LOCAL LIKELIHOOD FOR INTERVAL-CENSORED AND AGGREGATED POINT PROCESS DATA

by

Chun-Po Steve Fan

A thesis submitted in conformity with the requirements for the degree of Doctor of Philosophy
Dalla Lana School of Public Health
University of Toronto

©Copyright by Chun-Po Steve Fan 2009
Local Likelihood for Interval-Censored and Aggregated Point Process Data

Chun-Po Steve Fan

Doctor of Philosophy
Dalla Lana School of Public Health
University of Toronto
November 2009

Abstract

The use of the local likelihood method (Tibshirani and Hastie, 1987; Loader, 1996) in the presence of interval-censored or aggregated data leads to a natural consideration of an EM-type strategy, or rather a local EM algorithm. In the thesis, we consider local EM to analyze the point process data that are either interval-censored or aggregated into regional counts. We specifically formulate local EM algorithms for density, intensity and risk estimation and implement the algorithms using a piecewise constant function. We demonstrate that the use of the piecewise constant function at the E-step explicitly results in an iteration that involves an expectation, maximization and smoothing step, or an EMS algorithm considered in Silverman, Jones, Wilson and Nychka (1990). Consequently, we reveal a previously unknown connection between local EM and the EMS algorithm.

From a theoretical perspective, local EM and the EMS algorithm complement each other. Although the statistical methodology literature often characterizes EMS methods as ad hoc, local likelihood suggests otherwise as the EMS algorithm arises naturally from a local likeli-
hood consideration in the context of point processes. Moreover, the EMS algorithm not only
serves as a convenient implementation of the local EM algorithm but also provides a set of
theoretical tools to better understand the role of local EM. In particular, we present results
that reinforce the suggestion that the pair of local EM and penalized likelihood are analogous
to that of EM and likelihood. Applications include the analysis of bivariate interval-censored
data as well as disease mapping for a rare disease, lupus, in the Greater Toronto Area.
Acknowledgements

I am very much grateful to my supervisor Jamie Stafford for his encouragement, guidance and patience throughout the doctoral program. I thank my committee for many helpful suggestions, especially Patrick Brown for his support and advice in programming and computation, and Dr. Paul Fortin of Toronto Western Hospital for the permission to use the lupus dataset. Moreover, I would like to expression my gratitude to Paul Corey, Wendy Lou, Michael Escobar, Rafal Kustra, Jerry Brunner and Radu Craig for many intellectually intriguing discussions during the past four years. I also thank the examination committee for their careful review of the thesis.

I enjoyed many social events and took advantage of many interesting discussions with my fellow graduate students Angel Valov, Gerald Lebovic, Samuel Hikspoors and Ye Li. I also thank Christopher Meaney for editing my thesis and Longhai Li for providing the thesis template.

Finally, I must express my sincerest gratefulness to my family. I thank them for believing in me and supporting me. There is no word that can describe my uttermost gratitude to my wife for her unconditional love and the affectionate care in good and bad time during this long journey.
Contents

1 Introduction 1

1.1 Notations and Assumption ........................................... 3

1.2 Literature Review ......................................................... 5

1.2.1 Nonparametric Likelihood ........................................... 5

1.2.2 Local Likelihood ...................................................... 11

1.2.3 Piecewise Constant Approximation .............................. 16

1.3 Overview ................................................................. 17

2 One-Jump Point Processes 21

2.1 EMS Algorithm ........................................................ 23

2.2 Adaptive EMS Algorithm .............................................. 27

2.3 Partition Configurations .............................................. 29

2.4 Application: Bivariate Interval-Censored Data ................... 31

3 Temporal Point Processes 35

3.1 Local EM Algorithm .................................................. 36

3.2 EMS Implementation .................................................. 41

3.3 Application: Incidence Rate of Nausea in NCGS .................. 44

3.4 Simulation Study ...................................................... 48
# 4 Spatial Point Processes

4.1 Notations and Assumption ............................................. 51
4.2 Local Likelihood and Local EM ........................................ 53
4.3 EMS Implementation ....................................................... 55
4.4 Application: Disease Mapping .......................................... 57
  4.4.1 Study Objective ..................................................... 57
  4.4.2 Census Tracts ....................................................... 58
  4.4.3 Statistical Model ................................................... 60
  4.4.4 Kernel Matrix ..................................................... 61
  4.4.5 Cross Validation ................................................... 65
  4.4.6 Results and Discussion ............................................ 66
4.5 Simulation Study ........................................................ 68
4.6 Conclusions ............................................................. 70

# 5 Convergence and Optimality

5.1 Role of Local EM ......................................................... 74
  5.1.1 Modified EMS Algorithm and Equivalent Kernel ............... 74
  5.1.2 Convergence of EMS to Local EM .................................. 78
  5.1.3 Local EM and Penalized Likelihood ............................. 84
5.2 Convergence of Local EM Algorithm ................................ 86
  5.2.1 Fixed-Point Solution .............................................. 86
  5.2.2 Rate of Convergence .............................................. 86
5.3 Bivariate Interval-Censored Data ................................... 91

# 6 Concluding Remarks

# A Appendix

A.1 EMS Implementations of the Local EM Algorithm ................. 97
List of Figures

2.1 Each shaded rectangle represents a bivariate interval-censored observations. The sixteen intersections of the shaded rectangles are referred to as maximal intersections. ................................................................. 33

2.2 All three density estimates maximize the nonparametric likelihood (1.1). . . 33

2.3 Three local EM density estimates with different kernel functions. . . . . . . 34

3.1 Each row represents a patient. A rectangle in a row indicates the time interval between two consecutive visits, and its gray scale indicates the number of reported nauseas during the interval. Crosses indicate drop-out times, and dash lines represent the time elapsed between the last visit and drop-out. Two patients in the treatment group are excluded from the analysis. The x-axis shows the time points used to construct the partition $J$. . . . . . . . . . . . . 46

3.2 Local likelihood intensity estimates for the risk of nausea. . . . . . . . . . . . . 47

3.3 A NPMLE is smoothed by placing expected increments at the left-end, mid and right-end points of the $J_k$’s. By comparing with the three different versions of smoothed NPMLE’s, we show that the proposed local likelihood intensity estimator achieves a lower overall MISE with a small bandwidth of 0.5 (locally constant) and 0.6 (locally linear). Moreover, the bandwidth value with which the local likelihood estimator reaches its lowest MISE is close to that of Diggle’s edge-corrected intensity estimate, which is 0.65. . . . . . . . . . . . . . . . . 49
4.1 Superimposing \( M_1 \) and \( M_2 \) results in a partition, \( J \).

4.2 Each enclosed area represents a census tract. The census tract is small in the urban centre and large in the suburban area. Moreover, the census tract boundaries change from one map to another.

4.3 The region \( D \) is enclosed by three solid lines. Let \( \partial D_1 \), \( \partial D_2 \) and \( \partial D_3 \) be the horizontal, the arc and the vertical solid lines, respectively.

4.4 The GLM-adjusted pixel relative risk estimated by the local EM. The highest risk estimated equals approximately 3, indicating that the expected number of SLE cases in the area is three times higher than what the expected would be if there is no spatial variation.

4.5 The MISE of the smoothed nonparametric risk estimator is higher than that of the local likelihood risk estimator and kernel risk estimator for all bandwidth values shown in the plot. The local likelihood risk estimator achieves the lowest overall MISE with a small bandwidth value of 0.19. The kernel risk estimator achieves the lowest MISE among the three risk estimators since it is based on exact case locations.

5.1 The diagram illustrates how the results in §5.1 are related to reinforce the suggestion that local EM and the penalized likelihood may be paired in a manner analogous to the more familiar pairing of EM and likelihood.
4.1 The bandwidth that gives the smallest prediction error is chosen to be the optimal value. As shown in the table above, the prediction error is the smallest with the bandwidth of 1350.
Chapter 1

Introduction

When data arise from a point process, they could be a sequence of survival or event history data or a set of disease incidences in space. In either case, the estimation of intensity has long been the central focus in the theories and practices of nonparametric modeling.

When a temporal process can be continuously observed in time, event times are exactly recorded, and the data are said to be complete or directly observed. However, such a continuous observation design is often not realistic in many clinical studies. In clinical studies, subjects are periodically, rather than continuously, assessed. The subject’s assessment times could be determined by either a random or deterministic process, which may depend on the event process of interest. This sequence of assessment times can also vary among subjects. In addition, each subject may be right-censored due to drop-out or lost-to-followup. The event history, by no means, can be directly observed in this situation. As a result, this type of design can only provide investigators with incomplete or partially observed data, such as interval-censored data. An event time is said to be interval-censored if it is only known to lie in a time interval between any two consecutive visits. If multiple events are allowed, we refer to this type of data as panel counts.

In the case of spatial processes, observations consist of incidence locations, represented
by x- and y-coordinates. Here, we consider a less ideal scenario, in which collecting complete
data is not feasible due to budgetary or regulatory constraints. As a result, data are only
partially observed, and our primary concern is with the data that are either area-censored
or aggregated into regional counts. To visualize the area-censored or aggregated data, we
exploit the geographical feature of censoring regions and display the data as maps using
geographical information systems. Should the event of interest be the incidence of a certain
disease, the map is referred to as a disease map, and quantitative methods that model
the underlying disease process in space and/or in time are referred to as disease mapping.
When the data are obtained from multiple sources or longitudinally collected, they may have
different spatial resolutions, or censoring regions may be misaligned. For example, disease
incidence data may be collected over a long period of time in order to accumulate sufficient
information. During this collection period, the political boundary of geographical area under
investigation expands, or boundaries of censoring regions change. As a result, there could
be multiple disease maps that represent more or less the same geographical area, but they
consist of different collections of censoring regions. How one can model spatial variations
in disease risk by combining multiple disease maps remained unclear. In addition, we must
take into consideration regional population sizes in the analysis since the greater number of
disease incidences may be simply due to a larger population.

Should the underlying intensity be smooth in time or in space, the local likelihood pro-
posed by Tibshirani and Hastie [41] is well established and worthy of consideration. Although
the optimality and asymptotic properties of local likelihood estimators in the context of point
processes have been carefully scrutinized and well studied for right-censored data, much re-
 mains unknown when point process data are interval-censored or aggregated into regional
counts. Moreover, the data may have multiple resolutions, scales and alignments owing to
the censoring mechanism. The research interest is to extract useful information from a set of
multi-scaled and misaligned observations. In this thesis, we aim to develop and study a class
of local likelihood methods capable of flexible estimation in the presence of interval-censored or spatially aggregated data.

1.1 Notations and Assumption

Consider a study that consists of \(n\) independent subjects, each of which is assessed at a sequence of time points \(T_i = \{t_{ij} \mid j = 1, \cdots, K_i\}\), such that \(t_{i1} < \cdots < t_{iK_i}\) for all \(i\). While this sequence of assessment times could be random or deterministic, we assume it stochastically independent of the event process of interest. Denote the time interval between \(t_{ij-1}\) and \(t_{ij}\) by \(I_{ij}\), which we refer to as the \(j\)th panel of the \(i\)th subject. These panels may vary in length, and some may overlap with others.

Let \(T_{i\ell}\) denote the \(\ell\)th event time for subject \(i\). Suppose \(T_{i\ell}\) is interval-censored by \(T_i\), and the data only indicate if \(T_{i\ell} \in I_{ij}\) for all \(j\) and \(\ell\) and for each \(i\). To facilitate the methodological development, we assume that the event process is inhomogeneous Poisson and adopt notations used in the counting process literature. Let \(N_i(t)\) indicate the number of events for the \(i\)th subject up to time \(t\). Let the increment over \(I_{ij}\) be \(N_{ij} = N_i(t_{ij}) - N_i(t_{ij-1})\) and refer to \(N_{ij}\) as the \(j\)th panel count for the \(i\)th subject. When the event is recurrent, \(N_{ij}\) is a non-negative integer. We are interested in determining the expectation \(E N_i(t)\) common to all study subjects. In this setting, \(E N_i(t)\) is interpreted as the cumulative intensity \(\Lambda(t)\) of the event process and assumed to be differentiable with intensity \(\lambda(t)\).

Furthermore, each subject is allowed to drop out of the study for reasons stochastically independent of the event process. Denote subject \(i\)’s drop-out time by \(C_i\). If a subject remains in the study until the study terminates, \(C_i\) is set to be the study termination time. Let \(Y_i(t) = I(C_i > t), Y(t) = \sum_i Y_i(t), \mathcal{M}_i = \{t \mid Y_i(t) > 0\}\) and \(\mathcal{M} = \{t \mid Y(t) > 0\}\). The indicator \(Y_i(t)\) signals whether the \(i\)th subject is at risk at time \(t\), and \(Y(t)\) indicates the total number of at-risk subjects. Note that \(Y(t)\) becomes 0 once the study is terminated.
\( \mathcal{M}_i \) indicates subject \( i \)'s observational period, and \( \mathcal{M} \) is either the entire study period or the period prior to the drop-out of the last subject. Here, we assume a monotone drop-out process; that is, a subject cannot re-enter the study once he/she drops out of the study. It follows that \( Y_i(t) \) and \( Y(t) \) are monotonic processes in the sense that \( Y_i(t_1) \geq Y_i(t_2) \) and \( Y(t_1) \geq Y(t_2) \) for all \( t_1, t_2 \in \mathcal{M} \) such that \( t_1 \leq t_2 \). It should be emphasized that, unless a subject remains in the study from the beginning to the end, the sequence of assessment times also \textit{interval-censors} the drop-out time since he/she is only observable at discrete time points. In other words, if a subject drops out of the study, his/her exact drop-out time cannot be determined but only known beyond the last time seen. In this case, we assume that the subject drops out immediately after his/her last visit. The above details lead to the consideration of the nonparametric likelihood for the complete data:

\[
\mathcal{L}(\lambda) = \sum_{i} \log \lambda(T_{i\ell}) - \sum_{i} \int_{\mathcal{M}} Y_i(u) \lambda(u) \, du.
\]

Following the setup in Hu, Lagakos and Lockhart [20], we construct a partition based on all sequences of assessment times as follows:

1. \( \mathcal{T} = \bigcup_i \mathcal{T}_i \) and \( \mathcal{T} = \{t_0, t_1, \ldots, t_K\} \) with the time of origin \( t_0 \).

2. Construct a partition, denoted by \( \mathcal{J} = \{J_1, \ldots, J_K\} \) with \( J_j = (t_{j-1}, t_j] \).

Henceforth, we refer to the \( J_k \)'s as pixels and indicate if \( J_\ell \subseteq I_{ij} \) by \( I_{ij\ell} \). In addition, we refer to \( \Lambda_j = \int_{J_j} \lambda(u) \, du \) as the \( j \)th pixel intensity and denote the collection of pixel intensities by \( \Lambda = \{\Lambda_j \mid j = 1, \ldots, K\} \).

Let the event process be a general one-jump point process, rather than an inhomogeneous Poisson process. The process may be a survival process or a stochastic process that generates random samples of independent and identically-distributed (i.i.d.) observations. In this case, \( \ell = 1 \) for all \( i \) and \( N_{ij} \) equals either 0 or 1. Moreover, \( Y_i(t) = 1 \) for all \( i \) and all \( t \in \mathcal{M} \). Here, \( \mathcal{M} \) is either \( \mathbb{R} \) or \( \mathbb{R}^+ \). Assume \( \Pr(-\infty < T_i < \infty) = 1 \). Since \( \Pr(N_i(t) = 1) = \Pr(T_i \leq t) \),
\( \mathbf{E} N_i(t) \) is equivalent to the cumulative distribution function \( F(t) \) of \( T_i \) with density \( f(t) \). The event time \( T_i \) is known either to lie between two assessment times adjacent to each other or to be right-censored. If \( T_i \) is not right-censored, \( T_i \) manifests itself as an interval, \( I_i = (L_i, R_i] \) with \( L_i = \max_j(t_{ij} \mid t_{ij} < T_i) \) and \( R_i = \min_j(t_{ij} \mid t_{ij} \geq T_i) \). Otherwise, \( I_i = (L_i, R_i) \) and \( R_i = \infty \). In either case, the event time \( T_i \) is interval-censored, and \( T_i \in I_i \). Additionally, we denote the pixel probability \( \int_{J_j} f(u) \, du \) by \( p_j \) and the collection of \( p_j \)'s by \( p \).

### 1.2 Literature Review

#### 1.2.1 Nonparametric Likelihood

**Density Estimation**

We start with a one-jump point process, and the cumulative distribution function \( F \) is the quantity of interest. In the presence of interval censored data, Turnbull [43] considers the following likelihood in the search of a nonparametric maximum likelihood estimate (NPMLE) over a class of distribution functions on the real line.

\[
\mathcal{L}(F) = \sum_i \log \int_{I_i} dF(u),
\]

where \( \int_{I_i} dF(u) \) is meant to be \( F(R_i) - F(L_i) \). Turnbull [43] defines an innermost interval as an interval whose left- and right-end points are also the left- and right-end points of \( I_i \) for some \( i \). For example, if the observed intervals are \([0, 2]\) and \([1, 3]\), then the innermost interval is simply \([1, 2]\) since 1 is the left-endpoints of the latter interval and 2 is the right-endpoints of the former. Let \( S_j \) be the \( j \)th innermost interval and \( \mathcal{Q} = \{S_j\} \subseteq \mathcal{J} \). As discussed in Turnbull [43], any distribution estimate that puts positive masses outside the set \( \mathcal{Q} \) cannot be a NPMLE. It implies that \( p_\ell = 0 \) for any \( \ell \) such that \( J_\ell \in \mathcal{J} \setminus \mathcal{Q} \).
Let $T_{ij}'$ indicate if $S_j \subseteq I_i$. The details above reduce the likelihood (1.1) to

$$
\mathcal{L}(p) = \prod_i \left( \sum_j T_{ij}' p_j' \right),
$$

where $p_j' = \int_{S_j} F(u)$. Obviously, the maximum likelihood method estimates a distribution function, which constitutes pixel probabilities. Since the likelihood is independent of the behaviour of $F$ within each innermost interval, the NPMLE is only unique up to an equivalence class. For example, suppose that $\hat{p}$ is a NPMLE. Any distribution function $F$ such that

$$
\int_{J_{\ell}} dF(u) = \hat{p}_{\ell} \quad \forall \ell
$$

must also maximize the likelihood (1.1). Such functions form the equivalence class. As shown in Turnbull [43], an immediate consequence of the non-uniqueness of a NPMLE is that only the sum of $p_k$ and $p_\ell$ is identifiable if $I_{ik} = I_{i\ell}$ for some $k, \ell$ and for all $i$. As a result, the NPMLE does not provide any insight to the behaviour of $F$ within each innermost interval.

**Intensity Estimation**

When the event process is an inhomogeneous Poisson process in time, the number of events within the panel $I_{ij}$ follows a Poisson distribution. Since we assume a subject’s drop-out time is immediately after the last visit, $Y_i(t)$ is constant within each of the panels. Let $Y_i(t) = Y_{ij}$ for all $t \in I_{ij}$. In the situation in which an individual process is only observable whenever $Y_i(t) > 0$, the observable panel count is still Poisson distributed with mean equal to $Y_{ij} [\Lambda(t_{ij}) - \Lambda(t_{ij-1})]$, or

$$
Y_{ij} N_{ij} \sim \text{Poisson} \left( Y_{ij} [\Lambda(t_{ij}) - \Lambda(t_{ij-1})] \right).
$$
Note that $Y_{ij} \left[ \Lambda(t_{ij}) - \Lambda(t_{ij-1}) \right]$ can also be expressed in integral form as $Y_{ij} \int_{t_{ij}} d\Lambda(t)$. Wellner and Zhang [45] considers the following nonparametric likelihood for the panel count data:

$$L(\Lambda) = \sum_{ij} Y_{ij} N_{ij} \log \int_{I_{ij}} d\Lambda(u) - \sum_{i} \int_{\mathbb{R}} Y_{i}(t) d\Lambda(u) + \text{terms independent of } \Lambda. \quad (1.2)$$

Similar to the density estimation for interval-censored data, the NPMLE is only unique to an equivalence class. To see this, let $\hat{\Lambda} = \arg\max L(\Lambda)$ and $\hat{\Lambda}_k = \int_{J_k} d\hat{\Lambda}(u)$. Any non-negative function $\Lambda$ that satisfies

$$\int_{J_k} d\Lambda(u) = \hat{\Lambda}_k \quad \forall k$$

must also maximize the likelihood (1.2). These functions are members of the equivalent class. To put it briefly, the likelihood function is independent of the behaviour of $\Lambda$ within each $J_k$. Apart from the non-uniqueness, the NPMLE may not be identifiable in the sense that there may exist $\Lambda_1$ and $\Lambda_2$ such that $\Lambda_1 \neq \Lambda_2$ and $\Lambda_1 = \arg\max L(\Lambda) = \Lambda_2$.

**EM Algorithm**

Introduced by Dempster et al. [10], an EM algorithm is a general optimization procedure to search for a maximum likelihood estimate. The notion of EM can be succinctly described by two steps, namely, the $E$ (for expectation) and $M$ (for maximization) step. Denote the parameter of interest by $\theta$. The $E$-step calculates the expectation of the likelihood conditional on the observed data by introducing the following $Q$ function:

$$Q(\theta \mid \theta') \equiv E_{\theta'} \left[ L(\theta) \mid \text{observed data} \right],$$

where $L(\theta)$ is the likelihood for complete observations. At the $M$-step, one determines a new value of $\theta$ by maximizing $Q(\theta \mid \theta')$. The maximization step typically involve solving the
system of score equations
\[ \frac{\partial}{\partial \theta} Q(\theta | \theta') = 0. \]
The new value of \( \theta \), in turn, replaces the existing value of \( \theta' \) in the E-step of the next iteration. The cycle continues until the values of \( \theta \) and \( \theta' \) meet a convergence criterion.

When data are interval-censored, \( \theta = \mathbf{p} \). Providing the partition \( \mathcal{J} \), the data are “complete” if we observe the \( J_k \) in which \( T_i \) falls for all \( i = 1, \ldots, n \). This would be the case only if all subjects have the same assessment times. Suppose \( T_i \)'s are i.i.d. with density \( f \). The probability that \( T_i \) falls into \( J_k \) is simply \( p_k \). We apply EM to the following likelihood for the complete data:

\[ \mathcal{L}(\mathbf{p}) = \sum_{ik} I(T_i \in J_k) \log p_k \quad \text{with } p_k \geq 0 \text{ for all } k \text{ and } \sum_k p_k = 1. \tag{1.3} \]

At the E-step, we calculate

\[ Q(\mathbf{p} | \mathbf{p}') = \mathbf{E}_{\mathbf{p}'} [\mathcal{L}(\mathbf{p}) | I_1, \ldots, I_n] \]
\[ = \sum_{ik} \mathbf{E}_{\mathbf{p}'} [I(T_i \in J_k) | I_i] \log p_k \]
\[ = \sum_{ik} p'_k \log p_k \tag{1.4} \]

Maximizing (1.4) subject to the constraints results in an EM algorithm, the iteration of which is given by

\[ \hat{p}^{r+1}_j = \frac{1}{n} \sum_i \frac{I_{ij}\hat{p}^r_j}{\sum_k I_{ik}\hat{p}^r_k} \quad \text{for all } j = 1, \ldots, K. \tag{1.5} \]

Note that the self-consistent algorithm of Turnbull [43] can be seen as a variant of (1.5).

In the case of panel count data, \( \theta = \mathbf{\Lambda} \). If all subjects have the same assessment times, then the data are complete in the sense that we observe the panel count within each \( J_k \) for all subjects. Let \( N^*_{ik} = \#\{\ell \mid X_{i\ell} \in J_k\} \). Instead of the likelihood (1.2), we consider the
following likelihood to formulate an EM algorithm:

\[ L(\Lambda) = \sum_{ik} Y_i(t_k) N_{ik}^* \log (\Lambda_k) - \sum_{ik} Y_i(t_k) \Lambda_k + \text{terms independent of } \Lambda. \]  

(1.6)

\( N_{ik}^* \)'s are never observed. Having conditioned on the observables at the E step and ignored terms that does not depend on \( \Lambda' \) in (1.6), we have

\[ Q(\Lambda' | \Lambda) = \sum_{ik} Y_i(t_k) \mathbf{E}_\Lambda [N_{ik}^* | \{N_{ij}\}] \log \Lambda'_k - \sum_{ik} Y_i(t_k) \Lambda'_k \]

\[ = \sum_{ijk} Y_i(t_k) N_{ij} \frac{T_{ijk} \Lambda_k}{\sum_\ell T_{ij\ell} \Lambda'_\ell} \log \Lambda'_k - \sum_{ik} Y_i(t_k) \Lambda'_k \]  

(1.7)

The last equation follows since, given \( N_{ij} \), \( N_{ik}^* \) has a multinomial distribution. To maximize \( Q(\Lambda' | \Lambda) \) by differentiating with respect to \( \Lambda' \) leads to an EM iteration, given by

\[ \hat{\Lambda}_{k+1} = \sum_{ij} \frac{Y_i(t_k) N_{ij} \frac{T_{ijk} \hat{\Lambda}_k}{\sum_\ell T_{ij\ell} \hat{\Lambda}_\ell}}{Y(t_k)} \text{ for } k = 1, \ldots, K. \]  

(1.8)

Note that the EM algorithm (1.8) coincides with the self-consistent algorithm formulated by Hu, Lagakos and Lockhart [21] for Poisson count data.

One considerable advantage when using an EM algorithm is that its implementation is relatively easy and straightforward regardless of the dimensionality of data. Moreover, if an initial estimate complies with the nonnegativity constraint, then estimates in all subsequent iterations automatically comply with the constraint. Nonetheless, the EM algorithm is not foolproof. Wu [47] shows that, if EM iterations converge to a fixed-point solution, the fixed-point solution is only a stationary point, not necessarily a NPMLE. As a result, one ought to use the EM algorithm with caution by better understanding the likelihood function or, at very least, initiating the algorithm with different starting values. In addition, the EM algorithm is known for its slow convergence, especially when the dimensionality of the parameter space is high. In the context of interval-censored data, one may reduce the
dimensionality by first identifying innermost intervals. Identifying the innermost interval is not difficult in one dimension, but searching for its equivalent in two or more dimensions is complicated. Maathuis [27] refers to the innermost-interval equivalent in two or more dimensions as a maximal intersection\(^1\) and proposes a height-map algorithm to identify all maximal intersections for multivariate interval-censored data.

The maximum likelihood method for nonparametric estimation does not always result in a sensible solution. As described in Vardi et al. [44] and Silverman et al. [38], NPME’s are often too rough to provide investigators with any insightful results in the context of image reconstruction. In addition, Vardi et al. [44] demonstrates that the estimate becomes less and less smooth as the number of EM iterations exceeds a given threshold. As a result, the authors suggested to terminate the EM algorithm after a fixed but arbitrary number of iterations, say 50, in order to obtain a smoother intensity estimate. Surely, it is not very satisfactory, and the estimate may be premature and far away from the NPME.

To provide a smoother estimate while improving the algorithmic convergence, Silverman et al. [38] “philosophically prefers to abandon the aim of finding maximum likelihood estimates by explicitly acknowledging the needs for a smoothing procedure.” The authors proposed an EMS algorithm, which adds an extra S (for smoothing) step to the usual E and M step, as an alternative. In spite of the fact that the EMS iteration converges faster than the EM counterpart and results in a smoother and more useful estimate, Silverman et al. [38] characterized the algorithm “ad hoc.”\(^2\) We, however, demonstrate that the EMS algorithm arises naturally from a local likelihood consideration, and we are able to formally motivate its use in the local likelihood framework.

Another approach to obtain a smoother estimate is to consider penalized likelihood methods (see, for example, Green and Silverman [16]). First suggested by Good and Gaskin [14], the penalty acts as a way to regularize the behaviour of the parameter of interest. In general,

---

\(^1\)Refer to Wong and Yu [46] for the formal definition of maximal intersection.

