UiO : Department of Mathematics
University of Oslo

Cube Root Asymptotics

Maximal Scores and Irregular Histograms

Jonas Moss

Master's Thesis for the degree
Modelling and Data Analysis (MOD5960), November 2015

Abstract

Estimators with cube root asymptotics are typically the result of M-estimation with non-smooth objective functions. Aside from being inefficient, they are hard to calculate, have intractable limiting distributions, and are unamenable to the bootstrap. Manski's maximum score estimator and irregular histograms receive special attention. We investigate the geometry, algorithmics and robustness properties of Manski's maximum score estimator, a semiparametric estimator of the coefficients in the binary response model. We provide a new exact algorithm for its computation in covariate dimension one and two. This is faster than other algorithms described in the statistical literature. The breakdown point in covariate dimension one is derived, and we make progress towards finding it in higher dimensions. The breakdown points are highly dependent on the underlying data generating mechanism. Irregular histograms on the unit interval are also a major theme of this thesis. These are obtained through the minimisation of the Kullback-Leibler divergence and integrated squared distance. For smooth densities, we derive the limit distributions of the split points estimates for four classes of irregular histograms. Different conditions on the underlying density leads to different rates of convergence, with cube root being the norm. The computational challenges involved in finding these histograms are discussed, and some anomalies associated with them are investigated. Also, it is indicated how one can proceed in order to show consistency of these density estimators. Finally we derive the *CIC* (cube root information criterion), a cousin of the AIC.

Preface

The subject of this thesis was decided upon after Nils and I talked about a specific regression problem. Assume we have a regression $Y_i = m(x_i) + \epsilon_i$, where $x_i \in [0, 1]$ for simplicity. Questions in the social sciences sometimes take the form "is m an increasing function?". Let's take an example from the paper is "The Too-Much-Talent Effect: Team Interdependence Determines When More Talent Is Too Much or Not Enough" of Swaab et al. (2014). In team sports like football, baseball and basketball, it seems very plausible that teams get better as they get more talented players. If x_i is a measurement of the talentedness of team i, and Y_i is its objective outcome (e.g. number of games won), then $Y_i = m(x_i) + \epsilon_i$ is the model, and "m is monotonely increasing" is the (sensible!) null hypothesis. Swab et al. claim that the function m *isn't* increasing on $[0, 1]$ in the case of football, but reaches its maximum before 1. There is, to me, no obvious way to attack this problem, which is why I brought it up.

This discussion of monotone regression functions quickly led to *isotonic regression*, on to *Grenander's estimator.* Since these estimators are well-known for having cube root asymptotics, it suddenly became the theme of my thesis! This theme couldn't have been chosen if it weren't for the work Nils did in 2007 on the limiting distribution of the split points in so-called quantile histograms, which sparked the fires of this thesis.

General thanks to my adviser Nils Lid Hjort for giving me much to work on and for deluging me with wisdom. General thanks to my wife Kjersti Moss for proofreading, the verification of some equations and for taking care of the kids. I also thank Scott Bunting and Robert Bunting for proofreading parts of the thesis. Special thanks to Gudmund Hermansen for helping me out with the exact histogram algorithm. Special thanks to Jonas Lindstrøm for listening to me go on and on about histograms: Histograms is, frankly speaking, a dry subject. General thanks to Nadia Larsen and Dag Normann for being excellent advisers on a master project in C^*-algebras I never finished. General thanks to the people who authored of all those papers and books, cited or not. Many

of them are amazing! Unrelated thanks to Charles Darwin for being the most important scientist, and Yuki Kajiura for creating wonderful music. Very special bureaucratic thanks to the administration at Kvadraturen Skolesenter high school (in 2010) for giving me my high school diploma even though I never went to school there.

