bandwidth parameter. Another technique is to model the density with log-spline functions, using a smoothness penalty to select the number of knots (Kooperberg and Stone, 1992). While both of these techniques have been shown to reduce the variance of the estimated survival

curve for interval censored data (Pan 2000), they have their drawbacks as well. Both these estimators require selecting some form of an uninterpretable smoothness parameter, which can be non-trivial in the interval censored case. As an example, the logspline density estimator can be found in the CRAN package "logspline", with preset smoothing penalties. While this estimator can perform well with light censoring, in our experience with simulated data it behaved poorly under heavy censoring. In particular, it was often observed that either the estimate of the density jumps highly erratically or the algorithm fails to converge when used with current status data, even for large data sets (in fact, the estimator appears to perform *worse* for larger data sets). Similar problems were noted in Pan (2000). Likewise, the question of bandwidth selection for the kernel smoother in application to interval censored data is still an open question. While both these methods may prove fruitful if automated smoothness parameter selection methods can be developed specifically for interval censored data, in our study we found current implementations of such rules unreliable. This produces demand for a fully automated, theoretically justified smooth estimator.

Rather than selecting an uninterpretable smoothing parameter, we address this problem by applying a shape constraint on the density function which insures a smooth survival estimate. We will focus on the popular shape constraint of log concavity (originally published as Bagnoli and Bergstorm 1989, republished as Bagnoli and Bergstorm 2005) to meet these needs. A density function $f_X(x)$ is considered log-concave if $f_X(x) = e^{\varphi(x)}$, where φ is a concave function. This will insure that $f_X(x)$ will not have more than one peak, but can be flat, such as a uniform distribution. Because a straight line is the boundary of log-concave functions, it insures the tails of the distribution are at most exponential (*i.e.* log-linear). This assumption is considered very flexible as many parametric distributions fit into this category. The families of log-concave distributions include normal, gamma with shape parameter ≥ 1, all Weibull densities with exponent ≥ 1, all beta densities with both parameters ≥ 1

and the logistic density. Non-logconcave distributions include the t-distribution, the lognormal distribution and any multimodal distribution. However, the log-concave estimator has been shown to be useful in mixture modeling (Chang and Walther 2007) so it can be useful for multimodal data as well. This estimator is a consistent estimator of the density in the case of exact data (Dümbgen and Rufibach 2009) and an efficient algorithm for finding the log-concave NPMLE has been written for uncensored data (Dümbgen *et al* 2011).

Computation of the log-concave NPMLE for interval censored data has been considered in Dümbgen *et al.* (2011). They present a rough outline for an algorithm which discretizes the domain and applies an EM algorithm to find an approximation of the log-concave NPMLE (only an approximation due to the discretizing of the data). This algorithm has been implemented in the CRAN package "logconcens". While this algorithm behaves fairly well for right censored data, it behaves very poorly for interval censored data. We found this algorithm frequently failed to converge after 1,000 iterations (default settings on the package set the maximum iterations to 49) and in general, the algorithm was found to be too slow for most purposes (see table 3.2 for average speeds and frequency of failure to converge).

In this chapter, we provide an efficient method for finding the log-concave NPMLE for interval censored data. This algorithm was found to typically be over 100x faster than the algorithm used by the logconcens package and almost never failed to converge after 1,000 iterations ($< 0.1\%$ for simulated data with $n = 500$). In the illustrative example, a fairly large dataset of $n = 2,423$, our algorithm was almost 2,000 times faster (0.57 seconds vs 942 seconds). In addition, rather than discretize the domain, we provide a theorem which shows that the log-concave NPMLE can be characterized with a finite number of parameters ($2u - 1$, where u is the number of unique times appearing in the dataset). Using this parameter set, we provide an algorithm based

on the Active Set Algorithm presented in Dümbgen *et al.* (2011) for uncensored data, although several modifications are required to handle censored data. The most notable difficulty is the fact that the likelihood function is no longer concave. With our new algorithm, we will compare the operating characteristics of the log-concave NPMLE with the unconstrained NPMLE and the kernel smoother under a variety of different setting via Monte Carlo simulation.

The organization of the rest of this chapter is organized as follows. In Section 3.2, we formulate the likelihood function that we are interested in maximizing and state the conditions for which the likelihood function is bounded (proof is provided in appendix A). In Section 3.3, we prove a theorem which shows that the maximum likelihood can be achieved by a function of a finite number of parameters. In Section 3.4, we present the different parameterizations that will be used in the algorithm. In Section 3.5, we discuss the convergence criterion, which is a crucial part of our proposed algorithm. In Section 3.6, we describe the algorithm itself. In Section 3.7, we apply the algorithm to a classic sample data set and compare the results to those of the unconstrained NPMLE and the kernel smoother. In Section 3.8, we simulate data and compare the bias and standard deviations of the log-concave NPMLE, the unconstrained NPMLE and the kernel smoother across several different simulation scenarios. In Section 3.9, we discuss future research topics for the log-concave NPMLE.

3.2 Likelihood Function

We adopt the standard assumption of independent observations among subjects and an independent censoring mechanism. For the i^{th} subject, suppose the exact event time is known to be within the observation interval $[L_i, R_i]$. Let $\phi(x)$ represent the log density at time x. Because we are dealing with proper density estimates, we must

use the standard survival notation of $\delta_i = I_{\{L_i = R_i\}}$ (*i.e.* δ_i is an indicator that the exact event time was observed for subject i). Then the log likelihood function is

$$\ell(\phi) = \sum_{i=1}^{n} \delta_i \phi(L_i) + (1 - \delta_i) \log \left(\int_{L_i}^{R_i} e^{\phi(x)} \, \mathrm{d}x \right)$$

From this, the log-concave NPMLE is

$$\hat{\phi} = \arg\max_{\phi} \sum_{i=1}^{n} \delta_i \phi(L_i) + (1 - \delta_i) \log \left(\int_{L_i}^{R_i} e^{\phi(x)} \, \mathrm{d}x \right)$$

$$\text{which satisfies} \quad \frac{\phi(x_2) - \phi(x_1)}{x_2 - x_1} \geq \frac{\phi(x_3) - \phi(x_2)}{x_3 - x_2} \quad \forall x_1 < x_2 < x_3$$

$$\text{and} \int_{-\infty}^{\infty} e^{\phi(x)} \mathrm{d}x = 1$$

To ease the last restriction, we replace $e^{\phi(x)}$ with $\frac{e^{\phi(x)}}{\int e^{\phi(x)} \mathrm{d}x}$, so that $e^{\hat{\phi}(x)}$ is proportional to the log-concave NPMLE. Under this parameterization, the log-concave NPMLE is written as

$$\hat{\phi} = \arg\max_{\phi} \sum_{i=1}^{n} \delta_i \phi(L_i) + (1 - \delta_i) \log \left(\int_{L_i}^{R_i} e^{\phi(x)} \, \mathrm{d}x \right) - n \times \log \left(\int_{-\infty}^{\infty} e^{\phi(x)} \mathrm{d}x \right)$$

$$\text{which satisfies} \quad \frac{\phi(x_2) - \phi(x_1)}{x_2 - x_1} \geq \frac{\phi(x_3) - \phi(x_2)}{x_3 - x_2} \quad \forall x_1 < x_2 < x_3$$

Note that under this parameterization, $\hat{\phi}(x)$ is calculated up to an additive constant. For simplicity we set $\max_x \hat{\phi}(x) = 0$. We can then interpret $\hat{\phi}(x)$ as the estimated

log ratio between the density at x and the density at the mode.

Under minimal conditions, the likelihood function is bounded. In particular,

Theorem 1. *The likelihood function is bounded if one of the following three conditions is met:*

1. All the data are censored

2. At least two data points are uncensored, and they are not equal to each other

3. There exists one uncensored data point which is not contained in at least one of the censored intervals

See appendix A for proof of this theorem.

3.3 Solution Set

In the case of exact observations, it is known that the log-concave NPMLE must be a log piecewise linear function, with knots at the observed times and zero density outside the minimum and maximum observed times (Rufibach 2007). To prove this, consider that for exact times the log likelihood function can be written as

$$\sum_{i=1}^{n} \phi(x_i) - n \times \log \int e^{\phi(x)} \mathrm{d}x$$

where $x_1 < x_2 < \cdots < x_n$ are the ordered observations. For a fixed set of values of $\phi(x_i)$, the likelihood is maximized by minimizing $\int e^{\phi(x)} \mathrm{d}x$. For fixed $\phi(x_i)$ and $\phi(x_{i+1})$, $\int_{x_i}^{x_{i+1}} e^{\phi(x)} \mathrm{d}x$ under the concave restriction of $\phi(x)$ is minimized by linearly connecting $\phi(x_i)$ and $\phi(x_{i+1})$. Thus, $\hat{\phi}(x)$ is a log linear piecewise function in the case of exact observations. The concavity of the log-likelihood function with exact

observations insures that the solution is unique.

In the case of interval censored data, the log-concave NPMLE is not necessarily unique. As a trivial example, consider the case n = 1 and the one event is censored. In such a case, any log-concave function which places all of the mass inside of the censored interval would be a log-concave NPMLE. This is very similar to the problem of representational non-uniqueness in the unconstrained case (Gentleman and Vandal 2002). With such issues in mind, we will instead show that the maximum likelihood can be achieved via a log piecewise linear function with a finite number of knots, while recognizing that there may be other functions which have the same likelihood.

Define u = number of unique values of L_i and R_i.

Theorem 2. *The maximum likelihood for the logconcave NPMLE can be achieved with a piecewise linear function with at most* $2u - 1$ *knots.*

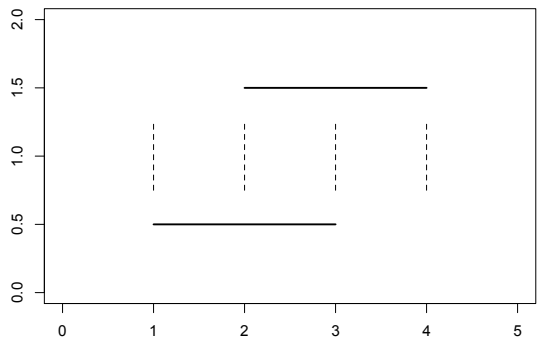

Figure 3.2: Observation intervals and contribution intervals

Proof. Let $\hat{\varphi}$ be a log-concave function which maximizes the likelihood function. We will prove that there exists $\hat{\phi}$ which is a log linear spline with at most $2u - 1$ such that $\ell(\hat{\varphi}) = \ell(\hat{\phi})$.

Define the i^{th} observation interval $[L_i, R_i]$ to be the interval in which the i^{th} event was known to have occurred. Define a contribution interval to be an interval such that both ends of the interval are ends of an observation interval, with no other ends of observation intervals in between. For example, Figure 3.2 shows two observation intervals, $L_1 = 1$, $R_1 = 3$ and $L_2 = 2$, $R_2 = 4$. This leads to 3 contribution intervals; [1,2], [2,3] and [3,4]. Note that exchanging mass *between* contribution intervals will affect the likelihood function, but exchanging mass *within* will not.

In any area that is not in a contribution interval, $\varphi(x)$ needs to be minimized for $\hat{\varphi}(x)$. Thus, $\hat{\varphi}(x)$ will be linear in areas that are between contribution intervals but are not contribution intervals and $\hat{\varphi}(x)$ will be $-\infty$ in areas that are not contribution intervals and not between contribution intervals. Therefore, we must only be concerned with whether ϕ can replicate the contribution of $\hat{\varphi}$ to the likelihood function in the contribution intervals while still respecting the constraints.

For the i^{th} contribution interval, define l_i and r_i to be the end points. Let us assume for now that $l_i > -\infty$ and $r_i < \infty$. For a given $\varphi(l_i)$, $\varphi'(l_i-)$ (left derivative at l_i), $\varphi(r_i)$ and $\varphi'(r_i+)$ (right derivative at r_i), this contribution interval affects the likelihood only through the total mass assigned to it, *i.e.* $p_i = \int_{l_i}^{r_i} e^{\varphi(x)}\mathrm{d}x$. We define $\hat{p}_i = \int_{l_i}^{r_i} e^{\hat{\varphi}(x)}\mathrm{d}x$

We will show that for a given $\hat{\varphi}(l_i)$, $\hat{\varphi}'(l_i-)$, $\hat{\varphi}(r_i)$ and $\hat{\varphi}'(r_i+)$ and $\hat{p}_i$, by placing one knot between l_i and r_i, we can set $\phi(l_i) = \hat{\varphi}(l_i)$, $\phi'(l_i-) = \hat{\varphi}'(l_i-)$, $\phi(r_i) = \hat{\varphi}(r_i)$ and $\phi'(r_i+) = \hat{\varphi}'(r_i+)$ while $\displaystyle\int_{l_i}^{r_i} e^{\phi(x)}\mathrm{d}x = \hat{p}_i$

Let us define a point inside the contribution interval

$$m_k = \frac{\hat{\varphi}(r_i) - \hat{\varphi}'(r_i+)r_i - \hat{\varphi}(l_i) + \hat{\varphi}'(l_i-)l_i}{\hat{\varphi}'(l_i-) - \hat{\varphi}'(r_i+)}.$$

In other words, m_i is the location of the intersection of the linear expansions of $\varphi(x)$ from l_i and r_i. Of all possibilities of φ, p_i is maximized by setting φ to be linear on the intervals $[l_i, m_i]$ and $[m_i, r_i]$ such that $\varphi(m_i)$ is equal to the height of the intersection of linear extension from each side. We can minimize p_i by setting in φ to be linear on $[l_i, r_i]$. This is illustrated in figure 3.3.

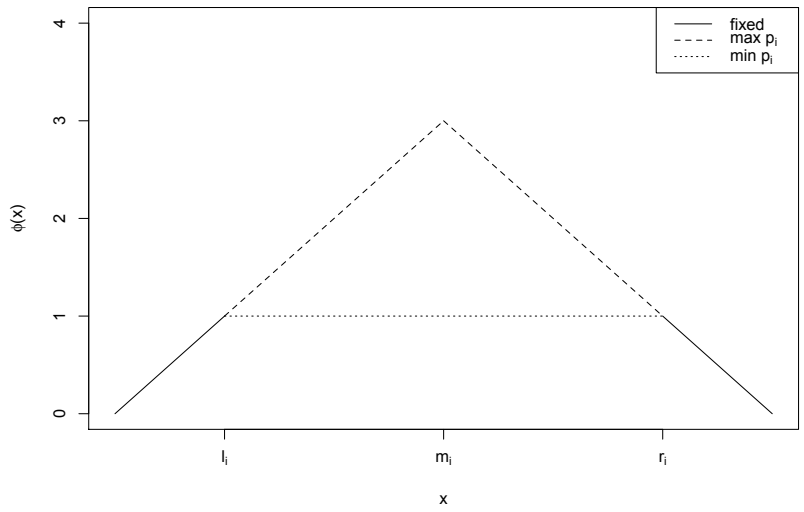

Figure 3.3: Maximizing and Minimizing p_i

Therefore, by adding a knot at m_i, we can set p_i to either the minimum or the maximum possible under the log concavity constraint, with given $\hat{\varphi}(l_i)$, $\hat{\varphi}'(l_i-)$, $\hat{\varphi}(r_i)$ and $\hat{\varphi}'(r_i+)$. In fact, we can set $\int_{l_i}^{r_i} e^{\phi(x)}\mathrm{d}x$ to any value between the minimum and maximum of p_i by setting $\phi(m_i)$ to a suitable value between its minimum and maximum allowed by the constraints. In particular, we will be able to set $\int_{l_i}^{r_i} e^{\phi(x)}\mathrm{d}x = \hat{p}_i$

This does not generalize to the case that $l_i = -\infty$ or $r_i = \infty$, as m_i will not be properly defined by our rule. Let us assume that $r_i = \infty$. In such a case, p_i is maximized by setting φ to be linear over $[l_i, \infty)$ with slope equal to $\hat{\varphi}'(l_i-)$ (if this value is non-negative, we will define $p_i = 1$). We can minimize p_i by setting the slope equal to $-\infty$ (*i.e.* $p_i = 0$). Any value between the minimum and maximum can be

achieved by setting this slope to a suitable value between $\hat{\varphi}'(l_i-)$ and $-\infty$. We can allow ϕ to span such choices of slope by setting $m_i = l_i + 1$. Likewise, if $l_i = -\infty$, we can set $m_i = r_i - 1$.

This implies any likelihood value which can be achieved by an arbitrary log-concave density can also be achieved by a piecewise log linear function with knots at the end points of contribution intervals and one knot inside each contribution interval. This leads to a maximum of $2u - 1$ knots. □

Theorem 2 is important as it clarifies the form of the NPMLE. However, without knowing $\hat{\phi}(x)$, one cannot know the exact location of m_i for the i^{th} contribution interval. In order to deal with this, m_i is replaced with the fixed location $mid_i = \frac{l_i + r_i}{2}$ for the main portion of the algorithm, as mid_i should be close to m_i. Once the solution is sufficiently close, Newton's method will be used to adjust the location of the knots between updates of parameter values.

For the rest of the thesis, the *support set* refers to all end points and mid points of all the contribution intervals.

3.4 Parameterizations

The algorithm presented in this chapter will use an active set algorithm, very similar to the algorithm used in the case of exact times (Rufibach, 2007). In order to clearly define the active set algorithm, several definitions are required. To start with, we define $\beta_i = \phi(x_i)$, where x_i is the i^{th} ordered support point as described in the theorem above. Let k be the number of support points. Because $\phi(x)$ is a linear spline with knots at x_i, the values of $\beta_1, \beta_2, ..., \beta_k$ and $x_1, x_2, ..., x_k$ will completely characterize $\phi(x)$. This will be referred to as the simple parameterization. Occasionally, we will

refer to the location of β_i. This refers to x_i. We will also denote $\Delta_i = \frac{\beta_{i+1}-\beta_i}{x_{i+1}-x_i}$. With this notation, we will note that the constraint of log concavity is equivalent to $\Delta_1 \geq \Delta_2 \geq ... \geq \Delta_{k-1}$. In principal, we would like to define x_i to be active if $\Delta_{i-1} > \Delta_i$ and inactive if $\Delta_{i-1} = \Delta_i$. However, due to numerical errors in calculating Δ_i, we define x_i to be active if $\Delta_{i-1} > \Delta_i + \xi$ and inactive if $\Delta_{i-1} \leq \Delta_i + \xi$. We set $\xi = 10^{-13}$ and found no problems resulting from this.

Similar to the case of exact times (Rufibach 2007), we noticed that the solution tends to have very few active points compared to total number of points considered. We can take advantage of this to create an efficient algorithm using an active set parameterization. Under this parameterization, we treat $\phi(x)$ as a linear spline with knots only at the active points and will adjust the β_i's as such. We will use the notation β_i^* to denote when we are using the active set parameterization. When using the active set parameterization, if we increase the active parameter β_i^*, we also increase the neighboring inactive β_j's as though the active points were the only knots of $\phi(x)$ *i.e.* the inactive β_j are determined by linear interpolation from the nearest active points. To demonstrate this, figure 3.4 demonstrates adding 1 to β_4^*. This starts with β_4 as an inactive point and makes it active as ϕ is now kinked at x_4 . It also increases the values of β_3 and β_5, as they are the surrounding inactive points.

To formally characterize addition under the active set parameterization, define $a(m)$ to be the index of the m^{th} active point. If $i = a(m)$ then $\beta_i^{*(t+1)} = \beta_i^{*(t)} + h$ is equivalent to

$$\beta_j^{(t+1)} = \begin{cases} \beta_j^{(t)} + h \times \frac{x_j - x_{a(m-1)}}{x_i - x_{a(m-1)}}, & \text{if } x_{a(m-1)} < x_j \leq x_i \\ \beta_j^{(t)} + h \times \frac{x_{a(m+1)} - x_j}{x_{a(m+1)} - x_i}, & \text{if } x_i < x_j < x_{a(m+1)} \\ \beta_j^{(t)}, & \text{otherwise} \end{cases}$$

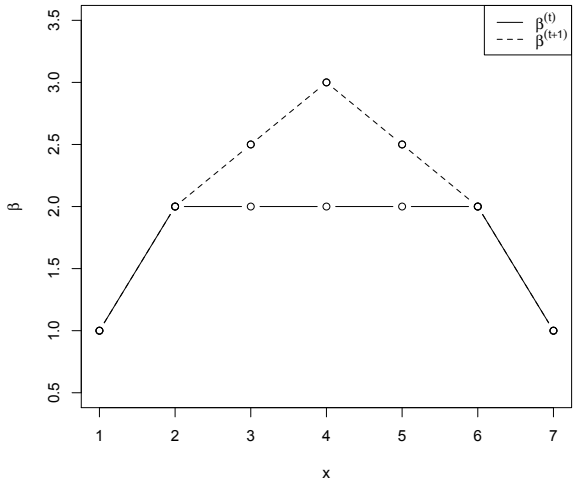

Figure 3.4: $\beta_4^{*(t+1)} = \beta_4^{*(t)} + 1$

Under the active set parameterization, all active points have a neighborhood for which they can both increase and decrease without violating the condition of log-concavity. Inactive points have a neighborhood in which they can increase and become an active point, but cannot decrease. In the simple parameterization, inactive points cannot decrease and all points can increase only if both neighboring points are active. For example, in figure 3.4, $\beta_4^{(t+1)} = \beta_4^{(t)} + 1$ violates log concavity, but $\beta_4^{*(t+1)} = \beta_4^{*(t)} + 1$ does not.

In a slight abuse of notation, let us also define $\Delta_{a(i)} = \frac{\beta_{a(i+1)} - \beta_{a(i)}}{x_{a(i+1)} - x_{a(i)}}$. If K = number of active points, then $\Delta_1 \geq \Delta_2 \geq ... \geq \Delta_{k-1}$ is equal to $\Delta_{a(1)} \geq \Delta_{a(2)} \geq ... \geq \Delta_{a(K-1)}$.

3.5 Stopping Criterion

We discuss the stopping criterion first before presenting the actual iterative algorithm because this discussion clarifies how we characterize the NPMLE, given that it is piecewise log-linear. Because of the log-concavity constraints, it is natural to consider the KKT conditions (Kuhn and Tucker 1951) which are necessary for an estimate to be a local maximum. Let us define

$$\text{KKT error} = \max \begin{cases} |\frac{\partial \ell}{\partial \beta_i^*}|, & \text{if } \Delta_{i-1} > \Delta_i + \xi \\ \max_i(\frac{\partial \ell}{\partial \beta_i^*}, 0), & \text{if } \Delta_{i-1} \leq \Delta_i + \xi \end{cases}$$

In other words, the error is the maximum of the absolute value of the derivatives associated with the active set parameterization of all the active points and the positive derivatives associated with the active set parameterization of the inactive points (because decreasing would violate the log concavity restraint).

After the KKT error requirement is met for fixed location of the knots, we allow the knots to move. When this happens, we will have two sources of error: KKT error and error associated with the location of the knots at the active points. Because active points always contain a neighborhood for which the knot can move in either direction, the location error will be the maximum absolute value of the derivative of the log likelihood function as a function of location of each of the active points. This will be referred to as "location error". At this point in the algorithm, err = max(KKT error, location error).

The algorithm terminates when either error < tolerance or iteration > max iterations. We set the tolerance to 10^{-4}. The reason this value was selected was setting the

tolerance to 10^{-5} usually required step sizes in the order of 10^{-6}, which was smaller than the precision of the quadratic programming package we used (see section 3.6.3 for more discussion). If greater precision was desired, the algorithm still achieves this as the univariate step of the algorithm functions properly even for extremely small step sizes. However, this is much more computationally inefficient and it was decided that the increase in precision was not worth the computational cost. In order to check that using tolerance 10^{-4} resulted in solutions sufficiently close to the mode, we simulated 200 data sets of size $n = 200$. In each data set, we set tolerance first to 10^{-4} and then 10^{-5} and compared final likelihoods of our solutions. In these data sets, it was found that the median increase from using tolerance 10^{-5} was 1.6×10^{-9} and the maximum increase was found to be 0.0020.

It is important to note that because the likelihood function is not always concave and potentially could be multi-modal, our stopping criterion is only a necessary condition for the global maximum, not a sufficient condition. However, when the algorithm was started from random starting points, it would always converge to the same solutions, suggesting the issue of potential multi-modality is not too severe.

3.6 Active Set Algorithm

3.6.1 Algorithm Outline

The algorithm includes four steps: one which selects new points to add to the set of active points (similar to the VDM algorithm of Fedorov 1972), one which efficiently increases the likelihood over the set of currently active points (similar to the methods used to compute the log-concave NPMLE with exact observations found in Dümbgen *et al.* 2011), one that fixes the tails and one that moves the location of the active points. The first step selects the index i with the maximal error and uses simple

univariate techniques to find an optimal β_i^*. The second step optimizes over the active set proceeds by approximating the log-likelihood function with a second order Taylor expansion, and then maximizing this approximation using quadratic programming, similar to the ICM algorithm (Jongbloed 1998). Because often $\beta_i = -\infty$ on the tails at the solution, it is occasionally necessary to adjust the tails of β for numerical stability so this is done in the fixing tails step. Finally, after the solution is sufficiently close, we allow the location of the active points to move via Newton's method. The basic form of the algorithm is as follows.

- Set initial values for β
- Set MOVEX = FALSE
- While ($err > \epsilon$ and $t <$ max iterations)

  {

  - Set $t = t + 1$
  - Select index i with maximal KKT error
  - Use univariate optimization to update β_i^*
  - Use quadratic programming to optimize over active set
  - Fix tails of β if necessary
  - Calculate err = KKT error
  - If($err < \epsilon$)
    - MOVEX = TRUE
  - If(MOVEX == TRUE)

    {

    - Use bivariate Newton's method to update knot location of active points which are mid points

* Calculate $err = \max(\text{KKT error, location error})$

}

}

- Convert β to $e^{\phi(x)} / \int e^{\phi(x)} \mathrm{d}x$
- Return $e^{\phi(x)} / \int e^{\phi(x)} \mathrm{d}x$

It is important to note that after the MOVEX part of the algorithm is activated, we will still need to update all values of β_i, as moving the location of the knots will have a change in the constraints. Because of this, when we calculate $err = \max(\text{KKT error, location error})$, we must recalculate the KKT error rather than reusing the earlier calculated KKT error in the earlier step of the algorithm.

There is one technical note about the initial values of β. It is very tempting to start with a uniform set, *i.e.* $\beta_1 = ... = \beta_k = 0$. However, if right censoring is infinite ($R_i = \infty$ or $L_i = -\infty$), this will result in an improper distribution, as the distribution Uniform(a, ∞) is improper. To insure the initial distribution is properly defined, we set $m = \frac{k+1}{2}$ and set $\beta_i = -|x_m - x_i|/\sigma_x^*$, where σ_x^* is the standard deviation of all finite support point locations. This leads to a Laplace distribution if defined on $\mathbb{R}$ and a truncated Laplace distribution otherwise. In general, we found that different valid starting values lead to the same results.

3.6.2 Univariate Optimization

Our algorithm selects the index i associated with the maximum KKT error. Once the index i is selected, β_i^* must be selected which increases the likelihood. Let $i = a(j)$

(if i is not an active point, it will be after optimization). From the constraints $\Delta_{a(j-2)} \geq \Delta_{a(j-1)}$, $\Delta_{a(j-1)} \geq \Delta_{a(j)}$ and $\Delta_{a(j)} \geq \Delta_{a(j+1)}$, we derive the constraints

$$\beta_i^* \leq \min\left(\beta_{a(j-1)} + \Delta_{a(j-2)} \times (x_{a(j)} - x_{a(j-1)}), \beta_{a(j+1)} - \Delta_{a(j+1)} \times (x_{a(j+1)} - x_{a(j)})\right)$$

$$\beta_i^* \geq \left(\frac{1}{x_{a(j+1)} - x_{a(j)}} + \frac{1}{x_{a(j)} - x_{a(j-1)}}\right)^{-1} \times \left(\frac{\beta_{a(j+1)}}{x_{a(j+1)} - x_{a(j)}} + \frac{\beta_{a(j-1)}}{x_{a(j)} - x_{a(j-1)}}\right)$$

Constraints which involve an undefined index, such as $a(0)$, can be ignored. If $\frac{d^2\ell}{d(\beta_i^*)^2} < 0$, then Newton's method was used, subject to the constraints. Half stepping was used to insure monotonic convergence. Rarely it was observed that $\frac{d^2\ell}{d(\beta_i^*)^2} \geq 0$, in which case the bisection method was used to update β_i^*. Although $\ell(\beta_i^*)$ was observed to be non-concave, it was not observed to be multimodal.

Much like the VDM algorithm, these steps alone insure that the algorithm will reach a local maximum, but is observed to do so very slowly. Using simulated data with $n = 200$, we observed that using these steps alone frequently failed to meet the convergence criterion after 1000 iterations, implying that an algorithm based only on univariate optimization would be insufficient. In section 3.6.3, we present a step that updates all active points simultaneously and greatly accelerates the algorithm. While the multivariate step can be forced to never decrease the likelihood function, the multivariate step alone is not insured to find a local maximum. Thus, we will include

both the univariate and multivariate steps in our algorithm to insure convergence.

3.6.3 Maximizing Over an Active Set

Once given an active set of points, we need a step which can maximize over the active set efficiently. Because of the linear constraints of concavity, Newton's method is difficult to apply as it would not respect the boundaries. Instead, an ICM (Jongbloed 1998) algorithm is used to maximize the over the active set, similar to the algorithm used in the exact case (Dümbgen *et al.*, 2011). An ICM algorithm works by approximating the target function with a second order Taylor expansion, in which the off diagonal partial derivatives are ignored. In this application, the parameters considered will be the log density at the active points, *i.e.* β_i's. It is worth noting that in the traditional ICM algorithm, the off diagonals are ignored due to the expense of computation. In this case the number of parameters considered is actually fairly low, making the number of off diagonals more manageable, but we have other reasons to ignore the off diagonals which will be discussed shortly. This approximation is maximized, according to the linear constraints, via quadratic programming. We will use the notation that a quadratic program minimizes

$$\frac{1}{2}d^T Q d + c^T d$$

Under the constraint

$$Ad \leq b$$

Because quadratic programming minimizes a program, we will minimize the Taylor

approximation of $-\ell(\beta^*)$), with the off diagonals of the Hessian ignored. In order to have $\frac{1}{2}d^TQd + c^Td$ be the given approximation, we define

$$d_i = \beta^{*(t+1)}_{a(i)} - \beta^{*(t)}_{a(i)}$$

$$Q_{i,i} = -\frac{\partial^2 \ell(\beta^{*(t)})}{\partial {\beta^{*(t)}_{a(i)}}^2}$$

$$Q_{i,j(i\neq j)} = 0$$

$$c_i = -\frac{\partial \ell(\beta^{*(t)})}{\partial \beta^{*(t)}_{a(i)}}$$

In order to preserve the constraint of $\Delta_{a(i)} \geq \Delta_{a(i+1)}$, we set

$$A_{i,i} = \frac{1}{x_{a(i+1)} - x_{a(i)}}$$

$$A_{i,i+1} = \frac{-1}{x_{a(i+1)} - x_{a(i)}} + \frac{-1}{x_{a(i+2)} - x_{a(i+1)}}$$

$$A_{i,i+2} = \frac{1}{x_{a(i+2)} - x_{a(i+1)}}$$

$$b_i = \Delta^{(t)}_{a(i+1)} - \Delta^{(t)}_{a(i)}$$

Similar to the ICM algorithm (Jongbloed 1998), half steps will be taken to insure the likelihood function does not decrease. In other words, if $\ell(\beta^{*(t)} + d) < \ell(\beta^{*(t)})$, then d is replaced with $d/2$ until $\ell(\beta^{*(t)} + d) \geq \ell(\beta^{*(t)})$. It was observed that these half steps were required very infrequently. However, if the tolerance was set very low, such as 10^{-6}, often this step would fail to increase the likelihood function, as the tolerance for the quadratic solver appeared larger than the necessary step size (the "solve.QP" function from the R package "quadprog" was used. Failure typically happened when the largest component of d (before half stepping) was on the order of 10^{-6}. In such cases, the univariate steps would still work and the algorithm was still observed to converge fairly quickly).

Perhaps the most novel part of this algorithm is in how we deal with the fact that $\ell(\beta^*)$ may not be locally concave. In particular, the quadratic function used to approximate the likelihood function will be unbounded if the Hessian is non-negative definite, leading to a degenerate purposed step. Because the ICM algorithm ignores

the off diagonals of the Hessian matrix, the only case of concern is when the diagonals of the Hessian are not negative. When the off diagonals are included in the quadratic approximation of the likelihood function, the quadratic approximation was much more likely to be unbounded. In order to remedy this, an approximation to the second derivative was used which would be insured to be negative. This approximation takes advantage of the fact that the likelihood function is bounded from above.

Suppose that the likelihood function is not locally concave as a function of one of the active points, *i.e.* $\frac{\partial^2 \ell}{\partial (\beta^{*(t)}_{a(i)})^2} \geq 0$. Let $\tilde{\beta}^*_{a(i)}$ be the value of $\beta^*_{a(i)}$ that maximizes $\ell(\beta^*_{a(i)})$ in the direction of $\frac{\partial \ell}{\partial \beta^{*(t)}_{a(i)}}$ (*i.e.* if $\frac{\partial \ell}{\partial \beta^{*(t)}_{a(i)}} > 0$ only consider $\tilde{\beta}^*_{a(i)} > \beta^{*(t)}_{a(i)}$, and if $\frac{\partial \ell}{\partial \beta^{*(t)}_{a(i)}} < 0$ only consider $\tilde{\beta}^*_{a(i)} < \beta^{*(t)}_{a(i)}$) with all other $\beta_{a(j)}$ held fixed. It is worth noting that we don't require $\tilde{\beta}^*_{a(i)}$ to respect the boundary set by the constraint of log concavity. If we then consider approximating $\ell(\beta^*_{a(i)})$ with a quadratic function whose first derivative at $\beta^*_{a(i)}$ is $\frac{\partial \ell}{\partial \beta^{*(t)}_{a(i)}}$ and whose maximum is reached at $\tilde{\beta}^*_{a(i)}$, this would imply that our approximation of the second derivative would be

$$\frac{-\left(\frac{\partial \ell}{\partial \beta^{*(t)}_{a(i)}}\right)}{\tilde{\beta}^*_{a(i)} - \beta^{*(t)}_{a(i)}}$$

This is guaranteed not to be positive.