\(^2\)Nychka [28] shows the EMS algorithm is less \textit{ad hoc} by showing its relationship to penalized likelihood.
the penalized likelihood can be written as follows:

\[ L_p(\theta; \alpha) = L(\theta) - R_\alpha(\theta). \] (1.9)

For example, \( L(\theta) \) is replaced with (1.1) and \( \theta \) becomes \( p \) in the context of interval-censored data. When data are panel counts, \( L(\theta) \) is replaced with (1.2) and \( \theta = \Lambda \). The functional \( R_\alpha \) penalizes the roughness of \( \theta \), and the tuning parameter \( \alpha \) controls the trade-off between the accuracy and smoothness of the estimate. Because the penalty functional can also be interpreted as the logarithm of a prior density in the Bayesian framework, one may appropriately choose a functional form to reflect how he/she views the qualitative features of \( \theta \). When data are partially observed or missing, one can apply EM to penalized likelihood, or rather an EMP algorithm, to search for a nonparametric maximum penalized likelihood estimate (see Green [17] and Titterington [42]). Instead of \( Q(\theta \mid \theta') \), the EMP algorithm aims to maximize the following function with respect to \( \theta \):

\[ Q_p(\theta \mid \theta'; \alpha) = Q(\theta \mid \theta') - R_\alpha(\theta). \] (1.10)

Green [15] shows that EMP iterations converge faster than their EM counterpart by proving that the spectral radius of the EMP mapping is smaller than that of EM. Nonetheless, this article also demonstrates that the introduction of \( R_\alpha \) could drastically increase the complexity and difficulty of the optimization.

### 1.2.2 Local Likelihood

**Complete Data**

Since Tibshirani and Hastie [41] introduced local likelihood as a method to flexibly estimate the parameter of interest, the local likelihood method for density estimation had been ex-
explored and pioneered by Hjort and Jones [19] and Loader [25]. The authors considered to locally estimate the density at point \( t \) by maximizing the following local likelihood function:

\[
\mathcal{L}_t(f) = \sum_{i=1}^{n} K_h(T_i - t) \log f(T_i) - n \int_{\mathbb{R}} K_h(u - t) f(u) \, du,
\]

where \( K_h(z) = K(z/h)/h \) is usually a symmetrical kernel with \( \int K(z) \, dz = 1 \). In particular, a polynomial of order \( q \) is used to approximate \( \log f(u) \) in the neighborhood of \( t \), such as

\[
\log f(u) \approx P(u - t) = \sum_{j=0}^{q} a_j (u - t)^j.
\]

Let \( \mathbf{a} \) denote the collection of the polynomial coefficients. The aim is to find the value of \( \mathbf{a} \) that maximizes \( \mathcal{L}_t(\mathbf{a}) \), i.e.

\[
\hat{\mathbf{a}} = \arg\min \mathcal{L}_t(\mathbf{a}).
\]

A local likelihood density estimator is set to be

\[
\hat{f}(t) \equiv \exp(\hat{a}_0) = \frac{n^{-1} \sum_i K_h(T_i - t)}{\int_{\mathbb{R}} K_h(u - t) \exp \left\{ \sum_{j=1}^{q} \hat{a}_j (u - t) \right\} \, du} \quad \text{for each } t,
\]

where \( \hat{a}_0 \) is the leading term of \( \hat{\mathbf{a}} \). For example, in the locally constant case in which the polynomial is truncated at the leading term, a local likelihood density estimate is given by

\[
\hat{f}(t) = n^{-1} \sum_i K_h(T_i - t).
\]

The local likelihood density estimate in this case coincides with the usual kernel density estimate.

When the event process in time is inhomogeneous Poisson, Loader [26] proposes the
following local likelihood for intensity estimation:

\[
\mathcal{L}_t(\lambda) = \sum_{i\ell} Y_i(T_{i\ell}) K_h(T_{i\ell} - t) \log \lambda(T_{i\ell}) - \sum_i \int_0^{\infty} Y_i(u) K_h(u - t) \lambda(u) \, du.
\] (1.11)

When \( \log \lambda(u) \cong \mathcal{P}(u - t) \), \( \mathcal{L}_t(\lambda) \equiv \mathcal{L}_t(\mathbf{a}) \). An estimate of the polynomial coefficients is set to be the one that maximizes the local likelihood \( \mathcal{L}_t(\mathbf{a}) \). Denote the estimate again by \( \hat{\mathbf{a}} \). A local likelihood intensity estimate is then established to be

\[
\hat{\lambda}(t) \equiv \exp\{\hat{\mathbf{a}}_0\} = \frac{\sum_{i\ell} Y_i(T_{i\ell}) K_h(T_{i\ell} - t)}{\sum_i \int_0^{\infty} Y_i(u) K_h(u - t) \exp\{\sum_{m=1}^{q} \hat{a}_m(u - t)^m\} \, du}.
\]

For example, the local likelihood intensity estimate in the locally constant case given by

\[
\hat{\lambda}(t) = \frac{\sum_{i\ell} Y_i(T_{i\ell}) K_h(T_{i\ell} - t)}{\sum_i \int_0^{\infty} Y_i(u) K_h(u - t) \, du}
\] (1.12)

is also recognized as the edge-corrected intensity estimator proposed by Diggle [11]. Consequently, Diggle’s edge-corrected intensity estimator is embedded in the local likelihood framework. Note that the parameter \( \mathbf{a} \) in both density and intensity estimation is indexed by \( t \) although we do not explicitly indicate its dependency for the sake of simplicity.

**Interval-Censored Data**

There are two possible extensions of local likelihood methods when data are interval-censored. One is to apply an EM strategy to local likelihood. The other is to apply a local likelihood method to a pseudo dataset created by replacing interval-censored observations with their imputed values.

Braun, Duchesne and Stafford [6] proposes a class of density estimators when data are interval-censored or aggregated into bins by introducing local weights with respect to the locations and the widths of observed intervals. These local weights, in turn, induce a natural
consideration of an EM-like strategy applied to local likelihood, which is referred to as local EM in the literature (see Betensky et al. [4, 5] for details). The authors formulated a local EM algorithm to estimate a density by repeating the expectation (E) and maximization (M) step. Let \( \hat{f}^r \) denote the value of \( f \) after \( r \) iterations. The E-step calculates

\[
Q_t(a | \hat{f}^r) = \mathbb{E}_{\hat{f}^r} [L_t | T_i \in I_i] = n \sum_{i=1} K_h(T_i - t) P(T_i - t) - n \int_{\mathbb{R}} K_h(u - t) \exp \left\{ \sum_{j=1}^q \hat{a}_j^{r+1}(u - t)^j \right\} du. \tag{1.13}
\]

The M-step involves maximizing \( Q_t(a | \hat{f}^r) \) to obtain an updated estimate of \( a \) and \( f \). The cycle of the E- and M-step can be summarized by

\[
\hat{f}^{r+1}(t) \equiv \exp(\hat{a}_0^{r+1}) = n^{-1} \frac{\mathbb{E}_{\hat{f}^r} [K_h(T_i - t) | T_i \in I_i]}{\int_{\mathbb{R}} K_h(u - t) \exp \left\{ \sum_{j=1}^q \hat{a}_j^{r+1}(u - t)^j \right\} du}. \tag{1.14}
\]

The iteration continues until the iterated density estimates converge to a fixed-point solution \( \hat{f}^* \) such that

\[
\hat{f}^*(t) = n^{-1} \frac{\mathbb{E}_{\hat{f}^*} [K_h(T_i - t) | T_i \in I_i]}{\int_{\mathbb{R}} K_h(u - t) \exp \left\{ \sum_{j=1}^q \hat{a}_j^*(u - t)^j \right\} du} \quad \text{with } \hat{a}^* = a(\hat{f}^*).
\]

for all \( t \) in the support of the underlying density \( f \). For example, when \( P(u - t) = a_0 \), the local EM algorithm is given by

\[
\hat{f}^{r+1}(t) = n^{-1} \sum_i \mathbb{E}_{\hat{f}^r} [K_h(T_i - t) | T_i \in I_i]. \tag{1.15}
\]

Local EM is different from a typical EM algorithm. As for the local EM algorithm, the conditional expectation is computed with respect to the *global* infinite dimensional parameter \( f \) while the M step only *locally* updates the parameter \( f \). As such, the typical arguments concerning the convergence of the EM algorithm cannot be brought to bear, and whether a
fixed-point solution achieves some optimality criteria remained unclear.

Alternatively, we may consider multiple imputation (MI) methods in Herzog and Rubin [18] and Rubin [35] to analyze the interval-censored data. MI is a general model-based procedure to stochastically impute incomplete or missing data, thus manufacturing a surrogate dataset with pseudo complete observations. Tanner and Wong [39] explores the notion of MI to model the hazard in the presence of interval-censored data. The idea is to simulate the unobserved quantities conditioning on the observed intervals and treat these simulated values as if interval-censoring never took place. Each simulation creates a surrogate dataset. Since the surrogate dataset contains no censored data, it is straightforward to employ local likelihood methods for hazard estimation. Tanner and Wong [39] prescribes a general-purpose algorithm to estimate the underlying hazard with interval-censored data. The algorithm involves an inner and a outer loop, which is described as follows.

**Inner Loop**

I.1 Simulate unobserved quantities based upon a density to obtain a surrogate dataset.

I.2 Apply a local likelihood method to the surrogate dataset and obtain a local likelihood hazard estimate.

I.3 Repeat step I.1 and I.2 multiple times.

**Outer Loop**

O.1 Average the local likelihood hazard estimates that are results of the inner loop.

O.2 Repeat the inner loop with an updated density until the averaged local likelihood hazard estimates between outer iterations are sufficiently close.

This formulation has two advantages. First, the algorithm can be rapidly implemented using existing software. Second, the within- and between-imputation variations can be empirically
estimated using the formula of Herzog and Rubin [18] and Rubin [35]. This, in turn, quantifies the uncertainty introduced by imputation.

As a final remark, the MI method is related with the local EM algorithm of Braun, Duchesne and Stafford [6]. To see how the two approaches are related, consider density estimation in the locally constant case. Let \( \tilde{f}^r \) denote the \( r \)th iterate of \( f \) and \( \tilde{f}^r_m \) be a density estimate based on the \( m \)th surrogate dataset after \( r \) iterations. Denote the imputation of the \( i \)th observation in the \( m \)th surrogate dataset by \( T_{ri}^{(m)} \). Given the condition that \( T_i \in I_i, T_{mi}^r \) has a distribution on \( I_i \) with density proportional to \( \tilde{f}^r(t) \). Tanner and Wong’s algorithm sets the \( r \)th iterate of \( f \) to be

\[
\tilde{f}^{r+1}(t) = M^{-1} \sum_m \tilde{f}^r_m(t) \\
= M^{-1} \sum_m \left( n^{-1} \sum_i K_h(T_{mi}^r - t) \right) \\
= n^{-1} \sum_i \left( M^{-1} \sum_m K_h(T_{mi}^r - t) \right) \\
\approx n^{-1} \sum_i \mathbb{E}_{\tilde{f}^r} [K_h(T_i - t) \mid T_i \in I_i] 
\]

Consequently, the MI method of Tanner and Wong [39] is, in fact, an implementation of the local EM algorithm of Betensky et al. [4].

**1.2.3 Piecewise Constant Approximation**

Let \( \psi \) be an absolutely integrable function on \( \mathcal{M} \). Given the partition \( \mathcal{J} \), a piecewise constant function \( g \) is defined to be

\[
g_{\psi}(t; \mathcal{J}) = \sum_j \mathcal{I}(t \in J_j) \|J_j\|^{-1} \int_{J_j} \psi(u) \, du,
\]
where $\| \cdot \|$ returns the Lebesgue measure of the pixel. In this context, $\| J_k \|$ returns the length of the $k$th pixel. Royden [34] formally refers to the piecewise constant function as the $\mathcal{J}$-approximant of $\psi$. Advantages of working with this piecewise constant approximation $g_\psi$ are that it is of finite-element and it converges to $\psi$ in $\mathcal{L}_p$ as $\max_j \| J_j \| \downarrow 0$ and $K \to \infty$ for $1 \leq p < \infty$.

1.3 Overview

In the thesis, we explore the local EM paradigm of Braun, Duchesne and Stafford [6] in the realm of point processes. We first consider density estimation for one-jump point processes in Chapter 2. When data are interval-censored, we propose simplifying the local EM algorithm (1.14) by replacing $\hat{f}^r$ with its $\mathcal{J}$-approximant at each iteration. The piecewise constant approximation leads to the collapse of the local EM into an EMS algorithm and reveals an unexpected linkage between local EM and the EMS algorithm. The EMS algorithm was first introduced in Silverman et al. [38] as an ad hoc method to obtain a smooth estimate by adding an extra s step to the usual EM algorithm. However, our local EM implementation suggests otherwise as the EMS algorithm arises naturally from a local likelihood consideration. It is noteworthy that the implementation of the local EM algorithm in Braun, Duchesne and Stafford [6] is also an EMS algorithm although the authors were not aware of it.

In Chapter 3, we focus on a more general temporal point process which allows multiple events for each subject. Here, we assume the process is inhomogeneous Poisson with intensity $\lambda$. When a subject’s event times are interval censored by a visit process, we follow the same strategy suggested by Braun, Duchesne and Stafford [6] and formulate a local EM method for intensity estimation. We set up a local EM algorithm and carefully derive a rather simple expression for the conditional expection at the E-step. Furthermore, we implement the local EM algorithm by replacing the $r$th iterate of $\lambda$ with its $\mathcal{J}$-approximant. It follows that the
local EM, again, collapses into an EMS algorithm. Through the use of the piecewise constant approximation, we are able to establish a previously unknown connection between local EM and the EMS algorithm. Since local likelihood in this context motivates the use of EMS algorithms, the EMS method for intensity estimation should no longer be considered as *ad hoc*. We apply the proposed local EM method to analyze the panel count data in Thall and Lachin [40]. While the result of our analysis is similar to the one in Thall and Lachin [40], the local EM method results in a much smoother intensity estimate than the one presented in Thall and Lachin [40]. Our analysis further suggests that the intensity functions between the treatment and placebo groups are qualitatively different in terms of the number of peaks and troughs. A simulation study is conducted to compare the proposed local likelihood intensity estimate with three different smooth nonparametric estimates (SNE’s). Since a NPMLE is only unique up to an equivalence class, we obtain the three SNE’s by directly smoothing a NPMLE with its masses placed at the left-end, middle and right-end points of the $J_k$’s. The empirical evidence suggests that the local EM requires fewer iterations than the EM algorithm to meet convergence criteria, and the local likelihood estimate achieves a smaller mean integrated squared error than the SNE’s.

Following the methodological development focusing on temporal processes, we consider extending the local EM method to analyze spatially aggregated data in Chapter 4. The development in this chapter differs from that in Chapter 3 because of the introduction of an offset surface. In particular, the intensity function is now parameterized as a product of a known offset surface and a spatially smooth relative risk function. Our aim is to estimate the risk function. Here, we establish a local EM algorithm for risk estimation and implement it as an EMS algorithm. We then apply the proposed local EM method to solve a problem in disease mapping. Here, disease incidences for each gender and age group are longitudinally collected. In compliance with the privacy policy, the data are spatially aggregated over census tracts and reported as regional counts. The primary issue in the data analysis is
that the census tract boundaries vary in time. Because of the time-varying boundaries, it was not clear how one may model the spatio-temporal structure of the disease process. We consider local EM to address this issue. Not only can the proposed local EM method easily accommodate the time-varying boundaries, but it can also account for covariate information as well as regional populations. In the situation in which there is only one map, the local EM algorithm iterates only once and the local likelihood risk estimator is recognized as the one suggested by Brillinger [7, 8, 9]. In conclusion, we obtain a local likelihood risk estimate by combining multiple disease maps while taking into account the spatial and temporal variations in regional population by gender and age group. As demonstrated in this application, the local EM method can also be thought as an extension of the image reconstruction technique of Silverman et al. [38] in a spatio-temporal setting.

We have shown that local likelihood formally motivates the use of EMS algorithms when data are either interval-censored or spatially aggregated. In Chapter 5, we study the optimality and the convergence of a local EM algorithm. Not only can the EMS algorithm guide the use as well as implementation of local EM, but it also provides a set of theoretical tools to establish the convergence and optimality of the local EM algorithm. Latham [24] shows the existence and uniqueness of the fixed-point solution of the EMS implementation, and Nychka [28] shows that a modified EMS algorithm maximizes the penalized likelihood (1.9). The former enables us to conclude that the fixed-point solution of the local EM algorithm is unique, whereas the latter suggests the role of local EM. In §5.1, we first show that employing a specific kernel in the local likelihood leads to a version of the local EM algorithm, the EMS implementation of which is equivalent to Nychka’s modified EMS algorithm. Thereby, we refer to the kernel as an equivalent kernel. Next, we show that the $r$th EMS iterate converges in the $L^1$ norm to its local EM counterpart as the pixel size shrinks toward zero. The $L^1$ convergence also holds with the modified EMS iterate. Finally, a careful examination of the limiting behaviour of the penalty leads us to speculate local EM penalizes the nonparametric
likelihood for departures of the target function from a class of eigenfunctions. Based on these results, we may conceptually pair local EM and penalized likelihood in the manner analogous to the pair of EM and likelihood. Moreover, by suggesting this role of local EM, we should no longer consider the local EM algorithm as a black box, and we can now interpret the local likelihood estimate in the context of interval-censored or aggregated data as a nonparametric maximum penalized likelihood estimate.

In the final concluding chapter, a comparison with the EMS method of Silverman et al. [38] provides us with an even deeper insight to the relationship between local EM and EMS algorithms. The EMS method in Silverman et al. [38] concerns issues related to data arising from a particle emission process modelled by a Fredholm integral equation of the first kind. The data that Silverman et al. considered are indirectly observed because the region where a particle is emitted can only be probabilistically described. In contrast to the context considered in this thesis, the data are directly observed in the sense that observed region is exactly the one where events occur. This observation leads to the suggestion that the local EM method can be well extended to analyze indirectly observed or mismeasured data, which is the direction of future work.
Chapter 2

One-Jump Point Processes

Following the literature review of local EM in Chapter 1, we consider to approximate the conditional expectation in (1.14) with respect to a piecewise constant function. The piecewise constant approximation leads to the collapse of the local EM (1.14) to an EMS algorithm. In addition, we discover an exciting connection between local EM and the EMS algorithm through the use of this piecewise constant approximation. The connection was not previously known, and it has significant implications. Not only can the EMS serve as a convenient and rapid implementation of local EM algorithms, but the implemention also suggests that the EMS algorithm is not ad hoc as previously thought in the literature. We demonstrate that the EMS algorithm arises naturally from a local likelihood consideration.

We start with a simple one-jump point process, such as a survival process. In the presence of interval-censoring, the ith subject’s event time $T_i$ is reported as an interval, $I_i = (L_i, R_i]$. $L_i$ is the lastest visit time prior to $T_i$, and $R_i$ is the earliest visit time after $T_i$. If we assume $\Pr(T_i < \infty) = 1$, a right-censored observation implies that $I_i = (L_i, R_i]$ with $R_i = \infty$. In either scenario, $T_i$’s are considered to be interval-censored. When $T_i$’s are i.i.d. with density $f$, the data consist of $n$ independent intervals, some of which may vary in length or overlap. Braun, Duchesne and Stafford [6] applied local EM to density estimation in the context of
interval-censored data and proposed a class of local likelihood density estimators that satisfy the following equation:

\[
\hat{f}^*(t) = \frac{n^{-1} \sum_i E_j \left[ K_h(T_i - t) \mid T_i \in I_i \right]}{\int_{\mathbb{R}} K_h(u - t) \exp \left\{ \sum_{j=1}^{q} \hat{a}_{j}^r(u - t) \right\} du}.
\]

The authors formulated the local EM algorithm (1.14) to obtain \( \hat{f}^*(t) \) for all \( t \in \mathbb{R} \) or \( \mathbb{R}^+ \).

Since the integrals in (1.14) involve the infinite dimensional parameter \( f \), we must rely on a quadrature rule, such as the trapezoid or Simpson’s rule, to numerically evaluate these integrals when we implement the local EM algorithm. Different quadrature rules lead to different versions of local EM implementations. For example, Braun, Duchesne and Stafford [6] uses the midpoint quadrature rule with an equally spaced mesh to implement the local EM algorithm (1.14). It will be shown in §2.3 that the author’s implementation is, in fact, an EMS algorithm although they were not aware of the connection between local EM and EMS algorithms.

There are 4 sections in Chapter 2. In §2.1, we start with the simple case of locally constant and implement the local EM algorithm (1.15) using the \( J \)-approximant of the \( r \)th iterate of \( f \). Such an implementation results in the collapse of the local EM to an EMS algorithm and allows us to discover an interesting relationship between local EM and the EMS algorithm. This relationship was not known in the literature, and it suggests that EMS algorithms are not \( ad \; hoc \) in this context. In §2.2, we generalize the development to local likelihood functions with higher order polynomials and show corresponding local EM algorithms, such as (1.14), can also be implemented as an EMS algorithm. The s-step, in this case, \( adapt \) itself in the sense that the smoother becomes dependent on the most recent iterate of \( f \). Thereby, we refer to this implementation as an adaptive EMS. In §2.3, we show that the partition configuration is rather arbitrary although it may depend on the data in a natural way. The freedom to choose a partition has significant practical and theoretical
implications, as we will show in the next three chapters. Finally, this chapter ends with an example of bivariate interval-censored data. The example illustrates how one may use the local EM algorithm to guide the estimation of the distribution function as well as to overcome difficulties that arise from the maximum likelihood method.

2.1 EMS Algorithm

Consider first the locally constant case since details are easier to follow. While the local EM algorithm (1.15) involves explicit numerical evaluations of conditional expectations with respect to $\hat{f}_r$, we propose to calculate the conditional expectation using the $J$-approximant of $\hat{f}_r$. Recall that $\hat{p}_r = \int_{J_r} \hat{f}_r(t) \, dt$ and $\mathcal{I}_{i\ell} = \mathcal{I}(J_\ell \subseteq I_i)$. Let $\hat{g}_r$ denote $g_{f_r}(t; \mathcal{J})$. It follows that

$$E_{\hat{f}_r} \left[ K_h(T_i - t) \mid T_i \in I_i \right] \approx E_{\hat{g}_r} \left[ K_h(T_i - t) \mid T_i \in I_i \right]$$

$$= \int_{I_i} K_h(u-t) \sum_j \mathcal{I}(u \in J_j) \|J_j\|^{-1} \hat{p}_j \, du$$

$$= \frac{\sum_j \int_{I_i} K_h(u-t) \mathcal{I}(u \in J_j) \, du \|J_j\|^{-1} \hat{p}_j}{\sum_k \int_{I_i} \mathcal{I}(v \in J_k) \, dv \|J_k\|^{-1} \hat{p}_k}$$

$$= \frac{\sum_j \int_{I_i} K_h(u-t) \mathcal{I}(u \in J_j) \, du \|J_j\|^{-1} \hat{p}_j}{\sum_k \int_{I_i} \mathcal{I}(v \in J_k) \, dv \|J_k\|^{-1} \hat{p}_k}$$

$$= \sum_j \|J_j\|^{-1} \int_{J_j} K_h(u-t) \, du \frac{\mathcal{I}_{ij} \hat{p}_j}{\sum_k \mathcal{I}_{ik} \hat{p}_k} \hat{p}_k$$

By substituting (2.1) into the local EM algorithm (1.15), we obtain the $(r+1)$st iterate of $f$, which is given by

$$\hat{f}_{r+1}(t) = n^{-1} \sum_{ij} \|J_j\|^{-1} \int_{J_j} K_h(u-t) \, du \frac{\mathcal{I}_{ij} \hat{p}_j}{\sum_k \mathcal{I}_{ik} \hat{p}_k}. $$
To update pixel probabilities by integrating $\hat{f}^{r+1}(t)$ over the $J_\ell$’s leads to an iterative algorithm for $p$ as follows:

$$\hat{p}_t^{r+1} = \int_{J_t} \hat{f}^{r+1}(t) \, dt = \int_{J_t} \left( n^{-1} \sum_{ij} |J_j|^{-1} \int_{J_j} K_h(u - t) \, du \frac{I_{ij} \hat{p}_j^r}{\sum_k I_{ik} \hat{p}_k^r} \right) \, dt$$

$$= \sum_j \|J_j\|^{-1} \int_{J_t} \int_{J_j} K_h(u - t) \, du \, dt \left( n^{-1} \sum_i \frac{I_{ij} \hat{p}_j^r}{\sum_k I_{ik} \hat{p}_k^r} \right)$$

(2.2)

The iteration in (2.2) can also be expressed in terms of matrices as

$$\hat{\mathbf{p}}^{r+1} = \mathcal{M}(\hat{\mathbf{p}}^r) \mathcal{K}_h.$$  

(2.3)

Here, $\mathcal{M}(\hat{\mathbf{p}}^r)$ is a row vector with the $j$th component equal to

$$\frac{1}{n} \sum_i \frac{I_{ij} \hat{p}_j^r}{\sum_k I_{ik} \hat{p}_k^r},$$

and $\mathcal{K}_h$ is a smoothing matrix with

$$[\mathcal{K}_h]_{j\ell} = \|J_j\|^{-1} \int_{J_t} \int_{J_j} K_h(u - t) \, du \, dt.$$

Since $\mathcal{M}(\mathbf{p})$ is recognized as the EM mapping in (1.5), this implementation of the local EM algorithm involves an expectation, maximization and smoothing step. That is, implementing the local EM algorithm using the $J$-approximant of $\hat{f}^r$ explicitly results in an EMS algorithm. Subsequently, we reveal an unknown but significant connection between local EM and the EMS algorithm. While the EMS method was previously described as “ad hoc” by Silverman et al. [38], local likelihood suggests otherwise. We have shown that the EMS algorithm naturally arises from the local EM framework. As a final note, Pan and Chappell [31] and Pan [30] also consider the EMS algorithm to obtain a smoothed nonparametric density estimate. However, the papers have no discussion of local likelihood, and the choice of the smoothing matrix is
Remarks:

1. If the EMS iteration in (2.3) converges to a fixed-point solution, denoted by \( \hat{p}^* \), a local likelihood density estimate is set to be

\[
\hat{f}^*(t) = \sum_j \|J_j\|^{-1} \int_{J_j} K_h(u - t) \, du \left( n^{-1} \sum_i \frac{I_{ij} \hat{p}_j^*}{\sum_\ell I_{i\ell} \hat{p}_\ell^*} \right),
\]

where \( \hat{p}_j^* = \int_{J_j} \hat{f}^*(u) \, du \).

2. When none of the observations is right-censored, \(-\infty < L_i \leq R_i < \infty\) for all \(i\). All of the intervals in \( J \) are of finite length. The entries of \( K_h \) can be easily calculated. When one of the \( T_i \)'s is right-censored, some care is required. Since some pixels in \( J \) are of infinite length, we need to calculate the double integral that involve an interval of infinite length with care. Let \( J_K = (t_{K-1}, t_K] \). We consider the behaviour of the integral as \( t_K \to \infty \).

\[
\lim_{t_K \to \infty} \|J_K\|^{-1} \int_{J_{t_K}} \int_{J_K} K_h(u - x) \, du \, dx = \int_{J_L} \lim_{t_K \to \infty} \int_{t_{K-1}}^{t_K} K_h(u - x) \, du \, dx = 0
\]

\[
\lim_{t_K \to \infty} \|J_K\|^{-1} \int_{J_K} \int_{J_K} K_h(u - x) \, du \, dx = 1 - \|J_{t}\|^{-1} \int_{J_L} \int_{-\infty}^{t_{K-1}} K_h(u - x) \, dx \, du
\]

\[
\lim_{t_K \to \infty} \|J_K\|^{-1} \int_{J_K} \int_{J_K} K_h(u - x) \, du \, dx = 1
\]

It follows that

\[
K_h = \begin{bmatrix} \mathcal{K}_h & d \\ 0 & 1 \end{bmatrix},
\]
The matrix $\mathbf{K}'_h$ is a $(K-1) \times (K-1)$ symmetric matrix, the column vector $\mathbf{d}$ is of length $K-1$, and $\mathbf{0}$ is a row zero vector of length $K-1$. Note that elements in $\mathbf{d}$ are of the form

$$1 - \|J_\ell\|^{-1} \int_{J_\ell} \int_{-\infty}^{t_{k-1}} K_h(u-x) \, dx \, du$$

and converge to 0 as $h \searrow 0$.

3. As a result of the previous remark and the following lemma, $\mathbf{K}_h$ converges to the identity matrix, and the local EM (2.3) becomes Turnbull’s EM algorithm [43]. This result is similar to that of Theorem 1 in Braun et al. [6].