Contents

Chapter 1

Setting the scene

1.1 Introduction

> No one has yet discovered any warlike purpose to be served by the theory of [cube root asymptotics], and it seems unlikely that anyone will do so for many years.
>
> - G. H. Hardy in *A Mathematician's Apology* (slightly paraphrased)

Among all warlike estimators in statistics, far most have square root asymptotics: An estimator $\widehat{\theta}_n$ of θ has square root asymptotics if $\sqrt{n}(\widehat{\theta}_n - \theta) \xrightarrow{d} Y$, where Y is a non-degenerate random variable. Typically, Y is normally distributed with some μ and Σ dependent on the features of the underlying distribution F. This kind of limiting distribution appears e.g. when we use maximum likelihood, Bayes estimators, Z-estimators and the generalised method of moments. Estimators with cube root asymptotics, on the other hand, satisfy $n^{\frac{1}{3}}(\widehat{\theta}_n - \theta) \xrightarrow{d} Y$ for some non-degenerate random Y. Instead of being normal, Y is typically distributed as the maximiser of a non-degenerate zero-mean Gaussian process with drift. Frequentely, Y is a scaled variant of $\arg\max_t [W(t) - t^2]$, where W is a two-sided Browian motion originating from 0. Such variables are distributed according to *Chernoff's distribution.* There is also at least one elementary case where we have *n-asymptotics*, namely $\widehat{\theta}_n = \max X_i$ when $X_i \overset{i.i.d.}{\sim} \mathcal{U}(0, \theta)$. Normalizing, we get the convergence $n(\frac{\widehat{\theta}_n - \theta}{\theta}) \to \exp(1)$. More generally, this phenomenon obtains whenever the underlying F is smooth all the way to a break point. There are no similarly simple examples of cube root asymptotics, however.

Cube root asymptotics fall into roughly three cases:

1. Maximum likelihood for distributions having specific features. Two examples are Grenander's estimator and isotonic regression. Under the assumption that f is a decreasing density, Grenander's estimator is the left derivative of the concave majorant of the empirical distribution function. If smoothness assumptions on f are satisfied, the estimator has cube root asymptotics. (Groeneboom et al., 2014)

2. Robustness. The *shorth* estimator (Andrews and Hampel, 2015) is a robust estimator for the mean. Manski's maximum score estimator is a model robust estimator for the binary choice model. This estimator is discussed at length in Chapter 3. The *least median squares* (Rousseeuw, 1984), which is discussed in Section 2.3 robustly estimates the βs in a linear regression model.

3. Approximation by step functions: The largest chapter (4) of this thesis is on the subject of *irregular histograms*, wherein the split points are shown to converge at the $n^{\frac{1}{3}}$ rate provided the underlying f possesses a derivative. A similar concept is that of decision trees, or regression histograms, discussed in Section 2.4.

A feature these estimators have in common is their attempt to measure something smooth by using something discrete. For instance, both Grenander's estimator and histograms approximate a smooth density with a step function. Manski's score estimator estimates the βs in a binary choice regression model by maximising a step function; binary decision trees estimate a smooth regression model with a step function. We can make this more precise. These estimators are typically M-estimators, with objective m_θ. The underlying mechanism is that $Pm_\theta = \int m_\theta dP$ is smooth in θ, but $P_n m_\theta = n^{-1} \sum_{i=1}^n m_\theta(X_i)$ is not. Most M-estimators are constructed in a manner which makes this impossible. The culprit in cube root asymptotics is the introduction of indicator functions in m_θ. For instance, $m_\theta = 1_{[x-\frac{1}{2}\theta, x+\frac{1}{2}\theta]}$ for Chernoff's mode estimator of Section 2.2.2 on page 12, a section contains more about the heuristics of cube root asymptotics.

Sometimes, when we replace the smoothness assumptions on Pm_θ with discreteness assumptions, we gain n-convergence instead. This happens for decision trees and is likely to happen for irregular histograms as well.

Other common features of cube root asymptotics are:

1. In higher dimensions, the limit distributions are intractable both analytically and numerically. In one dimension, the limiting distribution can

frequently be described as a rescaled Chernoff's distribution, typically involving several nuisance parameters.

2. Inconsistency of the bootstrap. Fortunately, the subsample and m-out-of-n bootstrap is still consistent. We will have more to say on this in Section 2.5.