In the case of a concave likelihood function, half stepping insures that the likelihood function will increase. Because our target function is not locally concave when using this approximation, even half stepping does not insure the likelihood function will increase. If after sufficient half steps (we chose 5), the likelihood still does not increase, this step of the algorithm is skipped. Theoretically, if the ICM step repeatedly failed, the algorithm could be extremely slow due to relying only on the univariate optimiza-

tion. However, this was not observed to occur and substituting this approximation into the ICM algorithm appears to work very well, usually reducing the number of iterations required to below 100 for simulated data of size $n = 200$. While finding $\tilde{\beta}^*_{a(i)}$ does come with some computational cost, this approximation was required infrequently and could be done efficiently enough that the computational costs were not significant in the speed of the algorithm.

3.6.4 Moving the Knots

Recall that we used mid_k as the location of the knots in the center of each contribution interval because we don't know the exact location defined as m_k in our theorem. To optimize the position of the knots, a bivariate Newton's method was used, with half-stepping of proposed steps to insure the new proposed estimate increases the likelihood function and respects the constraints. Two parameters considered were the location of the knot (*i.e.* x_i) and the log density at the knot (*i.e.* β_i). While the log likelihood function does appear consistently to be concave as a function of the location of the knots, we already know that the log likelihood is not always concave as a function of β_i. In the case where the Hessian matrix is not negative definite, we will perform Newton's method on only the location of the knot, as empirically this was always found to be locally concave. The algorithm includes a bisection step, should the likelihood function be non concave as a function of the knot location, but we have not observed it to be necessary.

It was found that the most computationally efficient way to implement this moving of the knots was to first let the algorithm run without moving the knots until the KKT error was below the tolerance. Once this occurred, then we would add the bivariate Newton's method into the algorithm, alongside both the univariate and the ICM steps, until both the KKT error and location error were below the tolerance.

Once this was satisfied, the algorithm was considered converged.

3.6.5 Fixing the Tails

When computing the log-concave NPMLE with exact observations, the area which must have positive mass is known in advance (*i.e.* the range of all observed times). When computing the log-concave NPMLE for interval censored data, this not the case. There will be several contribution intervals which receive 0 mass at the log-concave NPMLE. A trivial example to consider is if the data consisted of two overlapping intervals. The log-concave NPMLE would place all the mass in the overlap and no mass to the other two surrounding contribution intervals. In the case of the unconstrained NPMLE, the mass is known to be positive only in the set of Turnbull intervals. However, the log-concave constraint implies that this is not the case and often some of the contribution intervals outside of the minimum and maximum Turnbull intervals will receive positive mass at the log-concave NPMLE, while others will not.

Because of this, simply allowing the earlier steps to find the log densities at the tails of the estimated density can lead to numerical instability as $\phi(x) \to -\infty$. Numerical instability typically happened in the range of $\phi(x) = -1000$ for some x, as derivatives and second derivatives approached 0. Along with numerical issues there are computational costs. Consider that if $\phi(x) = -10$ for some x, then the density is 4.5×10^{-5} times the density at the mode, indicating $\phi(x)$ is very likely $-\infty$ at $\hat{\phi}(x)$. Even if numerical issues were not observed, allowing the algorithm to meet the stopping criterion can be very computationally expensive for a rather trivial transformation of $\phi(x)$.

To cope with this, our algorithm would periodically check if setting $\phi(x)$ to $-\infty$ increases the likelihood for x on the tails if $\phi(x) < -t$ for some threshold t. If this increased the likelihood, the β_i associated would be set to $-\infty$ and either β_{i+1} or β_{i-1}

(depending on which tail) would be checked next. If setting $\beta_i = -\infty$ did not increase the likelihood, the values would remain unchanged and this part of the algorithm would terminate. Interestingly, choosing t too small leads to local maxes which can be considerably less than the global max, even if optimization techniques are used to check if adding mass back to the tails which have been set to $-\infty$ could increase the likelihood. For poorly chosen t, such as $t = 1$, the difference in log likelihood was observed to be as high as 1.5 and the mass could be significantly different on the tails, despite the convergence criterion being satisfied. On the other hand, setting very large t, such as 500, was found to increase the number of iterations required several fold. Fortunately, there appears to be a large region for choices of t such that neither issues occurred. The performance across $t = 1$, 5, 10, 20 and 40 was examined. While it was found that $t = 10$ appeared satisfactory, for robustness $t = 20$ was selected as it led to a small increase in average iterations (about 15% more). With this fix, it was found that even with random starting values, the algorithm only led to the same estimate.

3.7 Algorithm Speeds

To investigate the speed of our algorithm, we computed the log-concave NPMLE across different scenarios. The complexity of each step of the algorithm is not only a function of sample size, but also the number of unique times in the data. Each step is of order $O(u^2)$, where u is the number of unique times. While given u, n does not affect the complexity of each iteration, we see that the number of iteration increases with n for a fixed u. Empirically, it appears the number of iterations required may be of order $O(\sqrt{n})$. On Table 3.1, we present average computation time in seconds across different sample sizes and different numbers of unique times in the data using simulated current status data in which both the event time and inspection time was

simulated from gamma(2,2) distribution. Binning was used to create the number of unique times.

	Unique Times						
n	10	50	100	500	1000	2000	5000
100	0.07	0.15	0.17	NA	NA	NA	NA
500	0.19	0.27	0.46	0.86	2.44	NA	NA
5000	0.42	0.86	1.11	5.07	7.59	31.6	156

Table 3.1: Average computation times in seconds for our algorithm

We compared this to the R package logconcens. We found that the algorithm in logconcens frequently failed to converge. We simulated datasets in the same manner as above, except that we only considered sample sizes $n = 25$, 50 and 100. In addition to average times, we present the proportion of datasets for which the algorithm failed to converge after 1,000 iterations.

	Unique Times		
n	10	25	50
25	9.29 (0.26)	20.4 (0.37)	NA
50	14.0 (0.15)	21.7 (0.27)	47.9 (0.67)
100	16.9 (0.20)	37.1 (0.43)	56.1 (0.47)

Table 3.2: Average computation times in seconds for logconcens. Values in parentheses are proportion of datasets failed to converge after 1,000 iterations

3.8 Illustrative Example

For a illustrative example, we will borrow data from MacMahon and Worcestor (1966).

Questionnaires were collected from $n = 2423$ participants regarding age at menopause. Because several of the subjects had not experienced menopause yet, this data set contained right censored data. However, MacMahon and Worcestor (1966) found that there was a marked terminal digit clustering in the response of reported time of menopause. Because of this, Krailo and Pike (1983) recommended only using the menopausal status of women at the time of the questionnaire, thus resulting in current status data. The data also contains two types of menopause; operative menopause and natural menopause.

Earlier analyses of this data used a competing risks model (Jewell *et al.* 2003). For demonstrative purposes, we will only examine the time to menopause, regardless of the type of menopause. For this example, we will examine the estimated densities and survival curves, although clearly it would be simple to also examine the estimated hazard as well. We wanted to compare the log-concave estimate to the unconstrained NPMLE, the logspline estimator and the kernel density estimator. However, using the standard settings found in the CRAN logspline package, the logspline estimator failed to converge with this dataset, a common problem for the logspline estimator with current status data. When using the kernel smoother, the problem of selecting bandwidth was non trivial. Braun *et al.* (2005) suggest using cross validation for selecting bandwidth. With a dataset of this size and current software options for the kernel smoother, this is not an option. Instead, midpoint imputation was used to determine bandwidth, as demonstrated in the CRAN ICE package. Other ad hoc fixes used were left endpoint and right endpoint imputation. All of these lead to relatively close bandwidths, which lead to very similar estimates.

Plotted estimates can be seen in figure 3.5 for the menopause data. We see is that the log-concave NPMLE has a much more jagged density estimator than the kernel smoother. This can be seen as a disadvantage for the log-concave NPMLE where

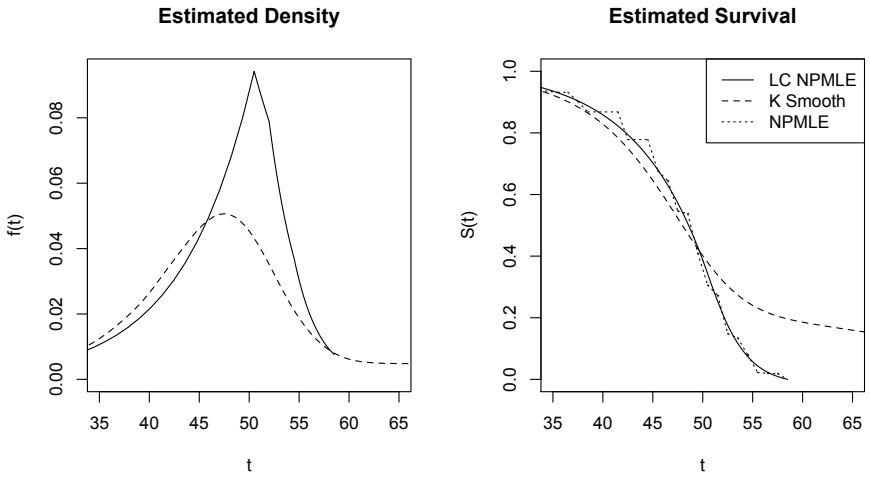

Figure 3.5: Estimated Functions

smoothness of the estimated density is a priority. There is no unconstrained NPMLE density estimate, as the unconstrained NPMLE does not provide valid density estimates. We also note that while the log-concave NPMLE and the unconstrained NPMLE survival estimates appear consistent with each other, there is a large disagreement with those estimates and the kernel smoother's. The log-concave NPMLE places 0 masses beyond t = 58.5, while the kernel smoother places mass as far as t = 100. The NPMLEs appear to agree with what is known about menopause, while the kernel smoother appears to be a less accurate portrayal of the distribution of time to menopause. For example, Treloar 1981 presents data from a longitudinal study. Excluding the cases lost to follow up, all 729 cases had experienced menopause by age 59 (and only one had experienced menopause after 58). However, the kernel smoother estimates that $S(59) = 0.19$. In contrast, both the log-concave NPMLE and unconstrained NPMLE place 0 mass beyond 58.5, which agree much better with Treloar's findings. The current implementation of the LC NPMLE algorithm took 0.57 seconds (221 iterations) to converge, although it is very important to note that this the best case scenario for a data set of this size (see appendix B on acceleration from ties). The algorithm in the logconcens package took 942 seconds to converge.

The kernel smoother took 16.5 seconds and the unconstrained NPMLE algorithm, as implemented in the CRAN package MLEcens, took 0.139 seconds to converge.

3.9 Simulations

Now that a reasonably fast algorithm has been created for computing the log-concave NPMLE, the finite sample size operating characteristics can be examined and compared to the competing estimators. In order to examine which estimator performed best in estimating quantiles, the bias and standard deviation of the estimated 0.1, 0.25, 0.5, 0.75 and 0.9 quantiles were compared across the estimators.

Current status data was used, as this simplified the process of censoring. The true time T was simulated along with the censoring time C. For simplicity, C followed the same distribution as T. The only information kept was C and whether the event was right or left censored.

A variety of different distributions were used to examine how the estimators worked in different scenarios. The distributions tested were gamma(shape = 2, rate = 2), gamma(shape = 100, rate = 2), weibull(shape = 6, scale = 4), lognormal(μ = 0, σ = 1) and a gamma mixture with p = (0.5, 0.5), component 1 = gamma(shape = 2, rate = 2), component 2 = gamma(shape = 5, rate = 1). It should be noted that the last two simulations violate the assumption of log concavity; the log normal distribution is mildly non log-concave due to heavy tails, while the gamma mixture model is heavily non log-concave due to bimodality. In each of the simulations, datasets were generated with n = 50, 200 and 800. For n = 50 and 200, MC = 400 simulations were generated. For n = 800, MC = 100 simulations were generated.

Tables of the results are given in Appendix C. We refer to the log-concave NPMLE as the LC NPMLE and the unconstrained NPMLE as the UC NPMLE. Some general

trends that were noted from the simulations were:

- For log-concave data, the LC NPMLE always outperforms the UC NPMLE in both bias and standard deviation.
- For log-concave data, the LC NPMLE consistently outperforms the kernel smoother in terms of bias in quantile estimation. Quite often, the bias from the kernel smoother was non-trivial.
- The kernel smoother shows extreme bias when the censored intervals cover vast regions where the density is very close to 0, such as the gamma (100,2) case. Again, this bias from the kernel smoother appears to increase as n increases. This bias is thought to be the cause of the discrepancy in estimates for the illustrative example.
- For log-concave data, the bias approaches 0 fairly quickly for the LC NPMLE. For samples over 200, the largest bias to standard deviation ratio was 0.4, although for $n = 50$, we observe a bias to standard deviation ratio of 0.7.
- For log-concave data, the LC NPMLE has higher standard deviation than the kernel smoother in small data sets. This trend reversed in larger datasets
- For log normal data, significant bias was seen in the estimation of the upper tail for the LC NPMLE. While this bias decreased as n increased, it did not appear to be converging to 0. While the UC NPMLE showed significant bias in small samples, the bias appeared to be converging to 0. The kernel smoother observed heavier bias than the LC NPMLE in estimating all cases except estimating the 0.9 quartile with $n = 50$.
- For mixture gamma data, significant bias was seen for almost all quantiles for the LC NPMLE, although this was about equal for the kernel smoother. While

the UC NPMLE suffered from significant bias in $n = 50$, by $n = 200$, these biases were mostly insignificant.

- For density estimation, the trend was very consistent; the log-concave NPMLE displayed lower bias but significantly higher standard deviation than the kernel smoother. The bias from the kernel smoother was often substantial. This bias often did not decrease with an increase in sample size.

- While the log-concave NPMLE did well in terms of bias, the standard deviation was often so high that it would be unreasonable to use for density estimation unless the data set was quite large ($n \geq 800$).

These simulation results suggest that the log-concave NPMLE would be the estimator of choice for quantile estimation with current status data that was believed to be log-concave or only mildly non log-concave. The advantage of the log-concave estimator increases as n increases. In large data sets or moderately sized data sets with light case II interval censoring, the log-concave NPMLE may be reasonable for density estimation, but it is not recommended for smaller data sets ($n < 800$ for current status data).

3.10 Future Work

In the case of the log-concave NPMLE for exact data, uniqueness is shown using the fact that the log likelihood is strictly concave (Rufibach 2007). This cannot be applied to the interval censored case, as the log likelihood function is not always concave. In the case of the univariate unconstrained NPMLE, it has been shown that the solution does not have mixture non-uniqueness (for a solution set of intervals, there is only one assignment of mass to each interval which maximizes the likelihood function) but

does suffer from representational non-uniqueness (for a mixed mass and interval, any assignment of mass within the interval leads to the same likelihood, Gentleman and Vandal 2001). The proof depends on the fact that the solution only assigns mass to the Turnbull intervals. In trivial problems, such as a single censored observation, it is clear that the log concave NPMLE can show representational non-uniqueness. However, it is not clear whether the estimator suffers from mixture non-uniqueness.

Although the algorithm presented in this chapter is acceptable for small to moderate sized data sets or larger data sets with large amounts of ties (see appendix), it would be too slow for larger data sets with large amounts of unique values. For example, in our simulations we found that if $n = 5,000$, all with unique times, the algorithm look over 2 minutes to converge on average (see table 3.1). In contrast, the CRAN package "MLEcens" can compute the unconstrained NPMLE for $n = 5,000$ in 3.85 seconds on average.

In section 3.6.5, it is noted that the algorithm can find local maxima, returning an estimate which can differ significantly from the true MLE on the tails. While an ad hoc fix is presented, it would be preferable to find a transformation of $\phi(x)$ such that the estimate could step away from the local max and toward the global max. Such a step would likely have to consider both the length of the tails of $\phi(x)$ and the log density of the tails of $\phi(x)$ simultaneously. Adding such a step would likely both accelerate the algorithm and help insure convergence to the global maximum.

Chapter 4

Inference for the Log-concave NPMLE

In Chapter 3, we presented a new algorithm for finding the log-concave NPMLE for interval censored data. We demonstrated that applying this new shape constraint significantly reduced the variance of the survival estimates in comparison to the unconstrained NPMLE.

In this chapter, we present methods for inference for the log-concave NPMLE for current status data. In section 4.1, we present a goodness of fit test for inspecting the validity of the log-concave assumption and investigate the power via Monte Carlo simulation. In section 4.2, we present two methods for construction of survival estimates using the log-concave NPMLE and compare their performance with the unconstrained NPMLE. In section 4.3, we present a Cox PH model with a log-concave baseline along with methods for confidence intervals, both for regression parameters and Cox PH survival estimates and compare with the unconstrained Cox PH model.

At this time, we have not implemented any inference methods for density estimation.

4.1 Goodness of Fit Tests

In comparison with the unconstrained NPMLE, the strength of the log-concave NPMLE is the reduced variance of survival estimates and properly defined density estimates. The disadvantage of the log-concave NPMLE is there is an assumption about the data being made which can lead to bias if the assumption is inappropriate. Because of this, it is very natural to want to formally test the validity of the assumption. In particular, we should be concerned with either the true distribution having multiple distinct modes or heavier tails than allowed by the constraint of log concavity (*i.e.* super exponential).

At this time, there is fairly little literature on goodness of fit tests for shape constrained estimation, especially for the log-concave assumption. Meister (2009) presents a test of local monotonicity of a function, although this does not generalize to a test of log concavity. Carroll *et al.* (2011) present a generalized test of shape constraints, which can be applied to the assumption of log concavity. This involves reweighting the observations and applying a kernel smoother such that a.) the estimate provided by the kernel smoother meets the shape constraints and b.) some measure of distance is minimized between the new weights and the uniform n^{-1} weights. A goodness of fit test is then based on comparing the minimum distance required to make the kernel smoothed estimate follow the shape constraint to the distribution of this distance under the null hypothesis.

Currently, there is no implementation of Carroll's work for interval censored data. While further investigation of such a test is warranted, there are reasons to believe it will not work as well for interval censored data. In particular, the simulations in chapter 3 show that using current suggested settings (*i.e.* bandwidth selection), the kernel smoother for interval censored data can lead to very heavy bias under certain

situations. Using a potential biased estimator for a goodness of fit test seems untrustworthy, especially given that the nature of the bias is currently not well understood.

In this section, we examine two alternative goodness of fit tests for current status data. Both of these tests are based on likelihood ratio tests of properly nested models. The first of these tests compares the likelihood of the unconstrained NPMLE to that of the log-concave NPMLE, as the models are properly nested and can be easily calculated without any new software (besides that presented in chapter 3). We found this test to be unsatisfactory in terms of power in certain conditions and so we propose a second test, in which we compare the log-concave NPMLE to a mixture model which uses two log-concave components. This new test significantly improves the power in all cases considered, although it still demonstrated low power in the case of a multimodal distribution with more than two modes. In application, we suggest both the use of this test and visual comparison of the unconstrained NPMLE and log-concave survival curve.

4.1.1 Log-concave vs. Unconstrained NPMLE Likelihood Ratio Test

The concept of the likelihood ratio test for the log-concave NPMLE vs the unconstrained NPMLE is straightforward, following a standard likelihood ratio test in which the log-concave NPMLE is the nested model within the unconstrained NPMLE. However, the null distribution of the test statistic does not follow any known distribution because the null hypothesis is on the boundary (Protassov *et. al.* 2002). This is further complicated by the fact that the estimators used are non-parametric, and so while the null hypothesis is properly nested within the alternative hypothesis, the difference in the dimension of the parameter space between in the two models is

undefined.

We use the same approach presented in Protassov *et. al.* (2002), which is to use Monte Carlo simulations to estimate the distribution of the test statistic under the null hypothesis. This is slightly more complicated in our problem, as both the distribution of the event times and the distribution of the censoring times will affect the likelihood. For simplicity, we focus on current status data, as it is easier to model the distribution of the censoring mechanism.

To implement the likelihood ratio test, we first find the log-concave NPMLE and the unconstrained NPMLE. We calculate

$$LR = -2\ell_{LC} + 2\ell_{UC}$$

where ℓ_{LC} = likelihood for the log-concave NPMLE and ℓ_{UC} = likelihood for the unconstrained NPMLE. This gives us our test statistic.

To sample a test statistic under the null hypothesis, we first sample n event times from a distribution for which the null hypothesis is true. The simplest way to do this is to draw n samples from the estimated log-concave NPMLE distribution. Then we censor these n times in a manner similar to the true censoring distribution. For current status data, this is quite easy. The only information necessary is the distribution of the inspection times, f_C. Because C is observed exactly, modeling of C is an easier problem than modeling the event times. In fact, we suggest using all n observed inspection times directly from the dataset itself. Using these inspection times, we censor our event times according to a current status mechanism. In other words, if T_i is the i^{th} sampled event time from the null distribution and C_i is the i^{th} inspection

time in the original dataset, then we set the i^{th} sampled interval to $(0, C_i)$ if $T_i < C_i$ and (C_i, ∞) if $T_i \geq C_i$.

We note that the power of the implemented test consists of two mechanisms. One is the difference in distribution of the log likelihood ratio under the null compared with under the alternative hypothesis. In particular, setting $\alpha = 0.05$, the power of the test is equal to

$$Pr(lr_a > q_{0.95})$$

where lr_a is the log likelihood ratio under the alternative hypothesis and $q_{0.95}$ is the 95^{th} quantile of the likelihood ratio under the null hypothesis. We call this the *potential power* of the test. In practice, we don't know what $q_{0.95}$ is so we use a bootstrap estimate of this value for each dataset. This leads to a further complication in that we may not be able to estimate $q_{0.95}$ well using bootstrap methods. For example, if the distribution of the likelihood ratio under the null hypothesis was highly sensitive to small changes to the true event distribution, the sampled bootstrap distribution of the test statistic may change significantly based on estimation error of the underlying distribution. Because this, we call the power observed by fully implementing the bootstrap method the *observed power*. Using Monte Carlo simulation to estimate the potential power is considerably cheaper than estimating the observed power, even though the observed power is what we are more interested in. We believe the potential power should be very close to the observed power, so we used the potential power to explore the operating characteristics of our tests. We used simulations to confirm that the potential power is approximately equal to the observed power, although this will be done on a smaller scale.

Case II interval censored data is much more difficult to model. In some cases, the censoring scheme is known in advance due to a planned inspection process. Assuming the true inspection times followed the distribution dictated by the study design, an investigator could censor the sampled event times under the null hypothesis following the distribution given by the study design. In other cases, even if distribution of the inspection process was not known exactly, the records of the inspection process many still exist and be helpful in modeling the censoring mechanism. If only the censoring intervals themselves are known to the investigator, modeling the censoring mechanism can be very difficult without making strong assumptions. Further work is required to examine how sensitive this test can be to various necessary assumptions. For now, we only consider the case of current status data.

While theoretically justified, we found the likelihood ratio test based on the unconstrained NPMLE to have unsatisfactory power for current status data, especially in the case of multimodal data. We came to this conclusion due to the results of Monte Carlo simulations, which will be discussed in more detail in section 4.1.3. We believe the poor behavior of this likelihood ratio test is due to the unconstrained NPMLE being too flexible. In particular, the unconstrained NPMLE allows for "unsmooth" estimated survival functions. This can have a large impact on the likelihood function, and yet most researchers are willing to confidently assume that the survival function is smooth (although accurately defining "smooth" can be quite difficult).

4.1.2 Log-concave vs. Mixture Log-concave

The low power of the previous test motivated us to try a likelihood ratio test based on an estimator which properly contained the log-concave NPMLE but was still restricted to be fairly smooth. A model we propose is a mixture model in which the components

are log-concave. This leads to a fairly smooth estimator which properly contains the single component log-concave estimator. With standard mixture models, selection of the number of components is often a difficult question. Because we are merely trying to test the fit of the single component model rather than properly estimate the underlying distribution, we will use a two component model to examine how much an extra component can help the fit, rather than attempting to find a selection rule for finding the optimal number of components. This removes the need for selection of a complexity parameter and greatly reduces the computational costs. Because adding further components tend to lead to a smaller and smaller increase in the likelihood function, only using two components is likely to have very similar power when compared to any sort of selection process, while both simplifying the process for the investigator and reducing the computational costs to a more manageable level.

The log-concave vs. mixture log-concave likelihood ratio test was implemented in the same manner as the likelihood ratio test with the unconstrained NPMLE, except the two-component maximum likelihood was used instead of the unconstrained maximum likelihood. To do this, we first needed to implement an algorithm for finding the two component log-concave mixture estimator.

Computation of the Two Component Mixture

In the case of the two component mixture model, the log likelihood function can be written as

$$\sum_{i=1}^{n} \log \left(p \int_{L_i}^{R_i} f_1(t)dt + (1-p) \int_{L_i}^{R_i} f_2(t)dt \right)$$

such that $f_1(t), f_2(t)$ are proper log-concave distributions

$0 \leq p \leq 1$

Note that the manner in which we have written the likelihood function does not allow for uncensored times. This is necessary, as many likelihood functions for a mixture model with two or more continuous components and at least one exact time will lead to a degenerate likelihood function, such as the Gaussian mixture model without constrained variances (Kiefer and Wolfowitz 1956, Day 1969). The likelihood function for the mixture of log-concave densities will be unbounded with uncensored times as well. However, if all the data is censored, the log likelihood function will be bounded from above by 0, so this problem will not occur. Because we are focussed on current status data, we sidestep the issue of exact times for now. If we wanted to expand this model to include exact times, we could implement modifications to the mixture estimator which bound the likelihood function and still properly contain the single component model. Examples of these methods for the Gaussian mixture model have been implemented in the Gaussian mixture case in Hathaway (1985) and Fujisawa and Eguchi (2006), among others. At this time, we know of no such methods implemented for the log-concave mixture.

We see that $\hat{f}_1$ and $\hat{f}_2$ can be properly described with the same support set as described by the theorem in chapter 3 by applying the same argument for the single component to the two component model. However, the addition inside the log function makes optimization even more difficult. To deal with this, an Expectation Conditional Maximization Either (ECME) algorithm (Liu and Rubin 1994) algorithm will be implemented to make the optimization tractable. The ECME algorithm is an extension of the Expectation Conditional Maximization (ECM) algorithm (Meng and Rubin 1993). The ECM algorithm is very similar to the EM algorithm, expect that

rather update all parameters simultaneously in the M step, the algorithm updates subsets of the parameters conditional on the both the missing data and the other parameters. The ECME further extends this algorithm by updating some subsets of the parameters conditional on the other parameters and the missing data, while other parameters are updated conditional on only the other parameters (*i.e.* the observed data likelihood is maximized conditional on the other parameters, rather than the complete data likelihood).

Much like the classic EM, if we knew which component each observation came from, the complete likelihood function could be written in the form of

$$\sum_{i=1}^{n}\left[\delta_i \log\left(\int_{L_i}^{R_i} f_1(t)dt\right) + (1-\delta_i)\log\left(\int_{L_i}^{R_i} f_2(t)dt\right)\right]$$

where $\delta_i = 1$ if the i^{th} observation came from component 1, and 0 if the i^{th} observation came from component 2. Based on the current estimates of p, f_1 and f_2, the E-step is the standard for mixture models, *i.e.*

$$\mathrm{E}[\delta_i | f_1^{(t)}, f_2^{(t)}, p^{(t)}] = \frac{p^{(t)} \int_{L_i}^{R_i} f_1^{(t)}(x)\mathrm{d}x}{p^{(t)} \int_{L_i}^{R_i} f_1^{(t)}(x)\mathrm{d}x + (1-p^{(t)}) \int_{L_i}^{R_i} f_2^{(t)}(x)\mathrm{d}x}$$

For the M-step, we note that the complete data likelihood can be written as

$$\sum_{i=1}^{n}\left[\delta_i \log\left(\int_{L_i}^{R_i} f_1(t)dt\right) + (1-\delta_i)\log\left(\int_{L_i}^{R_i} f_2(t)dt\right)\right] =$$

$$\sum_{i=1}^{n}\left[\delta_i \log\left(\int_{L_i}^{R_i} f_1(t)dt\right) + c(f_2, \delta_i, L_i, R_i)\right]$$

The second half of the summation can be ignored when optimizing over the parameters of f_1.This function can now be optimized using the method described in chapter 3, with the slight modification that each observation now has a weight δ_i or $1-\delta_i$, depending on whether f_1 or f_2 is to be optimized. For the M step of the algorithm, we ran one iteration of the algorithm which was insured to increase the complete data likelihood, rather than running to convergence for the complete data likelihood, as this was much more efficient.

In addition to updating $f_1^{(t)}$ and $f_2^{(t)}$, it is necessary to update $p^{(t)}$ as well. This was done with Newton's method (using the incomplete data likelihood), conditioning on current values of $f_1^{(t)}$ and $f_2^{(t)}$ and constraining $0 \leq p \leq 1$. This can also be handled by the EM algorithm, *i.e.* $p = \sum_{i=1}^{n} \delta_i/n$, but because this is only a one parameter problem with simple constraints, we found Newton's method to be more efficient.

To determine if the algorithm had converged, we set

$$\text{total error} = \max(\text{marg err}(f_1), \text{marg err}(f_2), |\partial\ell/\partial p|)$$

where "marg err" is the stopping criterion presented in chapter 3. We considered the algorithm to have converged if total error $< \epsilon$. We used $\epsilon = 10^{-4}$. Similar to our stopping criterion in chapter 3, this insures a local maximum and while it is a necessary condition for a global maximum, it is not a sufficient condition. A basic

outline of the algorithm is as follows.

- Set initial values for p, f_1 and f_2
- while (tol $>$ err & iterations $<$ max iterations)

  {

  - ECM1 Step
    - E step: calculate δ conditional on f_1, f_2 and p
    - M step: update f_1 conditional on δ, f_2 and p
  - ECM2 Step
    - E step: calculate δ conditional on f_1, f_2 and p
    - M step: update f_2 conditional on δ, f_1 and p
  - CM Step
    - update p conditional on f_1 and f_2
  - calculate err

  }

- return $\hat{p}$, $\hat{f}_1$ and $\hat{f}_2$

Unlike our results in the single component case, we found that random starting points often lead to different local maximums, sometimes with very different likelihood values. This is very typical of mixture models and is quite difficult to deal with even without the complication of interval censoring. A common method for coping with this is to start the algorithm from many different random starting points and choosing the estimate with the highest likelihood function. Unfortunately, the algorithm was too slow to consider doing this most situations, as we must also bootstrap from

the null hypothesis as well (this algorithm typically took ten times longer to converge than the algorithm for one component). We considered it a little too ambitious to find the global maximum at this time and instead merely used the value returned by this algorithm for the likelihood ratio test, acknowledging that some additional stochastic error was introduced by not necessarily finding the global maximum. Had we been able to find the global max, we would likely increase our power, although it is unclear by how much. However, using the first local max found, we still should have the correct significance level based on this test, as long as the same procedure was applied to both the actual data and the data sampled under the null.

While these methods from the single component model worked quite well when applied to the mixture model, there was some additional numerical complications. In particular, for certain observations, both $\delta_i = 0$ and $\int_{L_i}^{R_i} \hat{f}_1(x)\mathrm{d}x = 0$ (or $\delta_i = 1$ and $\int_{L_i}^{R_i} \hat{f}_2(x)\mathrm{d}x = 0$) in the solution. This complicated the complete data likelihood, as the term $\delta_i^{(t)} \times \log\left(\int_{L_i}^{R_i} f_1^{(t)}(x)\mathrm{d}x\right)$ is required in calculating the likelihood function. While it is simple enough to replace this value with the limit 0 if $\delta_i^{(t)}$ is 0, this can create numerical instability as $\delta_i^{(t)}$ and $\int_{L_i}^{R_i} f_1^{(t)}(x)\mathrm{d}x$ both approach 0 simultaneously. To remedy this, we replace $\delta_i^{(t)}$ with 0 if $\delta_i^{(t)} < \eta$ and 1 if $\delta_i^{(t)} > 1 - \eta$. We used $\eta = 10^{-3}$ and found this took care of any issues with numerical stability.

We implemented the log-concave likelihood ratio using the two component log-concave mixture in the same manner as the likelihood ratio using the unconstrained NPMLE described in section 4.1.1, replacing the alternative model of the unconstrained NPMLE with the two component log-concave mixture model.

4.1.3 Simulations

In our simulations, we investigated both the potential power of the test and the observed power of the test. In order to examine the potential power of the test, for each test statistic we first simulated the censored data, computed the likelihood ratio and created one bootstrap sample of the likelihood ratio under the null hypothesis. This gave us insight into the difference of the likelihood ratio under the null and alternative hypothesis. The values we examined in particular were the 95^{th} percentile of the test statistic under the alternative, the 95^{th} percentile of the bootstrapped statistics and the proportion of the statistics under the alternative which are greater than the 95^{th} percentile of bootstrapped statistics (*i.e.* the estimated potential power).

For the observed power, for each dataset we ran a full bootstrapped likelihood ratio test and record the estimated observed power over all the datasets we generated. Because of the high computational cost, we only examined one scenario under the null hypothesis to check that the significance level is approximately correct and one scenario under the alternative hypothesis to check that the potential power is approximately the same as the observed power.