**Lemma 2.1.** As $h \rightarrow 0$,

$$\int_{J_k} \int_{J_k} K_h(u-t) \, du \, dt \rightarrow \|J_k\| \delta_{tk}, \text{ where } \delta_{tk} = \begin{cases} 1 & \text{if } \ell = k \\ 0 & \text{otherwise} \end{cases}$$

**Proof.**

$$\int_{J_k} K_h(u-t) \, du = \int_{t_{k-1}}^{t_k} \frac{1}{h} K \left( \frac{u-t}{h} \right) \, du = \int_{t_{k-1}}^{t_k-1} \frac{1}{h} K(z) \, dz$$

On one hand, if $t \notin J_k$, then $\text{sign}\left(\frac{t_k-t}{h}\right) = \text{sign}\left(\frac{t_{k-1}-t}{h}\right)$. As $h \rightarrow 0$, the upper and lower bounds both approach infinity in the same direction, implying that $\int_{J_k} K_h(u-t) \, du \rightarrow 0$. On the other hand, if $x \in J_k$, then $\text{sign}\left(\frac{t_k-t}{h}\right)$ and $\text{sign}\left(\frac{t_{k-1}-t}{h}\right)$ are positive and negative, respectively. As $h \rightarrow 0$, the upper bound approaches positive infinity, and the lower bound approaches negative infinity, implying that $\int_{J_k} K_h(u-t) \, du \rightarrow \mathcal{I}(t \in J_k)$. Hence,

$$\lim_{h \rightarrow 0} \int_{J_k} \int_{J_k} K_h(u-t) \, du \, dt = \int_{J_\ell} \lim_{h \rightarrow 0} \int_{J_k} K_h(u-t) \, du \, dt = \int_{J_\ell} \mathcal{I}(t \in J_k) \, dt = \|J_\ell\| \delta_{tk}$$
4. When data are in the form of a histogram, \( J \equiv \{I_i\} \). Let \( N_j \) be the number of observations falling in the \( j \)th bin and \( \sum_j N_j = n \). The local EM algorithm only iterates once because \( M(p) \) is free of \( p \) and equal to \([N_1/n, \ldots, N_K/n]\). In this case, we set a local likelihood density estimate to be

\[
\hat{f}(t) = \sum_k \frac{(N_k/n)^{-1} \int_{J_k} K_h(u-t) \, du}{\int_{\mathbb{R}} K_h(u-t) \exp \left\{ \sum_{j=1}^q \hat{a}_j^r(\hat{p}^*)^j(u-t)^j \right\} \, du}.
\]

This type of density estimates, obtained by smoothing histograms, has been extensively studied by Bellhouse and Stafford [3], Scott [37] and Jones [22]. Moreover, the integrated kernel weight

\[
\int K_h(u-t) \, du
\]

has been suggested under various settings, for example, see Jones [22] in the context of the kernel density estimation and Gasser and Müller [13] in the regression context.

### 2.2 Adaptive EMS Algorithm

In general, when the polynomial \( P(z) \) is of order \( q \), replacing \( E_{jr} \left[ K_h(T_i - t) \mid T_i \in I_i \right] \) with \( E_{jr} \left[ K_h(T_i - t) \mid T_i \in I_i \right] \) in (1.14) results in the \((r+1)\)st iterate of \( f \) being

\[
\hat{f}^{r+1}(t) = \sum_k \left( \frac{\|J_k\|^{-1} \int_{J_k} K_h(u-t) \, du}{\int_{\mathbb{R}} K_h(u-t) \exp \left\{ \sum_{j=1}^q \hat{a}_j^{r+1}(u-t)^j \right\} \, du} \right) \left( n^{-1} \sum_i \frac{\mathcal{I}_{ij}\hat{p}_{j}^{r}}{\sum_{\ell} \mathcal{I}_{i\ell}\hat{p}_{\ell}^{r}} \right).
\]

Integrating \( \hat{f}^{r+1}(t) \) over the \( J_k \)'s leads to an iterative algorithm,

\[
\hat{p}_{\ell}^{r+1} = \sum_k \left( \int_{J_k} \frac{\|J_k\|^{-1} \int_{J_k} K_h(u-t) \, du}{\int_{\mathbb{R}} K_h(u-t) \exp \left\{ \sum_{j=1}^q \hat{a}_j^{r+1}(u-t)^j \right\} \, du} \, dt \right) \left( n^{-1} \sum_i \frac{\mathcal{I}_{ij}\hat{p}_{j}^{r}}{\sum_{\ell} \mathcal{I}_{i\ell}\hat{p}_{\ell}^{r}} \right).
\]
Since $\hat{a}^{r+1} = a(\hat{p}^r)$, we can express the iteration above in terms of matrices as

$$\hat{p}^{r+1} = M(\hat{p}^r)K_h(\hat{p}^r). \quad (2.4)$$

It can be seen that $M(\hat{p}^r)$ is the same as the EM mapping in (2.3); however, $K_h$ now depends on $\hat{p}^r$ through $\hat{a}^{r+1}$, and its $j\ell$th entry is given by

$$[K_h(\hat{p}^r)]_{j\ell} = \int_J \frac{||J_k||^{-1} \int K_h(u - t) du}{\int R K_h(u - t) \exp \left\{ \sum_{j=1}^q a_j(\hat{p}^r)(u - t)^j \right\} du} \, dt.$$

Because of the dependency, we refer to (2.4) as an adaptive EM implementation of the local EM algorithm. If the iteration in (2.4) converges, $\hat{p}^* = M(\hat{p}^*)K_h(\hat{p}^*)$. A local likelihood density estimate is set to be

$$\hat{f}^*(t) = \sum_k \left( \frac{||J_k||^{-1} \int K_h(u - t) du}{\int R K_h(u - t) \exp \left\{ \sum_{j=1}^q a_j(\hat{p}^*)(u - t)^j \right\} du} \right) \left( n^{-1} \sum_i \sum_j m_i \hat{p}_{ij}^* \right).$$

Remarks:

1. Since $\int R K_h(u - t) \exp \left\{ \sum_{j=1}^q a_j(\hat{p}^*)(u - t)^j \right\} du \to 1$ as $h \to 0$, the adaptive smoothing matrix approaches the identify matrix. Consequently, (2.4) becomes Turnbull’s EM algorithm [43].

2. The local likelihood estimate $\hat{f}^*$ is a non-negative function whenever $K(z)$ and $\hat{f}^0$ are non-negative. In the locally constant case,

$$\int_R \hat{f}^*(t) \, dt = n^{-1} \int R \int J_k K_h(u - t) du \, dt \sum_i \sum_j m_i \hat{p}_{ij}^* = n^{-1} \sum_i \sum_j m_i \hat{p}_{ij}^* = 1.$$

Thereby, $\hat{f}^*$ is indeed a valid density estimate. When the order of the approximating
polynomial is greater than 0, the local likelihood density estimate has to be renormalized.

2.3 Partition Configurations

While we use the partition that depends on data in the previous sections, this choice is, in fact, rather arbitrary. Suppose that $T_i$’s are i.i.d. with density $f$, the support of which is the positive real line. If we break the support of $f$ into $K$ intervals, denoted by $R_k$’s, in an arbitrary fashion, then the probability that any $T_i$ falls into the $k$th interval is simply equal to $p_k = \int_{R_k} f(u) \, du$. Let $Y_i = (I(T_i \in J_1), \ldots, I(T_i \in J_K)$. It follows that $Y_i$’s are also i.i.d. Each of them has a multinomial distribution with the parameters of 1 and $p_k$’s. The likelihood function, omitting parts that do not depend on $p_k$’s, is given by

$$\mathcal{L}(p) = \sum_{i,k} I(T_i \in R_k) \log p_k \quad \text{with } p_k \geq 0 \text{ for all } k \text{ and } \sum_k p_k = 1.$$ 

The likelihood function above is very much similar to (1.6), but $p_k$’s are defined differently. Here, the $p_k$ is defined as the integral of $f$ over the $R_k$; whereas, the one in (1.6) is the integral of $f$ over the $J_k$. Let $I_{ij} = I(R_j \subseteq I_i)$. Following the literature review, applying EM to the above likelihood results in an EM algorithm, written as

$$p_j^{r+1} = n^{-1} \sum_i I_{ij}p_j^r$$

for all $j$.

Evidently, formulating EM algorithms does not require any particular partition configuration. This also holds for the local EM algorithm owing to its EMS implementation. The property offers great computational convenience by allowing us to choose a configuration such that $K_h$ can be easily set up. For instance, we use a partition of squared pixels to set up the smoothing matrix for the disease mapping problem in Chapter 4 since $J_k$’s in this problem
are two-dimensional regions with irregular shape.

Let $\mathcal{C} = \{C_1, \ldots, C_K\}$ be a partition in which $|C_j| = |\mathcal{C}|$ for all $j = 1, \ldots, K$. Given $\mathcal{C}$, a local EM algorithm for density estimation is given by

$$\hat{p}_k^{r+1} = \sum_j |\mathcal{C}|^{-1} \int_{C_k} \int_{C_j} K_h(u - x) du dv \left( n^{-1} \sum_i \sum_{\ell} I_{ij} \hat{p}_j^r \right).$$

(2.5)

Note that $I_{ik} = I(C_k \subseteq I_i)$ and $\hat{p}_j^r = \int_{C_j} \hat{f}^r(u) du$. In this case, $K_h$ is a symmetric Toeplitz matrix with only $K$ distinct entries because the double integral depends on only the absolute difference between $k$ and $j$. In other words, if $|k - j| = |k' - j'|$, then

$$\int_{C_k} \int_{C_j} K_h(u - x) du dv = \int_{C_{k'}} \int_{C_{j'}} K_h(u - x) du dv$$

Thereby, we only have to calculate $K$, instead of $K(K + 1)/2$, double integrals.

It is noteworthy that the implementation of the local EM algorithm in Braun, Duchesne and Stafford [6] is, in fact, a version of the EMS algorithm with $\mathcal{C}$ although the authors were not aware of the connection between local EM and the EMS algorithm. To see this, consider partitioning a finite interval $\mathcal{S}$ such that $\mathcal{M} \subseteq \mathcal{S}$ with a set of equally-spaced points, $t_j$’s such that $0 = t_0 < t_1 < \ldots < t_K$. Let $C_j = [t_{j-1}, t_j)$ with $|C_j| = |C|$ for all $j$. Note that $|C| = |\mathcal{S}|/K$. Denote the midpoint of $C_j$ by $m_j$. Then substituting the following approximation

$$\hat{p}_k^r \approx \hat{f}^r(m_k)|\mathcal{C}| \quad \text{and} \quad \int_{C_k} \int_{C_j} K_h(u - x) du dv \approx K_h(m_j - m_k)|\mathcal{C}|^2$$

into (2.5) results in

$$\hat{f}^r(m_k)|\mathcal{C}| = \sum_j |\mathcal{C}|^{-1} K_h(m_j - m_k)|\mathcal{C}|^2 \left( n^{-1} \sum_i \sum_{\ell} I_{ij} \hat{f}^r(m_j) \frac{|\mathcal{C}|}{\sum_{\ell} |\mathcal{C}|} \right).$$
The cancellation of the $\|C\|$’s gives the expression (15) of Braun, Duchesne and Stafford [6]. A similar result also holds in the locally linear case.

In summary, the EMS implementation does not demand a particular partition configuration. This feature can not only reduce the amount of computation time but also eliminate complications brought by the use of the data-dependent partition.Unless specified otherwise, let $\mathcal{C}$ denote the partition with pixels of equal measure and $\mathcal{J}$ be the data-dependent partition in the rest of the thesis.

**Remark:** Recall that only the pixel probabilities induced by $\mathcal{J}$ are identifiable. Should there be multiple $\mathcal{C}_\ell$’s in the $\mathcal{J}_k$, then the EM step of (2.5) essentially re-distributes the probability weight of $\mathcal{J}_k$ to the $\mathcal{C}_\ell$’s contained in $\mathcal{J}_k$ in the accordance to the ratio of $\|\mathcal{C}_\ell\|$ and $\|\mathcal{J}_k\|$.

### 2.4 Application: Bivariate Interval-Censored Data

Suppose that $T_i$’s are i.i.d. bivariate random variables in $\mathfrak{M} = \mathbb{R}^2$. When data are doubly interval-censored, the maximum likelihood estimation of the joint distribution is complex. Using an EM algorithm to search for the maximum likelihood estimate requires the prior identification of maximal intersections, which are analogous to innermost intervals in one dimension. Finding all maximal intersection, however, is complicated and requires specialized algorithms, such as the height-map algorithm in Maathuis [27]. In this section, we demonstrate how local EM may ease these complexities and thus deserves further exploration. For example, the EMS implementation of local EM does not require the use of the height-map algorithm of Maathuis [27] to locate all maximal intersections. Moreover, the local likelihood estimate is unique while the NPMLE is not.

Consider a hypothetical example of bivariate interval-censored data, shown in Figure 2.1. The data consists of eight observations, represented by four horizontal and four vertical rectangles. Each rectangle indicates the region where $T_i$ lies. Overlaying these observations
forms a partition of eighty-one unit squares, and the sixteen intersections of these rectangles are the maximal intersections. Recall that NPMLE’s only assign probability to maximal intersections. As shown in Figure 2.2, a uniform weight of 1/16 on all maximal intersections in Figure 2.3(a), a weight of 1/4 on the positive diagonal intersections in Figure 2.3(b), and a weight of 1/4 on the negative diagonal intersections in Figure 2.3(c) all maximize the nonparametric likelihood (1.1). Consequently, the solution to which the EM iteration converges depends on how an initial value is chosen. However, it is not so when local EM is used. As we will show in §5.2, the local likelihood estimate is unique.

Figure 2.4(a) shows a density estimate when we use the local EM algorithm with a radially symmetrical Gaussian kernel. Here, the local EM iteration will always converge to the fixed-point solution that favours the NPMLE with uniform weighting even when initial values strongly favour the two diagonal solutions. Figure 2.4(b) gives a density estimate using a kernel with elliptical contours, such that bandwidth values are 1.5 and .15 in the respective x- and y-directions and rotated by 45 degrees. In this case, the local EM iteration always converges to the fixed-point solution that favours the positive diagonal NPMLE. Similarly, when the same kernel is rotated by -45 degrees, Figure 2.4(c) shows the local EM iteration always converges to the solution that favours the negative diagonal. As we will show in §5.1.3, this kind of behaviour can be explained using the Bayesian interpretation of penalized likelihood.
Figure 2.1: Each shaded rectangle represents a bivariate interval-censored observations. The sixteen intersections of the shaded rectangles are referred to as maximal intersections.

Figure 2.2: All three density estimates maximize the nonparametric likelihood (1.1).
(a) A density estimate using a radially symmetrical kernel.

(b) Kernel with elliptical contours with positive-sloped major axis.

(c) Kernel with elliptical contours with negative-sloped major axis.

Figure 2.3: Three local EM density estimates with different kernel functions.
Chapter 3

Temporal Point Processes

In this chapter, we extend the notion of local EM to the situation in which a unidimensional inhomogeneous Poisson process manifests itself in the form of indicators, \( \{I(T_i \in I_{ij})\} \). In particular, we extend the local EM paradigm of Braun, Duchesne and Stafford [6] to estimate the intensity of the process when event times are interval-censored. Here, multiple events are allowed. We proceed with the extension by first deriving the conditional expectation at the E-step using two well known properties of inhomogeneous Poisson processes. Next, we propose to approximate the conditional expectation using a piecewise constant function at the E-step. The piecewise constant approximation reveals an interesting and unknown connection between local EM and EMS algorithms. Not only does this connection guide how one may implement the local EM as an EMS algorithm, but it also implies that the EMS algorithm is not ad hoc from a local likelihood consideration. We then apply the proposed local EM method to analyze the panel count data studied in Thall and Lachin [40]. While our result is consistent with the one in Thall and Lachin [40], the local likelihood intensity estimates are considerably smoother than the ones of Thall and Lachin [40]. Moreover, our analysis suggests that the intensity function of the treatment group is different from that of the placebo group in terms of numbers of peaks and troughs.
Finally, the chapter ends with a simulation study, which is conducted to assess the empirical performance of the local likelihood intensity estimator for panel count data. Based on simulated samples, we obtain 300 local likelihood intensity estimates in both locally constant and linear cases for each bandwidth value. In addition, we compute a NPMLE using an EM algorithm for each sample. We further smooth the NPMLE with a kernel by placing its masses at the middle, left- and right-end points of the $J_k$’s and obtain three slightly different versions of smoothed nonparametric estimates (SNE’s). We then compare the local likelihood intensity estimates with the three SNE’s on the basis of the mean integrated squared error (MISE). The study result seems promising. Among all intensity estimators based on panel count data, the proposed local likelihood estimator has the lowest MISE. Moreover, it achieves the lowest MISE with a smaller bandwidth value.

### 3.1 Local EM Algorithm

Recall the local likelihood for intensity estimation is given by

$$
\mathcal{L}_t(a) = \sum_{i\ell} Y_i(T_{i\ell})K_h(T_{i\ell} - t)P(T_{i\ell} - t) - \sum_i \int_0^\infty Y_i(u)K_h(u - t)\exp\{P(u - t)\} \, du.
$$

Following Braun, Duchesne and Stafford [6], we apply a similar EM-like strategy to the above local likelihood when $T_{i\ell}$’s are interval-censored. At the E-step, we calculate the expectation of the local likelihood function conditional on the observed indicators with respect to the previous estimate of $\lambda$.

$$
Q_t(a \mid \hat{\lambda}^r) = \mathbb{E}_{\hat{\lambda}^r}[\mathcal{L}_t(a) \mid \{I(T_{i\ell} \in I_{ij})\}] \\
= \mathbb{E}_{\hat{\lambda}^r}\left[\sum_{i\ell} Y_i(T_{i\ell})K_h(T_{i\ell} - t)P(T_{i\ell} - t) \mid \{I(T_{i\ell} \in I_{ij})\}\right] \\
- \sum_i \int_{\mathbb{R}} Y_i(u)K_h(u - t)\exp\{P(u - t)\} \, du. \quad (3.1)
$$
The conditional expectation can be simplified using two properties of inhomogeneous Poisson processes\(^1\), which are summarized below. Drop the subscript \(i\) for the time being. Let \(N(A)\) denote the number of events in some set \(A \subseteq \mathbb{R}\) and \(T_\ell\) be the \(\ell\)th realization of an inhomogeneous Poisson process with intensity \(\lambda(t)\). Then

1. \(N(A)\) is independent of \(N(B)\) provided \(A \cap B = \emptyset\).

2. Let \(T = \{T_1, \ldots, T_{N(A)}\}\). The joint density of the ordered \(T\) conditional on \(N(A) = n\) is given by

\[
   f(t_1, \ldots, t_n \mid N(A) = n) = \begin{cases} 
   n! \prod_{i=1}^{n} \frac{\lambda(t_i)}{\int_A \lambda(u) \, du} & \text{if } t_i \in A \text{ for all } i \\
   0 & \text{otherwise}
   \end{cases}
\]

In other words, provided that \(n\) events occur in \(A\), the times \(T_1, \ldots, T_n\) at which events occur are considered as unordered random variables which are i.i.d. with density

\[
   f(u) = \begin{cases} 
   \frac{\lambda(u)}{\int_A \lambda(u) \, du} & \text{if } u \in A \\
   0 & \text{otherwise}
   \end{cases}
\]

Using the two properties, we derive the joint density of \(T_\ell\)'s conditional on \(N_{ij}\)'s as follows. First, since event times between subjects are independent, we may assume \(n = 1\) without the loss of generality. Because \(I_j\)'s do not overlap, Property 1 implies that \(N_j\)'s are independent Poisson random variables with a joint probability, given by

\[
   \mathbb{P}(N_1 = n_1, \ldots, N_K = n_K) = \prod_{j=1}^{K} \frac{\lambda_j^{n_j}}{n_j!} \exp(-\Lambda_j) = \exp\left(-\int_{\mathbb{R}} \lambda(u) \, du\right) \prod_{j=1}^{K} \frac{\Lambda_j^{n_j}}{n_j!},
\]

\(^1\)Refer to Ross [33] for details.
where \( \Lambda_j = \int_{I_j} \lambda(u) \, du \). Now suppose that the event times \( T_1, \ldots, T_m \) are fully observed over the period of \( \mathcal{M} \) and \( m = \sum_j n_j \). Their joint density is written as

\[
f(t_1, \ldots, t_m)(u_1, \ldots, u_m) = \exp \left( - \int_{\mathcal{M}} \lambda(u) \, du \right) \prod_{\ell=1}^m \lambda(u_\ell) \quad \text{with } u_1 < \ldots < u_m.
\]

By Property 2, the joint density of \( T_1 \ldots T_m \) conditioning on \( N_1 = n_1, \ldots, N_K = n_K \) is written as

\[
f(u_1, \ldots, u_m \mid N_1 = n_1, \ldots, N_K = n_K) = \prod_{j=1}^K n_j! \prod_{\ell=1}^{n_j} \frac{\lambda(u_\ell)}{\Lambda_j} \quad \text{(3.2)}
\]

with \( n_0 = 0 \) and \( t_0 < u_1 < \cdots < u_{n_1} < t_1 < u_{n_1+1} < \cdots < u_m < t_K \).

It follows that, given the condition that \( N_j = n_j \) for \( j = 1, \ldots, K \), Property 2 implies the times \( T_1 \ldots T_m \) at which events occur can be considered as \( m \) unordered random variables, the \( n_1 \) of which are i.i.d. with density \( \lambda(u)/\Lambda_1 \), the \( n_2 \) of which are i.i.d. with density \( \lambda(u)/\Lambda_2 \) and so forth. This allows us to re-index \( \{T_\ell\} \) to \( \{T_{ijk}\} \), where \( T_{ijk} \) is the \( k \)th event time in the \( j \)th panel for subject \( i \). With the new index, it follows that

\[
\sum_{ijk} \mathbb{E}_{\hat{\lambda}_r} \left[ Y_{i}(T_{ijk})K_h(T_{ijk} - t)P(T_{ijk} - t) \mid \{T_{ijk} \in I_{ij}\} \right]
\]

\[
= \sum_{ijk} \int \cdots \int \left( Y_{i}(u_{ijk})K_h(u_{ijk} - t)P(u_{ijk} - t) \prod_{\ell} \frac{\hat{\lambda}^r(u_{ij\ell})}{\int_{I_{ij}} \lambda^r(u) \, du} \right) \, du_{i1} \cdots du_{ijnj}
\]

\[
= \sum_{ijk} \int \cdots \int \left( Y_{i}(u_{ijk})K_h(u_{ijk} - t)P(u_{ijk} - t) \frac{\hat{\lambda}^r(u_{ijk})}{\int_{I_{ij}} \lambda^r(u) \, du} \prod_{\ell \neq k} \frac{\hat{\lambda}^r(u_{ij\ell})}{\int_{I_{ij}} \lambda^r(u) \, du} \right) \, du_{i1} \cdots du_{ijnj}
\]

\[
= \sum_{ijk} \left( \int_{I_{ij}} Y_{i}(u_{ijk})K_h(u_{ijk} - t)P(u_{ijk} - t) \frac{\hat{\lambda}^r(u_{ijk})}{\int_{I_{ij}} \lambda^r(u) \, du} \, du_{ijk} \right) \left( \prod_{\ell \neq k} \int_{I_{ij}} \frac{\lambda^r(u_{ij\ell})}{\int_{I_{ij}} \lambda^r(x) \, dx} \, du_{ij\ell} \right)
\]

integrates to 1.

\[
= \sum_{ij} N_{ij} \mathbb{E}_{\hat{\lambda}_r} \left[ Y_{i}(T)K_h(T - t)P(T - t) \mid T \in I_{ij} \right],
\]

\[
(3.3)
\]
where $\hat{\lambda}^r$ is the value of $\lambda$ after $r$ iterations. At the e-step, we have the following by substituting (3.3) into (3.1):

$$Q_t(a | \hat{\lambda}^r) = \sum_{ij} N_{ij} E_{\hat{\lambda}^r} \left[ Y_i(T) K_h(T - t) P(T - t) \mid T \in I_{ij} \right]$$

$$- \int_{\mathbb{R}} Y(u) K_h(u - t) \exp \{ P(u - t) \} \, du. \quad (3.4)$$

The m-step maximizes $Q_t(a | \hat{\lambda}^r)$ in (3.4) with respect to $a$. Let $\arg\max Q_t(\lambda(a) | \hat{\lambda}^r)$ denote by $\hat{a}^{r+1}$. After $(r + 1)$ iterations, $\hat{\lambda}^{r+1}(t) = \exp(\hat{a}_0^{r+1})$ for each $t$. The new value of $\lambda$ is then used at the e-step of the next iteration. The cycle between the e- and m-step can be summarized by

$$\hat{\lambda}^{r+1}(t) = \frac{\sum_{ij} N_{ij} E_{\hat{\lambda}^r} \left[ Y_i(T) K_h(T - t) \mid T \in I_{ij} \right]}{\Psi(t; \hat{a}^{r+1})}, \quad (3.5)$$

where $\Psi(t; \hat{a}^{r+1}) = \int_{\mathbb{R}} Y(u) K_h(u - t) \exp \{ \sum_{m=1}^q \hat{a}_m^{r+1}(u - t)^m \} \, du$. Recall that $I_{ijk} = I_{j_k \subseteq I_{ij}}$ and $Y_i(t) = Y_{ij}$ for all $t \in I_{ij}$ for each $j$ and for all $i$. Let $\sum_{ij} I_{ijk} Y_{ij} = Y(t_k)$.

We can rewrite $\Psi(t; \hat{a}^{r+1})$ as

$$\Psi(t; \hat{a}^{r+1}) = \sum_i \int_{\mathbb{R}} Y_i(u) K_h(u - t) \exp \left\{ \sum_{m=1}^q \hat{a}_m^{r+1}(u - t)^m \right\} \, du$$

$$= \sum_{ij} \int_{I_{ij}} Y_{ij} K_h(u - t) \exp \left\{ \sum_{m=1}^q \hat{a}_m^{r+1}(u - t)^m \right\} \, du$$

$$= \sum_{ijk} I_{ijk} Y_{ij} \int_{I_k} K_h(u - t) \exp \left\{ \sum_{m=1}^q \hat{a}_m^{r+1}(u - t)^m \right\} \, du$$

$$= \sum_k Y(t_k) \int_{I_k} K_h(u - t) \exp \left\{ \sum_{m=1}^q \hat{a}_m^{r+1}(u - t)^m \right\} \, du. \quad (3.5)$$

For example, in the locally constant case in which the polynomial is truncated at the leading
term, the iteration (3.5) is simplified to

\[
\hat{\lambda}^{r+1}(t) = \frac{\sum_{ij} N_{ij} E_{\hat{\lambda}^r} \left[ Y_i(T) K_h(T - t) \mid T \in I_{ij} \right]}{\sum_k Y(t_k) \int_{J_k} K_h(u - t) \, du}.
\]

In the locally linear case, (3.5) becomes

\[
\hat{\lambda}^{r+1}(t) = \frac{\sum_{ij} N_{ij} E_{\hat{\lambda}^r} \left[ Y_i(T) K_h(T - t) \mid T \in I_{ij} \right]}{\sum_k Y(t_k) \int_{J_k} K_h(u - t) \exp(\hat{\lambda}^{r+1}_1) \, du}.
\]
3.2 EMS Implementation

We propose to calculate the conditional expectation in (3.5) using the piecewise constant approximation, or rather the $J$-approximant of the $r$th iterate of $\lambda$. In particular, we express the conditional expectation (3.3) in terms of $\Lambda_k$’s. Not only does this use of the piecewise constant approximation lead to the collapse of the local EM to an EMS algorithm, but it also effectively reduces an infinitely dimensional problem to one of finite dimensions.