3. The estimates are computationally expensive to find, at least in higher dimensions. Finding them typically requires combinatorial optimisation, with methods like Newton-Raphson being next to useless.

4. Even for big n, the distance between the sampling distribution and the limiting distribution is large.

Add to all these the fact that cube roots give a tremendous loss in efficiency compared to other reasonable procedures, like local polynomial regression instead of decision trees, ordinary linear regression instead of least median of squares regression, and kernel smoothers instead of irregular histograms, and we have the reason why these estimators are not very popular. An attractive feature of these estimators is often that they have fewer assumptions, and sometimes other nice properties: Manski's maximum score estimator is consistent under very broad assumptions, it is robust, and it has been shown that there exist *no* $\sqrt{n}$-consistent estimator of the Betas in the binary response model under these assumptions (Chamberlain, 1986). Also, Rousseeuw's least median of squares estimator is the most outlier robust linear regression estimator there is, and irregular histograms are likely to be L_1-consistent for any underlying density on $[0, 1]$, are easy to interpret, and require little space to store (when compared to KDEs, which essentially require the entire set of observations). Still, the study of cube root asymptotics has mostly been theoretical, with some papers on the bootstrap appearing in later years.

The combination of the points 1.) and 2.) above is especially pernicious, as it leaves no obvious method for calculating confidence intervals and doing hypothesis tests. This can sometimes, as in the case of Manski's estimator and (probably) irregular histograms, be rectified by smoothing. Also, the m-out-of-n bootstrap and m-subsampling are consistent in general, but they require a choice of m. The choice of m is usually done by a kind of cross validation approach, which requires the calculation of countless estimates. In addition, these methods often require a large n in order to work well. This combined with 3.) makes it infeasible to use these two resampling approaches.

1.2 An outline

Roughly speaking, we will devote Chapter 2 to cube root asymptotics in general, while the the succeeding chapters are devoted to two more worked-out examples: Manski's maximum score estimator Manski (1975) and irregular histograms on the unit interval Rozenholc et al. (2010). These investigations are meant as theoretical exercises, and the estimators are not applied on any non-trivial data sets.

In Chapter 2 we briefly discuss some of the general theory required for proving cube root convergence. This is the theory of M-estimation and empirical processes, where most is taken from van der Vaart and Wellner's excellent book "Weak Convergence and Empirical Processes" (1996). The key result from this book is the rate theorem, which can be used to establish the limit distribution and prove the rate of convergence for any known estimator with cube root asymptotics. We will not be able to do this theory justice, but we will try to indicate where the results come from and what the difficulties are. We will also discuss the cube root heuristics of Kim & Pollard (1990) and Chernoff (1964), by means of a heuristic proof of the limiting distribution of Chernoff's mode estimator (ibid.), which is the most simple estimator in this category. This particular estimator illuminates the question of why cube root asymptotics occurs. Then we briefly describe the least median of squares estimator and binary decision trees (Banerjee and McKeague, 2007). Finally, we will discuss resampling.

The next chapter is about Manski's maximum score estimator (which we will often call "Manski's estimator"), a semiparametric estimator of the βs in a binary response model. We will not focus on the asymptotics in this Chapter, as the algorithmics will be at the centre of our attention. An algorithm for its computation in one and two dimensions is discussed in detail, and we supply and implementation of it in `C++` (Stroustrup, 1986), with a link to `R` (2014). In addition, we discuss its robustness properties and carry out some simulations.

In the Chapter 4, irregular histograms on the unit interval is the subject. This chapter forms the bulk of this thesis. First we discuss kernel density estimation on the unit interval, with a special focus on the Gaussian copula KDE of Jones and Henderson (2007*a*). Secondly, we define a class of histograms and discuss some of their properties. Then our attention goes the question of L_1-consistency for these histograms. In the succeeding section we find the limiting distribution of the *split points* of these histograms, which is our main application of theory of M-estimation from Chapter 2. Also, we will discuss three different algorithms for the computation of the split points, and demonstrate the considerable advantage

of pre-smoothing. We implement some of this in C++/R. Finally we discuss the *cube root information criterion* (CIC), an extension of the AIC to these classes of histograms.