Potential Power Simulations

To simplify the censoring process, we restricted the distributions to those with support on the interval [0,1]. All times were censored by a current status mechanism with the censoring distribution being uniform(0,1). We considered two scenarios under the null: a uniform(0,1) distribution and a beta(10, 10). The uniform distribution is on the boundary of log-concave, while the beta distribution is well within the boundary of log-concave. We also considered three scenarios under the alternative hypothesis. We considered a beta(2, 0.5), a two component mixture model with

probability vector $p = (0.5, 0.5)$ and components beta(4,12) and beta(12,4) and finally a three component mixture model with probability vector $p = (0.25, 0.5, .25)$ and components beta(12,6), beta(8,8) and beta(6,12). We considered sample sizes n = 100, 300 and 900. Results can be seen on table 4.1

Several interesting trends appeared in these simulations. First is that the distribution of the likelihood function does not appear constant under the null hypothesis for either tests. In particular, for both tests, as sample size increases, the critical value increases, although this effect is much heavier for the unconstrained NPMLE test. Also, we observe that when the true distribution is well within the constraints (*i.e.* Beta(10,10)), the likelihood ratio appears to be distributed much tighter than when the true distribution is on the boundary (*i.e.* Uniform(0,1)). It seems that this also lead to a lower significance level than expected in the Beta(10, 10) case. This is not necessarily a bad feature. This implies the test is less likely to reject the null hypothesis when the true distribution is well within the boundary than if it were close to the boundary.

We note the mixture test always displayed higher power than the unconstrained test under the alternative hypothesis. However, in the case of the three component beta mixture, even the mixture test does very poorly in large sample sizes.

Distribution	n	Log-concave Mixture			Unconstrained NPMLE		
		Samp 95^{th}	BS 95^{th}	Power*	Samp 95^{th}	BS 95^{th}	Power*
Uniform	100	8.62	8.56	0.054	22.5	21.3	0.103
	300	9.56	9.10	0.070	46.0	45.2	0.066
	900	10.4	10.1	0.058	108	107	0.097
Beta	100	4.46	5.13	0.024	15.3	15.4	0.068
(10,10)	300	5.03	5.73	0.027	37.7	36.7	0.050
	900	5.73	6.57	0.026	96.4	96.2	0.055
Beta	100	12.0	8.10	0.185	22.9	19.9	0.166
(2, 0.5)	300	16.7	9.12	0.311	50.3	42.9	0.265
	900	28.5	10.3	0.722	121	106	0.472
Beta Mixture	100	12.0	7.98	0.201	23.6	20.8	0.147
2 Components	300	16.8	8.97	0.376	51.2	45.9	0.247
	900	28.5	9.12	0.807	128	113	0.538
Beta Mixture	100	8.62	7.44	0.074	21.2	20.6	0.071
3 Components	300	11.1	8.51	0.127	46.4	45.7	0.070
	900	14.0	9.32	0.191	116	112	0.140

Table 4.1: Simulated Critical Values and Power for log-concave Goodness of Fit Test. Samp 95^{th} is the sampled 95^{th} percentile of the distribution, BS 95^{th} is the 95^{th} percentile of the bootstrapped null samples, Power* is the proportion of sample statistics greater than the BS 95^{th}. Power* is the theoretic power, not true power

Observed Power Simulations

The previous simulation samples one likelihood ratio test statistic and then samples one null bootstrap statistic from the log-concave fit. While the difference in distri-

butions of test statistic and bootstrap null test statistics should give us an idea of the power of the bootstrap test, but not necessarily the true power. To find the true power, we performed Monte Carlo simulations of the bootstrap performance. This was very computationally expensive, so we considered two scenarios. We checked the significance level with simulated data in which the true event distribution follows a beta(10,10) and we checked the power by considering the case where the true distribution is a beta mixture with probability vector $p = (0.5, 0.5)$ and beta components with shape parameters (12,4) and (4,12). The censoring mechanism used was current status with C_i Uniform(0,1). We set the sample size to $n = 300$. Each sample had $BS = 500$ bootstrap samples. We simulated $MC = 200$ simulations for each case.

In our first case, we estimated significance level of 0.04. Under the alternative hypothesis, the observed power was 0.39. Both these values are consistent with the potential power estimates, so bootstrapped distributions appear reliable.

4.1.4 Illustrative Example

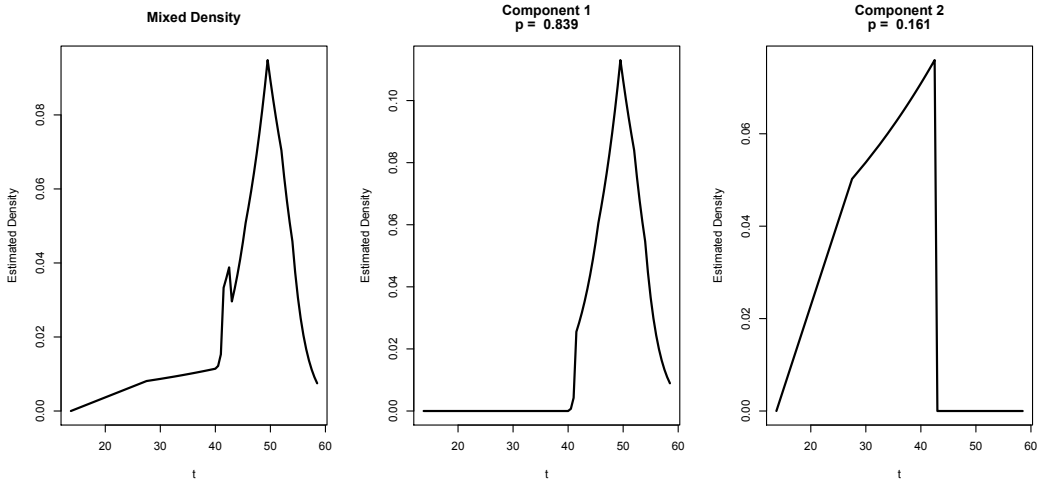

Figure 4.1: Two Component fit to Menopause Data

For an illustrative example, we revisit the menopause data which was used in chapter

3. To demonstrate the power of our algorithm for univariate data, we combined both types of menopause: natural and surgical. This leads to a natural mixture model, in which one component is natural menopause and the other surgical. Mixture models allow for multimodal distributions and so we should be concerned that perhaps a simple log-concave fit may not be appropriate. It is worth noting that there are plenty of examples of mixture models in which each component has a different mode, yet the mixture is still unimodal, so a log-concave fit may still be appropriate.

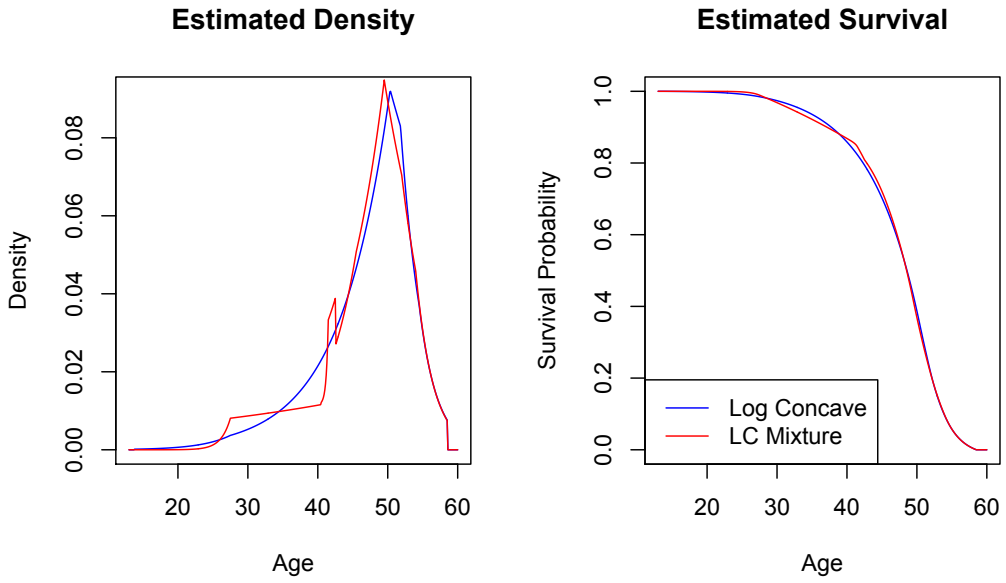

Figure 4.2: Log-concave NPMLE vs Two Log-concave Component Fit

We fit the two component mixture model to the data and present the estimated density in figure 4.1. Figure 4.2 compares log-concave NPMLE to the two component fit. We note that the density estimates disagree somewhat around $t = 25$. Other than that the density estimates are very similar. In addition, the estimated survival curves are fairly similar, implying the log-concave NPMLE is likely a reasonable fit. The test statistic we compute is 4.17. Based on 1,000 bootstrap simulations under the null, this resulted in a p-value of 0.367 (the simulated critical value for $\alpha = 0.05$ was 8.51). Therefore, we conclude that there is not significant evidence that the

log-concave assumption is a bad fit.

4.1.5 Log Concave Mixture Estimator for Interval Censored Data

Given the use of the log-concave mixture model for estimation with data containing no censoring, it seems natural to consider using the mixture estimator for interval censored data. However, we found the estimator to be overly flexible, leading to poor estimation. Not surprisingly, the estimator would almost always estimate a bimodal distribution (or even occasionally trimodal when the two components overlapped), even when the true distribution was unimodal. This lead to poor density estimation, along with an estimated survival curve that typically demonstrated one significant "kink".

To demonstrate this visually, we simulated three current status datasets. The distribution of event times in the first data set was beta(5,5), the second was beta(0.7, 5) and the third was a two component mixture with mixing probabilities = 0.5, 0.5, and two beta components with parameters 12,4 and 4,12 respectively. All datasets were censored with a current status mechanism in which $C_i \sim$ Uniform(0,1). The plots of the fitted densities and survival curves can be seen in figure 4.3. We can see that while the mixture estimator may be preferred when the true distribution is a mixture of two log-concave components, the mixture estimator does very poorly when the data is log-concave (beta(5,5)) or when the data is heavily tailed (beta(0.7, 5)). It is worth noting that in the case of heavy tailed data, even though we find the mixture log-concave to provide a very poor fit as demonstrated in figure 4.3, we found the likelihood ratio test using the mixture log-concave alternative still demonstrated high power in identifying that the data was not log-concave, as we saw in simulation

results in table 4.1.

Because the overly flexible nature of the log concave mixture estimator leads to poor estimation in most cases, we recommend the use of the log concave mixture fit as an alternative, more flexible model which can be used to check validity of the fit of the single component log-concave estimator via the likelihood ratio test, rather than a standalone estimator.

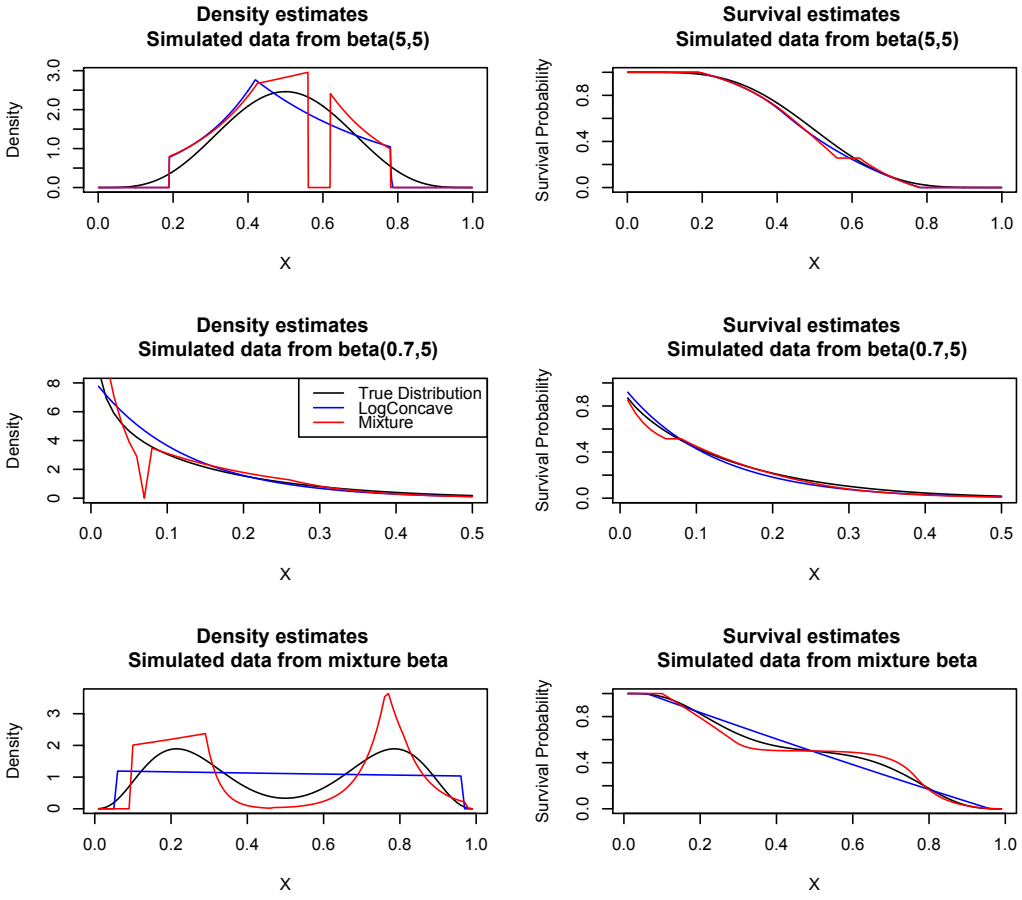

Figure 4.3: Log-concave NPMLE vs Two Log-concave Component Fit for Simulated Current Status Data with $n = 1,000$

4.2 Confidence Intervals

Confidence intervals are one of the most common tools for inference. In this section, we present methods for construction of confidence intervals for basic survival estimates (*i.e.* $S(t_o)$ or $S^{-1}(p_o)$) based on the log concave NPMLE for interval censored data.

Typically, confidence intervals are constructed from the theoretical distribution of the estimator. At this time, we have no distribution theory for the log-concave NPMLE for interval censored data. In addition, Banerjee and Wellner (2005) showed that confidence intervals for survival estimates based on distribution theory behaved very poorly in practice for the unconstrained NPMLE while profile likelihood confidence intervals behaved the best in comparison with other leading methods.

This motivates us to examine the behavior of profile likelihood confidence intervals for the log-concave NPMLE, as the construction does not require knowledge of the distribution of the estimator. Another method for construction of the confidence intervals not requiring distribution theory is bootstrapping. While the validity of bootstrapping for the unconstrained NPMLE has come into question recently (Abrevaya and Huang 2005), we argue that the bootstrap should lead to valid estimation for the log-concave NPMLE.

We implement both profile likelihood confidence intervals and bootstrap confidence intervals and compare the performance with confidence intervals constructed using the unconstrained NPMLE. Using simulations, we find both methods provide much narrower confidence intervals than the unconstrained method while still giving approximate coverage. We apply these methods to an illustrative example, comparing median time to tumor development among RFM mice placed in two different environments.

4.2.1 Confidence Intervals for the Unconstrained NPMLE

Recently, a review of confidence intervals for current status data was presented and the leading methods for creating confidence intervals were compared in Banerjee and Wellner (2005). The first method they considered, originally presented in Banerjee and Wellner (2001), was the profile likelihood confidence interval. Profile likelihood confidence intervals are an inversion of the likelihood ratio test. For the likelihood ratio test, the estimated survival curve under the null hypothesis is constrained so that $\hat{S}_0(t_0) = p_0$. The likelihood was maximized under this constraint, compared with the unconstrained maximum likelihood. It has been shown that

$$2(\hat{\ell} - \hat{\ell}_0) \sim \chi^2_1 \text{ under } H_0$$

where $\hat{\ell}$ is the maximum unconstrained log likelihood and $\hat{\ell}_0$ is the constrained log likelihood. Inversion of this test leads to the profile likelihood interval. It is worth noting that this is a fully automated procedure, in contrast with the other confidence interval procedures they examined.

The next method they considered was based on the asymptotic distribution of $\hat{F}$, the NPMLE for the cdf of the event time. Using the results of Groeneboom and Wellner 1992, they create an approximate 95% CI for $F(t_0)$ using

$$\hat{F}(t_0) - n^{-1/3}\hat{C}_n \times 0.99818, \hat{F}(t_0) + n^{-1/3}\hat{C}_n \times 0.99818$$

$$\text{where } \hat{C}_n = \left(\frac{4\hat{f}(t_0)\hat{F}(t_0)(1-\hat{F}(t_0))}{\hat{g}(t_0)} \right)^{1/3}$$

The constant 0.99818 is from the 97.5^{th} percentile of Chernoff's distribution (Groeneboom and Wellner 2001). Note that this is not a fully automated procedure. In particular, one must select a way to estimate $f(t_0)$ and $g(t_0)$, the density of the event time and the density of the inspection time for the current status censoring process. Because the censoring process itself is observed exactly, it is generally considered not very difficult to estimate $g(t_0)$, typically using either fully parametric methods or kernel smoothing. Estimating $f(t_0)$ is much more difficult due to censoring and the fact that the unconstrained NPMLE does not dictate what the density is at any point. It is worth noting that the log-concave NPMLE does dictate the density and could be a plug-in value in such a case, making for a fully automated non-parametric confidence interval. However, we did not try this, as it would still be making inference using the unconstrained NPMLE, thus gaining no power while still requiring the assumption of log concavity of the distribution of event times.

Finally, the third method considered is that of subsampling based on the work of Politis *et al.* (1999). Similar in motivation to the bootstrap method, the examines how the estimator performs in smaller subsets of the data. Then, knowing that the estimator has a convergence rate of $n^{-1/3}$, it uses this to extrapolate the performance of the estimator with the full data set. This method itself is not fully automated. In particular, there is the necessity to select the size of the subsamples, which can have a large effect on the estimator, and the optimal choice is not readily obvious. Thus, it is often recommended that a calibration algorithm, very similar to cross validation, be used to select the subsample size.

In their simulation studies, the investigators found that overall, the profile likelihood confidence intervals performed the most satisfactory. In particular, while the methods

based on the asymptotic behavior of the NPMLE produced similar average interval lengths to the profile likelihood methods, they displayed very low coverage probabilities; often 80-85%, while the likelihood ratio test typically had coverage probability approximately equal to 95%. The subsampling methods did appear to have correct coverage probability, but at the cost of considerably wider average confidence intervals than the likelihood ratio test.

Issues with Bootstrapping for the Unconstrained NPMLE

Recently, the use of bootstrap for cube root estimators has drawn concern in the literature (Abrevaya and Huang 2005, Leger and MacGibbon 2006, Sen *et al.* 2010). In particular, using the bootstrap for the Grenander (Grenander 1956) density estimator (*i.e* strictly decreasing density NPMLE) was shown to be inconsistent (Sen *et al.* 2010) and in Leger and MacGibbon (2006), it was shown that for Chernoff's estimator of the mode, basic bootstrap confidence intervals suffer from under coverage while the percentile bootstrap confidence estimator suffers from over coverage. In these cases, it has been shown that the issues because the rate of convergence is cube root and limiting distribution is Chernoff's distribution, *i.e.*

$$Z \equiv \text{argmax}_t(B(t) - t^2)$$

where $B(t)$ is a two sided Brownian motion with $B(0) = 0, -\infty < t < \infty$. The unconstrained NPMLE for interval censored data fits this description. The limiting distribution of $\hat{F}(t)$ involves Chernoff's distribution, as well as other nuisance parameters. In particular, for current status data,

$$n^{1/3}(\hat{F}(t_0) - F(t_0)) \to_d 2Z \left(\frac{1}{2} F(t_0)(1 - F(t_0)) \frac{f(t_0)}{g(t_0)} \right)^{1/3}$$

where $g(t_0)$ is the density of the censoring distribution at time t_0 and Z is Chernoff's distribution as described above. To our knowledge, there has been no study of the bootstrap estimator for the unconstrained interval censored NPMLE, but being in the class of estimators with cube rate convergence with the Chernoff's distribution as the limiting distribution, many authors are concerned about whether the bootstrap estimator will be consistent (see Delgado *et al.* 2001, Banerjee 2013 page 49). Several authors still have used the bootstrap for quantile estimation in several cases (see Pan and Chappell 2002, Frydman 1995, Rosenberg 1995), although it should be noted that all these papers were written before questions of the validity of bootstrapping cube root estimators were being asked.

Because of the cube root convergence, standard bootstrap methods are not considered valid for the unconstrained NPMLE. While we have not yet derived the theoretical distribution of the log-concave NPMLE, we used simulation to investigate the empirical convergence rate of the estimator.

To show that the estimator has faster than cube-root convergence, we assume that

$$(\hat{F}(t_0) - F(t_0))n^{\beta} \to_d U$$

where U is some unknown fixed distribution and β is the rate of convergence. Under this assumption, we note that

$$\sigma(n) \approx c \times n^{-\beta} \text{ and}$$

$$\log(\sigma(n)) \approx \log(c) - \beta \log(n)$$

where $\sigma(n)$ is the standard deviation of the estimate of interest at sample size n and c is some unknown constant. In order to estimate β, we first estimate $\sigma(n)$ via Monte Carlo simulation across a variety of sample sizes. Applying the log transformation to both the sample size and the estimated standard deviation, we use linear regression to estimate β and create confidence intervals of the rate of convergence.

In our simulations, we considered two scenarios. First, we considered when the event time was $T \sim \text{beta}(3, 3)$. It is important to note that this is a distribution which is well within the boundary of log-concave, as shape constrained estimators display lower variance if the distribution is on or outside the boundary. To examine how the estimator behaves on the boundary, we also simulated with $T \sim$ uniform(0,1). In both cases, we used a current status censoring mechanism in which the inspection times follow a uniform(0,1) distribution. The values of interest we inspected were the estimated median, 75^{th} and 90^{th} percentile. In addition, we also examined the performance of estimated survival at the true median, 75^{th} and 90^{th} percentile. We simulated data at n = 200, 400, 800, 1200, 1600, 2000 and 2400 with 400 Monte Carlo simulations at each sample size to estimate $\sigma(n)$. We also examined the rates of convergence of the unconstrained NPMLE, where we expected to see rates $n^{-1/3}$.

Estimated convergence rates, along with confidence intervals for the convergence rates, are presented in table 4.1. We note that in all cases for the log-concave NPMLE, the confidence interval for the convergence rates was strictly less than -1/3, implying faster than cube root convergence. In particular, when we were well within the

boundary (*e.g.* $T \sim$ beta(3,3)), the estimated rate of convergence appeared to be somewhere between -0.4 and -0.45. When we were on the boundary (*e.g.* $T \sim$ uniform(0,1)), we appeared to have about a -0.5 convergence rate, typical of fully parametric models. Finally, we note that the confidence intervals for the convergence rates of the unconstrained NPMLE all contained the theoretic rate of -1/3. Although further investigation is necessary to conclusively determine the rate of convergence for the log-concave NPMLE, we believe this is enough evidence to warrant an investigation of the performance of the bootstrap estimator for the log-concave NPMLE. The bootstrap procedure will be described in section 4.2.2 and the simulations and in section 4.2.3.

	Log-concave NPMLE		Unconstrained NPMLE	
Quantile Estimation for Beta(3,3) with Unif(0,1) Inspection				
	Estimate	95% CI	Estimate	95% CI
Median	-0.41	(-0.46, -0.36)	-0.34	(-0.38, -0.30)
75% Percentile	-0.42	(-0.43, -0.40)	-0.35	(-0.42, -0.29)
90% Percentile	-0.41	(-0.45, -0.37)	-0.30	(-0.36, -0.24)
Probability Estimation for Beta(3,3) with Unif(0,1) Inspection				
Median	-0.40	(-0.45, -0.34)	-0.33	(-0.36, -0.29)
75% Percentile	-0.43	(-0.45, -0.42)	-0.33	(-0.37, -0.29)
90% Percentile	-0.47	(-0.51, -0.42)	-0.33	(-0.37, -0.29)
Quantile Estimation for Uniform(0,1)				
Median	-0.48	(-0.53, -0.42)	-0.33	(-0.35, -0.31)
75% Percentile	-0.51	(-0.57, -0.45)	-0.33	(-0.39, -0.27)
90% Percentile	-0.52	(-0.60, -0.45)	-0.33	(-0.36, -0.30)
Probability Estimation for Uniform(0,1)				
Median	-0.52	(-0.58, -0.45)	-0.34	(-0.37, -0.21)
75% Percentile	-0.51	(-0.58, -0.44)	-0.32	(-0.36, -0.29)
90% Percentile	-0.51	(-0.56, -0.46)	-0.29	(-0.33, -0.26)

Table 4.2: Empirical Convergence Rates

4.2.2 Confidence Intervals for the Log-concave NPMLE

In this section, we present two novel methods for construction of confidence intervals using the log-concave estimator.

Profile Likelihood Confidence Interval

Construction of the profile likelihood confidence interval for the log-concave NPMLE follows the typical format for such confidence intervals. We start with a likelihood ratio test of our value of interest (either a survival probability or a quantile). The $(1-\alpha)\times 100\%$ confidence interval is equal to the set of all null hypotheses that fail to reject at the α significance level.

To conduct the likelihood ratio test, we first find the maximum likelihood of our standard estimator (in this case, the log-concave NPMLE), which we will call $\hat{\ell}$. Then we calculate the maximum likelihood of our estimator under the constraint that $\hat{S}_0(t_0) = p_0$, where $S_0(t_0) = p_0$ is the null hypothesis. We will call this value $\hat{\ell}_0$. Under the null hypothesis,

$$2(\hat{\ell} - \hat{\ell}_0) \dot{\sim} \chi^2_1$$

In order to perform the likelihood ratio test for the log-concave NPMLE we need to be able to maximize the log-concave NPMLE under the constraint that $\hat{S}(t_0) = p_0$. We use a Lagrangian penalty to enforce this constraint. That is to say, we replace our likelihood function with

$$\ell_0(S(t)) = \ell(S(t)) - \lambda \times (p_0 - S(t_0))^2$$

We maximize this likelihood and keep increasing λ until $\hat{S}(t_0)$ is sufficiently close

to p_0. We found this could be done quite easily by simply replacing the likelihood function ℓ used in the algorithm from chapter 3 with the likelihood function ℓ_0 given above. In addition, every time the KKT error became less than $(p_0 - S(t_0))^2 \times 100$, we would double λ (starting the algorithm with $\lambda = 1$). This insured that our final estimate had $|p_0 - S(t_0)| < 10^{-3}$ when the tolerance of the algorithm was set to 10^{-4}. Once we have this estimate, we can conduct a likelihood ratio test in the same manner stated above. We invert this likelihood ratio test to provide confidence intervals.

Bootstrap Confidence Intervals

Given the lack of distribution theory about the log-concave NPMLE, the bootstrap (Efron 1979) is a natural tool for construction of confidence intervals. The bootstrap method consists of three steps.

- Simulate B datasets
- Compute the estimate of interest for each dataset
- Use the B estimates to describe the distribution of the estimator and create a confidence interval

While step two is well defined, there is a lot of freedom in steps one and three. We have not attempted a review of different bootstrap methods for this problem, but rather implemented what we believe is the easiest method for use with interval censored data and confirmed via simulation that this method performs well.

For step 1, "case resampling" tends to be the simplest method. In this method, n subjects are sampled with replacement from the original dataset to make up each of the B bootstrap sample datasets. Other common methods for simulating each of

the datasets include simulating draws from the estimated distribution. In the case of interval censored data, drawing from estimated distribution is slightly more complicated because both the event time and the censoring method must be simulated. This was done in section 4.1.1 for current status data, but is very difficult for case II interval censored data, as it requires estimation of a complex and incompletely observed censoring mechanism. By using case resampling, we sidestep the issue of modeling the censoring mechanism and so our approach can be easily applied to either current status or case II interval censored data.

For step 3, two of the most common methods for creating confidence intervals are the percentile and the boot-t methods (Efron 1981). For the percentile bootstrap confidence interval, the $\alpha/2$ and $1-\alpha/2$ quantile of the B estimates are used to create a $(1-\alpha)\times 100\%$ confidence interval. For the boot-t, we calculate s_B = standard deviation of the B estimates and compute $\hat{\theta} \pm t_{B-1,\alpha_2} \times s_B$. We chose to use the boot-t because it requires fewer bootstrap simulations to get an accurate s_B compared to an accurate 0.025 and 0.975 quantile. In the simulations section below, we set $B = 250$, which would have led to poor estimation of quantiles but were sufficient for estimating the standard error. In our illustrative examples, we set $B = 1{,}000$.

4.2.3 Simulations

We used simulations to examine the finite sample performance of the profile and bootstrap confidence intervals for the log-concave NPMLE. In particular, we examined their average length and coverage probability. In addition, we compared the performance with the unconstrained NPMLE. Unfortunately, we did not have any software available to implement the confidence intervals procedures found in Banerjee and Wellner (2005). In order to compare the performance to the unconstrained

confidence intervals, we recreated the simulations presented in Banerjee and Wellner (2005) and compared to the results they presented for the profile confidence intervals for the unconstrained NPMLE.

We copied two of their simulation scenarios. In the first scenario, we created confidence intervals for the median in which the event time is distributed according to a exponential(1) and censored with a current status mechanism in which the inspection process is also an exponential(1). We note that this is a distribution on the boundary of log-concave, as exponential distributions are log linear. In scenario two, we simulated event times under the gamma(3, 1), in which the inspection process is a uniform(0,5) distribution. This scenario is well within the log-concave assumption. Scenarios 1 and 2 correspond with the simulated results presented on tables 4 and 2 respectively in Banerjee and Wellner (2005). Results are presented on table 4.3.

		Log-Concave Profile CI		Log-Concave Bootstrap CI		Unconstrained Profile CI	
Scenario	n	Average Length	Coverage Probability	Average Length	Coverage Probability	Average Length	Coverage Probability
1	50	0.653	0.950	0.649	0.960	0.962	0.938
	100	0.472	0.958	0.462	0.970	0.788	0.935
	200	0.338	0.954	0.328	0.948	0.626	0.941
2	50	0.393	0.966	0.427	0.962	0.501	0.938
	100	0.288	0.970	0.284	0.956	0.416	0.952
	200	0.210	0.970	0.199	0.950	0.333	0.949

Table 4.3: 95% Confidence Interval Performance

In our simulations, we find that for both methods of confidence interval construction, the log-concave intervals behave much better than the unconstrained NPMLE. In

particular, the average length is much shorter, while still maintaining approximately correct coverage probability. In addition, we notice that the relative advantage of the log-concave intervals increases as the sample sizes increases. For example, in scenario 1 the ratio of average length between the unconstrained NPMLE and the log-concave profile confidence intervals are 1.47, 1.67 and 1.85 for sample size $n = 50$, 100 and 200 respectively.

Comparing the two log-concave methods, we see that in 5 out of 6 of the scenarios, the average length of the bootstrap confidence intervals was less than the average length of the profile likelihood confidence intervals, although the differences were marginal. The profile likelihood confidence intervals displayed slight over coverage. In general, these methods seem approximately equivalent.

4.2.4 Illustrative Example

We consider the lung tumor study found in Hoel and Walberg (1972). In the study, 144 RFM mice (a line bred to have high rates of lung tumors) were placed in two environments: conventional environment (CE) and germ free environment (GE). The mice were sacrificed and examined for presence of lung tumors, leading to current status data. Of interest is comparing the rates of tumor between conventional environment and germ free environment. We did this by comparing medians and two year survival probabilities between the two groups. Medians were chosen because this is a standard measure. Two year survival probabilities were examined because most of the mice were sacrificed between 600 and 900 days, meaning we had the most information about this time (see figure 4.4).

Before fitting the log-concave confidence interval, we wished to check whether the log-concave was an appropriate fit. We did this first by examining the unconstrained

Figure 4.4: Sacrifice Times

NPMLE vs the log-concave NPMLE for each group and then by computing the likelihood ratio test presented in section 4.1. The plots can be seen on figure 4.5. The deviation of the unconstrained fits from the log-concave fits appears to be merely noise, without any systematic trends. In addition, the likelihood ratio statistics for the CE and GE group were 1.60 ($p = 0.80$) and 0.46 ($p = 0.96$), implying there were no statistically significant non log-concave trends.

Confident that the log-concave models are a reasonable fit for this dataset, we first constructed simple confidence intervals for the median and two year survival probabilities in each group, using both profile likelihood confidence intervals and bootstrap confidence intervals. In addition, we created confidence intervals for the difference in medians and two year survival probabilities. While we could create profile likelihood confidence intervals for this, it would require a significant amount of new coding to the algorithm we have, as we would have to hard-code the Lagrangian penalty for $(m_1 - m_2 = d_0)$ into the likelihood. Bootstrap confidence intervals require very little

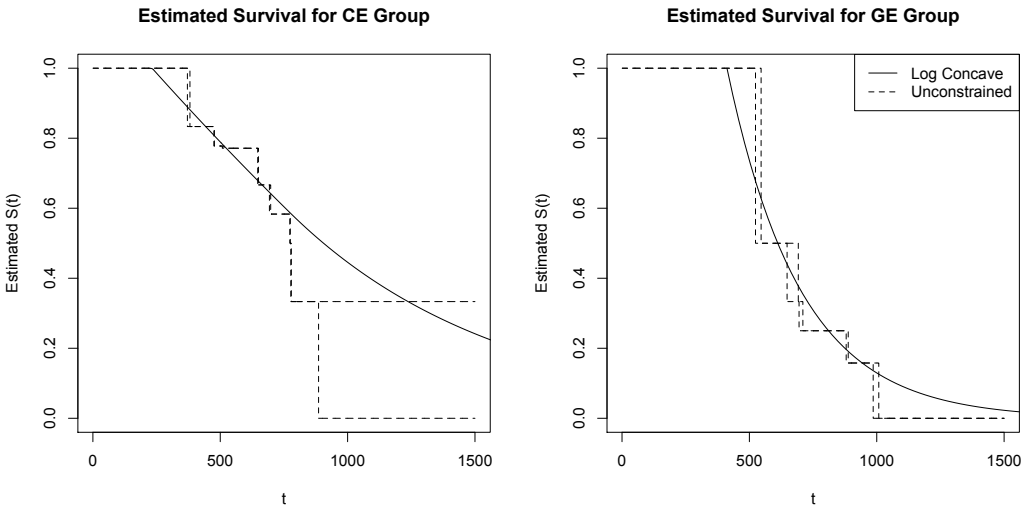

Figure 4.5: Logconcave and Unconstrained Survival Estimates

new code and have shown to be approximately equivalent to the profile likelihood confidence intervals. Because of this, we only looked at bootstrap confidence intervals for the differences between the two groups. For all bootstrap intervals in the illustrative example, we used 1,000 bootstrap samples. Confidence intervals for the median time and and two year survival probability can be see on tables 4.4 and 4.5.