Let $\tilde{g}^r$ denote $\tilde{g}^r(t) = g^r(t; J)$. Having replaced $\tilde{\lambda}^r$ with $\tilde{g}^r$ in (3.3), we have

$$E_{\tilde{g}^r} \left[ Y_i(T) K_h(T - t) \mid T \in I_{ij} \right] = \int_{I_{ij}} Y_i(u) K_h(u - t) \frac{\tilde{g}^r(u)}{\tilde{g}^r(u)} \, du$$

$$= \sum_k Y_{ij} \mathcal{I}_{ijk} \int_{J_k} K_h(u - t) \, du \frac{\hat{\Lambda}_k^r / \|J_k\|}{\sum_{\ell} \mathcal{I}_{ij\ell} \hat{\Lambda}_\ell^r}$$

$$= \sum_k Y_i(t_k) \int_{J_k} K_h(u - t) \, du \frac{\mathcal{I}_{ijk} \hat{\Lambda}_k^r / \|J_k\|}{\sum_{\ell} \mathcal{I}_{ij\ell} \hat{\Lambda}_\ell^r} \text{ with } Y_{ij} \mathcal{I}_{ijk} = Y_i(t_k). \quad (3.6)$$

By substituting (3.6) into (3.5), we have the $(r + 1)$st iterate of $\lambda$ given by

$$\dot{\lambda}^{r+1}(t) = \sum_{ijk} \frac{\|J_k\|^{-1} \int_{J_k} K_h(u - t) \, du}{\Psi(t; \hat{a}^{r+1})} Y_{ij}(t_{ik}) \mathcal{I}_{ijk} \hat{\Lambda}_k^r \sum_{\ell} \mathcal{I}_{ij\ell} \hat{\Lambda}_\ell^r$$

$$= \sum_k \left( \sum_{ij} Y_i(t_k) N_{ij} \frac{\mathcal{I}_{ijk} \hat{\Lambda}_k^r}{\sum_{\ell} \mathcal{I}_{ij\ell} \hat{\Lambda}_\ell^r} \right) \left( \|J_k\|^{-1} \int_{J_k} K_h(u - t) \, du \right) \frac{\Psi(t; \hat{a}^{r+1})}{\Psi(t; \hat{a}^{r+1})}. \quad (3.7)$$

After we integrate $\dot{\lambda}^{r+1}$ over the $J_k$’s, the following iteration for intensity estimation emerges from the local EM algorithm (3.5).

$$\hat{\Lambda}_\ell^{r+1} = \int_{J_\ell} \dot{\lambda}^{r+1}(t) \, dt = \int_{J_\ell} \sum_k \left( \sum_{ij} Y_i(t_k) N_{ij} \frac{\mathcal{I}_{ijk} \hat{\Lambda}_k^r}{\sum_{\ell} \mathcal{I}_{ij\ell} \hat{\Lambda}_\ell^r} \right) \left( \|J_k\|^{-1} \int_{J_k} K_h(u - t) \, du \right) \frac{\Psi(t; \hat{a}^{r+1})}{\Psi(t; \hat{a}^{r+1})} \, dt$$

$$= \sum_k \left( \sum_{ij} Y_i(t_k) N_{ij} \frac{\mathcal{I}_{ijk} \hat{\Lambda}_k^r}{\sum_k \mathcal{I}_{ijk} \hat{\Lambda}_k^r} \right) \left( \int_{J_\ell} \|J_k\|^{-1} \int_{J_k} K_h(u - t) \, du \frac{\Psi(t; \hat{a}^{r+1})}{\Psi(t; \hat{a}^{r+1})} \, dt \right). \quad (3.8)$$
Note that $\hat{a}^{r+1}$ is a function of $\hat{\Lambda}^r$ because $\hat{a}^{r+1}$ maximizes (3.4) with $\hat{\lambda}^r$ being replaced by $\hat{g}^r$. Here, we express (3.8) in matrix form as

$$\hat{\Lambda}^{r+1} = M(\hat{\Lambda}^r)K_h(\hat{\Lambda}^r).$$

(3.9)

The mapping, $M(\Lambda) : \mathbb{R}^K \rightarrow \mathbb{R}^K$, returns a row vector, the $\ell$th component of which equals

$$\sum_{ij} Y_i(t_k) Y(t_k) N_{ij} I_{ijk} \hat{\Lambda}_k \sum_k I_{ijk} \hat{\Lambda}_k.$$

This is identical to the EM mapping in (1.8), as well as the self-consistent mapping of Hu, Lagakos and Lockhart [21], for intensity estimation in the context of panel count data. Hence, $M(\Lambda)$ is recognized as a step in the EM algorithm, whereas $K_h(\Lambda)$ is a $K \times K$ smoothing matrix with

$$[K_h(\Lambda)]_{k\ell} = \int_{J_{\ell}} \frac{[J_k]^{-1} Y(t_k) \int_{J_k} K_h(u - t) \, du}{\sum_m Y(t_m) \int_{J_m} K_h(u - t) \exp \left\{ \sum_j \hat{a}_j^{r+1}(u - t) \right\} \, du} \, dt \quad \text{with} \quad \hat{a}_j^{r+1} = a_j(\hat{\Lambda}^r).$$

Because $K_h$ in (3.9) depends on $\Lambda$ through $a$, $K_h$ adapts itself to the new value of $\Lambda$ at each iteration. Consequently, we refer to (3.9) as an adaptive EMS implementation of the local EM algorithm. Thereby, using the $J$-approximant of $\hat{\lambda}^r$ in the local EM algorithm leads to an EMS algorithm for intensity estimation. The EMS method may be labelled as an ad hoc procedure in the literature. Nevertheless, we have shown that local likelihood endows the use of the EMS algorithm with a natural context.

Should the iteration of (3.9) converge, a local likelihood intensity estimate is set to be

$$\hat{\lambda}^*(t) = \sum_{ijk} \frac{[J_k]^{-1} \int_{J_k} K_h(u - t) \, du}{\sum_m Y(t_m) \int_{J_m} K_h(u - t) \exp \left\{ \sum_j \hat{a}_j^*(u - t) \right\} \, du} \frac{Y_i(t_k) N_{ij} I_{ijk} \hat{\Lambda}_k^*}{\sum_k I_{ijk} \hat{\Lambda}_k^*},$$

where $\hat{a}^* = a(\hat{\Lambda}^*)$. Note that the $\hat{\lambda}^{r+1}$ in (3.7) is obviously different from that in (3.5) but
they are related. Recall that $K$ is the number of pixels in $J$. Let $\hat{\lambda}_K^{r+1}$ and $\hat{\lambda}_\infty^{r+1}$ denote the $(r + 1)$st iterate of (3.7) and (3.5), respectively. We will show the $L^1$-convergence of $\hat{\lambda}_K^{r+1}$ to $\hat{\lambda}_\infty^{r+1}$ in Chapter 5. Finally, the numerical implementation of local EM algorithms with a constant and linear polynomial is given in detail in Appendix A.1, and the corresponding R [32] code is presented in Appendix A.3.

In summary, when the event process is inhomogeneous Poisson, using the $J$-approximant of $\hat{\lambda}^r$ in the implementation of the local EM algorithm explicitly results in an EMS algorithm. Similar to the local EM method for density estimation in Chapter 2, the EMS algorithm arises naturally from the local EM framework. Consequently, local likelihood suggests that the EMS algorithm is not ad hoc.
3.3 Application: Incidence Rate of Nausea in NCGS

We now use the proposed local EM method to analyze the panel count data presented in the Table 1 of Thall and Lachin [40]. The data consist of the number of nausea incidences during the first year of follow-up for a subset of 113 patients who participated in National Cooperative Gallstone Study (see Schoenfield and Lachin [36] for details). 65 patients were randomly assigned to receive high dosage of chenodeoxycholic acid, and 48 were randomly assigned to receive the placebo. Patients were scheduled to be assessed in approximately 1, 2, 3, 6, 9 and 12 months. The objective is to determine if there exists a significant difference in the number of nausea incidences between the high-dose and placebo groups during the first 60 months. Figure 3.1 provides a visualization of the data. Each row represents a patient, and a rectangle indicates the time interval between two adjacent visits and its gray scale indicates the number of reported nauseas between the two visits. Two patients in the treatment group are not shown because they dropped out of the study before the first visit.

We obtain two local likelihood intensity estimates for each group. One estimate is based on a constant polynomial, and the other is based on a linear polynomial. Here, the bandwidth is selected by applying leave-one-out likelihood cross validation to each group. The “optimal” bandwidth value is set to be the one that maximizes the following:

$$\sum_{ij} N_{ij}(C_{ij}) \log \left( \int_0^{C_{ij}} \hat{\lambda}_{ij|j}(t) \, dt \right) - \sum_{ij} \int_0^{C_{ij}} \hat{\lambda}_{ij|j}(t) \, dt,$$

where \( i = 1 \) or \( 2 \) indicating the treatment or control group. \( \hat{\lambda}_{ij|j} \) is an intensity estimate without the observations from the \( j \)th subject in the \( i \)th group.

As shown in Figure 3.2, the risk of nausea elevates either at the beginning or the end of the study and reaches the lowest point around week 20 for both high-dose and placebo groups. However, patients in the high-dose group have the highest risk of nausea between week 40 and 60, whereas the placebo group have highest nausea risk during the first 10 weeks.
of the study. Compared with the empirical intensity estimate shown in Figure 2 of Thall and Lachin [40], the local likelihood estimates are much more smoother. The finding is consistent with the empirical result shown in Figure 2 of Thall and Lachin [40], but our analysis further suggests that the intensities for high-dose and placebo groups are qualitatively different in terms of peaks and troughs. For example, Thall and Lachin’s intensity estimate for the placebo group during the first 20 weeks appears to be trimodal. However, the local EM method removes two bumps by smoothing and results in an intensity estimate, showing that during the first 20 weeks, the risk of nausea is unimodal and peaks in week 5.
Panel Count of Nausea Incidences

Figure 3.1: Each row represents a patient. A rectangle in a row indicates the time interval between two consecutive visits, and its gray scale indicates the number of reported nausea during the interval. Crosses indicate drop-out times, and dash lines represent the time elapsed between the last visit and drop-out. Two patients in the treatment group are excluded from the analysis. The x-axis shows the time points used to construct the partition $J$. 
Figure 3.2: Local likelihood intensity estimates for the risk of nausea.
3.4 Simulation Study

Assume that each subject has no events at time 0. We simulate an inhomogeneous Poisson process with intensity $\lambda(t)$ equal to a re-scaled gamma density function (shape = 9 and rate=3/4). This intensity achieves the maximum of 1 at $t = 32/3$. A subject’s event times are simulated by thinning a unit-intensity Poisson process in which event times are either accepted or rejected based the probability equal to $\lambda(t)$. Each subject is assumed to have a sequence of predetermined observation times $t_1, t_2, \ldots, t_K$ where $t_i = i$ and $K = 20$. However, subjects miss a visit with increasing probability; specifically, the probability of missing a visit at time $t$ equals $(t/20)^{1/4} - 0.05$. Finally, a subject’s panel counts are obtained by aggregating events times among his/her consecutive visits. Note that $\mathcal{J}$ may vary between samples.

For a sample, we compute a kernel intensity estimate, as in (1.12), as well as the proposed local likelihood estimate using a Gaussian kernel with various bandwidth values. Assuming no interval censoring has taken place, we use actual event times to obtain the kernel intensity estimate. For panel count data, we compute two local likelihood estimates in both locally constant and linear cases with the partition $\mathcal{J}$. We also obtain a NPMLE using an EM algorithm. Since the NPMLE is only unique up to an equivalence class, we smooth the NPMLE with a Gaussian kernel by placing its masses at the left-end, the middle and the right-end points of the $J_j$’s. This results in three slightly different versions of smoothed nonparametric intensity estimates

For each window size $h$, 300 samples are simulated. These samples are used to approximate each estimator’s mean integrated squared error (MISE). Let $\hat{\lambda}_k$ denote an intensity estimate for the $k$th sample. We consider the average integrated squared error, which is defined as $\sum_{k} \int (\hat{\lambda}_k(u) - \lambda(u))^2 \, du / 300$, to be an estimate of the MISE. This is performed for 40 equally spaced values of $h$ between 0.05 and 3.95. The study result considerably favours the proposed local likelihood intensity estimators, and the estimated MISE’s of the five estimators are plotted and shown in Figure 3.3.
Figure 3.3: A NPMLE is smoothed by placing expected increments at the left-end, mid and right-end points of the $J_k$'s. By comparing with the three different versions of smoothed NPMLE's, we show that the proposed local likelihood intensity estimator achieves a lower overall MISE with a small bandwidth of 0.5 (locally constant) and 0.6 (locally linear). Moreover, the bandwidth value with which the local likelihood estimator reaches its lowest MISE is close to that of Diggle’s edge-corrected intensity estimate, which is 0.65.

Among all intensity estimators, the kernel intensity estimator $\hat{\lambda}_{KE}$ achieves the smallest MISE because it is based on the actual event times. It seems that the proposed local likelihood intensity estimators track $\hat{\lambda}_{KE}$ quite closely in terms of MISE's. In addition, they attain the smallest overall MISE among all intensity estimators based on panel count data. This is perhaps not all that surprising given $\lambda(t)$ is quite non-linear. In the case in which $\lambda(t)$ is rather linear, the improvement in MISE for the local likelihood intensity estimators is not as dramatic.
Chapter 4

Spatial Point Processes

In this chapter, we apply local EM to the analysis of spatially aggregated data. We consider $n$ spatial processes, the $i$th of which is an inhomogeneous Poisson process with intensity $\lambda_i$. We define $\lambda_i$ to be the product of an offset surface and a relative risk function. While the offset surfaces may not be smooth and vary from one process to another, they are assumed to be known \textit{a priori}. The parameter of interest in this context is the unknown relative risk function that is spatially smooth and common to all $n$ processes.

There are five sections in the chapter. In §4.1, we introduce the notations frequently used throughout the chapter. We also present local likelihood for risk estimation when the point process data are fully observed. In the presence of aggregated data, we consider local EM and formulate a local EM algorithm in §4.2. Following the formulation of the local EM algorithm, we employ the piecewise constant approximation at the E-step and implement the local EM as an EMS algorithm. However, the local EM algorithm here is different from that in the context of panel count data because of the offset surfaces, and its EMS implementation is also modified by the surfaces.

In §4.4, we are interested in analyzing incidences of a rare autoimmune disease within the Greater Toronto Area (GTA). Since the risk of disease is believed to be influenced
by the physical environment and varying smoothly in space, we aim to obtain a spatially
smooth risk estimate, which can be used to identify areas with elevated risk, in the local EM
framework. Since the disease is rare, data are collected over four periods of 10 years in order
to accumulate sufficient information. Moreover, the data in each period are aggregated over
numbers of disjoint and bounded regions in order to protect patients’ medical privacy. The
regions could be census tracts or postal-code areas, and they together form a map of the GTA.
Here, we refer to the map as a disease map. Should the regions in all four disease maps
remain the same, it is fairly straightforward to estimate regional disease rates. However, in
the situation in which boundaries of the regions change from one map to another, it was
not clear how one may combine multiple disease maps to estimate the regional disease rates.
In this situation, we suggest a local EM method to model the disease process in a spatio-
temporal setting. In particular, we demonstrate how one may estimate covariate effects using
a generalized linear model (GLM) and incorporate the resulting effect estimates into a local
EM algorithm through the offset surface. In this disease mapping problem, we encounter
a daunting computational task when censoring regions have irregular shape. We propose
a computationally feasible solution to the aforementioned task using results from manifold
calculus.

4.1 Notations and Assumption

Consider a spatial point process $X_i = \{X_{ik} \mid k = 1 \ldots K_i\}$ during the $i$th period over a
geographical area, $\mathcal{M}_i \subseteq \mathbb{R}^2$. We assume $\mathcal{M}_i$’s are bounded and refer to them as maps.
Each map consists of numbers of disjoint and bounded regions, denoted by $R_{ij}$’s. Here,
$\mathcal{M}_i = \bigcup_j R_{ij}$. Let $N_{ij} = \#\{\ell \mid X_{i\ell} \in R_{ij}\}$, the number of cases that occur within $R_{ij}$. When
$X_i$ are aggregated, the data are reported in terms of $N_{ij}$’s. Suppose that the $i$th spatial
point process is inhomogeneous Poisson with intensity

\[ \lambda_i(x) = O_i(x)v(x). \]  \hfill (4.1)

Provided that \( v(x) = 1 \) for all \( x \in \mathcal{M} \), the offset function \( O_i(x) \) is interpreted as the expected case count at location \( x \) of the \( i \)th map. This function is not necessarily smoothly varying in space but assumed to be known \textit{a priori}. In particular, \( O_i(x) \) is assumed to be constant over the region \( R_{ij} \) for all \( i, j \) for the reason that will be explained in §4.3. For example, \( O_i(x) = O_{ij} \) for \( x \in R_{ij} \). The relative risk function, denoted by \( v \), is spatially smooth. The objective is to determine the spatial variation in terms of relative risk.

Denote the overall observation area by \( \mathcal{M} = \bigcup_i \mathcal{M}_i \). Analogous to the partition in the context of panel count data, a partition of \( \mathcal{M} \) can be constructed by superimposing \( \mathcal{M}_i \)’s; that is,

\[ \mathcal{J} = \{J_1, \ldots, J_K\} \text{ such that } \bigcup_{\ell} J_\ell = \mathcal{M} \text{ and } J_\ell \subseteq R_{ij} \text{ for some } i, j \text{ and for all } \ell. \]
For example, Figure 4.1 shows two maps that represent the same geographical area. One horizontally divides the area, and the other divides the area vertically. Superimposing the two maps results in a partition, \( J = \{ J_k \mid k = 1, \ldots, 4 \} \), where \( J_k \)'s are referred to as pixels. In addition, let \( \Upsilon_\ell = \int_{J_\ell} \nu(x) \, dx \) be the \( \ell \)th pixel risk, \( \Upsilon = \{ \Upsilon_1, \ldots, \Upsilon_K \} \) be the collection of pixel risks and \( I_{ij\ell} \) indicate if \( J_\ell \subseteq R_{ij} \).

### 4.2 Local Likelihood and Local EM

When the event process is a inhomogeneous Poisson process in space, we consider the following local likelihood:

\[
\mathcal{L}_x(\nu) = \sum_{i\ell} K_h(x_{i\ell} - x) \log \lambda_i(x_{i\ell}) - \sum_i \int_{\mathcal{M}_i} K_h(u - x) \lambda_i(x) \, du
\]

\[
= \sum_{i\ell} K_h(x_{i\ell} - x) \log \nu(x_{i\ell}) - \sum_i \int_{\mathcal{M}_i} \mathcal{O}_i(u) K_h(u - x) \nu(u) \, du
\]

\[+\text{ terms independent of } \nu, \quad (4.2)\]

However, estimating the parameter of interest \( \nu(x) \) requires some care. One of the reasons that local likelihood is appropriate is that the true intensity is sufficiently smooth to be locally approximated by a truncated Taylor series. In the situation in which the intensity function is not necessarily smooth nor continuous, locally approximating the intensity with a (smooth) polynomial would introduce unnecessary bias. Hence, instead of \( \lambda_i(x) \), we approximate its smooth component, namely \( \log \nu \), by a bivariate polynomial of order \( q \) in the neighborhood of \( x \). Denote the polynomial and its coefficients by \( \mathcal{P}(z) \) and \( a \), respectively. Here, \( z = u - x \) and \( z \equiv (z_1, z_2) \). For example, when \( q = 2 \),

\[
\mathcal{P}(z) = a_0 + a_1 z_1 + a_2 z_2 + a_3 z_1^2 + a_4 z_1 z_2 + a_5 z_2^2.
\]
Upon the substitution of $P(z)$ into the local likelihood function in (4.2), we maximize the local likelihood function with respect to $a$ for each $x$ and establish a local likelihood risk estimate at $x$ to be

$$
\hat{\nu}(x) \equiv \exp\{\hat{a}_0\} = \frac{\sum_{i\ell} K_h(X_{i\ell} - x)}{\Psi_h(x; \hat{a})},
$$

where $\hat{a}$ maximizes (4.2). Here, let $\tilde{O}_\ell = \sum_{ij} I_{ij\ell} O_{ij}$.

$$
\Psi_h(x; a) = \sum_i \int_{m_i} O_i(u) K_h(u - x) \exp \left\{ \sum_{m=1}^{q} a_m (u - x)^m \right\} du \\
= \sum_{ij} O_{ij} \int_{R_{ij}} K_h(u - x) \exp \left\{ \sum_{m=1}^{q} a_m (u - x)^m \right\} du \\
= \sum_{ij\ell} I_{ij\ell} O_{ij} \int_{J_{ij\ell}} K_h(u - x) \exp \left\{ \sum_{m=1}^{q} a_m (u - x)^m \right\} du \\
= \sum_{\ell} \tilde{O}_\ell \int_{J_{\ell}} K_h(u - x) \exp \left\{ \sum_{m=1}^{q} a_m (u - x)^m \right\} du. 
$$

When data are aggregated, only $N_{ij}$’s are reported. In this case, we apply local EM to (4.2) and formulate a local EM algorithm, which is given by

$$
\hat{\nu}^{r+1}(x) \equiv \exp\{\hat{a}_0\} = \frac{\mathbb{E}_{\hat{\nu}^r} \left[ \sum_{i\ell} K_h(X_{i\ell} - x) \mid \{N_{ij}\} \right]}{\Psi_h(x; \hat{a})}.
$$

We restate the two properties of inhomogeneous Poisson processes in the spatial context (see Diggle [12], page 67).

1. For process $i$, $N_{ij} \sim \text{Poisson} \left( \int_{R_{ij}} \lambda_i(u) \, du \right)$.

2. Given $N_{ij} = n$, the $n$ events in $R_{ij}$ form an independent random sample from the distribution on $R_{ij}$ with density $f$ equal to $\frac{\lambda_i(x)}{\int_{R_{ij}} \lambda_i(u) \, du}$.
Since $O_i(x) = O_{ij}$ for $x \in R_{ij}$, $f$ can be further simplified and given by

\[
f(x) = \begin{cases} 
\frac{v(x)}{\int_{R_{ij}} v(u) \, du} & x \in R_{ij} \\
0 & \text{otherwise.} 
\end{cases}
\] (4.5)

We carefully derive an expression for the conditional expectation in (4.4) using the two properties. Similar to the derivation in Chapter 3, the conditional expectation can be written as

\[
E_{\hat{\psi}r} \left[ \sum_{i} K_h(X_{it} - x) \mid \{N_{ij}\} \right] = \sum_{ij} N_{ij} E_{\hat{\psi}r} [K_h(X - x) \mid X \in R_{ij}]
\]

By substituting the expression above to (4.4), we obtain

\[
\hat{\psi}^{r+1}(x) = \sum_{ij} N_{ij} \frac{E_{\hat{\psi}r} [K_h(X - x) \mid X \in R_{ij}]}{\psi_h(x; \hat{a}^{r+1})}.
\] (4.6)

4.3 EMS Implementation

Recall the $J$-approximant of $\hat{\psi}^r(x)$ is written as

\[
g_{\hat{\psi}r}(x; J) = \sum_k \mathcal{I}(x \in J_k) \|J_k\|^{-1} \int_{J_k} \hat{\psi}^r(u) \, du = \sum_k \mathcal{I}(x \in J_k) \|J_k\|^{-1} \hat{\psi}^r_k.
\]

Let $\hat{g}^r$ denote $g_{\hat{\psi}r}(x; J)$. Following Chapter 3, we replace $\hat{\psi}^r$ with $\hat{g}^r$ in (4.6). It follows that

\[
\hat{\psi}^{r+1}(x) = \sum_{ij} N_{ij} \frac{E_{\hat{g}r} [K_h(X - x) \mid X \in R_{ij}]}{\psi_h(x; \hat{a}^{r+1})} = \sum_k \|J_k\|^{-1} \frac{\int_{J_k} K_h(u - x) \, du}{\psi_h(x; \hat{a}^{r+1})} \sum_{ij} N_{ij} \frac{\mathcal{I}_{ijk} \hat{\psi}^r_k}{\sum_m \mathcal{I}_{ijm} \hat{\psi}^r_m}.
\] (4.7)
Integrating $\hat{\varphi}^{r+1}(x)$ over the $J_\ell$'s leads to an iteration given by

$$\hat{\Upsilon}_{\ell}^{r+1} = \int_{J_\ell} \hat{\varphi}^{r+1}(x) \, dx = \sum_k \left( \sum_{ij} N_{ij} \frac{I_{ijk}}{\sum_m I_{ijm}} \hat{\Upsilon}_{k}^{r} \right) \left( \|J_k\|^{-1} \int_{J_\ell} \frac{\int_{J_k} K_h(u - x) \, du}{\Psi_h(x; \hat{a}^{r+1})} \, dx \right),$$

or, in terms of matrices,

$$\hat{\Upsilon}^{r+1} = M(\hat{\Upsilon}^r)K_h(\hat{\Upsilon}^r),$$  (4.8)

where

$$[M(\hat{\Upsilon}^r)]_\ell = \sum_{ij} N_{ij} \frac{I_{ij \ell}}{\hat{O}_\ell \sum_k I_{ijk}} \hat{\Upsilon}_{k}^{r} \quad \text{and} \quad [K_h]_{\ell s} = \frac{\hat{O}_\ell}{\|J_\ell\|} \int_{J_\ell} \frac{\int_{J_k} K_h(u - x) \, du}{\Psi_h(x; \hat{a}^{r+1})} \, dx.$$

$M(\Upsilon)$ is recognized as the EM mapping in Vardi et al. [44], whereas $K_h(\Upsilon)$ is a smoothing matrix that depends on $\Upsilon$ since $\hat{a}^{r+1} = a(\hat{\Upsilon}^r)$ whenever $q \geq 1$. Therefore, the iteration in (4.8) involves an expectation, maximization and smoothing step, or an EMS algorithm. To put it briefly, using the $J$-approximant of $\hat{\varphi}^r(x)$ at each iteration leads to an EMS implementation of the local EM algorithm for risk estimation in the spatial context. Note that multiplication and division by the quantity $\hat{O}_\ell$ ensures $M(\hat{\Upsilon}^r)$ is indeed the mapping in the usual EM algorithm. In addition, $K_h$ converges to an identity matrix since $\Psi_h(x; \hat{\varphi}^{r+1}) \to \hat{O}_\ell$ for $x \in J_\ell$ and $\int_{J_\ell} K_h(u - x) \, du \to I(x \in J_\ell)$ as $h \downarrow 0$. That is, the EMS implementation of the local EM becomes an EM algorithm as $h \to 0^+$. Upon the convergence of (4.8), $\hat{\Upsilon}^* = M(\hat{\Upsilon}^*)K_h(\hat{\Upsilon}^*)$. We establish a local likelihood risk estimate for the spatially aggregated data to be

$$\hat{\varphi}^*(x) = \sum_{ij \ell} \frac{\|J_\ell\|^{-1} \int_{J_\ell} K_h(u - x) \, du}{\Psi_h(x; \hat{a}^*)} N_{ij} \frac{I_{ijk}}{\sum_m I_{ijm}} \hat{\Upsilon}_{k}^{*} \quad \text{with} \quad \hat{a}^* = a(\hat{\Upsilon}^*).$$

Note that the local likelihood risk estimate in (4.7) is different from that in (4.6). However, the two estimates are related. Recall that $K$ is the number of pixels in $J$. Let $\hat{\varphi}_K^*(x)$ and
\( \hat{v}_r^\infty(x) \) denote the value of \( v(x) \) after \( r \) iterations of (4.7) and (4.6), respectively. It will be shown in Chapter 5 that, as \( K \to \infty \),

\[
\int_{\mathcal{M}} \left| \hat{v}^{r+1}_K(u) - \hat{v}^{r+1}_\infty(u) \right| \, du \longrightarrow 0.
\]

**Remark:** When there is only one map, the local EM algorithm only iterates once. In this situation, a local likelihood risk estimate in the locally constant case is given by

\[
\hat{\nu}(x) = \sum_j \|J_j\|^{-1} \frac{N_j \int_{R_j} K_h(u - x) \, du}{\sum \ell \mathcal{O}_\ell \int_{R_\ell} K_h(u - x) \, du}.
\]

The risk estimate is recognized as the one in (2.3) of Brillinger [9]. Hence, the estimate in (4.7) can be seen as a generalization of the risk estimate proposed in Brillinger [7, 8, 9] when there are multiple maps.