Chapter 2

Stochastic convergence

For many, abstract thinking is toil; for me, on good days, it is feast and frenzy.

- Martin Heidegger in *Nietzsche* (tr: D. F. Krell)

In the first section we briefly discuss the basic theory of weak convergence, including some of the technical difficulties that arises when working with stochastic variables on non-separable Banach spaces. The second section is devoted to M-estimation, a class which contains every known estimator with cube root asymptotics. A particularly important result is the monumental rate theorem. In the third section we discuss some estimators with cube root asymptotics. We will derive the limit distribution of Chernoff's mode estimator heuristically, and will discuss the heuristics of cube root asymptotics from Kim and Pollard (1990). We end the chapter with a discussion on resampling schemes in the context of cube root asymptotics.

2.1 Weak convergence

2.1.1 Theoretical basis

Given a probability space $(\Omega, \mathcal{F}, P)$, a stochastic variable is a map $X : \Omega \to \mathbb{R}^n$ which is $(\Sigma, \mathscr{B}^n)$-measurable, where $\mathscr{B}^n$ is the Borel σ-algebra on $\mathbb{R}^n$. The majority of classical results on convergence of stochastic variables depends on this measurability condition being satisfied. For instance, a sequence $X_n : \Omega \to \mathbb{R}$ converges weakly to X, denoted $X_n \xrightarrow{d} X$, if $P(X_n \leq x) \to P(X \leq x)$

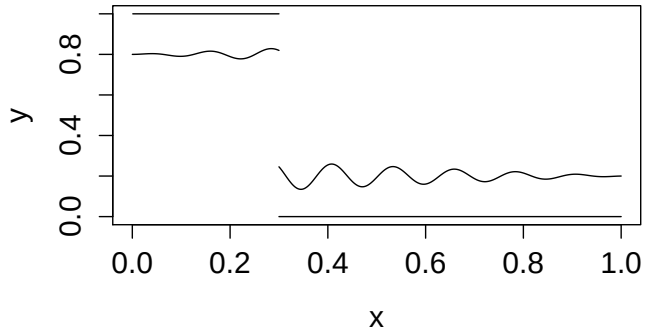

Figure 2.1.1: Example of an $l^\infty([0,1])$-function inside a_δ, with $\delta = 0.5$ and $a = 0.3$.

pointwise for every continuity point of $F(x) = P(X \leq x)$. This definition requires something like $(\Sigma, \mathcal{B})$-measurability, as we need to assign probabilities to every set of the form $X_n^{-1}(-\infty, x]$. Likewise, measurability is required for the concepts of convergence almost surely and convergence in probability.

In this thesis, we will encounter maps of the form $X_n : \Omega \to \mathbb{D}$, where $\mathbb{D}$ is some metric space, typically a non-separable Banach space. This lack of separability creates plenty of measurability issues. In the next example, it is illuminating to known that $l^\infty(T)$ is separable iff T is finite (Megginson, 2012, exercise 1.143).

Example 2.1.1. (Kosorok). Let $\Omega = [0,1]$ and $P = \lambda$ the Lebesgue measure on $[0,1]$. Let $U : \Omega \to [0,1]$ be the identity function on this interval, the uniform distribution on the unit interval. Now we define the function $X : [0,1] \to [0,1]$ by $X(x) = 1_{[U \leq x]}$. This is gives us a map

$$\begin{aligned} \Omega &\to l^\infty([0,1]). \\ \omega &\mapsto X. \end{aligned}$$

Consider the sets $a_\delta = \left\{ f \in l^\infty([0,1]) \mid \left\| f - 1_{[a \leq x]} \right\| < \delta \right\}$ and take $\delta = \frac{1}{2}$.

it is clear that $X^{-1}(a_\delta) = a$, so if $A_\delta = \cup_{a \in A} a_\delta$, then $X^{-1}(A_\delta) = A$. Now Let $A \subset [0,1]$ be a non-Borel subset, such as a complete analytic set (see e.g. Kechris (2012, p. 85)). As A_δ is open in $l^\infty([0,1])$, X isn't $(\Sigma, \mathcal{B}^n)$-measurable.