	CE median	GE median	Difference
Point Estimate (in days)	906	612	294
Profile Likelihood CI	(701, 1584)	(412, 747)	NA
Bootstrap CI	(702, 1421)	(325, 741)	(21, 812)

Table 4.4: Confidence intervals for Median Time until Lung Tumor

	CE 2 Year Survival	GE 2 Year Survival	Difference
Point Estimate	0.62	0.34	0.28
Profile Likelihood CI	(0.44, 0.84)	(0.18, 0.53)	NA
Bootstrap CI	(0.47, 0.77)	(0.16, 0.51)	(0.06, 0.51)

Table 4.5: Confidence intervals for 2 year survival probability

Examining the point estimates and confidence intervals, we see the median time to tumor development and two year survival probability is considerably less for the germ free environment than for the conventional environment. We note that for both the profile likelihood and bootstrap confidence intervals, the intervals overlap, but just barely. Because of this, these confidence intervals alone are inconclusive about whether the difference is statistically significant, so we will have to look at a confidence interval for the difference in medians for a conclusive answer. In comparing the two types of confidence intervals, for the conventional environment, the bootstrap confidence interval is considerably shorter, particularly on the upper end. However, because most mice do not live beyond 1,000 days (only one lived past 1,000 in our data set, 14 lived beyond 900 days), we have very little information about survival probabilities beyond this time. Thus, the fact that the profile confidence interval claims less information about survival probabilities beyond this time may be considered a good thing. In the germ free environment, the profile likelihood confidence interval is considerably shorter than the bootstrap confidence interval. Finally, viewing the bootstrap confidence interval for the difference of medians confirms that there is a statistically significant difference in medians between the two groups.

4.3 Regression

In this section, we present methods for using the log-concave NPMLE as the baseline distribution in a Cox PH model. We begin by reviewing the Cox PH model, present methods of computation with the log-concave NPMLE baseline and use simulations to compare estimating efficiency with the unconstrained Cox PH model. We find that while using the log-concave baseline does reduce the bias and variance of the regression estimates, the effect is marginal. However, estimates which involve the baseline distribution, such as estimated survival probability for subjects with a given set of covariates, are greatly improved by using the log-concave baseline in comparison to the unconstrained baseline. Finally, we apply this to the lung tumor study.

4.3.1 Review of Regression Models for Survival Data

In survival analysis, it is natural to be interested in regression analysis to compare treatment and other covariate effects. In fact, in many studies the baseline survival function is considered a nuisance parameter and the question of interest is in regards to the regression parameters. Several regression models for continuous data exist in the literature, including the Cox Proportional Hazard model (Cox PH, Cox 1972), which models the data according to

$$h(t|X,\beta) = h_0(t)e^{X^T\beta}$$

where $h(t|X,\beta)$ is the hazard function for a subject with covariates X and $h_0(t)$ is the baseline hazard function, or the hazard function for a subject with all covariates

= 0. Interpretation of the coefficients in this model is that a one unit increase in covariate X_1, with all other covariates remaining the same, is associated with a e^{β_1} fold increase in the hazard. An important relation which will be used later on to estimate the regression parameters is that because

$$S(t) = \exp\left(-\int_0^t h(x)\mathrm{d}x\right)$$

the proportional hazards relation is equivalent to

$$S(t|X,\beta) = S_0(t)^{e^{X^T\beta}}$$

Other popular models include the Accelerated Failure Time model (AFT), which models the data according to

$$S(t|X,\beta) = S_0(te^{X^T\beta})$$

where $S_0(t)$ is the baseline survival function. Interpretation of the coefficients in this model is that one unit increase in covariate X_1, with all other covariates remaining the same, is associated with a e^{β_1} fold decrease in average event time.

The last popular regression model we present is the Proportional Odds Model. According to the Proportional Odds model,

$$\frac{S(t|X,\beta)}{1-S(t|X,\beta)} = (e^{X^T\beta}) \times \frac{S_0(t)}{1-S_0(t)}$$

In other words, we can interpret the coefficients as saying that for a one unit increase in covariate X_1, there is an e^{β_1} increase in the odds of an event occurring by time t.

A more through description of the Cox PH, AFT and proportional odds model in application to interval censored data can be found in Sun, J. (2006).

By far, the most popular model is the Cox PH model and we will focus on this for the remainder of this chapter. This attention is due in part to the fact that Cox (1972) showed that by replacing the event times by their rankings, one can factor the likelihood function so as to separate the regression coefficients from the baseline distribution in the likelihood function for right censored data. This is done by replacing the the exact event times with the only slightly less informative ordinal rankings. This allows an investigator to make inference on the regression coefficients without having to specify a baseline hazard function. Some information is lost by replacing the event times with ranks, although it is generally considered a reasonable cost in exchange for robustness.

Being a semi-parametric model, there still are assumptions of the functional form which must be inspected in order to conclude that a Cox PH is sufficiently modeling the trends in the data. In particular, the hazard functions of individuals with different covariates must be a constant proportion of each other across the entire support considered. An example of this assumption failing dramatically is when survival curves cross for two groups being compared. This can be expected in cancer patients receiving chemotherapy treatment compared to control. Chemotherapy is a very

damaging treatment, killing vast amounts healthy cells along with cancerous cells. Immediately after receiving treatment the chemotherapy group will be at much higher risk for various health problems related to the treatment and we would expect a lower survival probability for these adverse events in the treatment group shortly after administration of treatment. However, once they recover from chemotherapy, the reduction (or elimination) of cancerous cells will lead to lower risk for adverse events and it would be expected that the survival probability for the control group would be lower than the treatment group later in the study. Thus, over the time span considered, the hazards of the two groups is not proportional and attempting to describe the relation with a proportional hazards model misses important aspects of the treatment effect.

This does not imply that a Cox PH model must be abandoned when hazard rates are non-proportional. When data contains non-proportional hazards, a Cox PH model still provides an estimate of the average hazard ratio over the entire time considered, similar to how a simple linear effect can estimate a first order trend when the true effect is non-linear in a least squares model. In addition, non-proportional hazards can be modeled with a Cox PH model by including an interaction effect with time. This is similar to fitting a smoothing spline in a least squares model.

In the context of interval censored data, the Cox PH model is still the most popular model, although this mostly is due to the popularity of the Cox PH model for right censored data rather than its practicality for interval censored data. Because the rank order in interval censored data is not known (assuming there are overlapping observation intervals), the partial likelihood of the regression coefficients cannot be separated from the baseline survival function as in the right censored case. However, the Cox PH estimates can still be found by maximizing the log likelihood function

$$\ell(S_0, \beta | X, L, R, \delta) = \sum_{i=1}^{n} \log \left[\delta_i f_0(t_i) e^{X^T \beta} S_0(t_i)^{e^{X^\beta} - 1} + (1 - \delta_i) \left(S_0(L_i)^{e^{X^T \beta}} - S_0(R_i)^{e^{X^T \beta}} \right) \right]$$

where δ_i is an indicator function for whether the i^{th} observation was observed exactly. In addition, one can avoid specifying the baseline survival function by using the unconstrained NPMLE. While this allows for semi-parametric modeling similar to the right censored case, the inability to factor the partial likelihood leads to a variety of complications, both in computation and inference. Computationally, one difficulty is that the classic EM algorithm for the NPMLE cannot be applied to the baseline survival function in a Cox PH model. Wei Pan (1999) presented an algorithm for fitting the Cox PH model with the unconstrained baseline. This is done via an ICM algorithm, similar to the algorithm used for univariate NPMLE, adding the regression parameters to the baseline parameters to be optimized by quadratic programming. Although there is a publicly available CRAN package "intcox" which implements Wei Pan's algorithm (not written by Wei Pan), we found several problems with this package. This is discussed in the appendix.

In addition to computational problems with the Cox PH model for interval censored data, there are also difficulties with inference. In particular, in a semi-parametric model, there are theoretically an infinite number of parameters to consider. Thus, it is impossible to invert the true Hessian (note: in the right censored case, because the partial likelihood function can be factored out, one can ignore the baseline parameters in the inverse of the Hessian, eliminating this issue). One solution which has been proposed instead is the use of sieve estimation. The basic concept is to model the infinite dimensional nuisance parameters (in this case the baseline survival function) with a finite dimensional parameter space which then allows us to use standard MLE

procedures for inference. In this problem, a very natural finite subset to use is all the maximal intersections with positive mass at the MLE as the finite subset of parameters to be considered. Although the likelihood function will be concave, and thus the Hessian is insured to be invertible, there are still problems as the Hessian matrix can be quite large, especially if a continuous censoring mechanism is used. As a result, inverting the Hessian may be subject to numerical instability. Other authors have used bootstrap estimates to achieve inference. While there are concerns of using the bootstrap estimator for the baseline survival function due to cube root convergence, as mentioned in section 4.2.1, the regression estimates are shown to have square root convergence to a Gaussian distribution (Huang 1996), so the bootstrap estimator is considered valid.

Wei Pan (1999) noted that the Cox PH regression model with the unconstrained NPMLE baseline was biased, in that it tended to overestimate effects.

4.3.2 Log-concave Cox PH

We chose to implement a Cox PH model, in which the baseline survival was estimated with the log-concave NPMLE. In this section, we define the estimator, briefly outline the algorithm used to find it and discuss methods for inference.

Definition of the Log Concave Cox PH Estimator

Our estimated distribution is defined as the maximum likelihood estimator of a Cox PH model with a log-concave baseline distribution. It can be characterized by the values of S_0^{LC}, β which maximize the likelihood function

$$\ell(S_0^{LC}, \beta | X, L, R, \delta) =$$

$$\sum_{i=1}^{n} \log \left[\delta_i f_0^{LC}(t_i) e^{X^T \beta} S_0^{LC}(t_i)^{e^{X^\beta} - 1} + (1 - \delta_i) \left(S_0^{LC}(L_i)^{e^{X^T \beta}} - S_0^{LC}(R_i)^{e^{X^T \beta}} \right) \right]$$

such that S_0^{LC} is the baseline survival function, constrained to having a log-concave density function (f_0^{LC} is the log-concave density function defined by S_0^{LC}). Note that this is the exact same likelihood as defined in general case above, except that we are constraining the baseline distribution. We can see that the parameterization for the log-concave baseline density function for the Cox PH model will be the same as we presented in section 3.3, *i.e* support points at the beginning and end of each interval, with one support point between each of these support points by applying the same reasoning as presented in the case without covariates.

Algorithm

To find the log-concave Cox PH estimator, we implemented a conditional maximization (CM) algorithm. This iterated between updating the regression coefficients and updating the baseline parameters. To update the regression coefficients, a modified Newton's method step was included, which included half-stepping if the proposed step did not increase the likelihood. To update the baseline parameters, the same algorithm as was implemented in chapter 3 was used. The stopping criterion err was the maximum of the stopping criterion of the baseline distribution, as described in section 3.5, and the maximum absolute derivative of the regression coefficients. In all applications of this algorithm, we set the tolerance to $\epsilon = 10^{-4}$ and terminated the algorithm when $err < \epsilon$. We outline the algorithm below, defining β to be the

regression parameters and φ to be the baseline distribution parameters.

- Set initial values of β, φ
- Set $err = \epsilon + 1$, iter $= 0$
- while($err > \epsilon$ AND iter $<$ max iterations) {
  - $err = 0$, iter = iter + 1
  - Update β via Newton's Method
  - Update φ using methods from chapter 3
  - err = max(KKT error (φ), maximum $|\frac{\partial \ell}{\partial \beta}|$)

  }

In practice, we found the regression coefficients converged much faster than the baseline parameters. However, the addition of regression coefficients implies that the algorithm will be unlikely to take advantage of ties in the data in same manner as the algorithm without covariates (see appendix B). Because of this, each step of the algorithm is of complexity $O(nu)$ (u = number of unique times in data set), rather than $O(u^2)$. Because in practice, it is often the case that $u << n$, this can have a significant effect on speed. We found the algorithm to be satisfactory for samples with $n < 500$, but may be insufficient for larger samples.

We found an interesting problem which can occur when using the log-concave Cox PH model with exact observation, although it is a problem easily remedied. Discussion of the issue and solution can be found in appendix D.

Inference Methods for the Log-concave Cox PH Model

For inference in the log-concave Cox PH model, we considered two methods for creating confidence intervals for regression parameters. First is the profile likelihood. Computation in this case requires very little modification of the general algorithm for the log-concave Cox PH model, as the likelihood under the null can be found by fixing the regression parameter to the null value and maximizing the remaining regression and baseline parameters. Secondly, we considered bootstrap confidence intervals. We will use case resampling methods and boot-t confidence intervals in the same manner as presented in section 4.2.2.

4.3.3 Simulations

We used simulations to examine the behavior of the log-concave Cox PH model and compare to the unconstrained Cox PH model. In our investigation, we found that using the log-concave baseline distribution reduced both the bias and the standard deviation of the estimated regression coefficients, but only mildly. The advantage of using the log-concave baseline distribution was much more pronounced when the estimate of interest involved a transformation of the baseline distribution, such as estimating the median survival time for an individual with a given set of covariates.

To examine the performance of the Cox PH models, we replicated the simulations found in Wei Pan (1999) and apply the Cox PH model with both the log-concave baseline and unconstrained baseline. We first examined the performance of the regression coefficients and then the estimated medians for subjects with given coefficients.

The sampling scheme is meant to replicate a standard treatment/control study with case II interval censoring. Under this scenario, the baseline survival function is a

Weibull with shape parameter equal to 2 and scale parameter equal to 1. Regression covariants are all binary, with equal probability to be 0 or 1. The regression parameters were $\beta = (1, 0, 0)$. Wei Pan (1999) used case II interval censoring, in which the first inspection time was given by $T_i \sim \text{uniform}(0, \theta)$, where θ will be determined later. After the first inspection time, follow up times were given by $T_i + lens * j$, where *lens* $= 0.5$ and $j = 1, ..., k$, where k will be determined later. If $T_i + lens * k < E_i$, where E_i is the event time, the observation was right censored. The following scenarios were considered.

- Case 1: $n = 100$, $k = 1$, $\theta = 2$
- Case 2: $n = 100$, $k = 2$, $\theta = 1$
- Case 3: $n = 200$, $k = 2$, $\theta = 1$
- Case 4: $n = 400$, $k = 2$, $\theta = 1$

In each scenario, MC $= 1000$ samples were taken. The regression parameter of interest is $\hat{\beta}_1$, where $\beta_1 = 1$. Results can be found on table 4.6.

	Case 1 $n = 100$ Heavy Censored		Case 2 $n = 100$ Light Censored		Case 3 $n = 200$ Light Censored		Case 4 $n = 400$ Light Censored	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Logconcave	1.12	0.39	1.07	0.27	1.03	0.18	1.02	0.12
intcox	1.21	0.47	1.19	0.29	1.06	0.19	1.05	0.13

Table 4.6: Estimated Mean and Standard Deviations for $\hat{\beta}_1$

The results from these simulations appear to show that when there is very little information (*i.e.* case 1), using the log-concave Cox PH model can sufficiently reduce

the standard deviation and the bias of the regression parameter. However, as the information increases (more informative censoring and larger sample size), the relative advantage of using the log-concave baseline reduces. This seems reasonable; small errors in the baseline survival function are unlikely to cause much of an effect to the regression parameters. This is evident in the fact that the unconstrained NPMLE has cube root convergence, while the regression parameters have square root convergence for the unconstrained Cox PH model (Huang 1996). Because of this, if the data set is weakly informative and the regression parameters are values of interest, the log-concave Cox PH model may be the estimator of choice. However, if the data set is moderately or strongly informative, the unconstrained Cox PH model may be preferred due to robustness.

As mentioned earlier, we found that the advantage of using the log-concave baseline was most significant when estimating values which required the baseline distribution. In order to investigate this, we again simulate cases 1, 2, 3 and 4 above. However, this time the estimate of interest was $\hat{F}_0^{-1}(0.5) - \hat{F}_1^{-1}(0.5)$ or the estimated difference in medians between the treatment and control groups. We compared the estimated difference between individuals with covariates $X = (1, 0, 0)$ (treatment) and $X = (0, 0, 0)$ (control). The true value of $F_0^{-1}(0.5) - F_1^{-1}(0.5) = 0.83 - 0.51 = 0.32$. Once again, we simulated MC = 1,000 samples and compared the differences. Results can be found on table 4.7.

In this case, we found that using the log-concave baseline cut the standard deviation approximately in half compared to the unconstrained case. In addition, the unconstrained case appears to display an upward bias in the difference in medians in the less informative cases, while the log-concave estimator appears to be approximately unbiased.

	Case 1 $n = 100$ Heavy Censored		Case 2 $n = 100$ Light Censored		Case 3 $n = 200$ Light Censored		Case 4 $n = 400$ Light Censored	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Logconcave	0.34	0.12	0.34	0.10	0.34	0.07	0.33	0.05
intcox	0.39	0.25	0.36	0.20	0.35	0.14	0.35	0.11

Table 4.7: Estimated Mean and Standard Deviations for $\hat{F}_0^{-1}(0.5) - \hat{F}_1^{-1}(0.5)$ with $F_0^{-1}(0.5) - F_1^{-1}(0.5) = 0.32$

4.3.4 Illustrative Example

For an illustrative example, we revisit the tumor study we analyzed in section 4.2. This time, we wanted to use a semi-parametric regression model to compare the two survival curves. Now we can compare the two curves using the proportional hazards parameter estimates. Because of this, we examined point estimates and confidence intervals for the regression parameter, using both profile likelihood confidence intervals and bootstrap confidence intervals for the log-concave Cox PH regression parameter. We compared this to the unconstrained baseline Cox PH model point estimate and bootstrap confidence interval. In addition, we created bootstrap confidence intervals for the difference in medians and two year survival probabilities between the groups using the log-concave Cox PH. However, we could not use the bootstrap procedure to create a confidence interval for the unconstrained Cox PH model because the bootstrap is not valid in such a case.

In the section 4.2, we decided that the log-concave fit was appropriate for fitting each survival curve. In this analysis, the assumption we must check is that of proportional

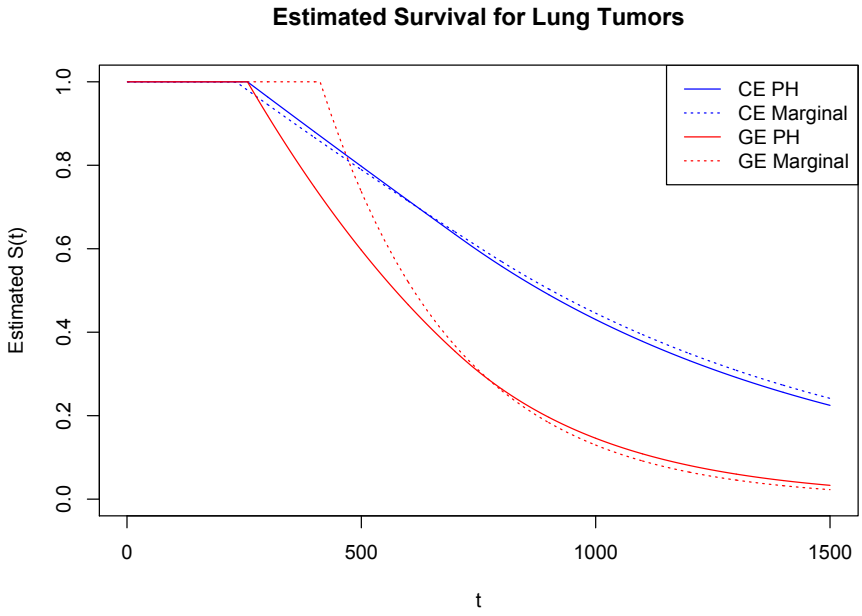

Figure 4.6: Cox PH Fits vs. Marginal Fits

hazards (given that we already concluded a log-concave baseline was a reasonable fit). To inspect the validity of the proportional hazards assumption, we visually compare the marginal fits of each group to of a proportional hazards fit. The estimated survival curves can be viewed on figure 4.8. We note that the estimates seem very similar, with the exception of early in the germ free environment. The proportional hazards model actually seems more reasonable than the marginal fit in this case, as it seems unreasonable that the true hazard is 0 until $t = 500$. Therefore, there is no strong evidence that the Cox PH model is a bad fit.

We present the point estimates and confidence intervals for the regression parameters on table 4.8. Using the log-concave Cox PH model, the hazard rate in the germ free environment is estimated to be 2.29 times higher than in the conventional environment, with a 95% CI for the this ratio being 1.10, 4.22 (using the profile confidence interval). From the log-concave Cox PH model, we conclude that the difference in rates of lung cancer between conventional environment and germ free environment

is statistically significant. We see that the log-concave Cox PH model estimates a slightly higher hazard ratio than the unconstrained Cox PH model. The confidence intervals for the log-concave Cox PH model are very similar for the profile likelihood and the bootstrap estimator. The confidence interval for the unconstrained Cox PH model is approximately the same length as well. However, the small difference in estimated values turns out to be the difference between statistical significance and and insignificance, as the log-concave Cox PH interval does not contain 0, while the unconstrained Cox PH does. The simulations presented 5.3.3 suggest that while the log-concave Cox PH model should have more power than the unconstrained Cox PH model, this difference is marginal, so we shouldn't expect that the use of the log-concave baseline will often be the difference between statistically significant and insignificant.

	Log-concave Cox PH β	Unconstrained Cox PH β
Point Estimate	0.83	0.69
Profile Likelihood CI	(0.10, 1.44)	NA
Bootstrap CI	(0.06, 1.41)	(-0.17, 1.28)

Table 4.8: Confidence Intervals for Log Hazard Ratio

On tables 4.9 and 4.10, we present confidence intervals for the difference in median times and two year survival probabilities between the two groups. In both cases, we see that there is a statistically significant difference between the two groups (although we already knew this would be true, because the regression coefficient was statistically significant). We see that the Cox PH model confidence interval is actually marginally wider (about 5%) than the simple model where we fit each survival curve independently. This came as a bit of a surprise to us, as we expected the Cox PH model to

have a lower variance than the marginal model. To test which estimator has lower variance in these general settings, we simulated data is a fairly similar environment. To do this, we used the marginal log-concave fits for each group to generate random draws from each distribution and then censored these event times with the observed censoring times in the data set. We took 1000 Monte Carlo samples to assess the difference in standard deviation of estimated median time between the Cox PH model and the two marginal models. As we expected, the Cox PH model had a lower standard deviation, although the reduction was very minimal (standard deviation for Cox PH model: 208, standard deviation for marginal model: 215).

	Cox PH Model	Marginal Model
Point Estimate (in days)	312	294
Bootstrap CI	(6, 840)	(21, 812)

Table 4.9: Confidence Intervals for Median Time to Lung Tumor

	Cox PH Model	Marginal Model
Point Estimate	0.28	0.28
Bootstrap CI	(0.01, 0.56)	(0.06, 0.51)

Table 4.10: Confidence Intervals for Two Year Survival Probability

4.4 Conclusion

In this chapter, we have presented novel methods for inference for the log-concave estimator for interval censored data. We presented a goodness of fit test based on a likelihood ratio test, two methods of confidence interval construction for survival estimates of the simple log-concave estimator and a Cox PH model which uses a base-

line log-concave distribution. We have shown that these methods can lead to efficient estimation of survival curves when compared with the unconstrained equivalent estimator, both for the Cox PH model and simple model. We found that there was some advantage to using the log-concave baseline when estimating regression coefficients in a Cox PH model, but it was much less pronounced than when estimating survival probabilities and quantiles.

Chapter 5

Inverse Convex Constraint

5.1 A New Shape Constraint: Inverse Convex

In chapters 3 and 4, we showed that applying the popular log-concave shape constraint to the non-parametric estimator for interval censored data can greatly improve the performance of survival estimates and allow for more efficient inference methods without having to specify a much more restrictive parametric model. However, the use of the log-concave estimator may not be as robust as we wish for survival analysis. The log-concave family allows for up to exponential tails. Given that the exponential distribution is often considered a standard model in survival analysis, we should not consider the log-concave constraint robust to heavy tailed survival data. In particular, we know that the log-concave NPMLE will perform inadequately if the true hazard is decreasing, as the log-concave constraint forces an increasing hazard function.

To address these concerns in survival analysis and other types of heavy tailed data, we introduce a new, more flexible shape constraint. We will call this constraint "inverse convex". In section 6.2, we examine some interesting characteristics of the family

of inverse convex distributions, including proving that the log-concave family is a proper subset of the inverse convex family, with inverse convex allowing for heavier tails. In section 6.3, we discuss characteristics of the likelihood function. Of particular interest is the fact that the likelihood function is unbounded, but local modes provide satisfactory estimates, similar to a Gaussian mixture problem without fixed variance components. In section 6.4, we briefly introduce the algorithm used to find the inverse convex estimator, although we skip the details as it is a simplified version of the algorithm presented in Chapter 3. We also demonstrate that our algorithm can reliably find the local mode of interest for reasonable sample sizes ($n > 25$). In section 6.5, we present simulation results to examine the performance of the inverse convex estimator as compared to the log concave estimator. In section 6.6, we apply the inverse convex to real data. In section 6.7, we discuss the findings of this study.

While we believe this estimator is very attractive for survival analysis for censored data, the algorithm we have implemented does not allow for censoring yet. Based on our work with the log-concave estimator, we believe such an algorithm should not require novel optimization methods beyond what we have already done. However, it will take some time to implement these tools.

5.2 Characteristics of the Inverse Convex Family

We start by defining the inverse convex family. We say that $f(x)$ is inverse convex if $f(x) = 1/g(x)$ such that $g(x)$ is a convex function. We will call $g(x)$ the inverse kernel. Much like the log-concave constraint, the inverse convex constraint implies a distribution to have no more than one peak (although both can be flat like a uniform distribution), as $g(x)$ can only have one locally minimum region. An attractive feature of the inverse convex family is that the log-concave family is a subset. To show this,

consider that $f(x)$ is log-concave if $f(x) = e^{\phi(x)}$ such that $\phi(x)$ is concave. If we can show that $e^{-\phi(x)}$ is convex, we have shown log-concave implies inverse convex. Because $\phi(x)$ is concave, $-\phi(x)$ is convex, making $e^{-\phi(x)}$ log convex. Log convex implies convex, therefore $e^{-\phi(x)}$ is a convex function. Thus, log-concave implies inverse convex.

In addition, several classic parametric distributions which are not log-concave are still inverse convex. For example, any t-distribution, the F-distribution with ν_1 (or df_1) ≥ 2 and log logistic distribution with $\beta \geq 1$ are inverse convex distributions but are not log-concave.

Several classic distributions are excluded from the family of inverse convex distributions. Any multimodal distribution cannot be inverse convex. In addition, any non-degenerate distribution with unbounded density, such as the F-distribution with $\nu_1 < 2$ or the log logistic with $\beta < 1$, cannot be an inverse convex distribution. We will use a proof by contradiction to show this.

Suppose $f(x)$ is a non-degenerate inverse convex distribution, such that $\lim_{x \to x_0} f(x) = \infty$ (or $\lim_{x \to x_0} g(x) = 0$) from at least one side . Because $f(x)$ is non degenerate, there must exist δ with $|\delta| > 0$ such that $f(x_0 + \delta) = a > 0$. For notational simplicity, we will assume $\delta > 0$, although this proof trivially generalizes to $\delta < 0$ as well. We can bound $\int_{x_0}^{x_0+\delta} f(x)\mathrm{d}x$ from below with a function which is inverse linear between x_0 and $x_0 + \delta$, as this is the boundary of an inverse convex distribution. Applying this bound,

$$f(x) \geq \left(g(x_0) + (x - x_0)\frac{g(x_0 + \delta) - g(x_0)}{x_0 + \delta - x_0} \right)^{-1} , \, x \in (x_0, x_0 + \delta)$$

Plugging in $g(x_0) = 0$, $g(x_0 + \delta) = a^{-1}$ and simplifying, we get

$$f(x) \geq \frac{a\delta}{x - x_0} \ , \ x \in (x_0, x_0 + \delta)$$

Integration leads to

$$\int_{x_0}^{x_0+\delta} f(x)\mathrm{d}x \geq \int_{x_0}^{x_0+\delta} \frac{a\delta}{x - x_0}\mathrm{d}x = a\delta \int_0^{\delta} \frac{1}{x}\mathrm{d}x = \infty$$

This leads to an improper distribution. Therefore, there are no proper inverse convex probability functions with unbounded density. This may be seen as a limitation of the inverse convex family. However, it also makes estimation tractable. In contrast, the unimodal shape constraint allows for unbounded density and as a result the likelihood function is unbounded, making estimation intractable (Wegman 1969). Interestingly, the lognormal distribution with $\sigma > 2$ is not inverse convex, despite having bounded density. Particularly, the distribution is non-inverse convex on the interval $(e^{\mu-\sigma^2/2-\sqrt{\sigma^4-4\sigma^2}}, e^{\mu-\sigma^2/2+\sqrt{\sigma^4-4\sigma^2}})$, where the density becomes extraordinarily peaked. The log normal is not log-concave for all values of $\sigma > 0$.

We emphasize that the key difference between the log-concave constraint and the inverse convex constraint concerns the heavy tails. The boundary of the log-concave constraint is an exponential tail, which may be insufficient in some cases. The boundary of the tails of an inverse convex distribution, as defined by the shape constraint alone, is an inverse linear function. However, because a positive inverse linear function will integrate to ∞ over $[a, \infty)$ for any a, in order for an inverse convex distribution to be a proper distribution function, the tails must not be on the boundary of the inverse convex constraint. Thus, the inverse convex restraint allows for as heavy tails

as possible for a distribution function.

This feature of heavy tails can be especially attractive for survival analysis with heavy censoring. For example, consider if investigators are conducting a study in which the study began at time $t = 0$ and ends at $t = 1$, so any events occurring after $t = 1$ are right censored. Suppose we found that the fit $\hat{f}_X(x) = 0.5e^{-x}$ describes the data quite well over $t = [0, 1]$. Although this function is log-concave over [0,1], we cannot fit a log-concave estimator to have this fit over [0,1], as it is not possible to assign enough probability mass over $[1, \infty)$ so that our estimate would be log-concave and a proper distribution function. This is because the mass assigned would be maximized by extending the function over $[1, \infty)$, leading to $\hat{f}_X(x) = 0.5e^{-x}$, $0 < x < \infty$, which integrates to 0.5. This has the unpleasant result that we cannot fit the data well because of the behavior over the area we have not observed, despite being a good fit over the area we have observed. On the other hand, if we used an inverse convex estimator and the fit appeared good over $t = [0, 1]$, then we will always be able to assign enough mass over $[1, \infty)$ to form a proper distribution function, as the amount of mass we can assign is unbounded. Thus, we only need to worry about the fit over the observed area rather than the fit over both the observed and censored data.

5.3 Characterization of the Inverse Convex Estimator

A displeasing characteristic of the inverse convex estimator is that the log likelihood function is unbounded, as we will show in this section. In particular, as the estimate approaches a degenerate distribution about a single observed value, the likelihood approaches infinity. However, in practice, we find that the algorithm we present in section 6.4 typically convergences to a non-degenerative, informative estimate. This

is very similar to the case of the Gaussian mixture model, in which the likelihood function is unbounded, but estimates which are local modes with finite likelihood lead to valid estimation. Empirically, we find our algorithm converges to a unique finite mode with very high probability for all but the smallest of samples, as will be demonstrated in section 6.4.