## 4.4 Application: Disease Mapping

### 4.4.1 Study Objective

Systemic lupus erythematosus (SLE) is a rare autoimmune disorder. There are approximately 30,000 Canadians with SLE, and 90% of patients are female. Although causes for the disease remain unknown, several potential risk factors have been identified, one of them being the physical environment (see Al-Maini [2]). As a result, the objective of our analysis is to use new SLE patients’ residential information to determine if there exists an area with elevated disease risk. Should such an area exist, the observed cases would appear to be spatially clustered and exhibit a spatial structure; otherwise, they should be randomly scattered over the observational area without any obvious spatial pattern.
4.4.2 Census Tracts

In this example, census data from 1971, 1981, 1991 and 2001 are available to us. Here, we assume the census is taken in the middle of the census period, thus defining four census periods to be 1966 ∼ 1975, 1976 ∼ 1985, 1986 ∼ 1995 and 1996 ∼ 2005. A census tract\(^1\) is a geographical unit used by Statistics Canada in the preparation and collection of census data. It is defined as a small, relatively stable geographic area that usually has a population of 2,500 to 8,000 residents. Figure 4.3(a), 4.3(b), 4.3(c) and 4.3(d) show census tracts that covered the GTA in 1971, 1981, 1991 and 2001, respectively. Two observations can be made based upon the inspection of these maps. First of all, the political boundaries of the GTA have been expanding during the 40-year observational period. Secondly, the census tracts tend to be small in the metropolitan urban centre and large in suburban peripheral areas. Additionally, the census tract boundaries vary from one period to another in order to accommodate growing regional populations. Since each of the four maps has its own coverage, resolution and alignment, how to combine multiple maps that have different coverages, resolutions and alignments remained unclear.

A patient’s exact residential information is concealed in order to protect his/her medical privacy. Therefore, only the case count at the census tract level is made available for this analysis. Since SLE is a rare disease, accumulating sufficient information requires a long period of data collection. In spite of the data-collection period being 40 years, the dataset only contains 872 new SLE cases in total. The statistical challenge in this problem is to model spatial variations in disease risk using multiple disease maps with time-varying boundaries while adjusting for gender and age group effects.

\(^1\)http://www12.statcan.ca/census-recensement/2006/dp-pd/prof/92-597/
(a) The layout of GTA census tracts in 1971.  
(b) The layout of GTA census tracts in 1981.  
(c) The layout of GTA census tracts in 1991.  
(d) The layout of GTA census tracts in 2001.

Figure 4.2: Each enclosed area represents a census tract. The census tract is small in the urban centre and large in the suburban area. Moreover, the census tract boundaries change from one map to another.
4.4.3 Statistical Model

We use the subscript $i$ to index either maps or census periods since there is only one map for a census period. Let $R_{ij}$ denote the $j$th census tract of the $i$th map. To take into account spatio-temporal variations in population disease rates for each age-gender group, consider $\beta$ to be a vector of disease rates by census period and $\theta$ to be a vector of disease rates by age-gender group. Note that $\beta$ is interpreted as the time effect for the entire period, instead of per unit time. Denote the population intensity of the $k$ age-gender group at location $x$ during the $i$th census period by $P_{ik}(x)$, and let $P_{ijk}$ denote the $R_{ij}$’s population of the $k$th age-gender group. Since the censuses only provide us with the total population data by age-gender group at the census tract level, we cannot possibly know how the population of each census tract was distributed without further information. In other words, the census tract is the finest spatial resolution based on the available information. As a result, it is plausible to assume that the population was distributed uniformly over each census tract and had remained constant throughout a census period. That is equivalent to say, during the $i$th census period,

$$P_{ik}(x) = P_{ijk}/\|R_{ij}\| \text{ for } x \in R_{ij}.$$ 

Finally, the disease process for the $k$th age-gender group during the $i$th period is assumed to be an inhomogeneous Poisson process with intensity

$$\lambda_{ik}(x) = v(x) \sum_j I(x \in R_{ij}) O_{ijk}, \text{ where } O_{ijk} = \|R_{ij}\|^{-1} \beta_i \theta_k P_{ijk}.$$ 

It follows that the number of SLE cases within $R_{ij}$ for the $k$th age-gender group, denoted by $N_{ijk}$, is distributed as

$$N_{ijk} \sim \text{Poisson} \left( O_{ijk} \int_{R_{ij}} v(x) \, dx \right). \quad (4.9)$$
Note that $\sum_{jk} N_{ijk}, \sum_{ij} N_{ijk},$ and $\sum_{ik} N_{ijk}$ are sufficient statistics for $\beta_i, \theta_k,$ and $\int_{R_j} u(x) \, dx,$ respectively.

We use a two-stage modelling approach to estimate $\beta,$ $\theta$ and $\Upsilon$. In the first stage, we ignore the spatial structure of the data and estimate $\beta$ and $\theta$ using a GLM with $P_{ijk}$'s as population offsets. In the second stage, we treat these GLM estimates of $\beta$ and $\theta$ as if they were known quantities and set $\check{O}_i = \sum_{ijk} I_{ijk} \hat{\beta}_i \hat{\theta}_k P_{ijk}$. Following the details described in the previous section, we set up the local EM algorithm as follows:

$$\check{\Upsilon}^{r+1} = \mathcal{M}(\check{\Upsilon}^r) \mathcal{K}_h(\check{\Upsilon}^r).$$

Upon its convergence, we set a local likelihood risk estimate to be

$$\hat{\upsilon}^*(x) = \sum_{ijs} J_{ijs} \frac{\int J_s K_h(u - x) \, du / \| J_s \| N_{ij} \cdot I_{ijs} \hat{\Upsilon}^*_s \sum_{q} I_{ijq} \hat{\Upsilon}^*_q}{\Psi_h(x; \hat{a}^*) \sum_{ij} J_{ijs} \hat{\Upsilon}^*_s},$$

(4.10)

where $N_{ij} = \sum_k N_{ijk}$ and $\Psi_h(x; \hat{a}^*) = \sum_{ij} \check{O}_i \int_{J_i} K_h(u - x) \exp \left\{ \sum_{m=1}^q a_m (\check{\Upsilon}^*)^m (u - x)^m \right\} \, du.

### 4.4.4 Kernel Matrix

With the data-driven partition $J$, the entry of the smoothing matrix is of the following form:

$$[\mathcal{K}_h]_{ls} = \frac{\check{O}_i}{\| J_i \|} \int_{J_s} \frac{\int J_l K_h(u - x) \, du}{\Psi_h(x; \hat{a}^{r+1})} \, dx.$$

Since the kernel function is a bivariate function, each entry of $\mathcal{K}_h$ involves a quadruple integral. When the partition $J$ consists of regions that are in irregular shape, to compute the integrals is difficult. To proceed with the computation, we tessellate all four maps$^2$, and the tessellation serves as a partition with pixels of equal size, denoted by $C = \{C_1, \ldots, C_{206957}\}$. With this partition, the dimension of the smoothing matrix is 206,957-by-206,957, and it is

$^2$Refer to the discussion about partition configurations in Chapter 2.
an intimidating computation task to numerically evaluate over 42 billion quadruple integrals to construct the smoothing matrix.

We take the following two steps in order to complete the taxing task in a timely fashion. First, we choose a radially symmetrical kernel with compact support. With a relatively small bandwidth value, the use of this kernel will result in a sparse smoothing matrix, in which most entries equal zero. Because we cannot even initiate a 206,957-by-206,957 matrix using usual R commands, we have to apply specialized procedures for sparse matrices, for example R’s SparseM package [23], to construct the smoothing matrix and perform necessary algebraic operations in a local EM iteration with minimal memory consumption. The R code to perform the local EM algorithm is included in Appendix A.3.

Next, we have to numerically evaluate the quadruple integrals in $\mathcal{K}_h$. A brute-force approach is to directly apply a quadrature rule, say Simpson’s rule, to the four dimensional function

$$\|C\|^{-1} \frac{\mathcal{O}_n K_h(u - x) \, du}{\sum_n \mathcal{O}_n \int_{C_n} K_h(u - x)}.$$ 

Due to the large number of pixels, the brute-force approach is infeasible. Instead, we choose a kernel simple enough that we can derive an analytical solution to the inner double integral. Once we obtain the analytical form of the inner double integral, we apply a Gaussian quadrature rule to approximate the outer double integral. This step did significantly shorten the computational time. Here, we choose a bivariate biweight kernel, given by

$$K(u, v) = \begin{cases} 
\frac{3}{\pi} \left(1 - (u^2 + v^2)\right)^2 & \text{where } u^2 + v^2 \leq 1 \\
0 & \text{otherwise.}
\end{cases}$$
For a fixed point $s$, we apply Green’s Theorem to simplify the double integral

$$\int_{C_p} K_h(x - s) \, dx$$

to a line integral, an indefinite integral that has an analytical formula. For instance, let $u = x_1 - s_1$ and $v = x_2 - s_2$. Without the loss of generality, set $h = 1$ so that the support of the kernel is a unit circle. Denote the intersection of the unit circle and the pixel $C_p$ by $D$ and the boundary of $D$ in a counter-clockwise orientation by $\partial D$. Then the analytical solution to the inner double integral is given by

$$\int_D K(u, v) \, du \, dv = \frac{3}{\pi} \int_{\partial D} f(u, v) \, du + g(u, v) \, dv,$$

where

$$f(u, v) = -(v^5/5 - 2v^3/3 + u^2v^3/3 + v/2) \quad \text{and} \quad g(u, v) = u^5/5 - 2u^3/3 + u^3v^2/3 + u/2.$$  

Equation 4.11 can be also expressed in polar coordinates with $r = 1$, which is given by

$$\frac{3}{\pi} \int_{\partial D} (5\theta - \sin 4\theta)/30 \, d\theta,$$

where $\partial D = \bigcup_i \partial D_i$, where $\partial D_i$ is either a segment of the pixel boundary or an arc of

---

\(^3\)Suppose that $D$ is a region in a plane with boundary $\partial D$. Let $\partial D$ be a positively oriented (counterclockwise), piecewise smooth and simple closed curve in the plane. Green’s theorem states

$$\int_{\partial D} f(x, y) \, dx + g(x, y) \, dy = \iint_D \frac{\partial g}{\partial x} - \frac{\partial f}{\partial y} \, dx \, dy.$$
Figure 4.3: The region $D$ is enclosed by three solid lines. Let $\partial D_1$, $\partial D_2$ and $\partial D_3$ be the horizontal, the arc and the vertical solid lines, respectively.

the unit circle, $\int_{\partial D} f(u,v) \, du + g(u,v) \, dv = \sum_i \int_{\partial D_i} f(u,v) \, du + g(u,v) \, dv$ and

$$\int_{\partial D_i} f(u,v) \, du + g(u,v) \, dv = \begin{cases} \frac{3}{\pi} \int_{u_R}^{u_L} f(u,v) \, du & \text{if } \partial D_i \equiv (u_R, v) \to (u_L, v) \\ \frac{3}{\pi} \int_{v_U}^{v_L} g(u,v) \, dv & \text{if } \partial D_i \equiv (u, v_U) \to (u, v_L) \\ \frac{1}{10\pi} \int_{\theta_L}^{\theta_U} 5\theta - \sin 4\theta \, d\theta & \text{if } \partial D_i \equiv \theta_L \to \theta_U. \end{cases}$$

For example, we can use Formula 4.11 to calculate area over the region $D$ enclosed by the three solid lines in Figure 4.3. Let $\partial D_1$, $\partial D_2$ and $\partial D_3$ be the horizontal solid line, the solid arc and the vertical solid line, respectively. Assume a counter-clockwise orientation.

$$\int_D K(u,v) \, du \, dv = \frac{3}{\pi} \int_{\partial D} f(u,v) \, du + g(u,v) \, dv$$

$$= \frac{3}{\pi} \int_{-\sqrt{.75}}^{.5} f(u,v) \, du + \frac{1}{10\pi} \int_{\pi-\sin^{-1}(.5)}^{2\pi-\cos^{-1}(.5)} (5\theta - \sin 4\theta) \, d\theta + \frac{3}{\pi} \int_{-\sqrt{.75}}^{5} g(u,v) \, dv.$$
Once the inner double integral can be analytically evaluated for any fixed point \( s \), the outer double integral is then numerically evaluated using a Gaussian quadrature rule (see Abramowitz and Stegun [1]). Here, we use the 25-point Gaussian quadrature rule (5 quadrature nodes in each dimension). Let \( \alpha_m \) and \( \omega_m \) denote the \( m \)th quadrature point in a square defined by \([-1, 1] \times [-1, 1]\) and its corresponding quadrature weight. Let \( s_{mi} \) be the \( i \)th quadrature point in \( C_m \). Mapping the square of \([-1, 1] \times [-1, 1]\) to \( C_m \) leads to the following linear transformation:

\[
\begin{bmatrix}
\frac{d_m}{2} & 0 \\
0 & \frac{d_m}{2}
\end{bmatrix}
\begin{bmatrix}
\alpha_i \\
c_m
\end{bmatrix} + \frac{d_m}{2}.
\]

where \( c_m \) and \( d_m \) are the centroid and edge length of \( C_m \), respectively. Since the Jacobian matrix of this transformation has determinant of \( \|C\|^2/4 \), it follows

\[
\|C\|^{-1} \int_{C_m} \frac{\tilde{O}_p \int_{C_p} K_h(x-s) \, dx}{\sum_q \tilde{O}_q \int_{C_q} K_h(x-s) \, dx} \, ds \approx \frac{1}{4} \sum_i w_i \frac{\tilde{O}_p \int_{C_p} K_h(x-s_{mi}) \, dx}{\sum_q \tilde{O}_q \int_{C_q} K_h(x-s_{mi}) \, dx}.
\]

We implement the computation of the smoothing matrix using the programming language C, and the C code is included in Appendix A.3. Note that the C program only outputs nonzero entries of the smoothing matrix as well as their corresponding row–column index. We then have to convert the output into a SparseM object, representing the sparse smoothing matrix.

### 4.4.5 Cross Validation

Leave-one-map-out cross validation is used to choose the optimal bandwidth value that minimizes the prediction error defined as the sum of squared differences between the observed and predicted values. Since \( \tilde{O}_\ell = \sum_{ijk} T_{ijkl} O_{ijk} \), excluding a map alters \( \tilde{O}_\ell \). Therefore, the cross validation requires the re-calculation of the smoothing matrix each time a map is left out. This step is extremely computationally involved since we are now required to construct four smoothing matrices of \( 206,957 \times 206,957 \) for each bandwidth value. We limit the
construction of the smoothing matrix to fourteen equally spaced bandwidth values. The fourteen bandwidth values are 150, 300, \ldots, 1950 and 2100. Moreover, geographical areas that the four maps represent are different as a result of city expansion. We restrict the estimation of prediction error over census tracts common to all four maps. There are in total 2,689 common census tracts. More specifically, we calculate the prediction error using the following formula:

\[
\text{PE}(h) = \frac{1}{4} \sum_{j \in \ell'} \left( N_{j\ell'} - O_{j\ell'} \hat{\Lambda}_{j\ell'}^{(-j)}(h) \right)^2,
\]

where \( \ell' \) is the index of the common census tracts.

### 4.4.6 Results and Discussion

As shown in Figure 4.4, a local likelihood risk is estimated with \( h = 1350 \), the bandwidth value that achieves the smallest prediction error based on the leave-one-out cross validation presented in Table 4.1. Using the two-stage modeling, we have identified several areas with elevated risk for SLE after accounting for the effects of gender and age group as well as the time trend. Overlaying the estimated disease map with major road arteries in the GTA using a geographical information system software allows us to geographically visualize the estimated relative risk, thus facilitating model interpretations. Figure 4.4 shows the resulting disease map. Because most of the observed cases occur in Toronto downtown, the relative risk is nearly flat and zero outside the downtown area. Consequently, we only show the area with interesting features. The area with the highest disease risk is near the intersection of Jarvis and Bloor Street, where the estimated relative risk is about 3. In other words, the expected SLE cases in this area is approximately three times higher than what we would expect if there is no spatial variation in disease risk.

Having consulted with clinicians, we doubt the disease map in Figure 4.4 is an accurate representation of the disease risk of SLE due to two types of biases. The two biases are
Table 4.1: The bandwidth that gives the smallest prediction error is chosen to be the optimal value. As shown in the table above, the prediction error is the smallest with the bandwidth of 1350.

omitted-variable and selection bias. First, there exists a clinically demonstrated predisposition of SLE in African Canadians, but the data do not contain ethnicity information. While there are many African Canadians living in two nearby communities, namely Regent park and old Cabbage town, it is not surprising that we observe more SLE cases in the communities with high proportion of African Canadians. As a result, the elevated risk may be induced by the population composition, instead of the physical environment.

As for the selection bias, note that, since not all patients with SLE in the GTA are referred to the lupus clinic. Depending on patients’ disease severity and the distance between their residences and the lupus clinic, some may be treated by their primary care physicians and some are referred to nearby community rheumatologists. Since a patient usually visits a primary care physician in his/her neighbourhood, we suspect that primary care physicians who practiced in close proximity to the lupus clinic were more likely to refer their patients to the clinic than others who practiced far away. As a result, most patients’ residences in the dataset are located in the downtown area. As a final note, we did not know where the clinic was located before the analysis. Having analyzed the data, we discovered that the lupus clinic had been located in the the dark-purple area before it was moved to Toronto Western Hospital.
Figure 4.4: The GLM-adjusted pixel relative risk estimated by the local EM. The highest risk estimated equals approximately 3, indicating that the expected number of SLE cases in the area is three times higher than what the expected would be if there is no spatial variation.

4.5 Simulation Study

A simulation study of 100 samples is conducted to assess the performance of the proposed local likelihood risk estimator in a simplified scenario. Here, we assume only one age-gender group and no temporal variations in population density. Each simulated sample consists of two set of observations on the same geographical area, denoted by \( \mathcal{M} \). Outer boundaries of the geographical area are identical, but the area is divided differently. More specifically, the first map consists of five vertically stacked horizontal rectangles, and the second map consists of five horizontally juxtaposed vertical rectangles. All regions have the same area of 5 squared-units. Overlaying these two maps forms 25 unit-squared pixels.

Given the region boundaries, disease cases are simulated in the following steps. First, a population is simulated according to a Poisson process with a piecewise constant intensity function. Specifically, the regions in the first map, from top to bottom, have intensity of 18, 28, 38, 28 and 18 per squared-unit. Likewise, the regions, from the left to right of the
second map, have intensity of 18, 28, 38, 28 and 18 per squared-unit. The absolute risk \( \rho(x) \) is set to be the product of two rescaled gamma density functions with the shape and scale parameters of 1.5 and 0.5, respectively. The risk function attains its maximum of 1 at \((0.25, 0.25)\). Next, for each simulated observation \( x_i \), a case label is randomly generated with the probability of \( \rho(x) \). Finally, total population and the number of cases by region are reported as a pair, and each simulated sample consists of 10 pairs of regional case count and population. Note that, in the simulation,

\[
O_i(x) = \left| R_{ij} \right|^{-1} P_{ij} \theta \quad \text{and} \quad v(x) = \frac{\rho(x)}{\int_{\mathcal{M}} \rho(x) \, dx / \left| \mathcal{M} \right|}.
\]

Here, \( \theta \) is interpreted as average incidence rate and equal to \( 1/25 \).

A Gaussian kernel and a sequence of 201 equally spaced bandwidth values are used in the simulation study. The bandwidth value ranges from 0 to 2. We compute three risk estimates for each simulated sample at each bandwidth value. First, we calculate a kernel risk estimate based on exact case locations with regional population as an offset. As for the spatially aggregated data, we compute the local likelihood risk estimate in §4.3 as well as a NPMLE using an EM algorithm. Since the NPMLE is only unique up to an equivalence class, we apply kernel smoothing to the NPMLE by placing its masses at the centre of the corresponding pixel. We refer to this estimator as a smoothed nonparametric risk estimator. Similar to the simulation study for panel count data in Chapter 3, we compare the three risk estimators on the basis of the mean integrated squared error

\[
\text{MISE} \equiv \mathbb{E}_v \left[ \int_{\mathcal{M}} \left( \hat{v}(u) - v(u) \right)^2 \, du \right].
\]

For each risk estimator, the MISE is estimated by averaging integrated squared errors of the 100 simulated samples at each bandwidth value and plotted in Figure 4.5.

As shown in Figure 4.5, the proposed local likelihood risk estimator attains a lower MISE
than the smoothed nonparametric risk estimator. The MISE of the kernel risk estimator is the lowest among all three because it is based on exact case locations, instead of the aggregated data. Finally, the bandwidth value that the proposed local likelihood risk estimator achieves its lowest MISE is 0.19, compared with 0.35 for the smoothed nonparametric risk estimator and the kernel risk estimator. Since aggregating or binning is a smooth operation, it is not surprising that the local likelihood risk estimator attains its lowest MISE with a smaller bandwidth value than the kernel estimator.

4.6 Conclusions

In Silverman et al. [38], the authors’s primary concern was with image reconstruction involving a single image with unit offsets. It was not clear how one could incorporate covariates into their methodological framework. As demonstrated in this chapter, local likelihood provides us with a natural framework to extend the applicability of the EM algorithm by allowing multi-scaled and misaligned maps/images as well as permitting the analysis of covariates. As a result, the local EM method in the application of disease mapping could be thought as an extension of the image reconstruction technique of Silverman et al. [38] to an epidemiological setting. Furthermore, although Silverman et al. did not advocate any particular type of smoothing matrix in their 1990 paper, the smoothing matrix of integrated kernels arises naturally from the local likelihood setting.
Figure 4.5: The MISE of the smoothed nonparametric risk estimator is higher than that of the local likelihood risk estimator and kernel risk estimator for all bandwidth values shown in the plot. The local likelihood risk estimator achieves the lowest overall MISE with a small bandwidth value of 0.19. The kernel risk estimator achieves the lowest MISE among the three risk estimators since it is based on exact case locations.
Chapter 5

Convergence and Optimality

What we have accomplished so far is to expose a previously unknown relationship between local EM and the EMS algorithm. This relationship has several important practical and theoretical implications. The local likelihood, on one hand, endows the EMS with a natural context. This suggests that we should no longer consider the EMS as *ad hoc*, as previously suggested by Silverman *et al.* [38]. On the other hand, the EMS algorithm may serve as a convenient implementation of the local EM in practice. More importantly, the EMS provides us with theoretical tools to better understand the convergence and optimality of the local EM algorithm, which was difficult to study previously.

Chapter 5 consists of three sections. In §5.1, we aim to strengthen the suggestion that local EM and the penalized likelihood may be paired in a manner analogous to the more familiar pair of EM and likelihood. We reinforce the analogy in three steps. First, we show that employing a specific kernel in the local likelihood results in Nychka’s [28] modified EMS algorithm that maximizes the penalized likelihood (1.9). As foreshadowed in Chapter 3 and 4, we next prove that the *r*th iterate of the EMS implementation approaches its local EM counterpart in the $L^1$ norm as the number of pixels increases and the maximum pixel size shrinks toward zero. The $L^1$ convergence also holds for the modified EMS iterate. Having
modified EMS $\Rightarrow$ distance between $\theta$ and the eigenspace of the maximal eigenvector of $\mathcal{K}_h$

§5.1.2 $\downarrow$ §5.1.1 $\uparrow$ §5.1.3 $\downarrow$ §5.1.3

Local EM $\leadsto$ distance between $\theta$ and $\mathcal{Z}_\infty$

Figure 5.1: The diagram illustrates how the results in §5.1 are related to reinforce the suggestion that local EM and the penalized likelihood may be paired in a manner analogous to the more familiar pairing of EM and likelihood.

considered the limiting behaviour of the penalty term, we speculate that the role of local EM is to penalize the usual nonparametric likelihood for the departure of the target function from the functional space spanned by the maximal eigenfunction of the kernel, denoted by $\mathcal{Z}_\infty$. The diagram in Figure 5.1 summarizes and illustrates how the results in the first section are connected. Finally, we revisit the example of bivariate interval-censored data in Chapter 2 to illustrate how local EM penalizes the nonparametric likelihood when different kernel functions are used. Henceforth, we should no longer consider the local EM algorithm as a black box in the sense that the local likelihood estimate in the presence of interval- and area-censored data can be interpreted as a nonparametric maximum penalized likelihood estimate.

Following §5.1, we shift the focus and study the convergence of local EM algorithms. Latham [24] shows that the fixed-point solution of an EMS algorithm is unique. This implies that, if the local EM algorithm is convergent, its iteration will converge to the unique fixed-point solution. However, to establish the global convergence of the local EM algorithm is not easy, and we rely on results, regarding the convergence of the EMS algorithm, given in Green [17]. Since the EM mapping is not a contraction over the entire parameter space, as mentioned in Silverman et al. [38], we study the rate of convergence by examining the spectral radius of the EMS mapping in the neighbour of the unique fixed-point solution. In particular, we derive
an upper bound for the spectral radius. In the case of density estimation, we show that the upper bound shrinks toward zero as the bandwidth value increases. This, in turn, provides us insights into the algorithmic convergence of the local EM in the context of interval-censored data. As for intensity and risk estimation, we can only conclude that the local EM algorithm is locally convergent based the proposition of Green [17] as well as the limiting behaviour of the upper bound.

5.1 Role of Local EM

In this section, we present results meant to strengthen the suggestion that local EM and penalized likelihood may be paired in a manner analogous to the pairing of the EM and likelihood for the context considered in this thesis. We focus primarily on the locally constant case as details are easier to follow.

5.1.1 Modified EMS Algorithm and Equivalent Kernel

Nychka [28] points out an interesting relationship between the EMS algorithm and penalized likelihood. Specifically, the author showed that, given a smoothing matrix, a modified EMS algorithm can be used to maximize the penalized likelihood. By exploring the notion of equivalent kernel in Green and Silverman [16], we use a specific kernel in the local EM such that the corresponding EMS algorithm is equivalent to Nychka’s modified EMS algorithm, hence maximizing the penalized likelihood.

Consider the following nonparametric likelihood for the spatially aggregated data:

$$L(v) = \sum_{ij} N_{ij} \log \left( \int_{R_{ij}} O_i(x)v(x) \, dx \right) - \sum_i \int_{\Omega_i} O_i(x)v(x) \, dx. \quad (5.1)$$

Following Nychka [28], we begin by first examining the situation in which $n = 1$ and $O(x) = 1$
for $x \in \mathcal{M}$. In this case, $\lambda \equiv \nu$. Reparameterizing the nonparametric likelihood (5.1) with respect to the square root of $\lambda$ results in the following penalized likelihood:

$$L_p(\theta) = L(\theta) - \theta^T R \theta,$$

where $L(\theta) = \sum_j N_j \log \left( \sum_k I_{jk} \theta_k^2 \right) - \sum_k \theta_k^2$. $R = (S^{-1} - I)$, and $S$ is any symmetrical smoothing matrix. Next, we consider the following function

$$\frac{1}{\lambda(u)}^{1/2} K_h(u - x),$$

where $\lambda$ is the true intensity. Here, we assume that $K_h(u - x)$ is any symmetric positive kernel with compact support. The renormalization of the function (5.2) is written as

$$K^*_h(u - x) = \left( \frac{\lambda(x)}{\lambda(u)} \right)^{1/2} K_h(u - x)$$

and, for any interior point $x^1$,

$$\int_{\mathcal{M}} K^*_h(u - x) \, du = 1 + o(h),$$

where $\mathcal{M}$ is the support of the kernel $K(z)$. We refer to $K^*_h$ as an equivalent kernel. Rather than the usual kernel $K_h$, consider implementing the local EM algorithm using the equivalent kernel with the partition $\mathcal{C}$. Calculating the conditional expectation $E_{\lambda^*} [K^*_h(X - x) \mid X \in R_{ij}]$

---

1 A point, $x$, is an interior point if $\int_{\mathcal{M}} K_h(u - x) \, du = 1$
with respect to the $C$-approximant of $\hat{\lambda}^r$ results in

$$
E_{\hat{\lambda}^r} \left[ K_h^r(X - x) \mid X \in R_j \right]
= \|C\|^{-1} \sum_k \int_{C_k} \left( \frac{\lambda(x)}{\lambda(u)} \right)^{1/2} K_h(u - x) \, du \frac{T_{jk} \hat{\lambda}_k^r}{\sum_m T_{jm} \hat{\lambda}_m^r}
\approx \|C\|^{-1} \sum_k \int_{C_k} \left( \frac{\hat{g}^r(x)}{\hat{g}^r(u)} \right)^{1/2} K_h(u - x) \, du \frac{T_{jk} \hat{\lambda}_k^r}{\sum_m T_{jm} \hat{\lambda}_m^r}
= \|C\|^{-1} \sum_k \left( \frac{\hat{\Lambda}_k^r}{\hat{\Lambda}_k} \right)^{1/2} \int_{C_k} K_h(u - x) \, du \frac{T_{jk} \hat{\lambda}_k^r}{\sum_m T_{jm} \hat{\lambda}_m^r}
$$

for $x \in J_\ell$. Substituting the expression above into the local EM iteration (3.5) leads to the following adaptive EMS iteration for $\Lambda$:

$$
\hat{\Lambda}_{\ell}^{r+1} = \|C\|^{-1} \sum_{jk} \left( \frac{\hat{\Lambda}_k^r}{\hat{\Lambda}_k} \right)^{1/2} \int_{C_{\ell}} \int_{\mathfrak{M}} K_h(u - x) \, du \, dx \frac{N_j T_{jk} \hat{\lambda}_k^r}{\sum_m T_{jm} \hat{\lambda}_m^r} \quad \text{for } \ell = 1, \ldots, K. \quad (5.3)
$$

When $h$ is sufficiently small or $\mathfrak{M} = \mathfrak{R}^2$, $\int_{\mathfrak{M}} K_h^r(u - x) \, du \approx 1$ for all interior points. In this case, the iteration (5.3) is approximately equivalent to

$$
\hat{\Lambda}_{\ell}^{r+1} = \|C\|^{-1} \sum_{jk} \left( \frac{\hat{\Lambda}_k^r}{\hat{\Lambda}_k} \right)^{1/2} \int \int_{C_{\ell}} K_h(u - x) \, du \, dx \frac{N_j T_{jk} \hat{\lambda}_k^r}{\sum_m T_{jm} \hat{\lambda}_m^r},
$$

or, in matrix form,

$$
\hat{\Lambda}_{\ell}^{r+1} = M(\hat{\Lambda}_k^r) \mathcal{K}_h(\hat{\Lambda}_k^r). \quad (5.4)
$$

$\mathcal{K}_h(\hat{\Lambda}_k^r) = (\hat{\Theta}_k^r)^{-1} K_h \hat{\Theta}_k^r$, where $\hat{\Theta}_k^r = \text{diag} \left( \hat{\theta}_k^r \right)$. The EMS algorithm (5.4) is the same as Nychka’s modified EMS algorithm with $S = \mathcal{K}_h$. Therefore, the fixed-point solution of (5.4) must maximize a penalized likelihood function upon the convergence. Note that, in density estimation, $\int_{\mathfrak{M}} K_h^r(u - x) \, du \approx 1$ for all $h > 0$ when $\mathfrak{M} = \mathfrak{R}$.