An easy modification of this example shows that the process

$$\sqrt{n}(\frac{1}{n}\sum_{i=1}^{n} 1_{[X_n \leq x]} - x)$$

isn't measurable when X_n are i.i.d. uniform on the unit interval. Hence we need a different framework to deal with such processes. The classic approach, used in Billingsley (2013), uses convergence in the Skorokhod topology. The modern approach, described in van der Vaart and Wellner (1996) and started by Hoffmann-Jørgensen, is quite different, and we will briefly describe it here.

It is a well known result that convergence in distribution can equivalently be formulated as weak*-convergence: A sequence $X_n \in \mathbb{R}$ converges in distribution to $X \in \mathbb{R}$ iff $Ef(X_n) \to Ef(X)$ for every bounded, continuous function $f : \mathbb{R} \to \mathbb{R}$, see e.g. Rosenthal (2006, p. 117, theorem 10.1.1). This definition extends nicely to the case when X_n are elements of separable Banach spaces, but the measurability issues mentioned above makes it impossible to use this same definition of convergence when X_n are elements of *non-separable* Banach spaces, like $l^\infty(\mathbb{R})$. An appropriate extension is to define $X^\star$ as the least measurable majorant of X, $E^\star(X) = E(X^\star)$: Let $\mathbb{D}$ be a (non-separable) Banach space, and let $X_n \in \mathbb{D}$ be random variables (not necessarily measurable). In addition, let $X \in \mathbb{D}$ be a measurable limit variable. Then $X_n \xrightarrow{d} X$ iff $E^\star f(X_n) \to Ef(X)$ for each bounded and continuous $f : \mathbb{D} \to \mathbb{D}$.

We will not make any direct use of this theory in this thesis, and we will mostly ignore measurability issues.

2.1.2 Glivenko-Cantelli classes

Throughout this thesis we use a functional notation for measures: When P is a measure on a measure space (X, Σ) and $f : X \to \mathbb{R}$ is a measurable function, we denote $\int f(x)dP(x) = Pf$. This notation reflects the fact that $\int \cdot dP$ is a functional mapping $f \to \int f(x)dP(x)$ for each f, and fits well into our approach. When P_n is the empirical measure obtained from n i.i.d. observations from P, $P_n f = n^{-1} \sum_{i=1}^n f(X_i)$. It follows from the law of large numbers that $P_n f \xrightarrow{a.s.} Pf$ for any measurable f. We will need *uniform variants* of this result for different classes of functions. When $\mathcal{F}$ is a class of functions, the norm $||\cdot||_\mathcal{F}$ is defined by $||X||_\mathcal{F} = \sup_{f \in \mathcal{F}} ||f(X)||$.

Definition 2.1.2. Let $\mathcal{F}$ be a class of functions and P a probability measure. If $||P_n - P||_\mathcal{F} \xrightarrow{p} 0$, the class $\mathcal{F}$ is *P-Glivenko-Cantelli.*

The name "Glivenko-Cantelli class" comes from the famous Glivenko-Cantelli theorem Billingsley (2008, p. 269), which states that the class

$$\mathcal{F} = \{(-\infty, x] \mid x \in \mathbb{R}\}$$

is Glivenko-Cantelli for any probability P on $\mathbb{R}$. Stated in a different way, it shows that $F_n(x) \xrightarrow{p} F(x)$ uniformly in x, where F_n is the empirical distribution function. The definition also makes sense when convergence in probability is replaced with convergence almost surely.

In order to show that a family is Glivenko-Cantelli, we will use the concept of *bracketing numbers* (van der Vaart and Wellner, 1996, p. 83), a stronger variant of the concept of compactness.