5.3.1 Parameterization of the Likelihood Function

The likelihood function can be written in the form

$$\ell(g|x) = \sum_{i=1}^{n} -\log(g(x_i))$$

$$\text{with constraints } \frac{g(x_2) - g(x_1)}{x_2 - x_1} \leq \frac{g(x_3) - g(x_2)}{x_3 - x_2} \; \forall x_1 < x_2 < x_3$$

$$\text{and } \int_{-\infty}^{\infty} g(x)^{-1} \mathrm{d}x = 1$$

To ease the last constraint, we will replace $g(x_i)^{-1}$ with $g(x_i)^{-1} / \int_{-\infty}^{\infty} g(x)^{-1} \mathrm{d}x$. This means that $g(x_i)^{-1}$ is proportional to the estimated density at x_i up to a multiplicative constant. Now we can rewrite the likelihood function as

$$\ell(g|x) = \sum_{i=1}^{n} -\log(g(x_i)) - n\log\left(\int_{-\infty}^{\infty} g(x)^{-1}\mathrm{d}x\right)$$

$$\text{with constraints } \frac{g(x_2) - g(x_1)}{x_2 - x_1} \leq \frac{g(x_3) - g(x_2)}{x_3 - x_2} \ \forall x_1 < x_2 < x_3$$

Similar to the log-concave estimator, the inverse convex estimator can will be described by an inverse linear function with knots at the observed times, as this is a necessary condition for our estimate $\hat{g}(x)$ to be a local max. We will use an identical argument to the argument presented in Rufibach (2007) for the log-concave estimator. Let us assume the x_i's are ordered, so that $x_1 \leq x_2 \leq ... \leq x_n$. For any set of values $g(x_1), ..., g(x_n)$, the likelihood function is maximized by minimizing $\int_{-\infty}^{\infty} g(x)^{-1}\mathrm{d}x$. That means zero mass will be placed below x_1 and above x_n. In addition, for a given $g(x_i)$, $g(x_{i+1})$, we maximize the likelihood function by minimizing $\int_{x_i}^{x_{i+1}} g(x)^{-1}\mathrm{d}x$. Because of the constraint of convexity of $g(x)$, this is minimized by making $g(x)$ linear between x_i and x_{i+1}. Therefore $\hat{g}$ will be a piecewise linear spline with knots at the observed values x_i. We can completely characterize the solution by $\beta_1, ..., \beta_n$, where $\beta_i = g(x_i)$. Under this parameterization, the likelihood function can be written as

$$\ell(\beta|x) = \sum_{i=1}^{n} -\log(\beta_i) - n\log\left(\sum_{j=1}^{n-1} \frac{\log(\beta_{j+1}) - \log(\beta_j)}{\left(\frac{\beta_{j+1} - \beta j}{x_{j+1} - x_j}\right)}\right)$$

$$\text{which satisfy } \frac{\beta_{i+1} - \beta i}{x_{i+1} - x_i} \leq \frac{\beta_{i+2} - \beta_{i+1}}{x_{i+2} - x_{i+1}}; 1 \leq i \leq n-2$$

$$\beta_i > 0;\ 1 \leq i \leq n$$

Note that if $\beta_{i+1} = \beta_i$, the i^{th} term in the summation is undefined. In this case, the integral over (x_i, x_{i+1}) is equal to $\beta_i^{-1}(x_{i+1} - x_i)$. In addition, the term is undefined if there are ties in the data. To account for this, we will redefine x_i = the i^{th} *unique* time, π_i = number of observations at time x_i and n^* is the number of unique times in the dataset. Note that $\sum_{i=1}^{n^*} \pi_i = n$. Then we can rewrite the inverse convex likelihood as

$$\ell(\beta|x) = \sum_{i=1}^{n^*} -\log(\beta_i)\pi_i - n\log\left(\sum_{j=1}^{n^*-1} \frac{\log(\beta_{j+1}) - \log(\beta_j)}{\left(\frac{\beta_{j+1}-\beta j}{x_{j+1}-x_j}\right)}\right)$$

Not only does this allow the likelihood function to handle ties in the data, but it also allows for easy implementation into a mixture model, as π_i could just as well represent probability weights.

5.3.2 Unbounded Nature of the Likelihood Function

Theorem 3. *The likelihood function for the inverse convex estimator is unbounded.*

Proof. In the case that the of only one unique observed value (or $\pi_1 = 1$), the proof is trivial. Let us consider at least two unique points. Recall that by representing the inverse kernel as a linear spline as we did above, the likelihood function can be

written as

$$\ell(\beta) = \sum_{i=1}^{n^*} -\pi_i \log(\beta_i) - n \log \left(\sum_{j=1}^{n^*-1} \frac{\log(\beta_{j+1}) - \log(\beta_j)}{\left(\frac{\beta_{j+1} - \beta j}{x_{j+1} - x_j}\right)} \right)$$

Now suppose we consider $g(x)$ to be strictly linear on $[x_1, x_n]$ with $g(x_n) = 1$. This leads to

$$\beta_i = \beta_1 + \frac{1 - \beta_1}{x_n - x_1} \times (x_i - x_1)$$

$$\sum_{j=1}^{n^*-1} \frac{\log(\beta_{j+1}) - \log(\beta_j)}{\left(\frac{\beta_{j+1} - \beta j}{x_{j+1} - x_j}\right)} = \frac{-\log(\beta_1)}{\left(\frac{1 - \beta_1}{x_{n^*} - x_1}\right)}$$

Plugging these values in, we can rewrite the likelihood function as

$$\ell(\beta_1) = -\pi_1 \log(\beta_1) - \sum_{i=2}^{n^*} \pi_i \log \left(\beta_1 + \frac{1 - \beta_1}{x_{n^*} - x_1} \times (x_i - x_1) \right) - n \log \left(\frac{-\log(\beta_1)}{\frac{1 - \beta_1}{x_{n^*} - x_1}} \right)$$

Assuming $\beta_1 < 1$,

$$= -\pi_1 \log(\beta_1) - \sum_{i=2}^{n^*} \pi_i \log \left(\beta_1 + \frac{1 - \beta_1}{x_{n^*} - x_1} \times (x_i - x_1) \right) - n \log \left(-\log(\beta_1) \right) - n \log \left(\frac{1 - \beta_1}{x_{n^*} - x_1} \right)$$

When taking the limit as $\beta_1 \to 0$, we note that $\sum_{i=2}^{n^*} \pi_i \log\left(\beta_1 + \frac{1-\beta_1}{x_{n^*} - x_1} \times (x_i - x_1)\right)$ and $n \log\left(\frac{1-\beta_1}{x_{n^*} - x_1}\right)$ both approach a constant. This leaves us with

$$\lim_{\beta_1 \to 0} \ell(\beta_1) = -\pi_1 \log(\beta_1) - n \log(\log(\beta_1)) + c$$

$$= -n \log(\beta_1^{(\pi_1/n)} \log(\beta_1)) + c$$

By L'Hôpital's rule, this approaches ∞ as $\beta_1 \to 0$. As $\beta_1 \to 0$, the estimated probability density approaches a degenerate distribution centered around x_1.

□

Because of this, the maximum likelihood estimate is undefined. However, we have found that using a local maximum of the likelihood function leads to a very useful estimation tool, similar to the Gaussian mixture problem. In addition, we found that in practice, avoiding the domain of attraction to the degenerate estimate was quite easy, especially as n becomes larger.

To present heuristic evidence of the shrinking domain of attraction, we present some toy datasets. We parameterize the estimator as was done in the theorem above, and plot the log likelihood as a function of the single parameter β_1. In these datasets, the observed values are set to the $\frac{1}{n+1}, ..., \frac{n}{n+1}$ quantiles of either a uniform(0,1) distribution or an $F_{1,1}$ distribution. We chose a uniform(0,1) for simplicity, as we the mode we are interested in will be known to be $\beta_1 = 1$. We chose an $F_{1,1}$ because this is a distribution which has unbounded density as $x \to 0$, so we expect the domain of attraction to the degenerate mode to be close to the local mode of interest and we were concerned the domain of attraction to the local mode may blend into the

domain for the degenerate mode. The plots for $n = 2$, 5, and 25 can been seen on figure 5.1.

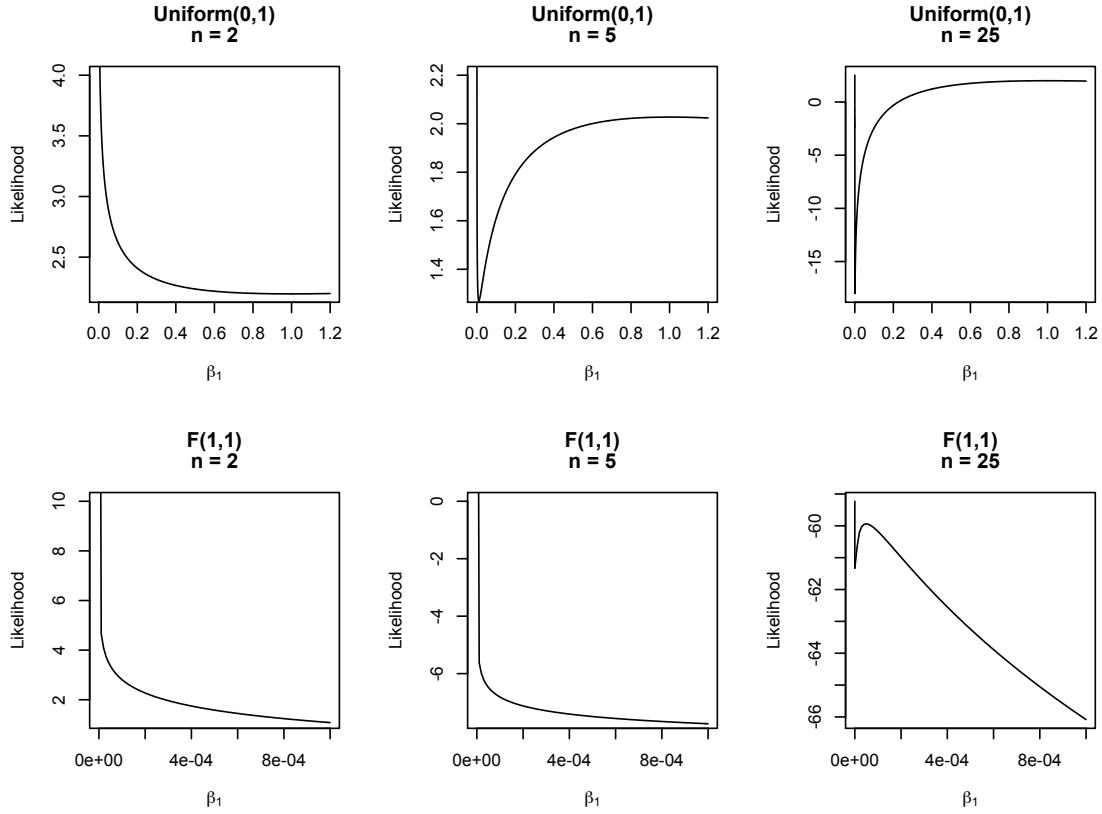

Figure 5.1: Likelihood as a function of β_1

We note a few things from these plots. First is that for both the uniform and $F_{1,1}$ distribution, the domain of attraction to the degenerate mode appears to be $(0, \infty)$ for the case of $n = 2$. However, for the uniform data points, by $n = 5$, the domain of attraction becomes quite small, approximately (0, 0.01). For the $F_{1,1}$ distribution, the domain of attraction appears to still be $(0, \infty)$ for $n = 5$. For $n = 25$, the domain of attraction became incredibly small: approximately $(0, 10^{-11})$ for the Uniform(0,1) and the $F_{1,1}$ datasets, although the local mode of interest for the $F_{1,1}$ is much closer than the Uniform (approximately 10^{-4} compared to 1).

We found that the algorithm we present in section 5.4 avoids the domain of attraction to the degenerate estimate with very high probability for all but the smallest of datasets. We present simulations to show this in the end of section 5.4. In addition, the algorithm avoided the domain of attraction to the degenerate mode in all 15,000 simulated datasets which were used for Monte Carlo evaluation of the inverse-convex estimator presented in section 5.5 and in the illustrative examples presented in section 5.6. Because of this, the issue of the unbounded likelihood is more of a theoretic problem than a practical problem.

5.3.3 Defining the Inverse Convex Estimator

Because the likelihood function is unbounded as it approaches a degenerate solution, we cannot simply classify the estimator as a maximum likelihood estimate. Instead, we will define the estimator as a local maximum of the likelihood function, much like the Gaussian mixture model. Defining a local maximum for the inverse convex is a little more tricky than the Gaussian mixture problem due to the shape constraints. Doing so will require the active set parameterization similar to those presented in section 3.4.

Let us define $\Delta_i = \frac{\beta_{i+1}-\beta i}{x_{i+1}-x_i}$. The inverse convex constraint implies $\Delta_{i-1} \leq \Delta_i$ In theory, we define the i^{th} point as active if $\Delta_{i-1} < \Delta_i$ and inactive if $\Delta_{i-1} = \Delta_i$. In practice, numeric error prevents exact evaluation of $\Delta_{i-1} = \Delta_i$, so we slacken this constraint by defining a point to inactive if $\Delta_{i-1} \geq \Delta_i + \xi$ and active if $\Delta_{i-1} < \Delta_i + \xi$ for a specified ξ. In our implementation, we define $\xi = 10^{-13}$.

Under the active set parameterization we treat $g(x)$ as a linear spline with knots only at the active points and will adjust the β_i's as such. We will use the notation β_i^* to denote when we are using the active set parameterization. When using the active

set parameterization, if we increase the active parameter β_i^*, we also increase the neighboring inactive β_j's as though the active points were the only knots of the linear spline *i.e.* the inactive β_j are determined by linear interpolation from the nearest active points. To demonstrate this, figure 5.2 demonstrates subtracting 1 from β_4^*. This starts with β_4 as an inactive point and makes it active as g is now kinked at x_4 . It also decreases the values of β_3 and β_5, as they are the surrounding inactive points.

To formally characterize addition under the active set parameterization, define $a(m)$ to be the index of the m^{th} active point. If $i = a(m)$ then $\beta_i^{*(t+1)} = \beta_i^{*(t)} + h$ is equivalent to

$$\beta_j^{(t+1)} = \begin{cases} \beta_j^{(t)} + h \times \frac{x_j - x_{a(m-1)}}{x_i - x_{a(m-1)}}, & \text{if } x_{a(m-1)} < x_j \leq x_i \\ \beta_j^{(t)} + h \times \frac{x_{a(m+1)} - x_j}{x_{a(m+1)} - x_i}, & \text{if } x_i < x_j < x_{a(m+1)} \\ \beta_j^{(t)}, & \text{otherwise} \end{cases}$$

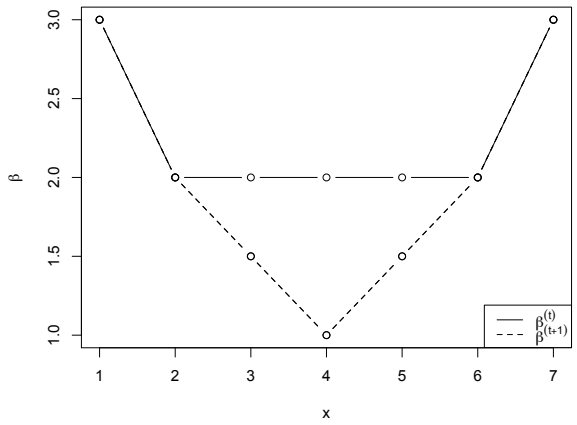

Figure 5.2: $\beta_4^{*(t+1)} = \beta_4^{*(t)} - 1$

Because of the inverse convex constraints, it is natural to consider the KKT conditions

(Kuhn and Tucker 1951) which are necessary for an estimate to be a local maximum. Let us define

$$\text{KKT error} = \max \begin{cases} |\frac{\partial \ell}{\partial \beta_i^*}|, & \text{if } \Delta_{i-1} < \Delta_i + \xi \\ \max_i(\frac{\partial \ell}{\partial \beta_i^*}, 0), & \text{if } \Delta_{i-1} \geq \Delta_i + \xi \end{cases}$$

We define the inverse convex estimator as having KKT error of 0 for all points. In addition, if i is an active point, then we further require that $\frac{\partial \ell^2}{\partial (\beta_i^*)^2} \leq 0$. In our implementation, we consider the solution to have been achieved if the KKT error was less than 10^{-4}.

It is not clear that the inverse convex estimator is unique. In the case of the Gaussian mixture, it is well known that there can be multiple distinct modes with finite likelihood and that different initial values often lead to different solutions. To test if the inverse convex estimator is unique, we simulated 50 sampled data points from a standard normal distribution and ran the algorithm presented in the next section from a uniform initial estimate, a initial estimate which concentrated mass toward the left side and finally a initial estimate which concentrated mass toward the right side. We recorded the maximum difference of estimated density at each of the data points in the data set. This was repeated 1,000 times. Over all 1,000 simulated datasets, the maximum difference in estimated density was 4.4×10^{-5}, suggesting that multiple modes is not likely to be an issue for this estimator.

5.4 Algorithm

In this chapter, we will only briefly describe the algorithm, as it is a simplified version of the algorithm found in chapter 3.

The nature of the problem is very similar to finding the log-concave estimator. The has been done efficiently with the Active Set Algorithm (ASA), presented in Dümbgen *et al.* (2011). The ASA procedure starts with a minimal set of active points and maximizes over this set, adding new active points and optimizing until the algorithm has converged. Constrained optimization over these sets is done via Sequential Quadratic Programming (SQP). This is a very efficient algorithm, as the number of active points in the solution tends to be considerably smaller than the total number of knots considered (n).

A further complication of the inverse convex estimator compared with the log-concave NPMLE without censoring is that the inverse convex likelihood function is not guaranteed to be concave. SQP will fail if the function is not locally concave at any step of the algorithm. A similar problem was faced when finding the log-concave NPMLE for interval censored data in chapter 3. We will use the same technique to address this problem. To insure convergence of the algorithm, we included a univariate optimization step. This step would optimize the active set parameters one at a time and could increase the likelihood function even when not locally concave via the bisection method. While this insured convergence, it was too slow to be used by itself, so our algorithm contained both an SQP step and a univariate optimization step. Typically, the likelihood function was not locally concave in only the first step or two of the algorithm, so rather than modifying the SQP step as we did in chapter 3, we merely skipped the SQP step if the likelihood function was found to be locally non-concave.

Two further modifications were required to deal with the inverse convex problem.

The first issue was that while that while the likelihood function was defined for all real values of the parameters in the log-concave case, the likelihood function for the inverse convex estimator is undefined if $\beta_i \leq 0$. This was dealt with by half-stepping if the proposed steps were beyond the boundary. The second issue is that we would like the algorithm to terminate if we are convinced that it has entered a domain of attraction to a degenerate distribution. To do this, we terminated the algorithm and returned an error report if the estimated density, normalized by the empirical standard deviation of the dataset, was ever greater than a pre-specified d_{max}. In our implementation, we set $d_{max} = 10^7$.

To test how often the algorithm would terminate early due to approaching a degenerate mode, we sampled data from an $F_{1,1}$ and an $F_{1,2}$ distribution. The $F_{1,1}$ was chosen because the unbounded density suggests it would behave worse than the $F_{1,2}$, which has bounded density but very heavy tails. The $F_{1,1}$ distribution is not inverse convex while the $F_{1,2}$ is on the boundary of inverse convex as mentioned in section 5.2. We tested data with samples sizes $n = 10$, 15, 20 and 25. For each n, we creating $MC = 1,000$ samples. We found that the estimated probability of terminating due to approaching a degenerate mode for each sample size was 0.353, 0.107, 0.003 and 0 respectively for the $F_{1,1}$ distribution and 0.302, 0.014, 0 and 0 for the $F_{1,2}$. In addition, the algorithm never terminated due to approaching a degenerate mode in our simulations in the next section. This suggests that while the unbounded likelihood is a theoretic issue with the inverse convex estimator, in practice our algorithm finds the local mode of interest reliably for datasets with $n \geq 25$.

We found this algorithm to be sufficiently fast for investigating the properties of the estimator for moderate sized data sets. For n = 1,000, the algorithm typically converged in under 2 seconds (2007 Macbook with 2 GHz Intel Core 2 Duo processor, 4 GB of RAM). However, we ran into problems with larger datasets. In sample sizes of

$n = 10{,}000$, we occasionally ran into numeric problems regarding the constraints in the quadratic programming routine. When the algorithm did converge, it typically took around 1 minute. We do note that our algorithm converged on all of the datasets we used in our real data applications in section 5.6, including 4 datasets of approximately $n = 7,000$ and one data set of $n = 28,155$. For the largest dataset, the algorithm converged in just under 5 minutes. We still caution that simulated data suggests that the algorithm will not generally be reliable for datasets of that size in its current implementation. Further work is necessary to develop a reliable algorithm for large datasets.

5.5 Simulations

One of the motivations for shape constrained inference is density estimation. Our motivation for investigating the inverse convex estimator began with estimating survival curves for interval censored data. Because of this, we will use simulations to compare both density estimation and survival estimation for the log-concave NPMLE and the inverse NPMLE.

We compared five different distributions. These were a standard normal, a gamma(2,2), a t-distribution with 3 degrees of freedom, an F-distribution with $\nu_1 = 3, \nu_2 = 3$ and a gamma(0.5, 0.5). Note that the normal and gamma(2,2) distribution are both log-concave but the remaining distributions are not. All of the distributions are inverse convex with the exception of the gamma(0.5, 0.5) which has unbounded density at 0. For each of these distributions, we will examine the estimated density at the true median, the estimated median and the estimated 90^{th} percentile. We consider sample sizes of $n = 50$, 100 and 500. For each sample size, we will generate $MC = 1,000$ simulated datasets and fit each estimator to each dataset. The means and standard deviations for the various estimates can be seen on tables 5.1 - 5.3. We also provide

sample plots of the estimated density and cumulative distributions function for the t and F distributions with $n = 1,000$ in figure 5.1.

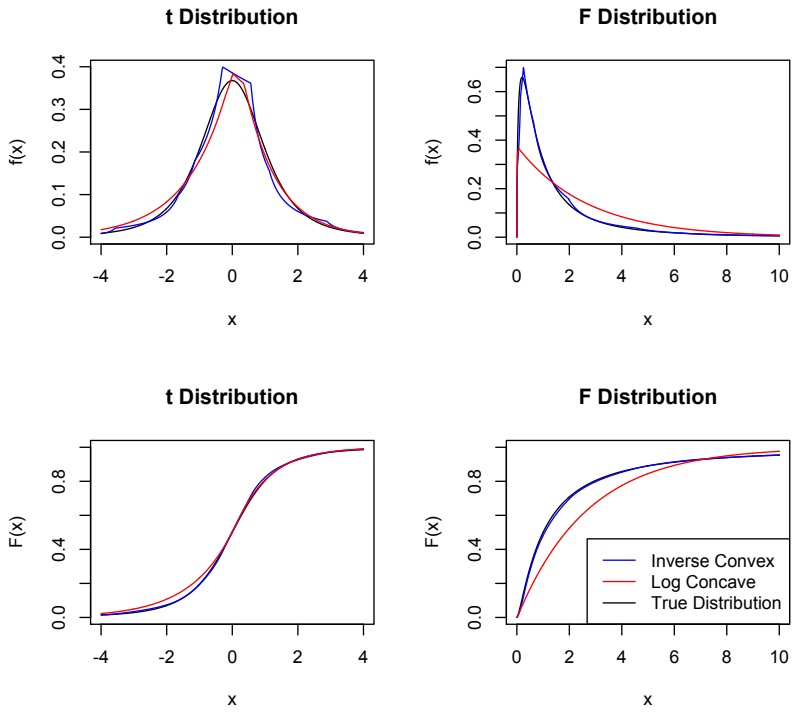

Figure 5.3: Inverse convex and log-concave estimators based on random sample of $n = 500$ from a t_3 and $F_{3,3}$ distribution

In our simulations, we saw a few trends. For log-concave distributions, the log-concave estimator had lower standard deviations for all estimators, although the magnitude of this advantage varied. For density estimation, the reduction was significant, but for survival estimation the reduction was marginal. In addition, as n increased, the relative advantage decreased. At $n = 500$ for log-concave simulated data, the estimators were nearly equivalent. We noticed that the log-concave estimator does a surprisingly good job of estimating the non log-concave t-distribution. We speculate that there are two reasons for this. First is that the t-distribution is only mildly non log-concave. In fact, the t-distribution is log-concave on the interval $[-\sqrt{\nu}, \sqrt{\nu}]$, where

n = 50		True Values	Inverse Convex	log-concave
N(0,1)	Density	0.40	0.42 (.090)	0.40 (.065)
	Median	0	0.00 (.166)	0.00 (.158)
	90^{th} Perc	1.28	1.22 (.222)	1.27 (.214)
Gamma	Density	0.63	0.65 (.132)	0.62 (.084)
(2,2)	Median	0.84	0.83 (.104)	0.86 (.097)
	90^{th} Perc	1.94	1.89 (.234)	1.94 (.221)
t_3	Density	0.37	0.38 (.098)	0.34 (.061)
	Median	0	0.00 (.186)	0.00 (.182)
	90^{th} Perc	1.64	1.55 (.366)	1.76 (.411)
$F_{3,3}$	Density	0.32	0.33 (.073)	0.27 (.063)
	Median	1	0.97 (.202)	2.00 (1.33)
	90^{th} Perc	5.39	5.21(1.68)	6.50 (4.42)
Gamma	Density	0.47	0.45 (.082)	0.64 (.066)
(0.5,0.5)	Median	0.45	0.35 (.122)	0.70 (.138)
	90^{th} Perc	2.71	2.53 (.542)	2.29 (.449)

Table 5.1: Simulated Comparisons with $n = 50$ based MC = 1000 simulations. Values in parentheses are standard deviations. "Density" refers to estimated density at the true median. 90^{th} Perc refers to the 90^{th} percentile.

ν = degrees of freedom. Secondly, the t-distribution is symmetric. This means the log-concave constraint "pulls" evenly from both sides of the mode. In contrast, we see the log-concave estimator suffers extreme bias when estimating the F and gamma(0.5, 0.5) distributions. The inverse convex estimator handles these distributions very well. Finally, we found that the inverse convex estimator did suffer from bias when estimating the gamma(0.5,0.5) distribution. However, these biases were very minor compared with biases the those that the log-concave estimator displayed. These simulations suggest that the inverse convex estimator would be preferred if the there were concerns of heavy tails, especially if the data were skewed.

5.6 Real Data Application

The characteristics of the inverse convex estimator can be summarized as unimodal, allowing for skew and very heavy tails. There are many areas of research for which

n = 100		True Values	Inverse Convex	log-concave
N(0,1)	Density	0.40	0.41 (.066)	0.40 (.053)
	Median	0	0.00 (.115)	0.00 (.112)
	90^{th} Perc	1.28	1.24 (.150)	1.28 (.145)
Gamma	Density	0.63	0.65 (.096)	0.62 (.065)
(2,2)	Median	0.84	0.84 (.076)	0.85 (.072)
	90^{th} Perc	1.94	1.90 (.170)	1.95 (.160)
t_3	Density	0.37	0.37 (.062)	0.35 (.046)
	Median	0	0.00 (.124)	0.00 (.124)
	90^{th} Perc	1.64	1.59 (.256)	1.77 (.309)
$F_{3,3}$	Density	0.32	0.33 (.052)	0.26 (.052)
	Median	1	0.99 (.144)	0.70 (1.01)
	90^{th} Perc	5.39	5.31 (1.24)	6.67 (3.34)
Gamma	Density	0.47	0.44 (.054)	0.63 (.048)
(0.5,0.5)	Median	0.45	0.36 (.090)	0.70 (.099)
	90^{th} Perc	2.71	2.63 (.438)	2.31 (.326)

Table 5.2: Simulated Comparisons with $n = 100$ based MC = 1000 simulations. Values in parentheses are standard deviations. "Density" refers to estimated density at the true median. 90^{th} Perc refers to the 90^{th} percentile.

n = 500		True Values	Inverse Convex	Log-concave
N(0,1)	Density	0.40	0.40 (.032)	0.40 (.030)
	Median	0	0.00 (.055)	0.00 (.054)
	90^{th} Perc	1.28	1.27 (.070)	1.28 (.069)
Gamma	Density	0.63	0.63 (.049)	0.63 (.039)
(2,2)	Median	0.84	0.84 (.036)	0.84 (.035)
	90^{th} Perc	1.94	1.93 (.077)	1.95 (.071)
t_3	Density	0.37	0.37 (.034)	0.37 (.030)
	Median	0	0.00 (.057)	0.00 (.057)
	90^{th} Perc	1.64	1.62 (.122)	1.75 (.151)
$F_{3,3}$	Density	0.32	0.33 (.052)	0.26 (.052)
	Median	1	0.99 (.144)	0.70 (1.01)
	90^{th} Perc	5.39	5.31 (1.24)	6.67 (3.34)
Gamma	Density	0.47	0.43 (.036)	0.64 (.022)
(0.5,0.5)	Median	0.45	0.36 (.051)	0.69 (.044)
	90^{th} Perc	2.71	2.75 (.494)	2.30 (.146)

Table 5.3: Simulated Comparisons with $n = 500$ based MC = 1000 simulations. Values in parentheses are standard deviations. "Density" refers to estimated density at the true median. 90^{th} Perc refers to the 90^{th} percentile.

this describes the type of data to be expected. At this time, we will examine wages and incomes. Moscarini (2005) describes an ideal wages model to have a unique interior mode, skewness and a long and fat right tail. Income may be further right skewed than wages, as it includes investments as well. This is the type of data we expect the inverse convex estimator to do quite well with.

We first considered a dataset collected by the Philippines' National Statistics Office. This dataset can be found in the publicly available CRAN package "ineq", titled "Ilocos". The dataset contains 632 subjects and their reported income, along with other covariates. One individual reported 0 income. We dropped this individual from our dataset for two reasons. First, typical parametric models for income, such as log normal, do not allow for 0 income. It should be noted that the log-concave and inverse convex models would not have such problems. However, it is more reasonable to model income as a mixture of those with no reported income and those with positive reported income. Therefore we will only model the income of individuals with positive reported income. Of these individuals, 116 came from the province La Union and 381 were from Pangasinan.

We would like to examine the fits of various models to the distribution of wealth in the two provinces. We will do this by comparing estimated probability densities and cumulative distribution functions with histograms and the empirical distribution functions. In addition, we will examine the Lorenz Curve and Gini coefficient, classic econometrics measures of disparity within a society. The Lorenz Curve states x proportion of the poorest subjects in a society own y proportion of the wealth (or in this case, income). The Gini coefficient is equal to $1-2\int_0^1 L(x)\mathrm{d}x$, where $L(x)$ is the Lorenz curve, or a one number summary of the Lorenz curve. The Gini coefficient ranges from 0 to 1 (assuming only non negative measures of wealth), with 0 being perfectly evenly distributed wealth in the society and $\frac{N-1}{N}$ if 100% of the wealth

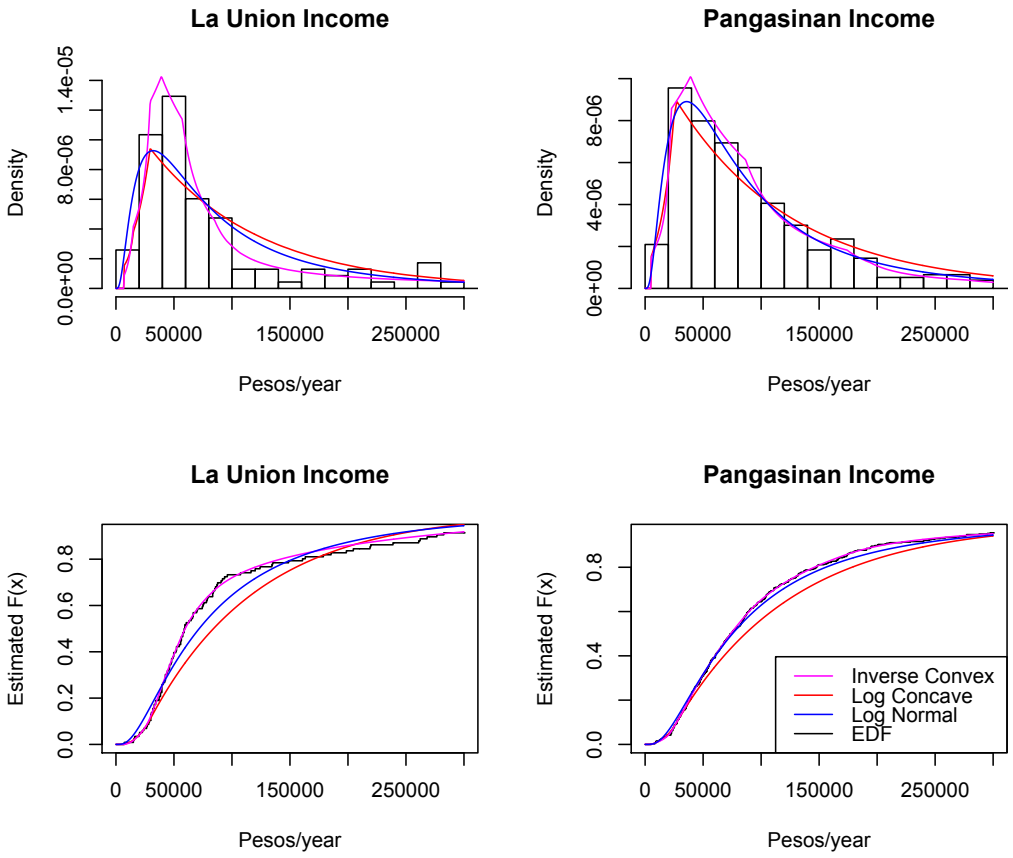

Figure 5.4: Fits of the Distribution of Income in the Pangasinan and La Union Provinces

belongs to only one individual. Larger values of the Gini coefficient corresponds with more disparity within a society. The estimated Lorenz curve and Gini coefficient can be calculated from an estimated cumulative distribution function.

We fit three models and compared the fits of each model. For shape constrained models, we fit the inverse convex and log-concave models. We also fit a log normal model, as this is a classic parametric model used for income. We used the Empirical Distribution Function to check the fits of our model. The fits can be seen on figure 5.2. We only plotted the fits up until 300,000 pesos/year to make the plots easier to

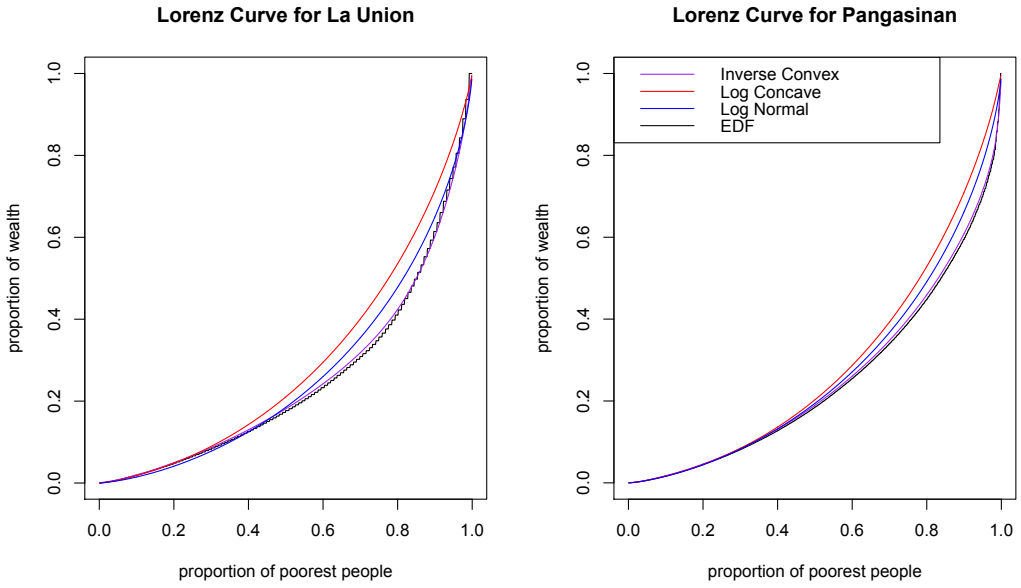

Figure 5.5: Lorenz Curves for the Pangasinan and La Union Provinces

examine.