When $n > 1$, consider the local EM algorithm in the spatial context. Since the offset
is not necessarily smooth, an equivalent kernel should be constructed based on the smooth component, namely $v$. Let $\theta_k^2 = v_k$. The corresponding equivalent kernel is then given by

$$K_h^*(u - x) = \left( \frac{v(x)}{v(u)} \right)^{1/2} K_h(u - x).$$

Following the same strategy, we can derive a version of (5.4) in terms of $\Upsilon$, such as

$$\hat{\upsilon}_{r+1}^\ell = \|C\|^{-1} \sum_k \left( \frac{\hat{\upsilon}_k^\ell}{\hat{\upsilon}_k^r} \right)^{1/2} \int_{C_k} \int_{C_k} K_h(u - x) \, du \, dx \sum_{ij} N_{ij} \mathcal{I}_{ijk} \hat{\upsilon}_k^r \sum_{m} \mathcal{I}_{ijm} \hat{\upsilon}_m^r,$$

or

$$\hat{\Upsilon}_{r+1} = \mathcal{M}(\hat{\Upsilon}^r) K_h^*(\hat{\Upsilon}^r).$$

When $h$ is small, $\sum_k \hat{\Theta}_k \int_{C_k} K_h^*(u - x) \, du$ is approximately equal to $\hat{\Theta}_k$ if $x \in C_k$ and zero otherwise. In this case, an approximated version of the iteration above is written as

$$\hat{\upsilon}_{r+1}^\ell = \|C\|^{-1} \sum_k \left( \frac{\hat{\upsilon}_k^\ell}{\hat{\upsilon}_k^r} \right)^{1/2} \int_{C_k} \int_{C_k} K_h(u - x) \, du \, dx \sum_{ij} N_{ij} \mathcal{I}_{ijk} \hat{\upsilon}_k^r \sum_{m} \mathcal{I}_{ijm} \hat{\upsilon}_m^r,$$

or

$$\hat{\Upsilon}_{r+1} = \mathcal{M}(\hat{\Upsilon}^r) K_h^*(\hat{\Upsilon}^r).$$

Again, $K_h^*(\hat{\Upsilon}^r) = \left( \hat{\Theta}^r \right)^{-1} K_h \hat{\Theta}^r$. Let $\hat{\Theta} = \text{diag}(\hat{\Theta}_k)$. As shown in Appendix A.2, the iteration (5.5) maximizes the penalized likelihood $L_p(\theta)$ with

$$L(\theta) = \sum_{ij} N_{ij} \log \left( \sum_k \mathcal{O}_{ij} \mathcal{I}_{ijk} \theta_k^2 \right) - \sum_{ij} \mathcal{O}_{ij} \mathcal{I}_{ijk} \theta_k^2$$

and $R = \left( K_h^{-1} - \hat{\Theta} \right)$. (5.6)

In summary, the use of the equivalent kernel in the local EM leads to an adaptive EMS algorithm, the algorithm that relates local EM with penalized likelihood.
5.1.2 Convergence of EMS to Local EM

Recall that \( C = \{ C_1, \ldots, C_K \} \) with \( \| C_k \| = \| C \| \) for all \( k \). In this section, we demonstrate that the iterates of the EMS implementation in the locally constant case converges to its local EM counterpart in \( L^1 \) as \( K \to \infty \). Denote the \( r \)th iterate of \( v \) in the local EM algorithm (4.6) and its corresponding EMS implementation (4.7) by \( \hat{\upsilon}_r^\infty \) and \( \hat{\upsilon}_r^K \), respectively. Recall that

\[
\hat{\upsilon}_r^{r+1}(x) = \sum_{ij} N_{ij} E_{\upsilon_r^\infty} [K_h(X - x) \mid X \in R_{ij}] \frac{\sum_i \int_{\mathcal{M}_i} \mathcal{O}_i(u) K_h(u - x) \, du}{\sum_i \int_{\mathcal{M}_i} \mathcal{O}_i(u) K_h(u - x) \, du}, \quad \text{and}
\]

\[
\hat{\upsilon}_r^K(x) = \sum_k \int_{J_k} K_h(u - x) \, du / \| J_k \| \frac{\sum_i \int_{\mathcal{M}_i} \mathcal{O}_i(u) K_h(u - x) \, du}{\sum_i \int_{\mathcal{M}_i} \mathcal{O}_i(u) K_h(u - x) \, du} \sum_{ij} N_{ij} I_{ijk} \hat{\upsilon}_r^k \sum_{im} I_{ijm} \hat{\upsilon}_r^m.
\]

Assume \( C = \{ C_1, \ldots, C_K \} \) with \( \| C_k \| = \| C \| \) for all \( k \). Let \( K(z) \) be a symmetric positive kernel with compact support and \( K_h(\cdot) = K(\cdot/h)/h \) for \( h > 0 \). The kernel integrates to one.

In addition, we assume \( \mathcal{M} \) to be compact with \( |\mathcal{M}| < \infty \) and \( R_{ij} \subseteq \mathcal{M} \) for all \( i, j \). Finally, define a norm on \( \mathcal{M} \) to be the \( L^1 \) norm, i.e. \( \| v \|_1 = \int_{\mathcal{M}} |v(u)| \, du \). We interpret \( f_K \xrightarrow{L^1} f \) to mean the \( L^1 \)-convergence of the function \( f_K \) to the limiting function \( f \) or \( \| f_K - f \|_1 \to 0 \) as \( K \to \infty \).

**Theorem 5.1.**

I. Define \( \mathcal{F}_1 = \{ v \in L^1(\mathcal{M}) \mid v \text{ is nonnegative with } v(x) > 0 \text{ for some } x \} \). For a common initial value \( \hat{\upsilon}_0 \in \mathcal{F}_1 \), we have, for all \( r = 1, 2, \ldots, \)

A. \( \hat{\upsilon}_K^r \xrightarrow{L^1} \hat{\upsilon}_r^\infty \) as \( K \to \infty \), and

B. \( \hat{\upsilon}_K^r, \hat{\upsilon}_r^\infty \in \mathcal{F}_1 \).

II. When the equivalence kernel of \( 5.1.1 \) is used, we instead define

\[
\mathcal{F}_2 = \{ v \in L^1(\mathcal{M}) \mid v \text{ is nonnegative with } v(x) > 0 \text{ for some } x \text{ and bounded} \}.
\]
For a common initial value $\hat{v}_0 \in \mathcal{F}_2$, the results $A$ and $B$ still hold for all $r = 1, 2, \ldots$, where $\mathcal{F}_1$ is replaced with $\mathcal{F}_2$ in $B$.

Let $O(u) = \sum_i O_i(u)$ and denote the square root of $v$ by $v^{1/2}$. Define $H_x$ to be $H_x : L^1 \mapsto L^1$ such that
\[
H_x(f) = \frac{\int_{\mathbb{R}} K_h(u - x) f(u) \, du }{\int_{\mathbb{R}} O(u) K_h(u - x) \, du}.
\]

Here, we assume that, for a fixed $h$,
\[
\int_{\mathbb{R}} O(u) K_h(u - x) \, du \geq c > 0, \quad \forall x \in \mathcal{M}.
\]

It is a reasonable assumption for the applications considered in this thesis. For example, in density estimation, $O(u) = 1$ and $\int_{\mathbb{R}} K_h(u - x) \, du = 1 > 0$. In intensity estimation, $O(u)$ is replaced with $Y(u) > 0$. Since $Y(u) \geq 1$ for all $u \in \mathcal{M}$,
\[
\int_{\mathbb{R}} Y(u) K_h(u - x) \, du \geq \int_{\mathbb{R}} K_h(u - x) \, du > 0.
\]

In risk estimation, the inequality holds as long as $O_i(x)$ is bounded away from 0 for all $x \in \mathcal{M}$. The proof of Theorem 5.1 relies on $H_x(u)$ being a bounded linear functional as well as some other basic results in operator theory stated as lemmas below. Proofs of the following lemma that may be found in Royden [34].

**Lemma 5.1.** Let $\nu \in L^1$. Then the $C$-approximant of $\nu$ converges in $L^1$ to $\nu$ on $\mathcal{M}$ as $K \to \infty$; that is, $g_\nu \xrightarrow{C^1} \nu$.

**Lemma 5.2.** If $\int_{\mathbb{R}} O(u) K_h(u - x) \, du \geq c > 0$, then $H_x$ is a linear bounded functional for all $f \in L^1$. That is, for all $x, a, b \in \mathbb{R}$, $H_x(af + b) = aH_x(f) + b$, and there exists a real number $M_h$ such that $H_x(f) \leq M_h \|f\|_1$.

**Lemma 5.3.** Let $\gamma_h(u, x) = g(x) K_h(u - x) f(u)$, where $f, g \in L^1$ and $K_h \in L^\infty$. Then
\( \gamma(u,t) \) is an \( \mathcal{L}^1 \) function on \( \mathbb{R}^4 \) with
\[
\iint |\gamma_h(u,x)| \, du \, dx \leq M_h \cdot \|g\|_1 \cdot \|f\|_1.
\]

**Main Result**

*Proof.* Consider a fixed \( h \) and \( n \). Let \( \bar{\upsilon}_r^x(x) = g_{\bar{\upsilon}_r^x}(x;\mathcal{C}) \).

1. Let \( r = 1 \). By the definition of \( \bar{\upsilon}_r^0 \), \( \int_{R_{ij}} \bar{\upsilon}_r^0(x) \, dx = \int_{R_{ij}} \hat{v}_0(x) \, dx \). Repeated use of the triangle inequality gives
\[
\left| \int_{\mathbb{R}^M} \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\upsilon}_0(u) \, du \right)^{-1} \frac{\iint_{R_{ij}} K_h(u-x)[\hat{\upsilon}_0(u) - \hat{\upsilon}_\infty(u)] \, du}{\iint_{R_{ij}} \mathcal{O}(u)K_h(u-x) \, du} \right| \, dx
\]
\[
\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\upsilon}_0(u) \, du \right)^{-1} \iint_{\mathbb{R}^M} \iint_{\mathbb{R}^M} \frac{K_h(u-x)}{\mathcal{O}(u)K_h(u-x)} |\hat{\upsilon}_0(u) - \hat{\upsilon}_0(u)| \, du \, dx
\]
\[
\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\upsilon}_0(u) \, du \right)^{-1} \iint_{\mathbb{R}^M} \iint_{\mathbb{R}^M} \frac{K_h(u-x)}{\mathcal{O}(u)K_h(u-x)} |\hat{\upsilon}_0(u) - \hat{\upsilon}_0(u)| \, du \, dx
\]
\[
\leq \|\mathcal{M}\| \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\upsilon}_0(u) \, du \right)^{-1} \left( \int_{R_{ij}} \hat{\upsilon}_0(u) \, du \right) \cdot M_h \|\bar{\upsilon}_0 - \hat{\upsilon}_0\|_1.
\]

The last inequality is due Lemma 5.2. By Lemma 5.1, the fact that \( \bar{\upsilon}_0 \overset{C^1}{\to} \hat{\upsilon}_0 \) implies \( \hat{\upsilon}_r^1 \overset{C^1}{\to} \hat{\upsilon}_\infty^1 \). Moreover, the positivity of the kernel ensures that \( \hat{\upsilon}_r^1 \) and \( \hat{\upsilon}_\infty^1 \) both belong to the class \( \mathcal{F}_1 \).
\textit{Induction Step:} Assume that $\hat{\nu}_K^r \xrightarrow{L^1} \hat{\nu}_\infty^r$ and $\hat{\nu}_K^r, \hat{\nu}_\infty^r \in \mathcal{F}_1$. Let $b_{ij}^r = \frac{\int_{R_{ij}} \hat{\nu}_\infty^r(v) \, dv}{\int_{R_{ij}} \hat{\nu}_K^r(v) \, dv}$. With the repeated use of the triangle inequality, we have

$$\|\hat{\nu}_K^{r+1} - \hat{\nu}_\infty^{r+1}\|_1 \leq \sum_{ij} N_{ij} \int_{\mathcal{M}} \int_{R_{ij}} \frac{K_h(u - x)}{\mathcal{O}(u)} K_h(u - x) \, du \left| \left( \frac{\int_{R_{ij}} \hat{\nu}_K^r(u) \, dv}{\int_{R_{ij}} \hat{\nu}_K^r(u) \, dv} - \frac{\hat{\nu}_\infty^r(u)}{\int_{R_{ij}} \hat{\nu}_\infty^r(v) \, dv} \right) \right| \, du \, dx$$

$$\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\nu}_\infty^r(v) \, dv \right)^{-1} \int_{\mathcal{M}} \int_{\mathcal{M}} \frac{K_h(u - x)}{\mathcal{O}(u)} K_h(u - x) \, du \left| b_{ij}^r \hat{\nu}_K^r(u) - \hat{\nu}_\infty^r(u) \right| \, du \, dx$$

$$\leq \|\mathcal{M}\| \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\nu}_\infty^r(v) \, dv \right)^{-1} M_h \left\| b_{ij}^r \hat{\nu}_K^r - \hat{\nu}_\infty^r \right\|_1.$$

The last inequality is again due to Lemma 5.2. Since the induction assumption and Lemma 5.1 imply $\hat{\nu}_K^r \xrightarrow{L^1} \hat{\nu}_\infty^r$ and $b_{ij}^r \to 1$, we have, for all $i, j$,

$$\left\| b_{ij}^r \hat{\nu}_K^r - \hat{\nu}_\infty^r \right\|_1 \to 0.$$

In addition, it is evident that $\hat{\nu}_K^{r+1}$ and $\hat{\nu}_\infty^{r+1}$ belong to $\mathcal{F}_1$ provided that $\hat{\nu}_K^r, \hat{\nu}_\infty^r \in \mathcal{F}_1$. Hence, we have (A) $\hat{\nu}_K^{r+1} \xrightarrow{L^1} \hat{\nu}_\infty^{r+1}$ on $\mathcal{M}$, and (B) $\hat{\nu}_K^{r+1}, \hat{\nu}_\infty^{r+1} \in \mathcal{F}_1$ by induction.
II. Let $r = 1$. By the triangle inequality and Lemma 5.3, we have

$$\| \hat{\upsilon}^1_K - \hat{\upsilon}^1_\infty \|_1 \leq \sum_{ij} N_{ij} \int_{R_{ij}} \int_{\mathfrak{M}} \frac{K_h(u - x)}{\mathcal{O}(u)K_h(u - x) du} \left| \frac{[\hat{\upsilon}^0(u)\hat{\upsilon}^0(x)]^{1/2} - [\hat{\upsilon}^0(u)\hat{\upsilon}^0(x)]^{1/2}}{\int_{R_{ij}} \hat{\upsilon}^0(v) dv} \right| du \, dx$$

$$\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\upsilon}^0(v) dv \right)^{-1} \left\{ \int_{\mathfrak{M}} \int_{\mathfrak{M}} \frac{K_h(u - x)}{\mathcal{O}(u)K_h(u - x) du} \left[ [\hat{\upsilon}^0(x)]^{1/2} [\hat{\upsilon}^0(x)]^{1/2} - [\hat{\upsilon}^0(x)]^{1/2} \right] du \, dx \\
+ \int_{\mathfrak{M}} \int_{\mathfrak{M}} \frac{K_h(u - x)}{\mathcal{O}(u)K_h(u - x) du} \left[ [\hat{\upsilon}^0(x)]^{1/2} [\hat{\upsilon}^0(x)]^{1/2} - [\hat{\upsilon}^0(x)]^{1/2} \right] du \, dx \right\}$$

$$\leq \| \mathfrak{M} \| \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{\upsilon}^0(v) dv \right)^{-1} M_h \left( \| (\hat{\upsilon}^0)^{1/2} \|_1 + \| (\hat{\upsilon}^0)^{1/2} \|_1 \right) \| (\hat{\upsilon}^0)^{1/2} - (\hat{\upsilon}^0)^{1/2} \|_1$$

The last inequality is due to Lemma 5.3. Moreover, the boundedness of $\upsilon$ implies that $\| (\hat{\upsilon}^0)^{1/2} \|_1 < \infty$ and $\| (\hat{\upsilon}^0)^{1/2} \|_1 < \infty$. Now by Lemma 5.1 and the continuous mapping theorem (CMT), $\| (\hat{\upsilon}^0)^{1/2} - (\hat{\upsilon}^0)^{1/2} \|_1 \rightarrow 0$. Therefore, $\hat{\upsilon}_K^1 \xrightarrow{\mathcal{L}^1} \hat{\upsilon}_\infty^1$. In addition, $\hat{\upsilon}_K^1$ and $\hat{\upsilon}_\infty^1$ are obviously bounded. Therefore, $\hat{\upsilon}_K^1, \hat{\upsilon}_\infty^1 \in \mathcal{F}_2$.

**Induction Step:** Assume that $\hat{\upsilon}_K^r \xrightarrow{\mathcal{L}^1} \hat{\upsilon}_\infty^r$ and $\hat{\upsilon}_K^r, \hat{\upsilon}_\infty^r \in \mathcal{F}_2$. The induction assumption and the CMT immediately imply that

$$c_{ij}^r = \frac{\int_{R_{ij}} \hat{\upsilon}_\infty^r(v) dv}{\int_{R_{ij}} \hat{\upsilon}_K^r(v) dv} \rightarrow 1 \text{ for all } i, j \text{ and } (\hat{\upsilon}_K^r)^{1/2} \xrightarrow{\mathcal{L}^1} (\hat{\upsilon}_\infty^r)^{1/2} \text{ on } \mathfrak{M}.$$
Similar to part (I), Lemma 5.1 and 5.3 imply that, as $K \to \infty$ and $\max_j \|C_j\| \to 0$,

$$\|\hat{v}_{r+1}^r - \hat{v}_\infty^r\|_1$$

$$\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{v}_\infty^r(x) \, dx \right)^{-1}$$

$$\int_{\Omega} \int_{R_{ij}} \frac{K_h(u-x)}{\mathcal{O}(u)K_h(u-x)} \, du \left| c_{ij} \left[ \hat{v}_K^r(u) \hat{v}_K^r(x) \right]^{1/2} - \left[ \hat{v}_\infty^r(u) \hat{v}_\infty^r(x) \right]^{1/2} \right| \, du \, dx$$

$$\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{v}_\infty^r(x) \, dx \right)^{-1}$$

$$\int_{\Omega} \int_{R_{ij}} \frac{K_h(u-x)}{\mathcal{O}(u)K_h(u-x)} \, du \left\{ c_{ij} \left( \hat{v}_K^r(x) - \hat{v}_\infty^r(u) \right) \right\}^{1/2}$$

$$+ \left| c_{ij} - 1 \right| \left[ \hat{v}_\infty^r(u) \hat{v}_K^r(x) \right]^{1/2} + \left[ \hat{v}_\infty^r(u) \hat{v}_\infty^r(x) \right]^{1/2}$$

$$\leq \sum_{ij} N_{ij} \left( \int_{R_{ij}} \hat{v}_\infty^r(x) \, dx \right)^{-1} M_h \left[ c_{ij} \left\| (\hat{v}_K^r)^{1/2} \right\|_1 \cdot \left\| (\hat{v}_K^r)^{1/2} - (\hat{v}_\infty^r)^{1/2} \right\|_1 \right.$$  

$$+ \left| c_{ij} - 1 \right| \left\| (\hat{v}_K^r)^{1/2} \right\|_1 \cdot \left\| (\hat{v}_\infty^r)^{1/2} \right\|_1 + \left\| (\hat{v}_\infty^r)^{1/2} \right\|_1 \cdot \left\| (\hat{v}_K^r)^{1/2} - (\hat{v}_\infty^r)^{1/2} \right\|_1 \right] \to 0.$$  

Provided that $\hat{v}_K^r$ and $\hat{v}_\infty^r$ are bounded, $\hat{v}_{r+1}^r$ and $\hat{v}_{r+1}^\infty$ are bounded. It follows that

(A) $\hat{v}_K^r \xrightarrow{\mathcal{L}^1} \hat{v}_\infty^r$, and (B) $\hat{v}_K^r$, $\hat{v}_\infty^r \in \mathcal{F}_2$ by induction.

$\square$
5.1.3 Local EM and Penalized Likelihood

In this section, we study the penalized likelihood of §5.1.1 under the conditions of §5.1.2. For the sake of clarity, assume $\mathcal{M}_i = \mathcal{M}$ and $O_{ij} = 1$ for all $i$ and $j$. Recall that the smoothing matrix of the EMS implementation in the locally constant case.

$$[\mathcal{K}_h]_{kj} = |C|^{-1} \int_{C_j} \int_{C_k} \frac{K_h(u-x)}{\mathcal{M}_k} K_h(u-x) \, du \, dx.$$

Let $I_K$ be a $K$-by-$K$ identity matrix. We begin by considering the values of $\theta$ for which the penalty $\theta^T R \theta$ is minimized. For such $\theta$, we have

$$R \theta = (\mathcal{K}_h^{-1} - I_K) \theta = 0$$

or rather

$$\theta = \mathcal{K}_h \theta. \quad (5.7)$$

This permits an interpretation of $\mathcal{L}_p(\theta)$ as penalizing the nonparametric likelihood on the basis of the proximity of $\theta$ to the maximal eigenvector of the smoothing matrix $\mathcal{K}_h$. To see this, let $\varrho(\ell)$ denote the $\ell$th largest eigenvalue of $\mathcal{K}_h$ with its corresponding eigenvector $\gamma(\ell)$.

Let $\Gamma = \begin{bmatrix} \gamma(K) & \gamma(K-1) & \cdots & \gamma(1) \end{bmatrix}$. Then the spectral decomposition of $R$ is $\Gamma D \Gamma^T$, where $D = \text{diag} \left( \varrho(1)^{-1} - 1, \varrho(2)^{-1} - 1, \ldots, \varrho(K)^{-1} - 1 \right)$. Since $\varrho(K) > \cdots > \varrho(1) \geq 1$, $R$ penalizes eigenvectors with small eigenvalues more than those with large ones.

We note that the $\theta$ satisfying Equation 5.7 also satisfies

$$\theta^T \theta - \theta^T \mathcal{K}_h \theta = 0. \quad (5.8)$$

We may consider the limiting behaviour of the left-hand side as $K \to \infty$. $[\mathcal{K}_h]_{jk}$ and $\theta_j$ may
be approximated as

$$[K_h]_{kj} = \delta^{-1} \int_{C_j} \int_{C_k} \frac{K_h(u-x)}{\int_{\mathcal{M}} K_h(u-x) \, du} \, dx \approx \frac{K_h(x_j - x_k)}{\int_{\mathcal{M}} K_h(u-x_j) \, du} \delta$$

$$\theta_j^2 = \left( \int_{C_j} v(u) \, du \right)^{1/2} \left( \int_{C_j} v(x) \, dx \right)^{1/2} = v(x_j) \delta$$

Therefore, we can rewrite the left-hand side of Equation 5.8 as

$$\sum_j v(x_j) \delta - \sum_{jk} v^{1/2}(x_j) \frac{K_h(x_j - x_k)}{\int_{\mathcal{M}} K_h(u-x_j) \, du} v^{1/2}(x_k) \delta \delta.$$  

If we let $\delta \downarrow 0$, the above expression becomes

$$\int_{\mathcal{M}} v(u) \, du - \int_{\mathcal{M}} \int_{\mathcal{M}} v^{1/2}(x) \frac{K_h(u-x)}{\int_{\mathcal{M}} K_h(u-x) \, du} v^{1/2}(u) \, du \, dx.$$

As a result, the penalty will equal 0 for any function $z$ belonging to the following class:

$$\mathcal{Z}_\infty = \left\{ z \left| z^{1/2}(x) = \int_{\mathcal{M}} \frac{K_h(u-x)}{\int_{\mathcal{M}} K_h(u-x) \, du} z^{1/2}(u) \, du \text{ for all } x \in \mathcal{M} \right. \right\}.$$  

Given this and the result of §5.1.2, we speculate that the local EM algorithm maximizes the nonparametric likelihood on the basis of the proximity of the function $v$ to the maximal eigenspace $\mathcal{Z}_\infty$. 
5.2 Convergence of Local EM Algorithm

5.2.1 Fixed-Point Solution

Consider the local EM algorithm with a locally constant polynomial in the spatial context.

\[
\hat{\mathbf{Y}}^{r+1} = \mathcal{M}(\hat{\mathbf{Y}}^r)K_h.
\]  

(5.9)

Our objective is to allow the algorithm to iterate until its iterations converge to a fixed-point solution, \(\hat{\mathbf{Y}}^*\). We rely on the results given in Latham [24] to demonstrate the uniqueness of the fixed-point solution. Since the smoothing matrix in (5.9) is independent of \(\mathbf{Y}\), Latham [24] shows the fixed-point solution is unique in the parameter space where \(v_k > 0\) for all \(k\). This implies that, if the EMS implementation in (5.9) is convergent at all, its iterations will converge to the unique fixed-point solution \(\hat{\mathbf{Y}}^*\). In other words, the local likelihood estimate is unique upon the convergence of (5.9). This result is consistent with the observation made in §2.4 of Braun, Duchesne and Stafford [6]. Note that the convergence of (5.9) does not guarantee that \(\hat{\mathbf{Y}}^*\) maximizes any penalized likelihood functions.