Definition 2.1.3. Given two functions l and u, define the bracket $[l, u] = \{f \mid l \leq f \leq u\}$. For a given norm $||\cdot||$, we define an ϵ-bracket as a bracket $[l, u]$ with $||l - u|| < \epsilon$. Let $\mathcal{F}$ be a class of measurable functions. The bracketing number of $\mathcal{F}$, denoted $N_{[]}(\epsilon, \mathcal{F}, ||\cdot||)$, is the minimal number of ϵ-brackets required to cover $\mathcal{F}$.

The following theorem is useful in proving that classes are Glivenko-Cantelli (van der Vaart and Wellner, 1996, p. 122, theorem 2.4.1):

Theorem 2.1.4. *Let $\mathcal{F}$ be a class of measurable functions that satisfy the bracketing number condition $N_{[]}(\epsilon, \mathcal{F}, L_1(P)) < \infty$. Then $\mathcal{F}$ is P-Glivenko-Cantelli.*

Now the original Glivenko-Cantelli result follows easily.

Corollary 2.1.5. *The class $\mathcal{F} = \{(-\infty, x] \mid x \in \mathbb{R}\}$ is Glivenko-Cantelli.*

Proof. Apply the previous theorem with brackets of the form $(-\infty, x_i]$, where $-\infty = x_0 < x_1, ... < x_m = \infty$ is chosen such that $P([x_i, x_{i+1}]) < \epsilon$. Then we need at most $2 + \frac{1}{\epsilon}$ brackets, yielding the desired result. □

Glivenko-Cantelli results are not of independent interest in this thesis, but will be used in order to establish consistency results through the Consistency Theorem (on page 21).

2.1.3 Donsker classes

While Glivenko-Cantelli results are uniform variants of the law of large numbers, Donsker results are uniform variants of the central limit theorem. These classes are named in honour of Monroe Donsker, who proved the uniform central limit theorem for the empirical distribution function. We are not interested in Donsker results in themselves, but they are needed in order to establish the limiting distributions of M-estimators by means of the rate theorem 2.2.4. A family of functions $\mathcal{F}$ is P-Donsker if

$$\sqrt{n}(P_n - P) \xrightarrow{d} Z$$

in $l^\infty(\mathcal{F})$, where Z is a non-degenerate and measurable random process.

The concept of bracketing entropy has its place here as well: The bracketing integral is

$$J_{[]}(\delta, F, L_2(P)) = \int_0^\delta \sqrt{\log N_{[]}(\epsilon, \mathcal{F}, L_2(P))} d\epsilon,$$

which we want to be bounded as $\delta \to \infty$. This is a condition on the growth of the entropy as $\epsilon \to 0$, for since $N_{[]}(\epsilon, \mathcal{F}, L_2(P) = 1$ when δ is large enough, it is only the values close to 0 that are of interest. This integral is important due to the following theorem, which is proved in Kosorok (2007, p. 148).

Theorem. *Let $\mathcal{F}$ be a class of measurable functions with a finite bracketing integral at infinity, $J_{[]}(\infty, F, L_2(P)) < \infty$. Then $\mathcal{F}$ is P-Donsker.*

2.2 M-estimation

2.2.1 Basics

Let θ_0 be some statistical quantity and $\widehat{\theta}$ be an estimator of θ_0. Then $\widehat{\theta}$ is an *M-estimator* if it is the point of maximum of a criterion function $P_n m_\theta$, where $m_\theta : \otimes \times \Theta \to \mathbb{R}$. The function m_θ is typically chosen in order to ascertain that $\arg\max_{\theta \in \Theta} P m_\theta = \theta_0$, which indicates $\widehat{\theta} \xrightarrow{p} \theta_0$ as $n \to \infty$. The most famous case of M-estimation is that of maximum likelihood estimation, where $m_\theta = l_\theta$, the log likelihood of a parametric model. M-estimation is often associated with robust statistics, where one can use criterion functions other than the log likelihood in order to obtain estimators with bounded *influence functions* (see Section 3.5) and easily understandable asymptotics. This approach was developed by Huber (1981), and is often constrasted with L-estimators and R-estimators in classical robust statistics. In the context of robust statistics, M-estimators has a more restricted meaning than ours, as only smooth choices of m_θ are used. In this setting, both the asymptotics of $\widehat{\theta}$ and its value can be obtained by differentiating $P_n m_\theta$, yielding estimators which can be understood as the solution of the equation $P_n \frac{d}{d\theta} m_\theta = 0$ for some m_θ. These estimators are often called Z-estimators in the empirical process literature van der Vaart and Wellner (1996), van der Vaart (2000, chapter 3.3; chapter 5), emphasising this property (Z is for zero). The following result is a classic. For a rigorous proof