Examining the fits, the inverse convex estimator matches the empirical distribution function much better than the log-concave estimator, while still providing a smooth estimated cdf and valid density estimates (the EDF does not). The log normal estimator fits the EDF considerable better than the log-concave estimator, but still not as well as the inverse convex estimator. The log-concave estimator is unable to properly model the heavy tails of the distribution. Although the overall fit of the inverse convex estimator is very good, we do note that there is some disagreement around 20,000 pesos/year in the La Union cdf between the inverse convex and EDF estimates. Given the smaller sample size of the La Union ($n = 116$), we believe the observed difference between the EDF and inverse convex fit is just noise rather than systematic mis-modeling. Despite this, the inverse convex still looks to be a much better fit than the other models for the La Union data. In the Pangasinan data, the

Model	La Union	Pangasinan
Empirical	0.514 (0.459, 0.569)	0.502 (0.436, 0.568)
Inverse Convex	0.500 (0.440, 0.560)	0.488 (0.427, 0.549)
log-concave	0.422 (0.426, 0.490)	0.433 (0.410, 0.456)
Log Normal	0.474 (0.420, 0.528)	0.458 (0.420, 0.496)

Table 5.4: Estimated Gini coefficients for each model. Values in parentheses are bootstrapped 95% CIs

inverse convex estimates look to be an unquestionably good fit.

We see a similar trend in the Lorenz curves. The inverse convex estimator appears to be a smoothed version of the EDF. We can see a strong systemic difference between the log normal and the EDF curves. The bias appears even higher for the log-concave curve. Finally, for each model we computed the Gini coefficient for both populations and created a 95% bootstrap CI using $B = 1,000$ bootstrap samples. These are presented on table 5.4. We note that the inverse convex estimates agree the most with the empirical estimate and we suspect that the greater observed differences for the log-concave and log normal models is due in part with these models not allowing for heavy enough tails. While the difference in Gini coefficients was not statistically significant in any of the groups, we note that the log-concave estimate reversed the ranking of the estimates compared to the other estimators. We were surprised that the empirical bootstrap confidence intervals for the La Union group ($n = 116$) were narrower than the inverse convex confidence intervals. A literature review on the topic revealed that bootstrap confidence intervals for the Gini coefficient typically are too narrow for smaller samples (Dixon *et al.* 1987), so we believe this is responsible for the narrower confidence interval based on the empirical estimator.

In this example, the inverse convex estimator appeared to fit the data very well, with the log normal providing a slightly worse fit but still much better than the log-concave fit. This was not always the case. We examined several income and wage data sets and

while the inverse convex fit always appeared to be the best fit, in some datasets the log-concave fit appeared a superior fit compared to the log normal. The general trend we found was that datasets with more disparity (*i.e.* higher Gini coefficients) tended to be better fit by the log normal distribution, while datasets with less disparity tended to be better fit by the log-concave fit. This trend would be expected, as the log-concave fit should be more flexible than log normal in general, but will break down for heavy tailed data. We emphasize again that in both situations, the inverse convex estimator appeared to be the best fit of the three considered.

To demonstrate this general trend we found, we present the data set "CPS1988", found in the CRAN package "AER". This dataset contains wage data collected from the March 1988 Current Population Survey conducted by the US Census Bureau. A total of 28,155 subjects are included in the dataset. They are divided into 4 different regions, Northeast ($n = 6,441$), Midwest ($n = 6,863$), South ($n = 8,760$) and West ($n = 6,091$). Wages are given in weekly salaries. We plotted the fitted cdf's for each region individually and also all regions aggregated using the same models as before. Again, we zoom into the interval (500, 2,000) to allow easier comparison of fits (maximum observed value was 18,777). We also list the Gini coefficient, as calculated by the empirical distribution model. Fits are presented on figure 5.6 and figure 5.7.

Figure 5.6: Estimated wage cdf from CPS1988 dataset

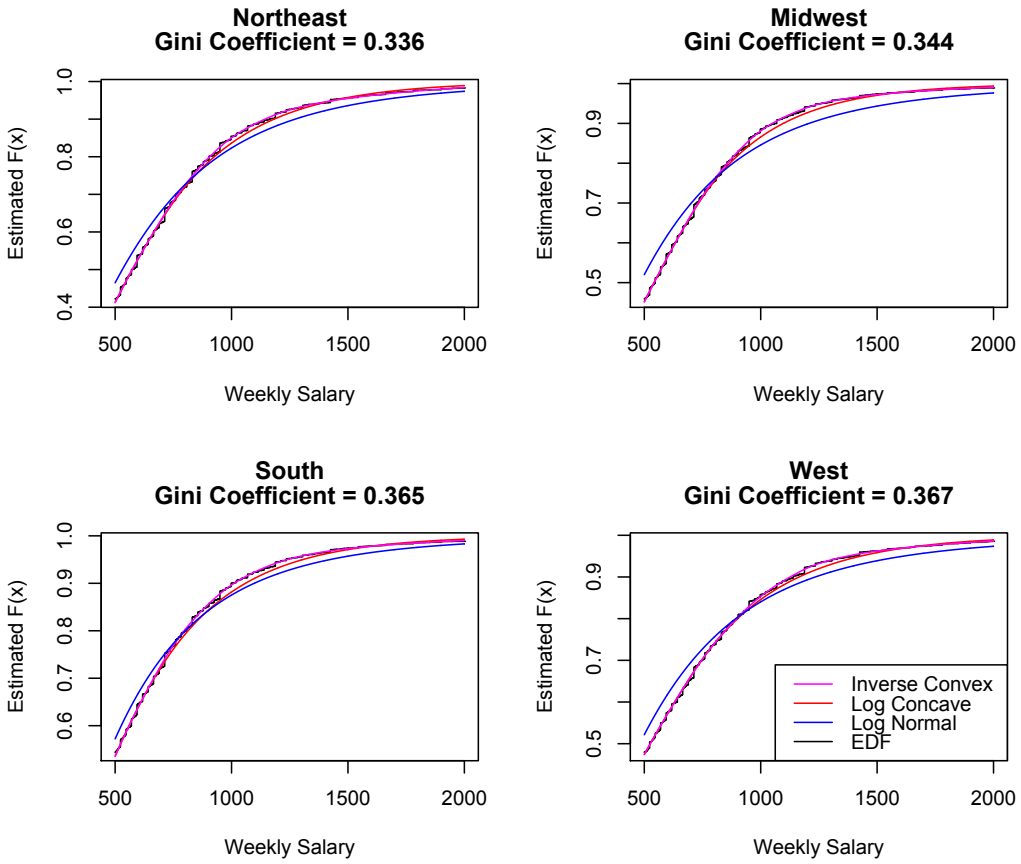

Figure 5.7: Estimated wage cdf's by region from CPS1988 dataset

We note that in this case, the inverse convex again appears to be the best fit in all scenarios. However, this time the log-concave does fairly well and the log normal does poorly. The fact that the inverse convex estimator always appears to fit best and both of the other estimators can suffer heavy bias under different scenarios makes the inverse convex estimator a very attractive choice.

5.7 Conclusion

The inverse convex constraint leads to a very flexible unimodal shape constraint, allowing for very heavy tails but not allowing for unbounded density. Theory dictates

that the family of inverse convex functions is more flexible than the log-concave family, as the log-concave family is properly contained within the inverse convex family. The inverse convex estimator does have the theoretic problem of an unbounded likelihood function leading to a degenerate maximum likelihood estimate. However, in practice it is quite easy to avoid these degenerate estimates and find a more useful local maximum as an estimate. Simulations show us that the inverse convex estimator is nearly as efficient as the log-concave estimator for quantile estimation when the true distribution is log-concave, although the log-concave estimator may be preferred for density estimation for smaller samples. We found the log-concave estimator suffered from heavy bias from heavy right skewed distributions. The inverse convex estimator behaved very well when estimating quantiles of unimodal heavy tailed distributions, even when the inverse convex assumptions was violated due to unbounded density. Applying the estimator to real data, we see that the inverse convex estimator can fit income and wage data much better than the log-concave or log normal distribution, both of which suffered heavy bias under different scenarios. We would recommend the inverse convex for modeling of distributions in which the investigator may be concerned with skewed heavy tails.

We are particularly interested in applying the inverse convex estimator to censored data, although currently an algorithm for such data has not been implemented. Another proposed research topic on the inverse convex estimator would be to find formal bounds on the domain of attraction toward the degenerate estimate. We are interested in finding a function such that when $f(\beta, x) < \xi$, β was insured to be outside the domain of attraction. This would both help our algorithm in smaller samples and help define the limitations of the inverse convex estimator.

Chapter 6

Efficient Computation of the Bivariate NPMLE

The author's exposure to interval censored data began with construction of an efficient algorithm for computing the NPMLE for bivariate interval censored data. This is not related to shape constrained estimation, so this contribution was moved to the last chapter. In addition, the recently published CRAN package "MLEcens" by Marloes Maathius implements an algorithm that is found to be slightly faster in typical cases of bivariate interval censored data, so this work is unlikely to be published in a statistical journal. The work is not without merit, though. The algorithm here was found to be significantly faster in worst case scenarios and needs no modification to be implemented for multivariate with dimensions greater than two, in which a worst case scenario is more probable. The algorithm can also be seen as a blue print for algorithms which need to rapidly determine which parameters will be set to 0 at the

MLE.

6.1 Introduction

We begin by introducing bivariate interval censored data and formal characterization of the bivariate NPMLE.

6.1.1 Bivariate Interval Censored Data

Bivariate survival data occurs when time to event data is recorded for two values on each subject. If the two variables are independent, the marginal distributions will be fully informative and and standard univariate techniques will suffice. However, if the relation of the two variables is of interest, they must be jointly modeled. In some cases, the event times will be strictly ordered, such as the time to infection of HIV and time to onset of AIDS (De Gruttola, Lagakos 1989). In other cases, the events will not necessarily be ordered, but are very likely to be correlated, such the development of different types of cataracts (Wong and Yu, 1999). Other classic data sets include time to CMV shedding in the blood and time to CMV shedding in the urine among HIV positive patients (Goggins and Finklestein 2000), time to onset of bacterial and viral infections and the relation between the two in AIDS Clinical Trials Groups study (Betnesky, Finkelstein 1999), emergence times and their relationship for permanent teeth (Bogaerts *et al.* 2002) and a competing risks dataset between age at natural menopause and age at artificial menopause (Maathius 2006).

In addition to examining associations between two event times, bivariate modeling can be used to analyze doubly censored data. For doubly censored data, both the time event time and time of origin may be censored. The earlier mentioned example of time to infection and time to onset of AIDS is an example of doubly censored data

when the scientific question of interest is about the time from infection of HIV to the onset of AIDS. This can be addressed by modeling the data using bivariate methods and extracting the estimate of interest from the bivariate estimate. If the time of interest is $T = Y - X$, and we have estimated the bivariate distribution of (X, Y), we can derive the estimated survival function for T by

$$\hat{S}_T(t) = 1 - \int_0^\infty \int_x^{x+t} \hat{f}_{(X,Y)}(x, y) dy dx$$

It is important to note that for these estimates to be fully informative, T must be independent of X. If not, we may be averaging over interesting patterns in the data in this simple model. For example, suppose X is the age at which a patient becomes HIV positive and Y is the development of AIDS. If time from HIV infection to development of AIDS is dependent on age, then using the above technique would miss this association. More advanced techniques could be used to adjust for X.

6.1.2 Bivariate NPMLE for Interval Censored Data

The bivariate NPMLE is considered the standard for bivariate estimation for interval censored data. This estimator has the advantage of flexibility, as it does not make any assumptions about of the underlying bivariate distribution of the true event times. However, it still requires the standard assumptions of independence of censoring mechanisms and event times (although the two censoring mechanisms can be dependent on each other). As is typical with such non-parametric models, this flexibility comes at the price of high variation in the estimates of interest. In addition, while the bivariate NPMLE can handle exact observations times in addition to censored times and is consistent under mild conditions, the estimator has the peculiar problem that

it is not consistent for singly censored data (in which one of the bivariate times is censored and the other is not), although this can be repaired by lightly censoring the exact observation (Laan, M. 1996).

Despite these problems, finding a non-parametric estimate of the bivariate survival function is of interest for several scientific purposes, including model checking of fully parametric or semiparametric models, such as the copula model (Clayton 1978). In order to find the bivariate NPMLE, we must first define the likelihood function. This requires some notation to be defined first. For the i^{th} observation, define L_{i1} to be the beginning of the interval known to contain the first event time, U_{i1} to be the end of the interval known to contain the first event time, and likewise L_{i2} and U_{i2} for the second event time. If the exact time of the first event is known for the i^{th} observation, then $L_{i1} = U_{i1}$. If the i^{th} observation is right censored for the first event, then $U_{i1} = \infty$. Noting that this set of four values determines a rectangle in two dimensions, define

$$P(O_i) = F(L_{i1}, L_{i2}) - F(L_{i1}, U_{i2}) - F(U_{i1}, L_{i2}) + F(U_{i1}, U_{i2})$$

In other words, $P(O_i)$ is the probability of observation i conditional on bivariate cdf F. This leads to likelihood function

$$\ell(F) = \prod_{i=1}^{n} P(O_i)$$

6.2 Support Set

When calculating the univariate NPMLE, Turnbull (1976) showed that all probability mass must be assigned to *Turnbull intervals* in the NPMLE. A Turnbull interval is an interval for which the left endpoint of the interval is the left endpoint of an observation interval (i.e. L_i) and the right endpoint is the right endpoint of an observation interval (i.e., R_j), with no other observation endpoints in between. To illustrate this, in figure 8.1, we see three observations intervals (in black), $L_1 = 1$, $U_1 = 4$, $L_2 = 3$, $U_2 = 9$, $L_3 = 6$ and $U_3 = 7$. This leads to two Turnbull intervals (in red): (3,4) and (6,7). How probability mass is assigned *between* Turnbull intervals in the NPMLE is unique. This is referred to as *mixture uniqueness* in Gentleman and Vandal (2002) and Vandal (1998) presented a proof of mixture uniqueness of the univariate NPMLE. On the other hand, how probability mass is assigned *within* a Turnbull interval does not affect the likelihood function and thus the NPMLE is not unique. This is referred to as *representative non-uniqueness* (Gentleman and Vandal 2002). Thus, we can parameterize the problem by treating the estimator as a discrete probability mass with $p_1, p_2, ..., p_k$, where p_i = probability mass assigned to the i^{th} Turnbull interval. From this, we need to find the unique $p_1, p_2, ..., p_k$ that maximize the likelihood function.

When calculating the multivariate NPMLE, Turnbull intervals are generalized by *maximal intersections* (Wong and Yu 1999). An intersection A_j of observation rectangles is a maximal intersection if $\forall\ i\ (A_j \cap O_i = \varnothing \text{ or } A_j \cap O_i = A_j)$ and $\exists\ i$ $A_j \cap O_i = A_j$. Another way to interpret maximal intersections is that if we considered a height map, in which the height of an area is given by the number of overlapping observation rectangles, the maximal intersections are the local maxima of heights (Maathius 2005). Much like the univariate NPMLE, the multivariate NPMLE suffers from representation non-uniqueness. Unlike the univariate NPMLE, the multivari-

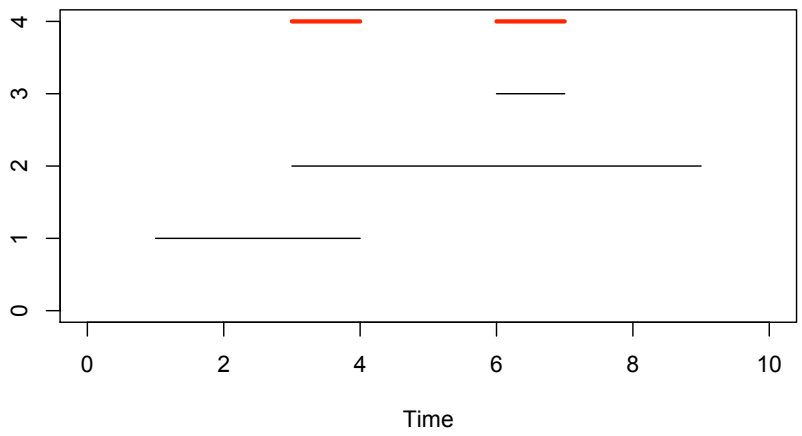

Figure 6.1: Example Turnbull Intervals

ate NPMLE also suffers from mixture non-uniqueness if the number of dimensions is greater than or equal to 2. This can be viewed in Figure 8.2. In this case, there are 4 overlapping observation rectangles, each one overlapping with two neighboring observational rectangles. This leads to a total of 4 maximal intersections, *i.e.* the overlaps of the observation rectangles. If we let p_1, p_2, p_3 and p_4 denote the probability assigned to each maximal intersection ordered clockwise, we can see that the NPMLE sets $p_1+p_2 = p_2+p_3 = p_3+p_4 = p_4+p_1 = 1/2$. Because there are only three linearly independent equations, there are an infinite number of solution which all lead to the same likelihood. Further discussion of the conditions that lead to mixture non uniqueness can be found in Gentleman and Vandal (2002).

In the univariate case, the definition of Turnbull intervals lead to a very natural method for finding all the Turnbull intervals. In the multivariate case, efficiently finding all the maximal intersections is not so simple. Thus, two types of algorithms are used in computing the multivariate NPMLE: *reduction algorithms*, which find all the maximal intersections and *optimization algorithms* which optimize the probability mass assigned to the maximal intersections. Recently, Maathius (2005) presented the

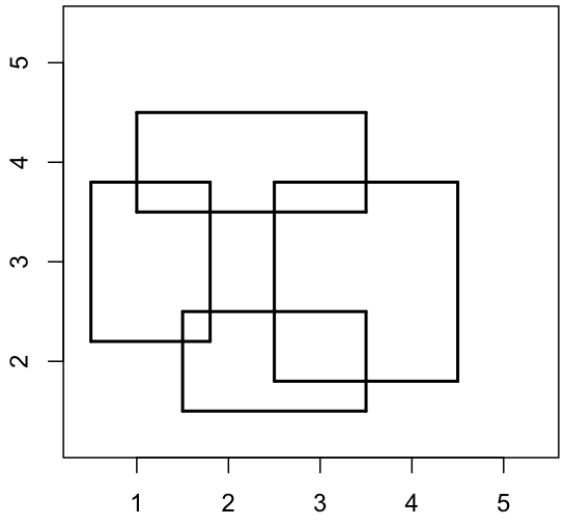

Figure 6.2: Example of Non-Uniqueness

HeightMap algorithm which finds all the maximal intersections in $O(n^2)$ time. As the name indicates, the algorithm works by creating a height map and sweeping through, recording all the local maxima of the height map. Currently, HeightMap is considerably faster than any optimization algorithm (including those presented in MLEcens and the one presented in this thesis)

Along with the location of the maximal intersections, the HeightMap algorithm returns an $m \times n$ clique matrix, in which the $(i, j)^{th}$ enter is equal to 1 if the i^{th} maximal intersection is contained within the j^{th} observational rectangle and 0 otherwise. Thus, if the clique matrix is $\mathbf{C}$ and $\mathbf{p}$ is a vector of length m, such that $p_i \geq 0$ and $\sum_{i=1}^{m} p_i = 1$, an optimization algorithm needs to find

$$\arg\max_{\mathbf{p}} \sum_{i=1}^{m} \log(\mathbf{C} \times \mathbf{p})_i$$

It is worth noting that at once the clique matrix is obtained, the dimension of the censored data is irrelevant as the data is handled in the same way (although higher dimensions will lead to a clique matrix with more rows). Gentleman and Vandal (2002) discuss how patterns in the clique matrix can differ for bivariate data and univariate data and this leads to the univariate NPMLE having mixture uniqueness but the bivariate NPMLE having mixture non-uniqueness.

6.3 Stopping Criterion

Many statistical algorithms will use the difference in likelihood function, *i.e.* $\ell(\theta^{(t+1)}) - \ell(\theta^{(t)})$, as a stopping criterion. For the case of the NPMLE for interval censored data, using such a stopping criterion can lead to premature termination of the algorithm. In particular, this is a big problem in the case of the basic EM algorithm presented in Turnbull (1976). The algorithm can make very little improvement in the likelihood function, despite being far away from the NPMLE.

Instead, we will use the stopping criterion presented by Böhning (1996). It was shown that if

$$\max\left(\frac{\partial \ell(\mathbf{p^{(t)}})}{\partial p_j^{(t)}} - n : j = 1, ..., m\right) \leq \epsilon$$

then $\ell(\hat{\mathbf{p}}) - \ell(\mathbf{p}^{(t)}) \leq \epsilon$. It is also worth noting that

$$\frac{\partial \ell(\mathbf{p^{(t)}})}{\partial p_j^{(t)}} = \sum_{j=1}^{n} \frac{\mathbf{C}_{i,j}}{n \sum_{k=1}^{m} \mathbf{C}_{k,j} p_k}$$

where $\mathbf{C}$ denotes the clique matrix produced by the HeightMap algorithm. This is

a value which will be used in another step of the algorithm. Thus, the only computational cost is finding the max of the m values. In contrast, calculating the log likelihood is $O(mn)$ and is not necessary for any of the steps of our proposed algorithm.

6.4 Current Algorithms

6.4.1 Basic EM Algorithm

The first optimization algorithm was introduced in Turnbull (1976) and was used to find the NPMLE for univariate data. This simple algorithm very naturally extends to the bivariate case. This algorithm by itself can be a stand alone optimization algorithm for the bivariate NPMLE. Empirically, it is found that the algorithm is unsatisfactory due to slow computation speeds. However, it is a very useful step in the algorithm we propose, so we will explain it in detail.

The observed log likelihood can be written as

$$\ell(\mathbf{p}) = \sum_{j=1}^{n} \log \left(\sum_{i=1}^{m} \mathbf{C}_{i,j} p_i \right)$$

The summation inside the log function makes optimization non-trivial. However, if the count of events that occurred within each maximal intersection was known, maximization would be in closed form. Define the missing data to be a vector $\mathbf{k}$, where k_i = number of events that occurred within maximal intersection i. In this case, the complete data log likelihood can be written as

$$\ell(\mathbf{p}|\mathbf{k}) = \sum_{j=1}^{m} k_j \log(p_j)$$

Noting that this is equivalent to the log likelihood function for a multinomial distribution. Thus, the complete data likelihood is maximized by

$$\hat{p}_i = \frac{k_i}{\sum_{j=1}^{m} k_j} = \frac{k_i}{n}$$

Conditional on $\mathbf{p}$, the probability that observation j occurred in maximal intersection i is

$$\frac{\mathbf{C}_{i,j} p_i}{\sum_{k=1}^{m} \mathbf{C}_{k,j} p_k}$$

This leads to

$$\mathbf{E}[k_i|\mathbf{p}] = \sum_{j=1}^{n} \frac{\mathbf{C}_{i,j} p_i}{\sum_{k=1}^{m} \mathbf{C}_{k,j} p_k}$$

Combining the E-step and the M-step, we get that

$$p_i^{(t+1)} = p_i \times \left(\sum_{j=1}^{n} \frac{\mathbf{C}_{i,j}}{n \sum_{k=1}^{m} \mathbf{C}_{k,j} p_k} \right)$$

Using this simple EM algorithm, one iteration can update all m parameters in $O(mn)$ time. To do this, it is necessary to first compute all n values of $\sum_{k=1}^{m} \mathbf{C}_{k,j} p_k$ for $j = 1, ..., n$. These values can be interpreted as the expected number of events to occur in observation rectangle j. This is done in $O(mn)$ time. Using these values, each of the m parameters can be updated in n time each.

Examining the combined EM step, one can see that if $p_i^{(t)} > 0$ then $p^{(t+1)} > 0$ (although $p_i^{(\infty)}$ can be 0). This will prove to be very inefficient in finding the bivariate NPMLE, as the number of maximal intersections can be of order $O(n^2)$, yet the minimal number of maximal intersections which must receive positive mass is at most n (Böhning *et al.* 1996). In addition, if $p_i^{(t)} = 0$, then $p^{(t+1)} = 0$. This implies that the classic EM can fail to find the NPMLE for poorly chosen initial values. An interesting note is that while the classic EM algorithm itself is quite slow, most of the more complicated algorithms can be greatly accelerated by adding a classic EM step.

6.4.2 VEM+ Algorithm

Böhning *et al.* 1996 showed the bivariate NPMLE can be viewed as a mixture model problem and proposed using a VEM algorithm to improve the rate of convergence. We will also use this algorithm, although we will modify the optimization technique used. The VEM algorithm selects indices u and l such that

$$\frac{\partial \ell(\mathbf{p^{(t)}})}{\partial p_u^{(t)}} = \max \left(\frac{\partial \ell(\mathbf{p^{(t)}})}{\partial p_i^{(t)}} \right)$$

$$\frac{\partial \ell(\mathbf{p^{(t)}})}{\partial p_l^{(t)}} = \min \left(\frac{\partial \ell(\mathbf{p^{(t)}})}{\partial p_i^{(t)}} : p_i^{(t)} > 0 \right)$$

The VEM step updates the parameters according to

$$p_i^{(t+1)} = \begin{cases} p_i^{(t)} + \delta & \text{if } i = u \\ p_i^{(t)} - \delta & \text{if } i = l \\ p_i^{(t)} & \text{otherwise} \end{cases}$$

At each step, δ is chosen to maximize $\ell(\mathbf{p}^{(t+1)})$ under the constraint that $\delta \leq p_l^{(t)}$. It is unnecessary to enforce the constraint that $\delta > -p_u^{(t)}$, as the concavity of $\ell(\mathbf{p})$ insures the proposed $\delta > 0$.

Any univariate optimization routine can be used to find δ. Newton's method is a natural choice. However, one downside with Newton's method is that while an ascent direction is insured, a proposed step is not guaranteed to increase the likelihood function without checking the likelihood function and half stepping. Given that computation of the likelihood function is of $O(mn)$ complexity, this comes at heavy computational cost. Alternatively, the classic EM algorithm could be used to chose δ, as an EM step is insured to increase the likelihood function. One problem with this is if $p_i^{(t)} = 0$, then the classic EM algorithm will always update to $p_i^{(t+1)} = 0$, even if $\hat{p}_i > 0$. In addition, the classic EM algorithm is observed to update inefficiently. We propose using a more efficient EM step which will be introduced in section 2.5.

The VEM+ algorithm is an algorithm which iterates back and forth between the VEM algorithm and the classic EM algorithm. Doing so was found to greatly increase the

speed of the algorithm.

6.4.3 ICM Algorithm

The ICM was originally proposed for the univariate NPMLE by Groeneboom (1991). To motivate the Iterative Convex Minorant (ICM) algorithm, we note that a standard Newton's method algorithm works very poorly in this problem for two reasons. First, the restrictions $\sum_{i=1}^{k} p_i = 1$ and $p_i \geq 0$ mean that the proposed step is very likely be outside the boundaries, although a reparameterization can be used to deal with the first restriction. Secondly, the larger number of parameters means inverting the Hessian matrix can be very computationally costly.

The ICM algorithm addresses both these issues by first approximating the likelihood function with a quadratic equation, given by the derivatives the likelihood function with the off diagonals of the Hessian ignored. Quadratic programming is used to maximize the approximated likelihood while still respecting the linear constraints on the estimated parameters. Much like Newton's method, this algorithm is guaranteed to find an ascent direction, although the proposed step may result in a lower likelihood function. Jongbloed (1998) suggest using half steps to insure monotonic convergence. Wellner and Zhan (1997) noted that combining the classic EM algorithm with the ICM algorithm greatly increased the speed of the algorithm.

6.4.4 SR Algorithm

When computing the bivariate NPMLE, the number of maximal intersections can be on the order of $O(n^2)$, while the number of support points necessary in the final solution cannot be greater than n. The Support Reduction (SR) algorithm (Groeneboom *et al.* 2008) is an active set algorithm which very rapidly determines the necessary

support points, not wasting computation time adjusting probability mass on support points that are not used in the NPMLE.

The SR algorithm has two steps: outer loops and inner loops. The outer loop step considers all support points, adding support points where necessary and removing unnecessary support points. Inner loops optimizes the function, only considering support points which already have positive mass. The CRAN package "MLEcens" implements an SR algorithm to find the bivariate NPMLE, using sequential quadratic programming (SQP) for the optimization steps in the SR algorithm.

The package MLEcens is currently the fastest readily available package for computing the bivariate NPMLE. It can also be applied to the univariate NPMLE. Theoretically, the SR algorithm should be of similar speed to the ICM algorithm for univariate data, as the SR algorithm is designed to take advantage of large numbers of unnecessary support points and this is much less of a problem in the univariate case. However, current implementations show the SR algorithm to be considerably faster. For example, a randomly generated data set of $n = 1000$ took 11.8 seconds for the EMICM algorithm and 0.50 seconds for the SR algorithm. This is likely due to efficient implementation rather than efficient computations.

6.5 Cocktail Algorithm

Introduced in Yu (2010), the cocktail algorithm utilizes three basic steps in each iteration for computing the univariate NPMLE. The first step is the classic EM algorithm, the same as presented earlier. The second step is the vetrex direction method (VDM; Fedorov 1972). This step selects a support point with the maximal derivative and places mass on this support by reducing mass from all other support points, proportional to the mass assigned to them. While the VDM algorithm by itself is

quite slow, it is guarantees convergence of the algorithm as long the initial starting point has finite log likelihood. Thus, if paired with algorithms which are monotonic in convergence, the algorithm is insured convergence for any non degenerate initial value. Finally, the last step of the algorithm is a nearest neighbor exchange (NNE) algorithm, in which mass is exchange between support points with positive mass which are located closest to each other. Mass is exchanged using the squeezing strategy, presented in section 6.6. Because the squeezing strategy may place 0 mass on support point which has positive mass at the NPMLE, the NNE step needs the VDM step to insure convergence to the NPMLE.

6.6 Squeezing Step

At the core of the algorithm we will present is an EM optimization step presented in Yu (2010). This is also the optimization procedure used in the cocktail algorithm. In its general form, the squeezing step is used to exchange probability mass between two support points in a mixture model. Without loss of generality, we can write the likelihood function in the form

$$\tilde{\ell}(\mathbf{p}) = \sum_{i=1}^{n} \log\left(p_1 f_{i1} + p_2 f_{i2} + r_i\right)$$

Note that $r_i = \sum_{j=3}^{k} p_j f_{ji}$, where $k =$ length of $\mathbf{p}$. Define

$$\beta_0 = p_1^{(t)} + p_2^{(t)}$$

$$g_i = \min\{f_{i1}, f_{i2}\}$$

$$\beta_1 = \min_{i:f_{i1}>f_{i2}} \frac{r_i + \beta_0 f_{i2}}{f_{i1} - f_{i2}},\ \beta_2 = \min_{i:f_{i2}>f_{i1}} \frac{r_i + \beta_0 f_{i1}}{f_{i2} - f_{i1}}$$

$$S_j = \left(p_j^{(t)} + \beta_j\right) \sum_{i=1}^{n} \frac{f_{ij} - g_i}{r_i + f_{i1} p_1^{(t)} + f_{i2} p_2^{(t)}},\ j = 1, 2$$

Using these values, we update

$$p_1^{(t+1)} = \max\{0, \min\{\beta_0, (\beta_0 + \beta_1 + \beta_2) S_1/(S_1 + S_2) - \beta_1\}\}$$

$$p_2^{(t+1)} = \beta_0 - p_1^{(t+1)}$$

In the case of the interval censored NPMLE, this simplifies somewhat. If the j^{th} maximal intersection is contained within the i^{th} observational rectangle, we say $m_j \in O_i$ and $m_j \notin O_i$ otherwise. Then we note that $f_{ij} = I\{m_j \in O_i\}$. Also, we will define $P(O_i) = \sum_{m_j \in O_I} p_j^{(t)}$ *i.e.* the likelihood of the i^{th} observation given the current estimate of $\mathbf{p}$.

We then note that

$$\beta_1 = \min_{i:m_1 \in O_i, m_2 \notin O_i} P(O_i) - p_1^{(t)}, \ \beta_2 = \min_{i:m_2 \in O_i, m_1 \notin O_i} P(O_i) - p_2^{(t)}$$

$$S_1 = \left(p_1^{(t)} + \beta_1\right) \sum_{i=1}^{n} \frac{I\{m_1 \in O_i, m_2 \notin O_i\}}{P(O_i)}, \ S_2 = \left(p_2^{(t)} + \beta_2\right) \sum_{i=1}^{n} \frac{I\{m_2 \in O_i, m_1 \notin O_i\}}{P(O_i)}$$

Once $P(O_i)^{(t)}$ has been calculated, we can update $P(O_i)^{(t+1)}$ in $O(n)$ time after an update of this type by setting

$$P(O_i)^{(t+1)} =$$

$$P(O_i)^{(t)} + (p_1^{(t+1)} - p_1^{(t)}) \times I\{m_1 \in O_i, m_2 \notin O_i\} + (p_2^{(t+1)} - p_2^{(t)}) \times I\{m_2 \in O_i, m_1 \notin O_i\}$$

When calculating $p_1^{(t+1)}$, the only part which takes $O(n)$ time is finding the positive values of $I\{m_1 \in O_i, m_2 \notin O_i\}$ and $I\{m_2 \in O_i, m_1 \notin O_i\}$. Thus, while initializing $P(O_i)$ is of $O(mn)$ complexity, applying the squeezing strategy to a chain of m pairs of parameters is only of $O(mn)$ complexity as well.