5.2.2 Rate of Convergence

Define the spectral radius of a vector-valued function from \(\mathbb{R}^K\) to \(\mathbb{R}^K\) to be the largest eigenvalue of its Jacobian matrix. The spectral radius is useful because a sufficient condition for a locally convergent algorithm is that the mapping of the algorithm has a spectral radius less than one at a fixed-point solution (see, e.g. Ortega [29]). Let \(\gamma(\mathbf{Y})\) be the spectral radius of the EMS mapping at \(\mathbf{Y}\). By the Perroni-Frobenius theorem,

\[
\gamma(\mathbf{Y}) \leq \max_s \sum_t [\partial_t \mathcal{M} K_h]_{ts},
\]
where $\partial M$ is a $K \times K$ matrix with

$$[\partial M]_{tk} = \frac{\partial M_k}{\partial v_t} = \begin{cases} \sum_{ij} \frac{N_{ij} \sum_{t \neq k} T_{ijk} T_{ij \ell} v_{\ell}}{\hat{O}_k (\sum_{\ell} T_{ij \ell} v_{\ell})^2} & \text{when } k = t \\ \sum_{ij} \frac{N_{ij} - T_{ijt} T_{ij k} v_k}{\hat{O}_k (\sum_{\ell} T_{ij \ell} v_{\ell})^2} & \text{otherwise} \end{cases}$$

It follows that

$$\gamma(\mathbf{Y}) \leq \max_s \sum_t [\partial M \mathcal{K}_h]_{ts} = \max_s \sum_k \sum_k [\partial M]_{tk} [\mathcal{K}_h]_{ks} = \max_k \sum_k [\mathcal{K}_h]_{ks} \sum_t [\partial M]_{tk}$$

Consider the local EM method for density estimation. In this context, our aim is twofold. First, we show that, for a fixed $\mathbf{Y}$, the upper bound can be made as small as possible by increasing $h$. Second, given a bounded parameter space, the local EM algorithm is globally convergent when $h$ is sufficiently large.
By expressing (5.10) in terms of \( p \), we have

\[
\sum_k \left| \sum_i \frac{1}{\|J_k\|} \sum_{t \neq k} I_{tk} I_{it} (p_t - p_k) \right| \frac{\max_s \int_{J_s} \int_{J_k} K_h(u - x) \, du \, dx}{(\sum_{t} I_{t} p_t)^2} \left( \sum_{t} I_{t} I_{t}^T \right) = \sum_k \left| \sum_{i} \frac{1}{\|J_k\|} \sum_{t \neq k} I_{tk} I_{it} (p_t - p_k) \right| \frac{\max_s \int_{J_s} \int_{J_k} K_h(u - x) \, du \, dx}{(\sum_{t} I_{t} I_{t}^T)} \left( \sum_{t} I_{t} I_{t}^T \right).
\] (5.11)

Since the kernel is symmetrical,

\[
\max_s \int_{J_s} \int_{J_k} K_h(u - x) \, du \, dx = \int_{J_k} \int_{J_k} K_h(u - x) \, du \, dx
\]

\[
= \int_{J_k} \int_{(tk - x)/h}^{tk - x)}/h} K(z) \, dz \, dx
\]

\[
= \|J_k\| \int_{(tk - x)}/h}^{tk - x)}/h} K(z) \, dz \quad \text{with some} \ x^* \in J_k. \quad (5.12)
\]

The last equality follows because of the mean-value theorem for integration.\(^2\) We obtain the following by taking the limit:

\[
\lim_{h \to \infty} \|J_k\| \int_{(tk - x)}/h}^{tk - x)}/h} K(z) \, dz = \lim_{h \to \infty} \|J_k\| \left( \int_{-\infty}^{(tk - x)}/h} K(z) \, dz - \int_{-\infty}^{(tk - x)/h} K(z) \, dz \right)
\]

\[
= \|J_k\| \left( \int_{-\infty}^{0} K(z) \, dz - \int_{-\infty}^{0} K(z) \, dz \right) = 0
\]

Thereby, the upper bound shrinks toward 0 as \( h \) increases. The shrinking spectral radius, in turn, accelerates the convergence of the local EM algorithm. Moreover, the upper bound

\(^2\)The mean value theorem for integration states that if \( f \) is continuous, then there exists a number \( c \in [a, b] \) such that

\[
\int_{a}^{b} f(u) \, du = f(c)(b - a).
\]
(5.11) is a decreasing function of $h$. To see this, suppose $0 < h_1 < h_2$. Then

\[
\int_{J_s} \int_{J_k} K_{h_1}(u - x) \, du \, dx - \int_{J_s} \int_{J_k} K_{h_2}(u - x) \, du \, dx
\]

\[
= \int_{J_s} \left( \int_{(t_k-x)/h_1}^{(t_k-x)/h_1} K(z) \, dz - \int_{(t_k-x)/h_2}^{(t_k-x)/h_2} K(z) \, dz \right) \, dx
\]

\[
= \int_{J_s} \left( \int_{(t_k-x)/h_1}^{(t_k-x)/h_1} K(z) \, dz + \int_{(t_k-x)/h_2}^{(t_k-x)/h_2} K(z) \, dz \right) \, dx \geq 0
\]

This implies that, provided that the parameter space is bounded, there exits an $H > 0$ such that, for all $h > H$, the upper bound (5.11) is less than one for all $p$. In this case, the EMS mapping in (5.9) is a contraction mapping, and the local EM algorithm is globally convergent in the parameter space, as conjectured in Braun, Duchesne and Stafford [6]. Note that the spectral radius does not need to be less than 1 for a convergent algorithm since it is not a necessary condition.

As for intensity estimation, the convergence of the local EM algorithm is more complicated for two reasons. First, the parameter space is not bounded. Second, the upper bound in (5.10) does not shrink toward zero in general, nor is it a monotone function of $h$. To see this, consider the local EM method for intensity estimation in the one dimensional case. Here, the upper bound can be written in terms of $\Lambda$, as follows.

\[
\sum_k \left| \sum_{i,j} \frac{N_{ij} \sum_{t \neq k} T_{ijk} T_{ij}(\Lambda_t - \Lambda_k)}{\|J_k\|^2 (\sum_{t} T_{ij} \Lambda_t)^2} \right| \max_s \int_{J_s} \frac{\int_{J_k} K_h(u - x) \, du}{\sum_{t} Y(t) \int_{J_k} K_h(u - x) \, du} \, dx. \tag{5.13}
\]

However, the term that involves the double integral in (5.13) does not vanish as $h$ increases.

\[
\int_{J_s} \frac{\int_{J_k} K_h(u - x) \, du}{\sum_{t} Y(t) \int_{J_k} K_h(u - x) \, du} \, dx = \int_{J_s} \frac{\int_{(t_k-x)/h}^{(t_k-x)/h} K(z) \, dz}{\sum_{t} Y(t) \int_{(t_k-x)/h}^{(t_k-x)/h} K(z) \, dz} \, dx
\]

\[
= \frac{\|J_s\| \int_{(t_k-x)/h}^{(t_k-x)/h} K(z) \, dz}{\sum_{t} Y(t) \int_{(t_k-x)/h}^{(t_k-x)/h} K(z) \, dz}. \tag{5.14}
\]
Again, the last equality follows due to the mean-value theorem for integration. By taking the limit, we have

$$\lim_{h \to \infty} \frac{\|J_s\| f_{(t_k-x_a)/h}(t_{k-1-x_a})/h K(z) dz}{\sum_{t} Y(t)} = \lim_{h \to \infty} \frac{\|J_s\| \left( \int_{-\infty}^{(t_k-x_a)/h} K(z) dz - \int_{-\infty}^{(t_{k-1-x_a})/h} K(z) dz \right)}{\sum_{t} Y(t) \left( \int_{-\infty}^{(t-x_a)/h} K(z) dz - \int_{-\infty}^{(t_{-1-x_a})/h} K(z) dz \right)}$$

$$= \lim_{h \to \infty} \frac{\|J_s\| \left( -K(\frac{t_k-x_a}{h})h - K(\frac{t_{k-1-x_a}}{h})(\frac{t_{k-1-x_a}}{h^2}) \right)}{\sum_{t} Y(t) \left( -K(\frac{t-x_a}{h})h - K(\frac{t_{-1-x_a}}{h})(\frac{t_{-1-x_a}}{h^2}) \right)}$$

$$= \frac{\|J_s\|\|J_s\|K(0)}{\sum_{t} Y(t)} = \frac{\|J_s\|\|J_s\|}{\sum_{i} \mathcal{M}_i}$$

(5.15)

since $\sum_{t} Y(t)\|J_s\| = \sum_{ij} Y_{ij} \sum_{t} I_{ijt} \|J_s\| = \sum_{ij} Y_{ij} \|I_{ijt}\| = \sum_{i} \mathcal{M}_i$. Note that the second line of (5.15) is due to l’Hôpital’s rule. By substituting (5.15) into (5.13), we have

$$\sum_k \left| \sum_{ij} N_{ij} \frac{\sum_{t \neq k} I_{ijk} I_{ijt}(\Lambda_t - \Lambda_k)}{(\sum_{t} I_{ijt} \Lambda_t)^2} \right| \max_s \left( \int_{J_k} \frac{\sum_{s} Y(t)}{\sum_{t} Y(t)} \int_{J_k} K_h(u-x) du \right) dx$$

$$\rightarrow \sum_k \left| \sum_{ij} \frac{N_{ij}}{\mathcal{M}_i} \frac{\sum_{t \neq k} I_{ijk} I_{ijt}(\Lambda_t - \Lambda_k)}{(\sum_{t} I_{ijt} \Lambda_t)^2} \right| \max_s \|J_s\| \text{ as } h \to \infty. \quad (5.16)$$

In this situation, we can only demonstrate that the local EM algorithm is locally convergent based on the result given by Green [17]. Let $\gamma$ denote the spectral radius of the EMS mapping in (5.9) at the fixed-point solution $\hat{\Upsilon}^*$ of (5.9). In this case, Green [17] shows that $\gamma < 1$ for all $h > 0$. This implies that local EM iterations converge at least in the neighbourhood of $\hat{\Upsilon}^*$. Our experience suggests that, when $h$ is sufficiently large, the iteration of the local EM algorithm never fails to converge. In addition, the local EM requires fewer steps than the EM algorithm to meet convergence criteria. Similar results hold for risk estimation in the context of Chapter 4.

\[\text{We also observe that the convergence of the EM is quicker than that of the local EM algorithm for some small } h > 0. \text{ A sufficient condition for a local EM converging at least as quickly as the EM algorithm is given in Green [17]. The condition is that the spectral radius of the EM mapping at } \hat{\Upsilon}^* \text{ is greater than } 1/2.\]
5.3 Bivariate Interval-Censored Data

Let us revisit the hypothetical example of the bivariate interval-censored data in Chapter 2. We applied local EM to model the distribution function in the analysis. While Maathuis [27] uses the height-map algorithm to identify maximal intersections for dimension reduction, the local EM method does not need the prior identification of the maximal intersection. Subsequently, we proceeded with parameter estimation without knowing the whereabouts of the maximal intersection. Moreover, while the NPMLE in this example is not identifiable in the sense that there are multiple NPMLE’s (see, for example, the three NPMLE’s shown in Figure 2.2), we made the suggestion that, provided a fixed kernel function, the iteration of the EMS implementation always converges to the same fixed-point solution regardless of initial values. In addition, we found that employing different kernel functions leads us to prefer one NPMLE to the others. The results in this chapter help us gain a better understanding of the phenomena that we observed in Chapter 2.

As demonstrated in §5.1, The local EM algorithm aims to maximize the penalized likelihood, and the penalty in this case depends on the orientation and symmetry of the kernel. For example, when the kernel is radially symmetrical, any deviations from the maximal eigenfunction are equally penalized. However, as the kernel becomes more elliptical, deviations in the direction of the major axis of the elliptical contour are penalized more than those in the direction of the minor axis. Consequently, the density estimate along the major axis is smoother than the minor axis. Since the penalty term can be interpreted as the log of the prior density in the Bayesian framework, the kernel function acts as a prior density and the choice of the kernel leads us to favour one NPMLE to another a priori. In addition, provided that $h$ is sufficiently large, the upper bound of the spectral radius is less than one for all $p > 0$ in the parameter space. The convergence of the algorithm and the uniqueness of the fixed-point solution explain why the iteration of the local EM algorithm always converges to the same solution.
Not only does this example illustrate the role of local EM, but it also highlights how one may use local EM algorithms to overcome difficulties in nonparametric maximum likelihood estimation.
Chapter 6

Concluding Remarks

A careful comparison of our local EM with the EMS algorithm of Silverman et al. [38] permits further insights beyond what has already been discussed. Silverman et al. in their 1990 paper suggested the EMS as an alternative method to address issues related to indirectly observed data in the sense that the data are contaminated according to an integral equation,

\[ h(x) = \int K(x, y)f(y)\,dy, \quad (6.1) \]

where \( K(x, y) \) is a known conditional density of \( X \) given \( y \). In the paper, Silverman et al. were interested in estimating \( f \), and the authors’ attempts to solve the integral equation by discretizing \( f \) led to the consideration of the EMS algorithm. Their version of the EMS algorithm is related to our implementation of the local EM, but they are different in terms of the conditional density \( K(x, y) \).

Silverman et al. [38] referred to quantities analogous to \( R_{ij} \)’s and \( J_\ell \)’s as observation and reconstruction bins, respectively, and defined a weight to be

\[ w_{ijk} = |J_k|^{-1} \int_{J_k} \int_{R_{ij}} K(x, y)\,dx\,dy. \]
The weight specifies how likely an event in $J_k$ is observed in $R_{ij}$. We may also interpret the weight $w_{ijk}$ as the amount of overall influence that an event in $J_k$ exerts on the observation in $R_{ij}$. In contrast to our EMS implementation (4.8), we have $I_{ijk}$’s in place of $w_{ijk}$’s. Here, $w_{ijk} = I_{ijk}$ only if $K(x, y)$ is a Dirac delta function. For example, let $K(x, y)$ be the Dirac delta function. If $I_{ijk} = 1$, then $J_k \subseteq R_{ij}$ and

$$w_{ijk} = \|J_k\|^{-1} \int_{J_k} \int_{R_{ij}} K(x, y) \, dx \, dy = \|J_k\|^{-1} \int_{J_k} 1 \, dy = 1.$$  

Likewise, if $I_{ijk} = 0$, $J_k \not\subseteq R_{ij}$ and $w_{ijk} = 0$. With $K(x, y)$ being the Dirac delta function, the integral equation becomes $h(x) = f(x)$. This implies that the interval-censored and spatially aggregated data that we have considered in the thesis are not indirectly observed in the sense of Silverman et al. [38].

The above observation provides us with an insight into extending local EM techniques to analyze indirectly observed data or mismeasured data in a clinical or epidemiological setting. For instance, glomerular filtration rate (GFR) is a useful measure to assess the health of the kidneys. If the GFR falls below a normal range, it is an early indication of substantial renal damages. Since the kidneys are most susceptible to disease activities of SLE, it is of clinical interest to determine the time to the first instance that the GFR dips below the normal range. A SLE patient is often subject to periodic medical testing to determine his/her renal health, and a sequence of test results from the patient is recorded. Should we assume the setting considered in Chapter 3, a negative test followed by a positive test indicates that the GFR falls below the normal threshold between the two test dates. However, it would only be the case if the measuring technique is perfect. Since the gold standard procedure to measure the GFR is too costly, it is common to approximate the GFR by other techniques. It is expected that the techniques other than the gold standard have lower sensitivity and specificity. As a result, the test results are more prone to mismeasurement. Under the periodic observation scheme, not only is the time of the first instance interval-censored, but it is also mismeasured.
The conditional density $K(x, y)$ in this setting is determined by the sensitivity and specificity of the test, and it can be shown that $w_{ijk}$ equals the probability of having the observed test results conditional on the time interval in which the first instance occurs.

In the spatial context, we may consider residential history data that involve disease incidences as an example of mismeasured data. Suppose that the primary cause of a certain type of disease is the \textit{cumulative} exposure of risk that is associated with the physical environment. Since a subject may move from one area to another over time, his/her (accumulative) risk exposure varies, depending on when and where he/she resides. For example, if a subject is diagnosed with the disease while living at location $x$, it may not be desirable to model the disease process using only the risk exposure at $x$. When the subject’s residential history data are available, it is more sensible to model the process using the exposure of risk at locations where the subject had lived. In this case, we may think of the movement in space and time as a major cause of mismeasured risk exposure. In the situation in which it is difficult to directly measure the level of risk exposure that varies in space and time for each subject, we can use his/her residential history as a proxy. For example, we may use the residential history data to specify $K(x, y)$ for each subject. Let $H_i$ denote the $i$th subject’s residential history. Suppose $H_i = \{y_1, y_2, y_3, y_4\}$, where $y_\ell$ is the $\ell$th residence of the $i$th subject. Given $H_i$, we can specify a conditional density, denoted by $K_i(x, y)$. Provided that $x, y \in H_i$, $K_i(x, y)$ gives the likelihood that the onset of the disease takes place at location $x$ given the condition that subject $i$ lives in $y$. In practice, the specification of $K_i(x, y)$ may be model-based or given by expert knowledge regarding the nature of the disease.

In summary, we can analyze mismeasured data, such as residential history data, in the local EM framework while the EMS procedure in Silverman \textit{et al.} [38] offers a convenient implementation of local EM algorithms that account for the mismeasurement. As a result, we are able to further explore the use of local EM methods in a much broader context.

In the context considered in this thesis, we have exposed an important relationship be-
tween local EM and the EMS algorithm in the sense that local EM and the EMS algorithm complement each other. On one hand, local EM motivates the use of the EMS algorithm in the local likelihood framework and present the EMS as a non *ad hoc* method. Moreover, local likelihood offers a coherent mechanism to incorporate covariates as well as offsets into an EMS algorithm, thus expanding the applicability of the EMS method. On the other hand, not only does the EMS method offer significant computational advantage in the implementation of the local EM algorithm, it also provides us with the theoretical rigour that ultimately reinforces the suggestion that the pair of local EM and penalized likelihood is analogously thought as that of EM and likelihood. By the analogy, we demonstrate that the local EM algorithm is not a black box, and local likelihood estimates for interval- and area-censored data can be regarded as NPMPLE’s. Moreover, the application of the EMS method in image reconstruction further suggests how one can extend local EM to analyze mismeasured data. Finally, the local likelihood can be seen as a semi-parametric method, providing a compromise between the power of parametric methods and the flexibility of kernel-base methods. Local EM provides a method for applying local likelihood to analyze interval- and area-censored data. By connecting the local EM with the EMS algorithm, we hope that the computational advantage offered by the EMS will lead to greater adoption of local EM methods.
Appendix A

A.1 EMS Implementations of the Local EM Algorithm

Assume there are $n$ study subjects and $K$ elements in the partition $\mathcal{J}$. For each subject, let $\vec{N}_i$ be a row vector of $N_{ij}$'s and $\mathbf{M}_i$ be an incidence matrix, the $j\ell$th entry of which equals $\mathcal{I}(J_{\ell} \subseteq I_{ij})$. Define $\mathbf{Y}_i$ and $\mathbf{V}_i(\Lambda)$ to be

$$\mathbf{Y}_i = \text{diag}(Y_i(t_1), \cdots, Y_i(t_K))$$

and

$$\mathbf{V}_i(\Lambda) = \text{diag} \left( \sum_{\ell} \Lambda_\ell \mathcal{I}_{i1\ell}, \cdots, \sum_{\ell} \Lambda_\ell \mathcal{I}_{iK\ell} \right),$$

respectively. Let

$$\vec{N} = \begin{bmatrix} \vec{N}_1 & \cdots & \vec{N}_n \end{bmatrix} \quad \text{and} \quad \mathbf{M}^T = \begin{bmatrix} \mathbf{M}_1^T & \cdots & \mathbf{M}_n^T \end{bmatrix}.$$
Let $V(\Lambda)$ be a block diagonal matrix with the $i$th block equal to $V_i(\Lambda)$. Then the EMS iteration can be implemented in the following matrix manipulation:

$$
\hat{\Lambda}^{j+1} = \tilde{N}V^{-1}(\hat{\Lambda}^j) M \text{diag}(\hat{\Lambda}^j) \left( \sum_i Y_i \right)^{-1} K_h(\hat{\Lambda}^j). 
$$

(A.1)

In the case of binary process, the iteration (A.1) becomes

$$
\hat{p}^{j+1} = n^{-1} \tilde{1}_n V^{-1}(\hat{p}^j) M \text{diag}(\hat{p}^j) K_h(\hat{p}^j),
$$

where $\tilde{1}_n$ is a row vector of $n$ 1’s.

When the polynomial is truncated at the leading term, the smoothing matrix $K_h$ is free of $\Lambda$. Hence, we only have to calculate the smoothing matrix once, and the calculation is fairly straightforward. However, whenever the order of the polynomial is greater than 0, updating the smoothing matrix at each iteration becomes necessary. The following example describes in detail the EMS implementation of the local EM algorithm in the locally linear case with a Gaussian kernel.

**Locally Linear Example:** Let $P(z) = a_0 + a_1 z$. Recall that

$$
\Psi(t; \hat{\alpha}^{j+1}) = \sum_{\ell} Y(t_\ell) \int_{J_\ell} K_h(u-t)e^{\hat{\alpha}^{j+1}(u-t)} du.
$$

The EMS implementation in this case requires numerically integrating the function

$$
B_j(t; K_h, \hat{\alpha}^{j+1}) = \frac{\int_{J_j} K_h(u-t) du}{\Psi(t; \hat{\alpha}^{j+1})}
$$

over the $J_j$’s for all $j = 1, \ldots, K$. Since the inner integral usually has an analytical form, its calculation is not difficult. Following the evaluation of the inner integral, a numerical procedure, or rather a quadrature rule, is used to calculate the outer integral since its analytical
solution is often not available.

Before applying any quadrature rules, we need to determine \( \hat{\mathbf{a}}^{t+1} \) at each \( t \in \mathcal{M} \). Let \( \mathbf{a}_t = [a_0(t) \ a_1(t)]^T \) for a fixed \( t \). Consider the E-step (3.1) with \( \hat{\lambda}^r \) being replaced by \( \hat{g}^r \).

\[
Q_t(\lambda_t \mid \hat{g}^r) = \sum_{ij} N_{ij} E_{ij} \left[ K_h(T-t)(a_0(t) + a_1(t)(T-t)) \mid T \in I_{ij} \right]
- \sum_i \int_0^\infty Y_i(u) K_h(u-t) \exp(a_0(t) + a_1(t)(u-t)) \, du
= a_0(t) \sum_{ij} N_{ij} E_{ij} \left[ K_h(T-t) \mid T \in I_{ij} \right] + a_1(t) \sum_{ij} N_{ij} E_{ij} \left[ K_h(T-t)(T-t) \mid T \in I_{ij} \right]
- e^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} K(z) e^{a_1(t)hz} \, dz,
\]

where \((J_\ell - t)/h = [(t_{\ell-1} - t)/h, (t_\ell - t)/h]\). Differentiating with respect to \( \mathbf{a}_t \) results in the following system of score equations:

\[
0 = \nabla Q_t = \left[ \begin{array}{c}
\sum_{ij} N_{ij} E_{ij} \left[ K_h(T-t) \mid T \in I_{ij} \right] - e^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} K(z) e^{a_1(t)hz} \, dz \\
\sum_{ij} N_{ij} E_{ij} \left[ K_h(T-t)(T-t) \mid T \in I_{ij} \right] - he^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} zK(z) e^{a_1(t)hz} \, dz
\end{array} \right]
\]

For any given \( t \), to obtain a numerical solution for \( \mathbf{a}_t \) inevitably requires a root-finding algorithm or an inner loop. Index the inner loop by \( \iota \). We solve the system above using a Newton-Raphson algorithm that requires second-order partial derivatives or the Hessian matrix. Let \( \mathcal{H}_t \) denote the Hessian matrix and

\[
\mathcal{H}_t(\mathbf{a}_t) = \left[ \begin{array}{cc}
-he^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} K(z) e^{a_1(t)hz} \, dz & -he^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} zK(z) e^{a_1(t)hz} \, dz \\
h^2 e^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} z^2 K(z) e^{a_1(t)hz} \, dz & -h^2 e^{a_0(t)} \sum_\ell Y(t_\ell) \int_{J_{\ell-1}^h} z^2 K(z) e^{a_1(t)hz} \, dz
\end{array} \right].
\]

Provided a set of evaluation points, denoted by \( e_m \)'s, the Newton-Raphson iteration can be
expressed as

\[ \hat{A}^{j+1,ι+1} = \hat{A}^{j+1,ι} - \left[ \begin{array}{ccc} \mathcal{H}_e(\hat{a}_1^{j+1,ι}) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{H}_m(\hat{a}_m^{j+1,ι}) \end{array} \right]^{-1} \left[ \begin{array}{c} \nabla Q_e(\hat{a}_1^{j+1,ι}) \\ \vdots \\ \nabla Q_m(\hat{a}_m^{j+1,ι}) \end{array} \right], \tag{A.2} \]

where \( A^T = [a^T_1 \cdots a^T_m] \), a row vector of the polynomial coefficients ordered by the evaluation points. Upon the convergence of (A.2), we set \( \hat{A}^{j+1} \) to be \( \hat{A}^{j+1,∞} \).

Next, define

\[ m(t; a, K) = \int_{-∞}^{a} e^{tz} K(z) \, dz. \]

This function is recognized as the incomplete moment generating function for a random variable with the density \( K(z) \). It can be shown that \( m'(t; a, K) = \int_{-∞}^{a} z e^{tz} K(z) \, dz \) and \( m''(t; a, K) = \int_{-∞}^{a} z^2 e^{tz} K(z) \, dz \). Assume a Gaussian kernel. Let \( \phi_\mu(x) \) denote the Gaussian density function with the mean \( \mu \) and standard deviation of 1 and \( \Phi_\mu(x) = \int_{-∞}^{x} \phi_\mu(s) \, ds \).

Using the properties of the incomplete moment generating function, it can be shown that

\[
\begin{align*}
m(t; a, \phi) &= e^{t^2/2} \Phi_t(a) \\
m'(t; a, \phi) &= e^{t^2/2} [t \Phi_t(a) - \phi_t(a)] \\
m''(t; a, \phi) &= e^{t^2/2} [(1 + t^2) \Phi_t(a) - (a + t) \phi_t(a)]
\end{align*}
\]

It follows that the integrals in \( Q_t \) and \( \mathcal{H}_t \) have analytical forms. For example,

\[
\int_{t_{\ell-1} \over h}^{t_\ell \over h} e^{a_1hz} K(z) \, dz = m \left( a_1h; \frac{t_\ell - t}{h}, \phi \right) - m \left( a_1h; \frac{t_{\ell-1} - t}{h}, \phi \right) \\
= e^{(a_1h)^2/2} \left[ \Phi_{a_1h} \left( \frac{t_\ell - t}{h} \right) - \Phi_{a_1h} \left( \frac{t_{\ell-1} - t}{h} \right) \right]
\]
\[
\int_{J_{\ell}^{-1}} z e^{a_1 h z} K(z) \, dz = m'(a_1 h; \frac{t_\ell - t}{h}, \phi) - m'(a_1 h; \frac{t_\ell - 1 - t}{h}, \phi) = \left\{ a_1 h \left[ \Phi_{a_1 h} \left( \frac{t_\ell - t}{h} \right) - \Phi_{a_1 h} \left( \frac{t_\ell - 1 - t}{h} \right) \right] - \left[ \phi_{a_1 h} \left( \frac{t_\ell - t}{h} \right) - \phi_{a_1 h} \left( \frac{t_\ell - 1 - t}{h} \right) \right] \right\}
\]

\[
\int_{J_{\ell}^{-1}} z^2 e^{a_1 h z} K(z) \, dz = m''(a_1 h; \frac{t_\ell - t}{h}, \phi) - m''(a_1 h; \frac{t_\ell - 1 - t}{h}, \phi) = \left\{ [1 + (a_1 h)^2] \left[ \Phi_{a_1 h} \left( \frac{t_\ell - t}{h} \right) - \Phi_{a_1 h} \left( \frac{t_\ell - 1 - t}{h} \right) \right] - \left( \frac{t_\ell - t}{h} + a_1 h \right) \phi_{a_1 h} \left( \frac{t_\ell - t}{h} \right) + \left( \frac{t_\ell - 1 - t}{h} + a_1 h \right) \phi_{a_1 h} \left( \frac{t_\ell - 1 - t}{h} \right) \right\}
\]

We can then solve for \( a_\ell \) by substituting the three formulae to (A.2).