Key aspects of this squeezing strategy are that it is very efficient at exchanging mass between two support points and that being an EM algorithm, it is unnecessary to check the likelihood function to insure monotonic convergence. Also, unlike the classic EM algorithm, the squeezing strategy can add mass to a support point which has mass 0 and can set mass to a support point to 0 in one step. However, this strategy can

only be used to exchange mass between two support points and in many problems, there is not a clear matching scheme. As stated earlier, this squeezing strategy can be used can be used for the optimization step of the VEM algorithm and we found doing so sped up the algorithm compared to using Newton's method. However, since initializing $P(O_i)$ for bivariate data takes $O(mn)$ time, but each squeeze and update of $P(O_i)$ takes $O(n)$ time, using long chains of the squeezing strategy, such as the NNE in the cocktail algorithm, is more efficient.

6.7 Folding Strategy

In the univariate NPMLE, the support points have a natural ordering, leading to a natural implementation found in the cocktail algorithm above. For the bivariate NPMLE, there is not a natural ordering. Several ordering schemes were considered, including zig zagging and pairing support points which shared large amounts of observation rectangles. None of these schemes proved fruitful and the nearest neighbor exchanges strategy was abandoned.

Instead, a strategy with similar motivation to the SR algorithm was used (the author was not aware of the SR algorithm at the time). Because the number of maximal intersections can be in order of $O(n^2)$, but the number necessary support points cannot be above n, an algorithm which can rapidly reduce the number of support points with positive mass can greatly accelerate the algorithm. The classic EM algorithm can update all support points simultaneously in a direction that is insured to increase the likelihood function. However, in a finite number of classic EM steps, none of those support points will have reached 0 probability mass. The squeezing EM step can set the probability mass of a support to 0 in one step, although pairing of support points must be done in an intelligent way. Combining these aspects of the two algorithms lays the groundwork for the Folding Strategy.

In the Folding Strategy, we start with probability mass distributed uniformly across all maximal intersections. The folding step begins with one classic EM algorithm step. After this, we use the squeezing step, matching support points with mass above the median with support points below the median. Specifically, if there are k support points with mass greater than the median, then we order the support points by mass, matching the 1^{st} point with the $(k+1)^{th}$, the 2^{nd} with the $(k+2)^{th}$, etc. Occasionally, the squeezing steps will set mass to 0 prematurely, so after the last squeezing step is applied, the vector of derivatives is checked and any support point with 0 mass and positive probability is matched with a support point with positive mass and a negative derivative and the squeezing step is applied.

The Folding Strategy very quickly eliminates support points which do not require positive mass at the NPMLE, typically cutting the number of support points with positive mass in half with each fold until the number of support points with positive mass is very close to the number at the NPMLE. We found it optimal to allow the Folding Strategy to run until no new support points were set to 0 probability mass.

Once this happened, we terminated the Folding Strategy and switched to a VEM+ style of algorithm, but limited our scope to only the support points that were assigned positive mass in by the Folding Strategy. As we noted in the Squeezing Strategy section, it is more efficient to use long chains of the squeezing strategy, rather than just one pair, as proposed by the standard VEM algorithm. Thus, along with pairing the support points with the largest positive and negative derivatives, we also paired all support points with positive derivatives with support points with negative derivatives. We did not update the derivatives during this chain, as an update of derivatives would cost $O(mn)$ time. This implies the pairing was suboptimal, but we found this strategy to still be very helpful. We will refer to this as the Chained VEM step. A Classic EM step was added to the end of the Chained VEM step. Finally, when

the Chained VEM algorithm converged over this subset of support points, we again used the Chained VEM algorithm, except now considering all support points, not just those presented by the Folding Strategy. We found that the majority of the time, no more iterations were required at this point, but not always, *e.g.* occasionally when the Folding Strategy terminated, it had set a support point to have 0 mass which had positive mass at the NPMLE. Considering all support points in the last Chained VEM section of the algorithm allows us to double check that we were not missing any support points.

Proving convergence of this algorithm is trivial. The VEM+ algorithm has been shown to converge for any non degenerate initial value. Because the uniform distribution of probability mass across all maximal intersections is not degenerate and we only use EM steps, we know that the estimate proposed to the final VEM+ algorithm (which considers all support points) must not be degenerate. If we consider the estimate proposed to the final VEM+ as an initial value, we see the algorithm is insured to converge.

In theme, this algorithm is very similar to the SR algorithm: they both aim to rapidly reduce the number of support points considered. Some key differences are that the Folding Strategy uses an EM algorithm, while the SR algorithm in its current implementation uses sequential quadratic programming (SQP). In addition, the Folding Strategy starts with positive mass at all points, and so is a "top down" approach. The current implementation of the SR algorithm starts with the minimal number of support points and adds more when necessary, and so is a "bottom up" approach, although Groeneboom *et al.* (2008) presents the generalized SR algorithm for both methods. Also, being a purely EM algorithm, the Folding Strategy has the advantage that checking the likelihood function to insure monotonicity is not necessary, unlike the SR algorithm which uses a Newton based algorithm.

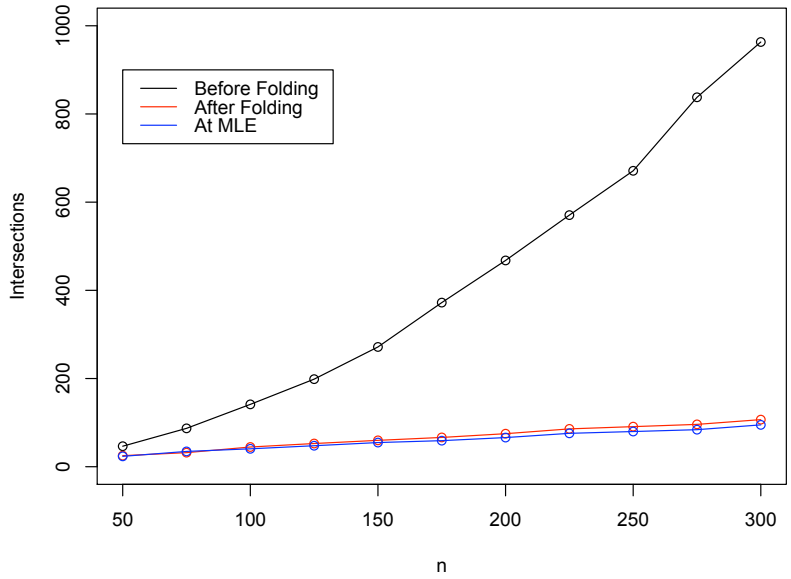

Figure 6.3: Average number of support points with positive mass before folding, after folding and at NPMLE

We applied the Folding Strategy on simulated data to show how effective it is at removing unnecessary support points. Data was simulated across a variety of different sample sizes. Description of the simulated data can be found in greater detail in section 2.7. In figure 8.3, we see the average number of maximal intersections found by the HeightMap algorithm in black. In red, we see the average number of maximal intersections with positive mass after the Folding Strategy terminates. In blue, we see the average number of maximal intersections with positive mass at the NPMLE. We will note that the number of maximal intersections appears to grow quadratically with n. In contrast, the number of maximal intersections with positive mass at the NPMLE is closer to $n/2$. The folding strategy appears to remove almost all of the

unnecessary maximal intersections.

6.8 Sparse Data Implementation

For all algorithms mentioned, calculation of the NPMLE uses calculations based on the clique matrix produced by the HeightMap algorithm. The clique matrix can be thought of a special case of general mixture matrices in mixture models. However, the clique matrix has the characteristic that all entries are either 0 or 1. Unless the censoring mechanism is extreme, the vast majority of the entries will be 0's. We can take advantage of this structure of the data by representing the clique matrix with a sparse data matrix. Recalling that the $i^{\text{th}}, j^{\text{th}}$ entry of the clique matrix is an indicator that the i^{th} maximal intersection is within the j^{th} observational rectangle, we access the values of the matrix via the rows, not the columns, in our algorithm. We can summarize the rows of the clique matrix by a matrix in which the first entry of each row is the count of 1's and then each entry after this an index for the 1's in the row. This can accelerate all of the given algorithms by quickly reporting which entries are 1's, rather that checking each entry every time.

6.9 Algorithm Speeds

We tested the Folding algorithm, with and without the sparse data matrix implementation, the SR algorithm, the VEM+ and EM algorithms. We found that the sparse data implementation always increased the speed of the algorithm, although more in some situations than others. We only examined sparse data implementations of the VEM+ and EM algorithm, as they were still not capable of competing with a standard implementation of the more advanced algorithms.

While we found that the SR algorithm and Folding algorithm dominated the VEM+

and EM algorithm, how they compared to each other was not as consistent. In particular, the Folding strategy did much better when the number of maximum intersections reported by the HeightMap algorithm was very large, while the SR algorithm seemed to do better at quickly optimizing over a smaller number of maximal intersections. Because of this, the Folding algorithm does significantly better in the worst case scenario, specifically when the censoring distribution is continuous and thus the number of maximal intersections is of order $O(n^2)$. In addition, typically the more informative censoring (*i.e.* shorter intervals) lead to more maximal intersections, which lead to further advantage of the Folding algorithm. However, in practice, this worst case scenario is very rare, as the censored times are typically rounded to the nearest day, year, etc. In this case, if k_1 = number of unique censoring times for T_1 and k_2 = number of unique censoring times for T_2, then the number of maximal intersections is of order $O(k_1 k_2)$. In these cases, we found that the SR algorithm slightly outperformed the Folding algorithm.

To illustrate our findings, we simulated true event times in a simple manner. For each observation, we first simulate $X_1, X_2, X_3 \sim_{\text{iid}} \text{Uniform}(0, 1)$. We set $T_1 = X_1 + X_2$ and $T_2 = X_1 + X_3$, creating a moderate correlation between T_1 and T_2 ($\rho = 0.5$). We then used three different censoring mechanisms to demonstrate how the algorithms perform differently under different censoring scenarios. In the first situation, we generate $C_1 \sim \text{Uniform}(0, 1)$ and then generate $C_2 \sim \text{Uniform}(C_1, 2)$. This creates a "window" (C_1, C_2). If $T_j \in (C_1, C_2)$, then (C_1, C_2) is reported. If $T_j < C_1$, then $(0, C_1)$ is reported. If $T_j > C_2$, then $(C_2, 2)$ is reported. This represents continuous heavy censoring. In the second scenario, we censor the data in the same manner, except that we round the censoring times to the nearest 0.1. This leads to a discrete censoring distribution with $k_1 = k_2 = 20$, which we found to be fairly typical of datasets which appear in the literature. In the third situation, we set $C_1 = 0$ and $C_2 = \text{Uniform}(0, 0.5)$. If $T_j \in (C_1, C_2)$, we would return (C_1, C_2) as the censoring

times. If not, we would set $C_1 = C_2$ and $C_2 = C_2 + \text{Uniform}(0, 0.5)$ and repeat until $T_1 \in (C_1, C_2)$. If $C_2 > 2$, we would truncate $C_2 = 2$. This represented more informative censoring. We test across a variety of sample sizes (n = 100, 200, 400, 800 and 1200 for continuous censoring, n = 100, 200, 400, 800, 1600 and 3200 for discrete censoring).

The tolerance was set to 10^{-10}, as this insures that the log likelihood is within 10^{-10} of the max and in practice is very conservative. For each sample size, 100 simulations were run. The Folding, VEM+ and EM algorithm were all implemented by the author. The SR algorithm was the algorithm found in "MLEcens" CRAN package. This lead to bias in the comparison of algorithms. Although the author does not know the details of the implementation of the SR algorithm, it was assumed that it did not use a sparse data implementation. However, there seems no reason why the SR algorithm could not be built around a sparse data matrix. Because of this, two Folding Algorithms were written: one based on a sparse data matrix, which showed the potential of the Folding Algorithm and one based on the standard clique matrix, which allowed for a fair comparison with the SR algorithm. Because the VEM+ and classic EM algorithm were so slow, there was no need to implement the standard clique matrix for a fair comparison. The average times are presented on tables 8.1, 8.2 and 8.3. Tables 8.1, 8.2 and 8.3 use the first, second and third censoring scheme respectively as described in the earlier paragraph. All algorithms with a "*" use a sparse data matrix implementation, while those without use the standard clique matrix.

	n				
	100	200	400	800	1200
Folding*	0.01	0.07	0.41	2.49	7.95
Folding	0.01	0.09	0.58	6.05	24.1
SR	0.01	0.09	0.78	7.70	32.9
VEM+*	0.08	1.42	25.8	-	-

Table 6.1: Average time in seconds for heavy continuous censoring (censoring scheme 1)

	n					
	100	200	400	800	1600	3200
Folding*	0.02	0.04	0.14	0.42	1.79	7.95
Folding	0.02	0.06	0.23	0.80	3.68	24.1
SR	0.01	0.03	0.11	0.33	1.02	3.02
VEM+*	0.08	0.43	1.95	6.89	31.5	-

Table 6.2: Average time in seconds for heavy discrete censoring (censoring scheme 2)

	n				
	100	200	400	800	1200
Folding*	0.02	0.08	0.49	2.05	5.56
Folding	0.04	0.21	1.71	22.3	69.2
SR	0.02	0.16	1.63	18.9	83.5
VEM+*	-	-	-	-	-

Table 6.3: Average time in seconds for light continuous censoring (censoring scheme 3)

We note that the Folding Algorithm with the sparse data matrix implementation is the fastest when the censoring distribution is continuous, especially in the case of lightly censored data. While the standard implementation of the Folding algorithm is still faster than the SR algorithm in many of the continuous censored scenarios, especsially with larger datasets, the sparse data matrix implementation appears to be responsible for most of the gains in speed over the SR algorithm. In the most extreme case, this lead to an increase in speed by a factor of 15.

However, in the case of a discrete censoring mechanism, the SR algorithm does better than even the sparse data implementation of the Folding Algorithm, although the increase was never more than threefold in the scenarios we tested. Because bivariate interval censored data is typically seen in this format, it does not seem wise to recommend the Folding Algorithm over SR algorithm at this time.

This is not to say the Folding Algorithm is without merit. As more efficient algorithms allow for handling of a wider variety of datasets, more uses of such methods may propagate and the need to handle continuous censoring distributions may become necessary. In addition, it is worth noting that for higher dimensional data, such as trivariate, the basic structure of the optimization would be the same and the Folding Algorithm could be applied to such problems without any modification, as it only needs a clique matrix as input which would be in the same form regardless of dimension (although the HeightMap algorithm for finding the clique matrix would have to be modified). Because the number of maximal intersections grows exponentially with the number of dimensions, and the Folding Algorithm has a competitive advantage over the SR algorithm for large number of maximal intersections, we speculate that the Folding Algorithm may considerably outperform the SR algorithm higher dimensional data, even when the censoring distribution is discrete. However, because we do not have an algorithm for finding the maximal intersections for higher dimensions,

we were not able to investigate the performances of the algorithms in these scenarios.

6.10 Illustrative Example: Hemophiliac Data

A classic dataset found in the literature presented by Kim *et al.* (1993), this study involved 257 hemophiliac patients treated with HIV contaminated blood at two French hospitals beginning in 1978. Of these 257 patients, 188 contracted HIV by August 1988. Of these 188, 41 progressed to develop AIDS. The patients were placed into two groups: lightly treated and heavily treated, depending on how much blood had been received. There were 104 subjects classified as heavily treated, while 153 were classified as light treated. We are interested in comparing survival curves between the light treated and heavy treated groups.

Interval censoring occurs because the patients are not continually monitored, but rather only observed during doctor visits. Before HIV was detected, the censoring was very heavy, as patients had their blood checked only during standard check ups. Once they had been diagnosed with HIV, they were still subject to interval censoring, although the check ups were more frequent so the censoring of the development of AIDS is lighter. In particular, the average length of intervals for which HIV infection were known to occur was 2.1 years, while the average length of intervals for which AIDS was known to have developed was 0.56 years. We note that the time that we are interested in is doubly censored, as we are interested in time from seroconversion to development of AIDS, both of which are interval censored.

In order to deal with the double censoring, we will first model the joint cdf of the two event times via the bivariate NPMLE. We will call X (seroconversion) and Y (development of AIDS). The bivariate NPMLE gives a list of rectangles with probability mass associated with these rectangles. How probability mass is assigned within

these rectangles does not affect the likelihood function. Thus, in order to get lower bound of the NPMLE, for each rectangle, we take min(Y) - max(X; $X \leq Y$) and for an upper bound, we take max(Y) $-$ min(X). This will provide an "indifference area", *i.e.* an area for which the survival function estimate has maximal likelihood. Estimated survival curves can be found on figure 8.4.

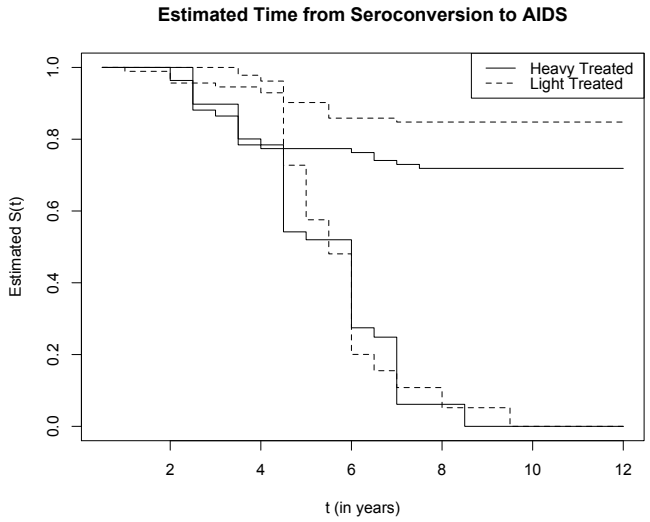

Figure 6.4: Estimated NPMLE Survival Curves for Time from Seroconversion to Development of AIDS

Examining the plots, we first notice that we have very little information about the survival curves for both groups after 4.5 years, as the indifference area becomes quite large. It is very important to remember that these are not confidence regions but rather areas in which the likelihood is equivalent. Comparing the times before 4.5 years, we see the heavy treated group appears to be progressing from HIV to AIDS at a higher rate.

We ran both the Support Reduction algorithm and the Folding Strategy to estimate the NPMLE. For both the heavily treated ($n = 104$) and the lightly treated ($n = 153$). For both groups, both algorithms converged in an insignificant amount of time; for the heavily treated group, the SR algorithm took 0.002 seconds and the Folding

Strategy took 0.001 seconds. For the lightly treated group, both algorithms took 0.002 seconds. Both algorithms lead to nearly identical solutions: the maximum difference in estimated survival probability in all situations considered in this data set was 6.5×10^{-13}.

Chapter 7

Future Work

The work done in this dissertation leads to many future research topics. One area that has not received sufficient coverage for these topics is theoretical results about the distribution of the estimators. At this time, the only known results for both the log-concave NPMLE for interval censored data and the inverse convex estimator for exact times are based on simulations. While we have presented several methods for inference for the log-concave estimator which sidestep the question of distribution theory, without theoretic results it is very difficult to compare the performance with competing estimators without computationally intensive simulations which provide very limited insight into the global behavior of an estimator. In addition, it is has not been shown that the algorithm for the interval censored log-concave NPMLE finds global maximum, nor has it been shown that the local maximum obtained by the algorithm for the inverse-convex estimator is unique. While multiple runs of both algorithms from random starting points lead to the same estimates, suggesting that there is a unique solution in each case, it would be more satisfying to prove this analytically.

The next clear topic to address is implementing an algorithm for the inverse convex estimator for interval censored data. Our motivation for the inverse convex estimator is application to survival analysis, so naturally we would like an algorithm which can handle censored data. We believe that the outline of the algorithm for finding the log-concave NPMLE for interval censored data will easily adapt to the inverse convex estimator, as the parameterization of the two problems is very similar. This would also lead to similar applications to those presented in chapter 4: a goodness of fit test, profile and bootstrap confidence intervals and a Cox PH model.

Finally, a novel estimator we are very interested in is an augmentation of the shape constrained estimator for both the log-concave and inverse convex estimator. In the case of exact observations, both shape constraints place no probability mass beyond $x_{\min}$ and $x_{\max}$. In application to interval censored data, similar problems occur, although when and where this occurs is not as well defined. While this has little effect on central quantile estimates for large samples, it can cause heavy bias in hazard estimation. In particular, $\hat{h}(x_{\max}) = \infty$, as $\hat{S}(x_{\max}) = 0$ and $\hat{f}(x_{\max}) > 0$. In general, this causes upward bias as $x \to x_{\max}$. We speculate that this may be responsible for the upward bias for Cox PH regression parameters for both the log-concave and unconstrained baseline case, as the unconstrained NPMLE has exhibits similar problems.

In addition, this can cause a variety of problems for mixture models with shape constrained components. For the mixture model, we observed that using log-concave components often results in components which assign 0 estimated density to observed values, leading to an estimated 0 conditional probability of an observation being generated from that component. This can be problematic for a variety of reasons. First, it can create numerical instability in the EM algorithm as the estimated conditional probabilities and component density simultaneously approach 0, as the contribution

to the complete data likelihood includes the term $p_{ij}\log(f_j(x_i))$ where p_{ij} is the conditional probability of observation i coming from component j and f_j is the density of component j. In addition, it can be very unsatisfying for an estimator to assign conditional probability 0 to an observation which is narrowly outside of a data driven boundary, especially when the conditional probability may be quite high were to it be on the other side of the boundary. Finally, it can lead to degenerate classification estimates if we use the model to predict assignment of a new observation with extreme values of x (*i.e.* the density for *every* component could be 0, leading to undefined conditional probability).

A solution we propose is to only allow the shape constrained estimator to assign $\pi < 1$ probability mass and then place the remaining $1-\pi$ probability mass to the tails, forcing $\hat{f}_X(x) > 0$, $x \in (0,\infty)$ (or $x \in \mathbb{R}$ depending on the problem). While this seems ad hoc, there is theoretic justification for this. In the case of n observations with no censoring, the expected value of x_{min} = the $1/(n+1)$ quantile and the expected value of x_{max} = the $n/(n+1)$. With this in mind, it seems natural to set $\pi = \frac{n-1}{n+1}$. In spirit, this is very similar to replacing the MLE of σ with the unbiased estimator s.

Applying this new augmented estimator to the hazard estimates on $[x_{min}, x_{max}]$ for exact observations is very straightforward. We will force $1/(n+1)$ probability mass below x_{min} and $1/(n+1)$ above x_{max}. Thus we will define $\hat{f}_{AUG}(x) = \hat{f}_{SC}(x) \times \left(\frac{n-1}{n+1}\right)$ and $\hat{F}_{AUG}(x) = \frac{1}{n+1} + \hat{F}_{SC}(x) \times \frac{n-1}{n+1}$ for $x \in [x_{min}, x_{max}]$, where f_{SC} is the shape constrained density and f_{AUG} is our new augmented estimator. For now, we will not worry about what to assign $f_{AUG}(x)$ below x_{min} or x_{max}, as it is not reasonable to be able to make efficient hazard estimates on (x_{max}, ∞) without unreliable fully parametric assumptions.

To illustrate the effect of our augmented estimator, we sampled 100 values from an exponential distribution with rate = 1. We fit both the inverse convex estimator

and the augmented inverse convex estimator and plotted the estimated hazards in Figure 6.1. We note that on [0,2], the estimators appear almost identical as we would expect for areas with high survival probabilities. However, on [2,3], we note that the inverse convex estimator appears to be significantly drifting upward and on [3, 4+) the estimate is clearly uninformative. The augmented estimator does not show any definitive biases, although more theoretic work is required to examine its behavior.

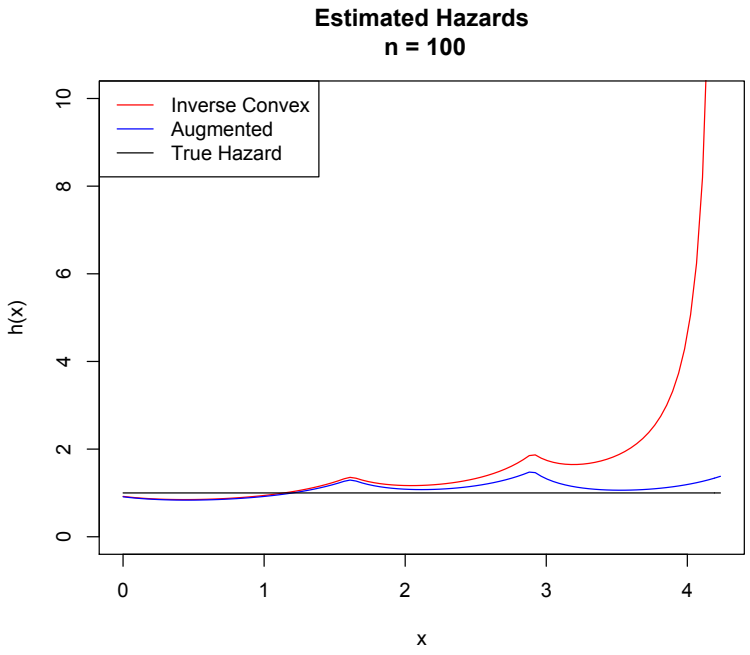

Figure 7.1: Estimated hazards based on sample of 100 exponential random variables

In this simple example, implementation of the augmented estimator was quite easy and appears favorable to the standard shape constrained estimator for hazard estimation. However, there are many applications where we believe it could be useful and implementation is not so simple. For example, it is not readily apparent what value of π to select in the case of censoring. Furthermore, we are interested in implementing this into a Cox PH model to see if this reduces the bias. Finally, while the matter of how the probability mass is assigned beyond x_{max} is of little concern for

hazard estimation, it will be important in the case of mixture modeling for reasons mentioned above. While we believe attempting unbiased density estimation beyond x_{min} and x_{max} is foolish (although consistent estimation is implied as long as $\pi \to 1$ as $n \to \infty$), we would like to create a rule which automatically assigns the probability density such that a.) a theoretically justified choice of $1 - \pi$ is automatically assigned without input from the user b.) $\hat{f}_{AUG}(x) = \pi \hat{f}_{SC}(x)$ on $[x_{min}, x_{max}]$ c.) the estimated density is positive on either $\mathbb{R}$ or $(0, \infty)$ d.) the shape constraints are respected on $\mathbb{R}$. We expect that some favorable rules may work with some shape constraints but not others.

Chapter 8

Conclusion

Shape constrained density estimation offers a compromise between parametric estimation methods, which have limited flexibility, and traditional non-parametric estimators which can inefficient. By enforcing the assumptions that are known to be robust for the data type, we can regain some of the efficiency of parametric models while limiting our bias.

Interval censored data presents an excellent application where the shape constrained niche is quite useful. The limited informative nature of each data point makes model checking very difficult and so parametric fits run the risk of introducing heavy bias without providing adequate diagnostic tools for the investigator. On the other hand, the classic NPMLE is notoriously inefficient, displaying an unfavorable $n^{-1/3}$ convergence rate.

The main focus of this work has been on shape constrained estimation with application to interval censored data. We presented an efficient algorithm for finding the log-concave NPMLE for interval censored data and used this new algorithm to investigate the properties of the log-concave NPMLE. We found that it greatly reduced the

variance in comparison to the classic NPMLE.

We developed new methods of inference for our estimator, including a likelihood ratio goodness of fit test based on a two component mixture model, two methods for confidence intervals for quantile and survival probabilities and a Cox PH model which uses a baseline log-concave distribution. We found that the use of the log-concave constraint greatly reduced confidence intervals for quantile and survival probabilities, although it only marginally improved estimation for regression coefficients.

While the application of the log-concave constraint showed the potential of shape constrained estimation when applied to interval censored data, we caution that the log-linear tail boundary may not allow for heavy enough tails for some survival analysis problems. To meet this need we presented a new shape constraint which we call inverse convex. While still insuring the distribution is unimodal, this new constraint allows for much heavier tails. We implemented an algorithm for finding the inverse convex estimator for uncensored data and demonstrated that this new shape constraint can fit heavy tailed data much better than the log-concave estimator, both for simulated and real data.

Bibliography

Abrevaya, J., and Huang, J. (2005), On the Bootstrap of the Maximum Score Estimator, *Econometrica*, Vol 73, 1175-1204

Bagnoli, M., Bergstorm, T. (2005), Log-concave Probability and its Applications, *Economic Theory* Vol 26 No. 2, 445-469

Balabdaoui, F., Rufibach, K. and Wellner, J. (2009), Limit Distribution Theory for Maximum Likelihood Estimation of a Log-Concave Density, *Annals of Statistics*, Vol 37, 1299-1331

Banerjee, M. and Wellner, J. (2001), Likelihood Ratio Tests for Monotone Functions, *Annals of Statistics*, Vol 29, 1699-1731

Banerjee, M., and Wellner, J. (2005), Confidence Intervals for Current Status Data, *The Scandinavian Journal of Statistics*, Vol 32 405-424

Bebchuk, J.D. and Betensky, R. A. (2000), Multiple Imputation for Simple Estimation of the Hazard Function Based on Interval Censored Data, *Statistics in Medicine*, Vol 19, 405-419

Betensky, R. A., and Finkelstein, D. M. (1999), A Nonparametric Maximum Likelihood Estimator for Bivariate Censored Data, *Statistics in Medicine*, Vol 18, 3089-3100

Betensky, R., Lindsey, J., Ryan, L., Wand, M. (1999), Local EM Estimation of the Hazard Function Interval-Censored Data, *Biometrics* Vol 55 238-245

Bogaerts, K., Leroy, R., Lesaffre, E. and Declerck, D. (2002), Modeling Tooth Emergence Data Based on Multivariate Interval-Censored Data, *Statistics in Medicine*, Vol 21, 3775-3787

Bogaerts, K., and Lesaffre, E. (2004), A New Fast Algorithm to Find the Regions of Possible Support for Bivariate Interval-Censored Data, *Journal of Computational and Graphical Statistics*, Vol 13, 330-340

Böhning D., Schlattmann, P. and Dietz E. (1996), Interval Censored Data: A Note on the Nonparametric Maximum Likelihood Estimator of the Distribution function, *Biometrika*, Vol 83, 462-466

Braun, J., Duchense, T., Stafford, J. (2005), Local Likelihood Density Estimation for Interval Censored Data, *The Canadian Journal of Statistics*, Vol 33 No.1, 39-60

Carroll, R., Delaigle, A., and Hall, P. (2011), Testing and Estimating Shape Constrained Nonparametric Density and Regression in the Presence of Measurement Error, *Journal of the American Statistical Association*, Vol 106, 191-202

Chang, G., Walther, G. (2007), Clustering with Mixtures of Log-Concave Distributions, *Computational Statistics and Data Analysis*, Vol 51, No. 12, 6242-6251

Chen, D., Sun, J. and Peace, Karl, *Interval-Censored Time-to-Event Data: Methods and Applications*, Florida: CRC Press

Clayton, D. G. (1978), A Model for Association in Bivariate Life Tables and its Application in Epidemiological Studies of Familial Tendency in Chronic Disease Incidence, *Biometrika*, Vol 65, 141-151

Cox, D. R. (1972), Regression Models and Life-Tables, *Journal of the Royal Statistical Society, Series B*, Vol 34, 187-220

Day, N. E. (1969), Estimating the Components of a Mixture of Normal Distributions, *Biometrika*, Vol 56, 463-474

De Gruttola, V. and Lagakos, S. W. (1989), Analysis of Doubly-Censored Survival Data, with Application to AIDS, *Biometrics*, Vol 45, 1-12

Delgado, M., Rodriguez-Poo, J. and Wolf, M. (2001) Subsampling Inference in Cube Root Asymptotics with an Application to Manski's Maximum Score Estimator, *Economics Letters*, Vol 73, 241-250

Dixon, P., Weiner, J., Mitchell-Olds, T., Woodly, R. (1987), Bootstrapping the Gini Coefficient of Inequality, *Ecology*, Vol 68, 1548-1551

Dümbgen, L., Hüsler, A., Rufibach, K. (2011), Maximum Likelihood Estimation of a Log-Concave Density Based on Censored Data, preprint

Dümbgen, L., Rufibach, K. (2009), Maximum Likelihood Estimation of a Log-Concave Density and its Distribution Function: Basic Properties and Uniform Consistency, *Bernoulli*, Vol 15 No. 1 40-68

Efron, B. (1979), Bootstrap Methods: Another Look at the Jackknife, *Annals of Statistics*, Vol 7, 1-26

Efron, B. (1981), Non Parametric Standard Errors and Confidence Intervals, *Canadian Journal of Statistics*, Vol 9, 139-158

Fujisawa, H., and Eguchi, S., (2006), Robust Estimation in the Normal Mixture Model, *Journal of Statistical Planning and Inference*, Vol 136, 3989-4011

Frydman, H. (1995), Nonparametric Estimation of a Markov "Illness-Death" Process from Interval-Censored Observations, with Application to Diabetes Survival Data, *Biometrika*, Vol 82, 773-789

Fedorov, V. (1972), *Theory of Optimal Experiments*, Academic Press, 1972

Finkelstein, D. M. and Wolfe, R. A. (1986), Isotonic Regression for Interval Censored Survival Data Using an E-M Algorithm, *Communications in Statistics: Theory and Methods*, Vol 15, 2493-2505

Gentleman, R., Geyer, C. (1994), Maximum Likelihood for Interval Censored Data: Consistency and Computation, *Biometrika*, Vol 81, No. 3, 618-623

Gentleman, R., and Vandal, A. (2001), Computational Algorithms for Censored-Data Problems Using Intersection Graphs, *Journal of Computational and Graphical Statistics*, Vol 10 No. 3, 403-421

Gentleman, R., and Vandal, A. (2002), Nonparametric Estimation of the Bivariate CDF for Arbitrarily Censored Data, *The Canadian Journal of Statistics*, Vol 30 No. 4, 556-571

Goggins, W. B. and Finkelstein, D. M. (2000), A Proportional Hazards Model for Multivariate Interval-Censored Failure Time Data, *Biometrics*, Vol 56, 940-943

Grenander, U. (1956), On the Theory of Mortality Measurement. Part II, *Scandina-*

vian Actuarial Journal Vol 39, 125-131

Groeneboom, P. (1987), Asymptotics for Interval Censored Observations, *Technical Report* 87-18. Department of Mathematics, University of Amsterdam

Groeneboom, P. (1991), Nonparametric Maximum Likelihood Estimation for Interval Censored Data, *Technical Report*, Statistics Department, Stanford University

Groeneboom, P. and Wellner, J. A. (1992), *Information Bounds and Non-parametric Maximum Likelihood Estimation*, DMV Seminar, Band 19, Birkhauser, New York

Groeneboom, P. and Wellner, J. A. (2001), Computing Chernoff's Distribution, *Journal of Computational and Graphical Statistics*, Vol 10, 388-400

Groeneboom, P., Jongbloed, G. and Wellner, J. (2008), The Support Reduction Algorithm for Computing Non-Parametric Function Estimates in Mixture Models, *Scandinavian Journal of Statistics*, Vol 35, 385-399

Hathaway, R.J. (1986), Another Interpretation of the EM Algorithm for Mixture Distributions, *Statistics and Probability Letters*, Vol 4, 53-56

Hoel, D. G. and Walburg, H. E. (1972), Statistical Analysis of Survival Experiments, *Journal of National Cancer Institute*, Vol 49, 361 - 372

Huan, J., Wellner, J. (1997), Interval Censored Survival Data: A Review of Recent Progress, *Procedings of the First Seattle Symposium in Biostatistics: Survival Analysis*, 123-169

Huang, J. (1996), Efficient Estimation for the Proportional Hazards Model with Interval Censoring, *The Annals of Statistics*, Vol 24, 540-568

Jewell, N., Laan, M., Henneman, T. (2003), Nonparametric Estimation from Current Status Data with Competing Risks, *Biometrika*, Vol 90, 183-197

Jongbloed, G. (1998), The Iterative Convex Minorant Algorithm for Nonparametric Estimation, *Journal of Computational and Graphical Statistics*, Vol 7, No. 3, 310-321

Kiefer, J., and Wolfowitz, J. (1956). Consistency of the Maximum Likelihood Estimates in the Presence of Infinitely Many Incidental Parameters, *Annals of Mathematical Statistics*, Vol 27, 887-906

Kim, M., De Gruttola, V., and Lagakos, S. W. (1993), Analyzing Double Censored Data with Covariates, with Application to AIDS, *Biometrika*, Vol 49, 12-22

Krailo, M. and Pike, M. (1983), Estimation of the Distribution of Age at Natural Menopause from Prevalence Data, *American Journal of Epidemiology*, Vol 117, 356-361

Kopperberg, C., and Stone, C. (1992), Logspline Density Estimation for Censored Data, *Journal of Computational and Graphical Statistics*, Vol 1, 301-328

Kuhn, W., Tucker, W. (1951), Nonlinear Programming, *Proceedings of 2nd Berkeley Symposium*, 481-492

Laan, M. J. (1996), *Efficient and Inefficient Estimation in Semiparametric Models*, CWI-tract #114, Centre for Mathematics and Computer Science, Amsterdam

Leger, C., and MacGibbon, B. (2006), On the Bootstrap in Cube Root Asymptotics, *The Canadian Journal of Statistics*, Vol 34, 23-44

Lindsey, J.C. and Ryan, L. M. (1998), Tutorial in Biostatistics: Methods for Interval-Censored Data, *Statistics in Medicine*, Vol 17, 219- 238

Liu, C. and Rubin, D. (1994), The ECME Algorithm: A Simple Extension of EM and ECM with Faster Monotone Convergence, *Biometrika*, Vol 81, 633-648

Maathuis, M. (2005), Reduction Algorithm for the NPMLE for the Distribution Function of Bivariate Interval-Censored Data, *Journal of Computational and Graphical Statistics*, Vol 14, 352-362

Maathuis, M. (2006), Nonparametric Estimation for Current Status Data with Competing Risks, Ph.D. thesis, University of Washington

Maathius, M. and Hudgens, M. (2011), Nonparametric Inference for Competing Risks Current Status Data with Continuous, Discrete and Grouped Observation Times, *Biometrika*, Vol 98 325-340

MacMahon B. and Worcester J., (1966), Age at Menopause. United States – 1960-1962, *Nation Center for Health Statistics. Vital and Health Statistics*, Vol 11, No. 19

Meister, M. (2009), On Testing for Local Monotonicity in Deconvolution Problems, *Statistics and Probability Letters*, Vol 79, 312-319

Meng, X. and Rubin, D. (1993), Maximum Likelihood Estimation via the ECM Al-

gorithm: A General Framework, *Biometrika*, Vol 9-, 267-278

Moscarini, G., (2005), Job Matching and the Wage Distribution, *Econometrica*, Vol 73, No. 2 481-516

Pan, W., (1999), Extending the Iterative Convex Minorant Algorithm to the Cox Model for Interval-Censored Data, *Journal of Computational and Graphical Statistics*, Vol 8, 109-120

Pan, W., (2000), Smooth Estimation of the Survival Function for Interval Censored Data, *Statistics in Medicine*, Vol 19, No. 19, 2611-2624

Pan, W., (2000a), A Multiple Imputation Approach to Cox Regression with Interval-Censored Data, *Biometrics*, Vol 56, 199-203

Pan, W. and Chappell, R. (2002), Estimation in the Cox Proportional Hazards Model with Left Truncated and Interval Censored Data, *Biometrics*, Vol 58, 64-70

Politis, D. N., Romano, J. P. and Wolf, M. (1999), *Subsampling*, Springer-Verlag, New York

Rosenberg, P. S. (1995), Hazard Function Estimation Using B-splines, *Biometrics*, Vol 51, 874-887

Protassov, R., van Dyk, D., Connors, A., Kashyap, V. and Siemiginowska, A. (2002), Statistics, Handle with Care: Detecting Multiple Model Components with the Likelihood Ratio Test, *The Astrophysical Journal*, Vol 571, 545-559

Rubin, D. B. (1978), Multiple Imputations in Sample Surveys - A Phenomenological Bayesian Approach to Nonresponse", *Proceedings of the Survey Research Methods Section, American Statistical Association*, 20-34

Rufibich, K. (2006), Log-Concave Density Estimation and Bump Hunting for i.i.d. Observations. Ph.D. dissertation, Univ. Bern and Göttingen

Rufibach, K. (2007), Computing Maximum Likelihood Estimators of a Concave Density, *Journal of Statistical Computation and Simulation*, Vol 77, No. 7, 561-574

Satten, G. A., Datta, S., and Williamson, J.M. (1998), Inference Based on Imputed Failure Times for the Proportional Hazards Model with Interval-Censored Data, *Journal of the American Statistical Association*, Vol 93, 318-327

Sen, B., Banerjee, M. and Woodroofe, M., (2010), Inconsistency of Bootstrap: The Grenander Estimator, *Annals of Statistics*, Vol 38, 1953-1977

Silvapulle, M. and Sen, P. (2004), *Constrained Statistical Inference: Inequality, Order and Shape Restrictions* New: Wiley

Sun, J. (2006), *The Statistical Analysis of Interval-Censored Failure-Time Data.* New York: Springer

Treloard, A. (1981), Menstrual Cyclicity and the Pre-menopause, *Maturitas*, Vol 3, 249-264

Turnbull, B., (1976):,The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data, *Journal of the Royal Statistical Society. Series B*

(Methodological), Vol 38, No. 3 290-295

van Eeden, C. (1958) *Testing and Estimating Order Parameters of Probability Distributions*, Ph.D. thesis, University of Amsterdam

Wegman, E. (1969), A Note on Estimating a Unimodal Density, *The Annals of Mathematical Statistics*, Vol 40 No. 5, 1661-1667

Wellner, J. A. and Zhan, Y. (1997), A Hybrid Algorithm for Computation of the Nonparametric Maximum Likelihood Estimator from Censored Data, *Journal of the American Statistical Association*, Vol 92, 945 - 959

Wong, G. Y., and Yu, Q. (1999), Generalized MLE of a Joint Distribution Function with Multivariate Interval-Censored Data, *Journal of Multivariate Analysis*, Vol 69, 155 - 166

Appendices

A Proof of Theorem 1

In chapter 3, we stating the following theorem about the likelihood function for the log-concave NPMLE for interval censored data:

The likelihood function is bounded if one of the following three conditions is met:

1. All the data are censored

2. At least two data points are uncensored, and they are not equal to each other

3. There exists one uncensored data point which is not contained in at least one of the censored intervals

Proof. We shall see that case 1 is trivial and case 2 is easily proven true using the proof in the uncensored case presented in Rufibach (2006). Case 3 will require a little more work.

To prove case 1, first note that the contribution to the likelihood function for any censored interval can be bounded by

$$\log\left(\int_{L_i}^{R_i} e^{\phi(x)}\mathrm{d}x\right) \leq 0$$

This clearly implies the likelihood function will be less than or equal to 0.

(Note: in these proofs, we will treat $e^{\phi(x)}$ as though it were a proper distribution function, unlike our earlier parameterization which allow it to be proportional to a proper distribution function)

For case 2, we can break up the log likelihood into the contribution of the observed points and the censored intervals. The contribution of the censored is non-positive from the reasoning above. The contribution of uncensored points is bounded from above, from the fact that the likelihood function for two or more unique observed uncensored times is bounded (Rufibach 2006).

For case 3, first we note that the only complication is when there is only one unique time. If there is more than one unique time, this could be considered case 2 which has already been established. It is worth noting that there can be multiple observations all occurring at the same time, so one could have multiple exact observations without qualifying as case 2. If there are n_1 uncensored observations at time x_1 and n_2 censored observations, the log likelihood function can be written as

$$\ell(\phi) = n_1\phi(x_1) + \sum_{i=n_1+1}^{n_1+n_2} \log\left(\int_{L_i}^{R_i} e^{\phi(x)}\mathrm{d}x\right)$$

Suppose L and R are the end points of an interval such that the unique uncensored time is not within it. Then because the contribution of the other censored observations

is less than 0, we have

$$\ell(\phi) \leq n_1\phi(x_1) + \log\left(\int_L^R e^{\phi(x)}\mathrm{d}x\right)$$

Noting that the right side of the above equation is bounded below 0, it suffices to show that $\ell(\phi)$ is bounded if $\ell(\phi) \to -\infty$ whenever $\phi(x_1) \to \infty$.

Without loss of generality, let us assume that $x_1 = 0$ and $L > 0$. Then we have that

$$\ell(\phi) \leq n_1\phi(0) + \log\left(\int_L^R e^{\phi(x)}\mathrm{d}x\right)$$

$$\ell(\phi) \leq n_1\phi(0) + \log\left(\int_L^\infty e^{\phi(x)}\mathrm{d}x\right)$$

We note that for any choice of $\phi(0)$, $\log\left(\int_L^\infty e^{\phi(x)}\mathrm{d}x\right)$ is maximized by setting $\phi(x)$ to be an exponential distribution with rate $\lambda = e^{\phi(0)}$. This can be seen readily from the fact that the exponential distribution is the limit of the log-concave constraint. This means we can use the cdf of the exponential distribution to further bound the likelihood, *i.e.*, setting $\phi(0) = \log(\lambda)$, we get

$$\ell(\phi) \leq n_1 \log(\lambda) + \log(e^{-\lambda L}) = n_1 \log(\lambda) - \lambda L$$

Because $L > 0$, as $\lambda \to \infty$ the above equation approaches $-\infty$. Therefore, the likelihood function is bounded.

□

B Efficient Likelihood Functions for the Log-Concave NPMLE

For this algorithm, efficient calculation of the likelihood and its derivatives is paramount for an efficient algorithm. The first step in calculating the likelihood function is calculating a vector $\mathbf{p}$ in which $p_k = \int_{t_k}^{t_{k+1}} e^{\phi(x)} \mathrm{d}x$, or the mass placed within the k^{th} contribution interval. Because $\phi(x)$ is linear between the knots, the solution is in closed form. Define $\phi(t_k) = \phi_k$ and $\Delta x_k = x_{k+1} - x_k$. Integrating gives us

$$p_k = \frac{\Delta t_k}{\Delta \phi_k} \times (e^{\phi_{k+1}} - e^{\phi_k})$$

This is undefined if $\phi_{k+1} = \phi_k$ and so the limit as $\phi_{k+1} \to \phi_k$ will be used instead. The limit is $e^{\phi_k} \times \Delta t_k$. It was also observed that calculating p_k suffers from numeric instability as $\phi_k \to \phi_{k+1}$. In order to deal with, if $|\Delta \phi_k| < 10^{-5}$, p_k will be replaced with a first order Taylor approximation, *i.e.*,

$$p_k \approx e^{\phi_k} \times \Delta t_k + \frac{\partial p_k}{\partial \phi_{k+1}} \times \Delta \phi_k$$

$$= e^{\phi_k} \times \Delta t_k + e^{\phi_k} \times \frac{\Delta t_k}{2} \times \Delta \phi_k$$

$$= e^{\phi_k} \times \Delta t_k \times \left(1 + \frac{\Delta \phi_k}{2}\right)$$

In order to efficiently calculate the likelihood function, we calculate the cdf at each of the knots and store it in a vector $\mathbf{F}$, *i.e.* $\mathbf{F}[m] = \sum_{i=1}^{m-1} p_i$. If u_i = the index of the support point x_j which is equal to the upper limit of the i^{th} observation and l_i is the index for the lower point, then

$$\sum_{i=1}^{n} \log\left(\int_{L_i}^{R_i} e^{\phi(x)} \mathrm{d}x\right) - n \times \log\left(\int_{-\infty}^{\infty} e^{\phi(x)} \mathrm{d}x\right)$$

$$= \sum_{i=1}^{n} \log\left(\mathbf{F}[\mathrm{u}_i] - \mathbf{F}[\mathrm{l}_i]\right) - n \times \log\left(\mathbf{F}[k]\right)$$

Using these methods, the likelihood function could be computed in $O(n)$ time. For current status data with a large number of ties in the times, several of the pairs u_i and l_i are identical. This can further accelerate the calculation of the likelihood by writing it in the form

$$= \sum_{i=1}^{n^*} c_i \log\left(\mathbf{F}[\mathrm{u}_i^*] - \mathbf{F}[\mathrm{l}_i^*]\right) - n \times \log\left(\mathbf{F}[k]\right)$$

where u_i^*, l_i^* are the indices for the i^{th} unique observation, c_i is the number of times that the pair u_i^*, l_i^* appears in the data set and n^* is the count of unique pairs in the data set. Using this, the likelihood can be computed in $O(k)$ time. Due to the fact that large numbers of ties are often observed for current status data, this

can lead to a significant improvement. In the illustrative example, $n = 2{,}423$ and $k = 55$. Similar methods were used calculate the vector of derivatives, which could be computed in $O(k^2)$ time. Without using the final modification, the vector of derivatives is computed in $O(kn)$ time.

C Simulation Results for LC NPMLE

Quantile Estimation

The tables below give the results of the simulation for quantile estimation. The top of the table shows the 0.1, 0.25, 0.5, 0.75 and 0.9 quantile, respectively, of the distribution of the simulated data. The simulated data is current status data, in which the distribution of the censoring distribution followed the same distribution as the true times. The main table shows the bias and standard deviation found for each of the estimators, organized by sample size. LC = log-concave NPMLE, UC = unconstrained NPMLE and KS = Kern Smoother.

	Q(p)	0.27 (0.1)	0.48 (0.25)	0.84 (0.5)	1.35 (0.75)	1.94 (0.9)
n		Bias / Standard Deviation				
50	LC	−0.04/0.11	−0.04/0.11	−0.04/0.14	0.01/0.22	0.21/0.31
	UC	−0.11/0.14	−0.07/0.17	−0.05/0.22	0.02/0.33	0.27/0.4
	KS	0.03/0.07	−0.03/0.09	−0.07/0.13	−0.1/0.24	−0.11/0.37
200	LC	−0.01/0.07	−0.01/0.06	−0.02/0.08	−0.01/0.13	0.07/0.21
	UC	−0.05/0.09	−0.02/0.12	−0.03/0.14	0/0.22	0.12/0.35
	KS	0.02/0.04	−0.01/0.05	−0.02/0.07	−0.05/0.14	−0.28/0.35
800	LC	0/0.03	0/0.03	−0.01/0.04	−0.01/0.06	0.03/0.11
	UC	0/0.05	−0.02/0.06	−0.01/0.08	0.01/0.14	0.05/0.21
	KS	0.01/0.02	0/0.02	0/0.04	−0.02/0.07	−0.47/0.29

Table C.1: Quantile Estimation for Gamma(2, 2)

	Q(p)	43.7 (0.1)	46.5 (0.25)	49.8 (0.5)	53.3 (0.75)	56.5 (0.9)
n		Bias / Standard Deviation				
50	LC	−0.4/2.55	−0.08/1.54	−0.05/1.14	0.06/1.44	0.72/2.14
	UC	−0.67/3.73	−0.23/2.08	−0.21/1.75	0.03/2.03	0.93/2.41
	KS	11.48/7.22	3.36/3.15	−0.1/1.51	−2.24/1.56	−3.89/1.92
200	LC	−0.1/1.13	0.08/0.73	−0.02/0.67	−0.1/0.81	0.18/1.19
	UC	−0.17/1.51	0.04/1.1	−0.11/1.06	−0.01/1.23	0.14/1.82
	KS	12.64/5.73	1.88/1.33	−0.22/0.67	−1.69/0.93	−3.83/1.6
800	LC	0.14/0.68	0.04/0.44	−0.07/0.33	−0.12/0.44	0/0.61
	UC	0.1/0.97	−0.03/0.64	−0.1/0.51	−0.05/0.75	−0.04/0.98
	KS	15.94/3.23	1.18/0.66	−0.25/0.3	−1.22/0.48	−4.57/1.35

Table C.2: Quantile Estimation Gamma(100, 2)

	Q(p)	2.75 (0.1)	3.25 (0.25)	3.76 (0.5)	4.22 (0.75)	4.6 (0.9)
n		Bias / Standard Deviation				
50	LC	−0.12/0.33	−0.02/0.21	0/0.15	0.01/0.18	0.06/0.25
	UC	−0.24/0.38	−0.09/0.29	−0.03/0.23	0.05/0.24	0.17/0.26
	KS	0.5/0.44	0.19/0.26	0.01/0.16	−0.13/0.17	−0.25/0.19
200	LC	−0.03/0.18	0.01/0.12	0/0.09	0/0.09	0.02/0.14
	UC	−0.1/0.25	−0.05/0.2	−0.02/0.16	0.02/0.15	0.08/0.19
	KS	0.51/0.31	0.12/0.14	0.01/0.09	−0.09/0.1	−0.25/0.14
800	LC	0/0.11	0.01/0.06	0/0.05	−0.01/0.05	0/0.07
	UC	−0.03/0.19	−0.01/0.12	−0.02/0.11	0/0.1	0.02/0.12
	KS	0.55/0.17	0.08/0.06	0/0.04	−0.07/0.05	−0.3/0.1

Table C.3: Quantile Estimation for Weibull(6, 4)

	Q(p)	0.28 (0.1)	0.51 (0.25)	1 (0.5)	1.96 (0.75)	3.6 (0.9)
n		Bias / Standard Deviation				
50	LC	−0.04/0.13	−0.09/0.15	−0.14/0.23	0.02/0.43	0.77/0.73
	UC	−0.12/0.17	−0.11/0.23	−0.12/0.4	−0.04/0.94	0.59/1.95
	KS	0.18/0.21	−0.11/0.13	−0.35/0.29	−0.49/0.7	−0.54/1.56
200	LC	0/0.07	−0.03/0.07	−0.08/0.12	−0.02/0.25	0.49/0.46
	UC	−0.05/0.09	−0.03/0.13	−0.07/0.24	−0.01/0.5	0.27/1.05
	KS	0.16/0.16	−0.08/0.07	−0.26/0.18	−0.28/0.37	−0.58/1.09
800	LC	0/0.03	−0.02/0.03	−0.07/0.06	−0.04/0.12	0.39/0.22
	UC	−0.03/0.06	−0.03/0.07	0/0.14	−0.04/0.33	0.08/0.83
	KS	0.1/0.06	−0.08/0.03	−0.19/0.11	−0.13/0.18	−0.78/0.91

Table C.4: Quantile Estimation for Lognormal(0, 1)

	Q(p)	0.41 (0.1)	0.84 (0.25)	2.17 (0.5)	4.68 (0.75)	6.72 (0.9)
n		Bias / Standard Deviation				
50	LC	−0.14/0.22	−0.36/0.31	−0.27/0.53	0.51/0.79	0.95/1.03
	UC	−0.26/0.33	−0.28/0.57	−0.3/1.02	0.17/1.31	0.96/1.43
	KS	−0.02/0.21	−0.42/0.28	−0.43/0.51	0.09/0.9	0.01/1.26
200	LC	−0.03/0.11	−0.21/0.14	−0.14/0.27	0.39/0.43	0.32/0.63
	UC	−0.09/0.18	−0.09/0.3	−0.2/0.77	0.07/0.89	0.43/1.09
	KS	−0.04/0.09	−0.28/0.13	−0.15/0.26	0.23/0.53	−0.51/1.01
800	LC	0.01/0.05	−0.15/0.07	−0.08/0.13	0.38/0.22	0.04/0.35
	UC	−0.05/0.11	−0.05/0.15	−0.04/0.45	0.04/0.53	0.07/0.79
	KS	−0.05/0.04	−0.18/0.06	0.03/0.14	0.31/0.26	−0.97/0.64

Table C.5: Quantile Estimation for Gamma Mixture

Density Estimation

The tables below give the results of the simulation for density estimation at the quantiles. The top of the table shows the density of the simulated data at the 0.1, 0.25, 0.5, 0.75 and 0.9 quantiles. The simulated data is current status data, in which the distribution of the censoring distribution followed the same distribution as the true times. The main table shows the bias and standard deviation found for each of the estimators, organized by sample size. LC = log-concave NPMLE and KS = Kern Smoother. The unconstrained NPMLE is excluded as it does not provide density estimates.

	f(p)	0.62(0.1)	0.74(0.25)	0.63(0.5)	0.36(0.75)	0.16(0.9)
n		Bias / Standard Deviation				
	LC	0.11/0.35	−0.01/0.31	−0.04/0.21	−0.09/0.22	0.01/0.13
50	KS	0.15/0.1	0.12/0.11	0.01/0.1	−0.02/0.08	−0.02/0.07
	LC	−0.05/0.27	0/0.15	0/0.11	−0.03/0.08	−0.02/0.09
200	KS	0.09/0.08	0.05/0.07	−0.01/0.08	0.02/0.05	0.02/0.03
	LC	−0.05/0.12	0.01/0.09	0.01/0.07	−0.01/0.04	−0.02/0.04
800	KS	0.05/0.04	0.01/0.04	−0.01/0.04	0.05/0.03	0.05/0.02

Table C.6: Density Estimation at Quantiles for Gamma (2,2)

	f(p)	.038(0.1)	.067(0.25)	.08(0.5)	.061(0.75)	.032(0.9)
n		Bias / Standard Deviation				
	LC	.007/.029	−.007/.036	−.003/.029	−.006/.034	.003/.026
50	KS	.01/.005	.027/.007	.032/.008	.016/.007	−.003/.006
	LC	−.004/.022	−.002/.018	.003/.016	−.002/.015	−.006/.018
200	KS	.012/.003	.023/.005	.023/.007	.012/.005	0/.004
	LC	−.002/.01	.003/.009	0/.013	−.001/.009	−.004/.009
800	KS	.017/.002	.022/.003	.014/.006	.01/.004	.005/.002

Table C.7: Density Estimation at Quantiles for Gamma(100,2)

	f(p)	0.21(0.1)	0.4(0.25)	0.55(0.5)	0.49(0.75)	0.3(0.9)
n		Bias / Standard Deviation				
	LC	0.02/0.16	−0.05/0.23	−0.02/0.19	−0.04/0.27	0.02/0.23
50	KS	0.02/0.04	0.07/0.07	0.11/0.09	0.09/0.07	0/0.06
	LC	−0.04/0.11	−0.01/0.09	0.01/0.1	−0.02/0.13	−0.03/0.17
200	KS	0.04/0.03	0.06/0.05	0.08/0.06	0.08/0.05	0.03/0.04
	LC	−0.01/0.05	0/0.06	0.01/0.07	0/0.08	−0.03/0.08
800	KS	0.05/0.02	0.06/0.03	0.05/0.04	0.08/0.03	0.06/0.02

Table C.8: Density Estimation at Quantiles for Weibull(6,4)

	f(p)	0.63(0.1)	0.62(0.25)	0.4(0.5)	0.16(0.75)	0.05(0.9)
n		Bias / Standard Deviation				
	LC	0.17/0.28	0.06/0.21	−0.04/0.12	−0.06/0.09	0/0.04
50	KS	0.35/0.07	0.28/0.09	0.03/0.08	−0.05/0.04	−0.01/0.03
	LC	0.03/0.16	0.07/0.09	0/0.05	−0.04/0.03	−0.01/0.02
200	KS	0.32/0.06	0.25/0.08	0.01/0.06	−0.05/0.03	0/0.02
	LC	0.02/0.08	0.06/0.05	0.01/0.02	−0.03/0.01	−0.01/0.01
800	KS	0.29/0.05	0.21/0.07	−0.02/0.04	−0.04/0.03	0.01/0.01

Table C.9: Density Estimation at Quantiles for Lognormal(0,1)

	f(p)	0.36(0.1)	0.32(0.25)	0.11(0.5)	0.09(0.75)	0.05(0.9)
n		Bias / Standard Deviation				
	LC	0.13/0.11	0.08/0.08	−0.07/0.04	−0.01/0.04	0.01/0.03
50	KS	0.21/0.03	0.13/0.04	−0.07/0.03	0.01/0.02	0.01/0.02
	LC	0.09/0.05	0.07/0.03	−0.06/0.01	0.01/0.01	0.01/0.01
200	KS	0.18/0.03	0.08/0.03	−0.07/0.02	0.02/0.02	0.01/0.01
	LC	0.08/0.02	0.07/0.02	−0.06/0.01	0.01/0.01	0.01/0.01
800	KS	0.13/0.02	0.04/0.02	−0.05/0.01	0.02/0.01	0.02/0

Table C.10: Density Estimation at Quantiles for Gamma Mixture

D Problems with the CRAN "intcox" Package

As mentioned earlier, Wei Pan presented an algorithm for finding the Cox PH MLE estimate with the baseline survival estimation done by the unconstrained NPMLE. This algorithm was later implemented in the CRAN package "intcox", although this was not written by Wei Pan. We found three problems with this package. First, it does not report standard error in the summaries, but rather reports "NA". While a nuisance, this problem is not of great concern, as inference can still be done via bootstrapping. In fact, we believe bootstrapping is likely to be more reliable than sieve estimation based on our experiences in other applications.

Secondly, the algorithm always failed to converge if there were ties in the data. This would be a significant problem in real world problems. Virtually all data sets which appear in the literature follow a discrete (rounded) censoring mechanism, meaning that there are several ties in the data. However, for simulated data, we can use a continuous censoring mechanism to insure no ties in the data.

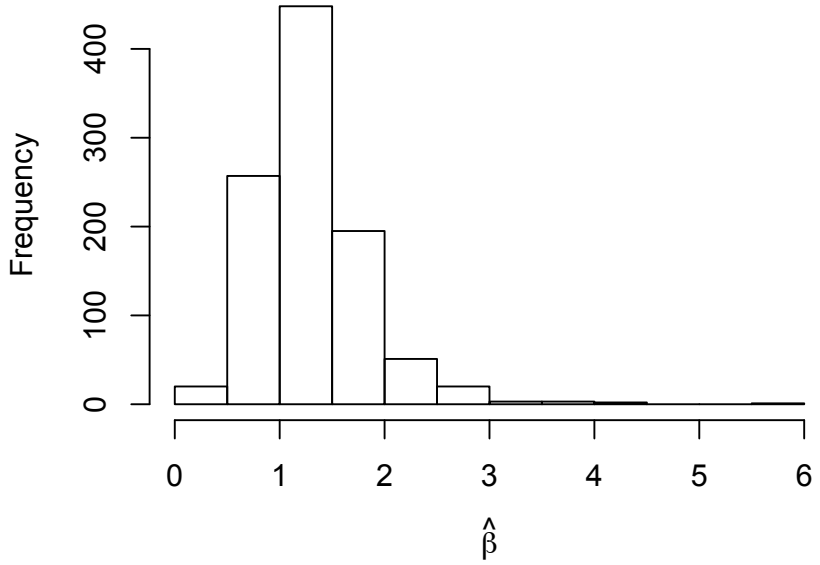

Figure D.1: 1000 Samples of $\hat{\beta}$ for Cox PH Model with Unconstrained Baseline Survival

Finally, the algorithm appears to terminate prematurely. The stopping criterion instituted in the package is the difference in log-likelihood for each iteration. This differs from Wei Pan (1999), which used a max absolute difference in regression estimates. The default set in intcox is $\epsilon = 10^{-4}$. We found that this typically lead to a premature termination of the algorithm. For example, we ran simulated 1000 datasets in the same manner as in section 4.3.3. For each dataset, we first used intcox with $\epsilon = 10^{-4}$ and then ran the same data set using $\epsilon = 10^{-6}$. When doing this, we found the median difference in likelihood function when using the tighter stopping criterion was 0.266 and the maximum difference was 1.83. The difference in estimated regression coefficient was substantial as well; the median absolute difference in estimated regression coefficient was 0.037, with maximum difference 2.16 (the true $\beta = 1$). Because of this, this stopping criterion should not be considered valid.

A counter intuitive result that occurs is that using the looser stopping criterion lead to both a slight reduction in bias and a substantial reduction in standard deviation. With the looser stopping criterion, the mean of our sample was 1.29 with standard deviation 0.425, while with the tighter stopping criterion, the mean of our sample was 1.31 with standard deviation 0.517. When we look at the sample of values in figure 4.7, we see that the distribution of $\hat{\beta}$ is right skewed, leading to upward bias and higher standard deviation. Because the algorithm starts at 0 for the regression parameters, stopping prematurely likely leads to lower estimated values. Thus this issue happens to help in this scenario. However, it is certainly not theoretically justified and would not be recommended to do intentionally. Wei Pan (1999) suggests that Lasso could be used to reduce the bias and variance. In their implementation of Lasso, the standard deviation is reduced, but the bias is reversed (*i.e.* has downward bias of the same magnitude instead of upward bias). It's not clear that this would be the desired operating characteristics.

E Complications with Exact Times for the Log-Concave Cox PH Model

We found one problem with the log-concave Cox PH model. In general, the density function in a Cox PH model can be defined as

$$f(t|X,\beta) = f_0(t)e^{X^T\beta}S_0(t)^{e^{X^T\beta}-1}$$

This can cause a problem in certain situations. To illustrate this, consider if the

dataset has no censored data. If we order the event times $t_1, ..., t_n$ such that $t_1 \leq ... \leq t_n$ we know that $\hat{S}_0(t_n) = 0$ and $\hat{f}_0(t_n) > 0$ for the log-concave estimator (a technical note is that this assumes $\hat{S}_0$ is defined, and we are about to show that in some situations, it is not). Because of this, the contribution of t_n to the log likelihood function with a Cox PH model will be $-\infty$ if $X_n^T\beta > 0$ and ∞ if $X_n^T\beta < 0$, leading to degenerate estimation. We note that this problem does not occur with the contribution of censored data to the likelihood function. In addition, this problem is not faced in the Cox PH model with unconstrained baseline distribution, as the distribution is treated as though it were discrete in that case.

We provide two methods for avoiding this problem. We note that if $X_n^T\beta = 0$, then we can replace $f(t_n|X, \beta)$ with the limit as $t \to t_n$, leading to $\hat{f}(t_n|X_n, \beta) = \hat{f}_0(t_n)$. We can insure that $X^T\beta = 0$ by centering X about X_n, *i.e.* $X_i = X_i - X_n$ for $i = 1, ..., n$. Typically, re-centering in regression problems only leads to shifts in the intercepts and does not affect the slope coefficients, so re-centering is often done without concern. In this case, re-centering does affect all coefficients in that it makes them estimatible.

Doing this requires the maximum value of t to belong to only one observation, *i.e.* $t_{n-1} \neq t_n$. While theoretically this should happen with probability 1 for a continuous distribution, in practice this doesn't always happen yet we may still want to model the data as though it were from a continuous distribution. If it is the case that $t_n = t_{n-1}$, we will not be able to use our re-centering trick from above if $X_{n-1} \neq X_n$. In such a case, one solution could be to manually censor these maximum values. While it is very unsatisfying to censor data that was reported exactly, there is some justification for this. Because ties should happen with probability 0, if there are ties in the data and the event times are distributed continuously, then one should conclude that the dataset has been rounded or binned. This is a form of censoring. It is up to the investigator to determine the width of the censoring interval.

We also suggest another repair to the estimator in chapter 7 which would resolve this issue by forcing $\hat{S}(t_n) > 0$. This would be a more satisfying method, as it repairs the problem causing this issue, rather than sidestepping the issue. We also hope this repair would reduce the bias of the regression parameters, although this fix has not been implemented for the Cox PH model at this time.