What is left is to evaluate the outer integral. We use a \( q \)-point Gaussian quadrature rule\(^1\) to numerically integrate \( B_j(t; K_h, \hat{a}^{j+1}) \) over all \( J_\ell \)'s and for \( j = 1, \ldots, K \). Let \( \alpha_m \) and \( \omega_m \) denote the \( m \)th quadrature node between -1 and 1 and its corresponding weight, respectively. Let \( s_{\ell m} \) be the \( m \)th node of \( J_\ell \) and

\[
s_{\ell m} = \frac{\| J_\ell \|}{2} \alpha_m + \frac{t_\ell + t_{\ell - 1}}{2}
\]

The \( q \)-point Gaussian quadrature rule approximates the outer integral over \( J_\ell = [t_{\ell - 1}, t_\ell] \) as follows.

\[
\int_{t_{\ell - 1}}^{t_\ell} B_j \left( t; K_h, \hat{a}_t^{j+1} \right) \, dt = \frac{\| J_\ell \|}{2} \int_{-1}^{1} B_j \left( \frac{\| J_\ell \|}{2} u + \frac{t_\ell + t_{\ell - 1}}{2}; K_h, \hat{a}_u^{j+1} \right) \, du \approx \frac{\| J_\ell \|}{2} \sum_{m=1}^{q} \omega_m B_j(s_{\ell m}; K_h, \hat{a}_{s_{\ell m}}^{j+1}).
\]

\(^1\)Refer to Abramowitz and Stegun [1].
Denote the kronecker product by $\otimes$. Let $L = \text{diag}(\|J_k\|)$. It follows

$$K_h(\hat{\Lambda}) = \frac{1}{2} \text{diag} \left( \bar{O}_1, \ldots, \bar{O}_K \right) L^{-1} B \left[ L \otimes \bar{w} \right],$$

where $\bar{w}^T = \begin{bmatrix} \omega_1 & \cdots & \omega_m \end{bmatrix}$ and

$$B = \begin{bmatrix}
B_1(s_{11}; K_h, \hat{a}_{s_{11}}^{j+1}) & \cdots & B_1(s_{Kq}; K_h, \hat{a}_{s_{Kq}}^{j+1}) \\
\vdots & \ddots & \vdots \\
B_K(s_{11}; K_h, \hat{a}_{s_{11}}^{j+1}) & \cdots & B_K(s_{Kq}; K_h, \hat{a}_{s_{Kq}}^{j+1})
\end{bmatrix}.$$

\section{A.2 Adaptive EMS and Penalized Likelihood}

Recall that $\nu_k = \theta_k^2$ and $\Theta = \text{diag}(\theta_k)$. We claim in §5.1.1 that the fixed-point solution of the EMS implementation in (5.5) maximizes the penalized likelihood $L_p(\theta)$ with

$$L(\theta) = \sum_{ij} N_{ij} \log \left( \sum_k O_{ij} I_{ijk} \theta_k^2 \right) - \sum_{ijk} O_{ij} I_{ijk} \theta_k^2$$

and $R = \left( K_h^{-1} - \bar{O} \right)$.

To see this, consider rewriting the EMS implementation in (5.5) in terms of the notions introduced in the previous appendix. Additionally, let $\theta$ be a row vector of $\theta_k$’s. Note that $\text{diag}(\nu_k) = \Theta^2$ and $\Upsilon = \theta \Theta$. It follows

$$\hat{\theta}^{r+1} \hat{\Theta}^{r+1} = \bar{N} V^{-1}(\hat{Y}^r) M \bar{O}^{-1} \text{diag}(\hat{\nu}_k^r) \left( \hat{\Theta}^r \right)^{-1} K_h \hat{\Theta}^r$$

$$= \bar{N} V^{-1}(\hat{Y}^r) M \bar{O}^{-1} \hat{\Theta}^r K_h \hat{\Theta}^r.$$

On one hand, any fixed-point solution of (5.5) such that $\hat{Y}^* = \mathcal{M}(\hat{Y}^*) K_h^* \left( \hat{Y}^* \right)$, must also satisfy

$$\hat{\theta}^* \hat{\Theta}^* = \bar{N} V^{-1}(\hat{Y}^*) M \bar{O}^{-1} \hat{\Theta}^* K_h \hat{\Theta}^* \quad \text{or} \quad \hat{\theta}^* = \bar{N} V^{-1}(\hat{Y}^*) M \bar{O}^{-1} \hat{\Theta}^* K_h.$$
On the other hand, $\hat{\theta}^*$ solves the system of the penalized score equations since

\[
\nabla L^T_p = \left[ \frac{\partial L_p}{\partial \theta_1} \ldots \frac{\partial L_p}{\partial \theta_K} \right] = 2 \left( \frac{N}{V - 1} (\Upsilon) M \tilde{O} - \hat{\theta} \hat{\Theta} - \hat{\theta} R \right) = 2 \left( \frac{N}{V - 1} (\Upsilon) M \tilde{O} - \hat{\theta} \hat{\Theta} \hat{K}^{-1} \right)
\]

\[
\Rightarrow \left. \nabla L^T_p \right|_{\theta = \hat{\theta}^*} = \left[ 0 \ldots 0 \right]
\]

If $\hat{\theta}^*$ is the NPMPLE, the fixed-point solution of the EMS implementation in (5.5) also maximizes the penalized likelihood.

### A.3 R and C Code

#### A.3.1 Kernel Matrix Computation

The following code is used to calculate the kernel matrix with a bivariate biweight kernel in Chapter 4.

```c
#include <math.h>
#include <stdio.h>
#include <S.h>
#define THREE_PI (3*M_1_PI)
#define R64

#ifdef R64
typedef int intType;
#else
typedef long intType;
#endif

inline double ifx(double y, double lb, double ub);
inline double igy(double x, double lb, double ub);
inline double ifgxy(double lb, double ub);
double condA(double x, double y);
```
double condB(double x, double y);
double condC(double x, double y);
double condD(double x, double y);
double condE(double x, double y);
double condF(double x, double y);
double condG(double x, double y);
double condH(double x, double y);
double pbivarbiwgt(double x, double y);

extern "C" void smoother(double *cx,
                        double *cy,
                        intType *ncell,
                        double *quadPts,
                        double *quadWgt,
                        intType *nquadPts,
                        double *bandwidth,
                        double *cellwidth,
                        double *offsets,
                        double *resultValue,
                        intType *colNonzero,
                        intType *rowIndex,
                        intType *nNonzero) {

  intType rowCell, colCell, R, C;
  double x, y, xdist, ydist, dist, xleft, ybottom, xright, ytop, denominator, intwgt, tmp_integral;
  double maxdist = *bandwidth + 2*M_SQRT2*(cellwidth), halfcw = (*cellwidth)/2;
  double *tmp_innerInt, *tmp_outerInt;

  tmp_innerInt = (double *) S_alloc(*ncell, sizeof(double));
  tmp_outerInt = (double *) S_alloc(*ncell, sizeof(double));

  for (rowCell=0; rowCell<*ncell; rowCell++) {
    for (C=0; C<*nquadPts; C++) {
      for (R=0; R<*nquadPts; R++) {
        denominator = 0;
        intwgt = 0;
        x = cx[rowCell] + halfcw*quadPts[R];
        y = cy[rowCell] + halfcw*quadPts[C];
        intwgt = quadWgt[R]*quadWgt[C];

        ...
      }
    }
  }
}
for (colCell=0; colCell<*ncell; colCell++) {
    tmp_innerInt[colCell] = 0;
    xdist = cx[colCell] - x;
    if (fabs(xdist) > maxdist) {
        ydist = cy[colCell] - y;
        if (fabs(ydist) > maxdist) {
            dist = sqrt(pow(xdist, 2) + pow(ydist, 2));
            if (dist > maxdist) {
                xleft = (xdist - halfcw)/(*bandwidth);
                ybottom = (ydist - halfcw)/(*bandwidth);
                xright = (xdist + halfcw)/(*bandwidth);
                ytop = (ydist + halfcw)/(*bandwidth);
                tmp_integral = (pbivarbiwgt(xright, ytop) + pbivarbiwgt(xleft, ybottom)) -
                              (pbivarbiwgt(xleft, ytop) + pbivarbiwgt(xright, ybottom));
                if (tmp_integral > 0) {
                    tmp_innerInt[colCell] = tmp_integral;
                    if (offsets[colCell] > 0) denominator += (offsets[colCell]*tmp_integral);
                }
            }
        }
    }
}
for (colCell=0; colCell<*ncell; colCell++) {
    if (tmp_outerInt[colCell] > 0) {
        tmp_outerInt[colCell] += (tmp_innerInt[colCell]*intwgt/(4*denominator));
    }
}
for (colCell=0; colCell<*ncell; colCell++) {
    if (tmp_outerInt[colCell] > 0) {
        colNonzero[rowCell]++;
        rowIndex[*nNonzero] = colCell;
    }
}
resultValue[*nNonzero] = tmp_outerInt[colCell];
*nNonzero += 1;
}

// reset the outer integral to zero for the next loop.
tmp_outerInt[colCell] = 0;
}
}

inline double ifx(double y, double lb, double ub) {
  return(-1*((pow(y, 5)/5 - 2*pow(y, 3)/3 + y/2)*(ub-lb) + (pow(ub,3)-pow(lb,3))*pow(y, 3)/9));
}

inline double igy(double x, double lb, double ub) {
  return((pow(x,5)/5 - 2*pow(x, 3)/3 + x/2)*(ub-lb) + (pow(ub,3)-pow(lb,3))*pow(x, 3)/9);}

inline double ifgxy(double lb, double ub) {
  return((5*(ub-lb) -( sin(4*ub)-sin(4*lb) ))/30);
}

double condA(double x, double y) {
  double result;
  result = igy(x, -1*sqrt(1-pow(x, 2)), y) + ifx(y, x, -1*sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI-acos(x));
  return(THREE_PI*result);
}

double condB(double x, double y) {
  double result;
  result = igy(x, -1*sqrt(1-pow(x, 2)), sqrt(1-pow(x, 2))) + ifx(y, sqrt(1-pow(y, 2)), -1*sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI-acos(x));
  return(THREE_PI*result);
}

double condC(double x, double y) {
  double result;
  result = igy(x, -1*sqrt(1-pow(x, 2)), sqrt(1-pow(x, 2))) + ifgxy(acos(x), 2*M_PI-acos(x));
  return(THREE_PI*result);
}
double condD(double x, double y) {
    double result;
    result = ifx(y, sqrt(1-pow(y, 2)), -1*sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI+asin(y));
    return(THREE_PI*result);
}

double condE(double x, double y) {
    double result;
    result = igy(x, -1*sqrt(1-pow(x, 2)), y) + ifx(y, x, -1*sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI-acos(x));
    return(THREE_PI*result);
}

double condF(double x, double y) {
    double result;
    result = igy(x, -1*sqrt(1-pow(x, 2)), y) + ifx(y, x, -1*sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI-acos(x));
    return(THREE_PI*result);
}

double condG(double x, double y) {
    double result;
    result = igy(x, -1*sqrt(1-pow(x, 2)), sqrt(1-pow(x, 2))) + ifgxy(acos(x), 2*M_PI-acos(x));
    return(THREE_PI*result);
}

double condH(double x, double y) {
    double result;
    result = igy(x, -1*sqrt(1-pow(x, 2)), y) + ifx(y, x, -1*sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI-acos(x));
    return(THREE_PI*result);
}

double condi(double x, double y) {
    double result;
    result = ifx(y, sqrt(1-pow(y, 2)), -sqrt(1-pow(y, 2))) + ifgxy(M_PI-asin(y), 2*M_PI+asin(y));
    return(THREE_PI*result);
}

double pbivarbiwgt(double x, double y) {
    double xsq = pow(x, 2), sqa = sqrt(1-xsq), ans;

A.3.2 EMS Implementation

Gaussian Quadrature Rule

quad5pt<- function(J, JintervalWidth=diff(c(0, J)), nJ=length(J)) {
  absc<- c(-0.906179845938664,
       -0.538469310105683, 0.000000000000000, 0.538469310105683, 0.906179845938664);
  wgt<- c(0.236926885056189, 0.478628670499366, 0.568888888888889, 0.478628670499366, 0.236926885056189);
quad10pt<- function(J, JintervalWidth=diff(c(0, J)), nJ=length(J)) {
  absc<- c(-0.973906528517172,
    -0.86506366688985,
    -0.679409568299024,
    -0.433395394129247,
    -0.14887438891631,
    0.14887438891631,
    0.433395394129247,
    0.679409568299024,
    0.86506366688985,
    0.973906528517172);
  wgt<- c(0.066671344308688,
    0.149451349150581,
    0.219086362515982,
    0.269266719309996,
    0.295524224714753,
    0.295524224714753,
    0.269266719309996,
    0.219086362515982,
    0.149451349150581,
    0.066671344308688);

  nQuadpts<- length(absc);
  halfw <- .5*JintervalWidth;
  midpts<- .5*(J + c(0, J[-nJ])
  x<- rep(midpts, each=nQuadpts) + rep(halfw, each=nQuadpts)*rep(absc, nJ);
  return(list(nQuadpts=nQuadpts, lattice=x, wgt=wgt));
}

quad10pt<- function(J, JintervalWidth=diff(c(0, J)), nJ=length(J)) {
  x<- rep(midpts, each=nQuadpts) + rep(halfw, each=nQuadpts)*rep(absc, nJ);
  return(list(nQuadpts=nQuadpts, lattice=x, wgt=wgt));
}
Locally Constant Case for Panel Count Data

ems_constructor_constant <- function(panel.data, bandwidth, kernel="norm", quad.order=NULL, J=NULL) {

  ck <- paste("p", kernel, sep="")
  if (is.null(J)) J <- sort(union(panel.data$L, panel.data$R))
  J <- setdiff(J, c(0, Inf))
  nJinterval <- length(J)
  JintervalWidth <- diff(c(0, J))
  if (quad.order == 10) quadRule <- quad10pt(J)
  else quadRule <- quad5pt(J)
  ## Step 1: construct the incidence matrix
  censor.mat <- outer(tapply(panel.data$CT, panel.data$ID, unique), J, ">=");
  offsets <- apply(censor.mat, 2, sum)
  tmp.data <- panel.data[which(panel.data$M != 0),]
  incidence.mat <- outer(tmp.data$R, J, ">=") - outer(tmp.data$L, J, ">=");

  ## Step 2: construct the smoothing matrix at different x (for the final intensity estimate)
  lattice <- c(quadRule$lattice, (quadRule$lattice + max(J))]
  tmpK <- outer(J, lattice, function(x, y) eval(call(ck, q=(x-y)/bandwidth))) -
                  outer(c(0, J[-nJinterval]), lattice, function(x, y) eval(call(ck, q=(x-y)/bandwidth)))
  K_x.mat <- diag(offsets/JintervalWidth) %*% tmpK %*% diag(1/as.vector(offsets %*% tmpK))
  rm("tmpK");

  ## Step 3: construct the smoothing matrix using Gaussian quadrature rule.
  K.mat <- K_x.mat[, 1:length(quadRule$lattice)] %*% kronecker(diag(.5*JintervalWidth), quadRule$wgt)

  return(list(J=J,
              incidence.mat=incidence.mat,
              offsets=offsets,
              lattice=lattice,
              K_x.mat=K_x.mat,
              K.mat=K.mat))
}

local.constant.nhpp <- function(panel.data, bw, tol=1.0E-8, quad.order=10, partition=NULL) {
  ems_input <- ems_constructor_constant(panel.data, bandwidth=bw, quad.order=quad.order, J=partition)
  offsets <- ems_input$offsets
  ## I can ignore the panel with zero count ####
tmp.data<- panel.data[which(panel.data$M!=0),]

Lambda.old<- tmp.data$M %*% ems_input$incidence.mat %*% diag(1/offsets);
Lambda<- matrix(0, nrow=1, ncol= length(ems_input$J));
niter<- 1;
repeat {
  V.inv<- diag(1/as.vector(Lambda.old %*% t(ems_input$incidence.mat)));
  tmpEM<- tmp.data$M %*% V.inv %*% ems_input$incidence.mat %*% diag(as.vector(Lambda.old)) %*% diag(1/offsets)
  Lambda<- tmpEM %*% ems_input$K.mat;
  if (sum((Lambda.old-Lambda)^2) > tol) { Lambda.old<- Lambda; niter<- 1+niter; }
    else { lambda<- tmpEM %*% ems_input$K_x.mat; break; }
}
return(list(Lambda.est=as.vector(Lambda), lambda.est=as.vector(lambda), nIter=niter, lattice=ems_input$lattice));

Locally Linear for Panel Count Data

iMGF_gaussian<- function(ub, t=0) exp(.5*t^2)*pnorm(ub, mean=t)

iMGF_gaussian_first_derivative<- function(ub, t=0) {
  exp(.5*t^2)*(t*pnorm(ub, mean=t)-dnorm(ub, mean=t))
}

iMGF_gaussian_second_derivative<- function(ub, t=0) {
  exp(.5*t^2)*((1+t^2)*pnorm(ub, mean=t) - (ub+t)*dnorm(ub, mean=t))
}

### L and R are the left- and right-endpoints of those J intervals. ###
### lattice is a set of the evaluation points based on gaussian quadrature rule. ###
### entry1 and entry 2 come from HESSIAN.INV ###
score.fun<- function(a0, a1, lattice, bandwidth, L, R, em_estimate, entry1, entry2) {
  if (any(L>R)) {
    stop("Right endpoints have to be greater than left ones")
  }

  if (length(R)==length(L)) {
    nrow<- length(R)
    hessian.mat<- diag(0, 2*length(lattice))
    indexA0<- 2*(1:length(lattice))-1
    indexA1<- 2*(1:length(lattice))
} else {
    stop("Lengths of left- and right-end points are different.")
}

if (length(a0) != length(lattice)) {
    stop("Constant coefficient and grid points are of different length.")
}

if (length(a1) != length(lattice)) {
    stop("Linear coefficient and grid points are of different length.")
}

tmpA1<- rep(a1, each=nrows)*bandwidth;
### first entry of the score function ###
firstTerm<- outer(R, lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth)) -
        outer(L, lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth));
firstTerm<- as.vector(as.vector(em_estimate) %*%
        diag(1/(R-L)) %*% firstTerm) + as.vector(entry1);

### second entry of the score function ###
secondTerm<-
        outer(R, lattice, function(x, y) iMGF_gaussian_first_derivative(ub=(x-y)/bandwidth)) -
        outer(L, lattice, function(x, y) iMGF_gaussian_first_derivative(ub=(x-y)/bandwidth));
secondTerm<- as.vector(as.vector(em_estimate) %*%
        diag(bandwidth/(R-L)) %*% secondTerm) + as.vector(entry2);

score<- as.vector(rbind(firstTerm, secondTerm));
return(score);

hessian.fun<- function(a0, a1, lattice, bandwidth, L, R, offsets) {
    if (length(R)==length(L)) {
        nrows<- length(R);
        hessian.mat<- diag(0, 2*length(lattice));
        indexA0<- 2*(1:length(lattice))-1;
        indexA1<- 2*(1:length(lattice));
    } else stop("Lengths of left- and right-end points are different.")
    }
if (length(a0) != length(lattice)) {
    stop("Constant coefficient and grid points are of different length.")
}

if (length(a1) != length(lattice)) {
    stop("Linear coefficient and grid points are of different length.")
}

tmpA1 <- rep(a1, each=nrows)*bandwidth
### second derivative w.r.t a0 ###
tmp11 <- outer(R, lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth, t=tmpA1)) -
    outer(L, lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth, t=tmpA1));
tmp11 <- -exp(a0)*as.vector(offsets %*% tmp11);
diag(hessian.mat)[indexA0] <- tmp11;

### cross derivatives ###
tmp12 <- outer(R, lattice, function(x, y) iMGF_gaussian_first_derivative(ub=(x-y)/bandwidth, t=tmpA1)) -
    outer(L, lattice, function(x, y) iMGF_gaussian_first_derivative(ub=(x-y)/bandwidth, t=tmpA1));
tmp12 <- -bandwidth*exp(a0)*as.vector(offsets %*% tmp12);
hessian.mat[(row(hessian.mat)==col(hessian.mat)+1) & (row(hessian.mat) %% 2==0)] <- tmp12;
hessian.mat[(row(hessian.mat)==col(hessian.mat)-1) & (row(hessian.mat) %% 2==1)] <- tmp12;

### second derivative w.r.t a1 ###
tmp22 <- outer(R, lattice, function(x, y) iMGF_gaussian_second_derivative(ub=(x-y)/bandwidth, t=tmpA1)) -
    outer(L, lattice, function(x, y) iMGF_gaussian_second_derivative(ub=(x-y)/bandwidth, t=tmpA1));
tmp22 <- -(bandwidth^2)*exp(a0)*as.vector(offsets %*% tmp22);
diag(hessian.mat)[indexA1] <- tmp22;

if (det(hessian.mat)!=Inf) hessian.inv <- solve(hessian.mat)
else {
    hessian.inv <- matrix(0, nrow=nrow(hessian.mat), ncol=ncol(hessian.mat))
    for (k in 0:(length(R)-1)) {
        tmpInd <- (20*k+1):(20*k+20)
        hessian.inv[tmpInd, tmpInd] <- solve(hessian.mat[tmpInd, tmpInd])
    }
}

### tmp11 and tmp 12 are needed to calculate the score function ###
return(list(hessian=hessian.mat, hessian.inv=hessian.inv,
entry1=tmp11, entry2=tmp12, entry3=tmp22));
}

esms_constructor_linear<- function(J, lattice, bandwidth, a1, offsets, quadwgts) {
    nJinterval<- length(J);
    JintervalWidth<- diff(c(0, J));
    tmpA1<- rep(a1, each=nJinterval)*bandwidth;
    ## Step 1: Inner integral at different x ##
    num<- outer(J, lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth)) -
        outer(c(0, J[-nJinterval]), lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth));
    den<- outer(J, lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth, t=tmpA1)) -
        outer(c(0, J[-nJinterval]), lattice, function(x, y) iMGF_gaussian(ub=(x-y)/bandwidth, t=tmpA1));
    K_x.mat<- diag(offsets/JintervalWidth) %*% num %*% diag(1/as.vector(offsets %*% den));
    rm(list=c("num", "den"));
    ## Step 2: the smoothing matrix using Gaussian quadrature rule.
    K.mat<- K_x.mat %*% kronecker(diag(.5*JintervalWidth, nrow=nJinterval), quadwgts);
    return(list(K_x.mat=K_x.mat, K.mat=K.mat))
}

local.linear.nhpp<- function(panel.data, bw, tol=1.0E-8, max.iter=100, quad.order=NULL, partition=NULL, Lambda.ini=NULL) {
    if (is.null(partition)) J<- sort(union(panel.data$L, panel.data$R))
    else J<- partition
    if (any(J==0)) J<- setdiff(J, 0)
    nJinterval<- length(J)
    JintervalWidth<- diff(c(0, J))
    if (quad.order == 10) quadRule<- quad10pt(J)
    else quadRule<- quad5pt(J);

    # Step 1: construct the incidence matrix
    censor.mat<- outer(tapply(panel.data$CT, panel.data$ID, unique), J, ">=");
    offsets<- apply(censor.mat, 2, sum)

    ### since only the panel with nonzero count matter ###
    ### get rid of the zero panel ###
    tmp.data<- panel.data[which(panel.data$M!=0),]
incidence.mat <- outer(tmp.data$R, J, ">=") - outer(tmp.data$L, J, ">=")
if (is.null(Lambda.ini)) {
    Lambda.old <- tmp.data$M %*% incidence.mat %*% diag(1/offsets)
} else Lambda.old <- t(as.matrix(Lambda.ini))
a0.old <- kronecker(log(2 + as.vector(Lambda.old)), rep(1, quadRule$nQuadpts));
a1.old <- rep(0, length(a0.old))
a.old <- as.vector(rbind(a0.old, a1.old))
a0 <- a1 <- rep(0, length(a0.old))
a <- as.vector(rbind(a0, a1))
Lambda <- matrix(0, nrow=1, ncol=nJinterval)
niter <- 1

repeat {
    V.inv <- diag(1/as.vector(Lambda.old %*% t(incidence.mat)));
    tmpEM <- tmp.data$M %*% V.inv %*% incidence.mat %*% diag(as.vector(Lambda.old));
    #### Newton Raphson Algorithm ####
    nr.iter <- 0
    repeat {
        a0.old <- a.old[seq(1, length(a), by=2)];
        a1.old <- a.old[seq(2, length(a), by=2)];
        hessian <- hessian.fun(a0=a0.old,
            a1=a1.old,
            lattice=quadRule$lattice,
            bandwidth=bw,
            L=c(0, J[-length(J)]),
            R=J,
            offsets=offsets
        );
        score <- score.fun(a0=a0.old,
            a1=a1.old,
            lattice=quadRule$lattice,
            bandwidth=bw,
            L=c(0, J[-length(J)]),
            R=J,
            em_estimate=tmpEM,
            entry1=hessian$entry1,
            entry2=hessian$entry2
A Appendix

function(){
    a <- a.old - as.vector(hessian$hessian.inv %*% score);

    if ((sum((a.old-a)^2) > tol) & (nr.iter <= max.iter)) {
        a.old <- a;
        nr.iter = nr.iter + 1;
    } else {
        a0 <- a[seq(1, length(a), by=2)];
        a1 <- a[seq(2, length(a), by=2)];
        break;
    }
}

##### construct the smoothing matrix #####
ems_input <- ems_constructor_linear(J=J, lattice=quadRule$lattice,
    bandwidth=bw, a1=a1, offsets=offsets, quadwgts=quadRule$wgt)

##### Smoothing Step #####
Lambda <- (tmpEM%*% diag(1/offsets)) %*% ems_input$K.mat;

if (sum((Lambda.old-Lambda)^2) > tol) {
    Lambda.old <- Lambda
    niter <- 1+niter
} else {
    lambda <- (tmpEM%*% diag(1/offsets)) %*% ems_input$K_x.mat; break;
}
return(list(Lambda.est=as.vector(Lambda),
            lambda.est=as.vector(lambda),
            nIter=niter, lattice=quadRule$lattice));
}

Local Constant for Area-censored Data

ems_constructor <- function(centroid, offsets=NULL, bandwidth=0, cellwidth, quadPts, quadWgts) {

    maxNonzero <- (1 + 4*(1+floor(bandwidth/cellwidth))*(2+floor(bandwidth/cellwidth)) -
                   floor(bandwidth*(2-sqrt(2))/(2*cellwidth))^-2));
    dimMat <- length(centroid)*maxNonzero;

    }
if (is.null(offsets)) offsets <- rep(1, length(centroid));

dyn.load(paste("smoother", .Platform$dynlib.ext, sep=""));
tmpD <- .C("smoother", cx=as.double(Re(centroid)), cy=as.double(Im(centroid)),
        ncell=as.integer(length(centroid)), quadPts=as.double(quadPts),
        quadWgt=as.double(quadWgts), nquadPts=as.integer(length(quadPts)),
        bandwidth=as.double(bandwidth), cellwidth=as.double(cellwidth),
        offsets=as.double(offsets), resultValue=as.double(rep(0, dimMat)),
        colNonzero=as.integer(rep(0, length(centroid))),
        rowIndex=as.integer(rep(-1, dimMat)), nNonzero=as.integer(0));

K.mat.csr <- as.matrix.csr(0);
K.mat.csr@ra <- tmpD$resultValue[1:tmpD$nNonzero];
K.mat.csr@ia <- cumsum(c(as.integer(1), tmpD$colNonzero));
K.mat.csr@ja <- as.integer(1 + tmpD$rowIndex[1:tmpD$nNonzero]);
K.mat.csr@dimension <- rep(length(centroid), 2);
return(K.mat.csr);

intKernel <- function(centroid, bandwidth=0, cellwidth, quadPts, quadWgts) {
  dyn.load(paste("intKernel", .Platform$dynlib.ext, sep=""));
tmpD <- .C("intKernel", cx=as.double(Re(centroid)), cy=as.double(Im(centroid)),
          ncell=as.integer(length(centroid)), quadPts=as.double(quadPts),
          quadWgt=as.double(quadWgts), nquadPts=as.integer(length(quadPts)),
          bandwidth=as.double(bandwidth), cellwidth=as.double(cellwidth),
          intResult=as.double(rep(0, 15625)), xevalpt=as.double(rep(0, 625)),
          yevalpt=as.double(rep(0, 625)));

  kernel.mat <- matrix(tmpD$intResult, ncol=25, byrow=T);
  quadWgts.mat <- t(diag(1/4, 25) %x% as.vector(outer(quadWgts, quadWgts, "*")));

  return(list(kernel.mat=kernel.mat, quadWgts.mat=quadWgts.mat));
}

intEstimate.exact <- function(location, centroid, offsets, bandwidth=0, cellwidth, quadPts, quadWgts) {
  dyn.load(paste("smoother_exact", .Platform$dynlib.ext, sep=""))
tmpD <- .C("smoother_exact", cx=as.double(Re(centroid)), cy=as.double(Im(centroid)),
          ncell=as.integer(length(centroid)), quadPts=as.double(quadPts),
          ...)
quadWgt=as.double(quadWgts), nquadPts=as.integer(length(quadPts)),
bandwidth=as.double(bandwidth), cellwidth=as.double(cellwidth),
offsets=as.double(offsets), Lambda=as.double(rep(0, length(centroid))),
xobs=as.double(Re(location)), yobs=as.double(Im(location)), nobs=as.integer(length(location));
}
Bibliography


119