can be found van der Vaart and Wellner (1996, theorem 3.3.1)

Theorem 2.2.1. *Assume Pm_θ is differentiable at θ_0 with a non-singular Hessian J, and $K = Pm_\theta m_\theta^T$ exists. Also assume some regularity conditions, most importantly that m_θ is differentiable at θ_0. Then $\sqrt{n}(\widehat{\theta} - \theta) \xrightarrow{d} N(0, J^{-1}KJ^{-1})$.*

Proof. Let $h = \theta - \theta_0$. Under some regularity and smoothness conditions we have

$$\begin{aligned} nP_n m_\theta &= \sum_{i=1}^{n} m_\theta(X_i) \\ &\approx \sum m_{\theta_0}(X_i) + U(X_i;\theta_0)h + \frac{1}{2}h^T J h + o(h^2), \end{aligned}$$

where $U = \frac{d}{d\theta} m_\theta \mid_{\theta=\theta_0}$. This yields the local centred likelihood process

$$\begin{aligned} M_n(\theta) &= n(P_n - P)(m_{\theta_0 + sn^{-\frac{1}{2}}} - m_{\theta_0}) \\ &= \sqrt{n}s \left[\frac{1}{n} \sum_{i=1}^{n} U(X_i;\theta_0) \right] - \frac{1}{2} s^T J s + o(|s|^2). \end{aligned}$$

This process converges to $t^T Z - \frac{1}{2} t^T J t$, with $Z \sim N(0, K)$ and $K = \mathrm{Var} U(X, \theta_0)$. Taking the derivative, we get $KZ = Jt$, hence $t = J^{-1}Z$ maximises it, and $\sqrt{n}(\widehat{\theta} - \theta_0) \xrightarrow{d} N(0, J^{-1}KJ^{-1})$. □

The matrix $J^{-1}KJ^{-1}$ is often called the *sandwich matrix*, and be put to use as a model robust covariance matrix in maximum likelihood estimation. In the special case when $m_\theta = \log g(\theta)$, for some density g, we obtain the classical limit result on maximum likelihood estimators as a corollary. If $g = f$, where f is the true model density, $J = K$ and we obtain the well-known limit $\sqrt{n}(\widehat{\theta} - \theta_0) \xrightarrow{d} N(0, J^{-1})$.

Our investigations will deal with M-estimators where the smoothness assumption on m_θ does not hold. The estimators will have associated criterion functions m_θ such that $P_n m_\theta = n^{-1} \sum_{i=1}^{n} m_\theta(X_i)$ is very jagged and far from being differentiable. Still, importantly, we will require that the "true" objective $Pm_\theta = \int m_\theta dP$ is differentiable. In order for this to be the case, P will have to be sufficiently smooth.

2.2.2 Heuristics and examples

Now we will turn to a discussion of the heuristics of cube root asymptotics, as discussed in Kim and Pollard (1990, p. 193). In the process we derive the limit

distribution of the sample median and *Chernoff's mode estimator*. The median is natural to spend some time on: First of all, it appears non-smooth, it has a small solution space, is very robust, and forms the basis of the least median of squares regression estimator (which has cube root asymptotics). Chernoff's mode estimator is the simplest cube root estimator, and the basis of Kim and Pollard's (ibid.) cube root intuition.

We begin by characterising the sample median as an arg min. The mean is the arg min of $\theta \mapsto E(X-\theta)^2$: Use the decomposition $E(X-\theta)^2 = E(X-\mu)^2 + 2(\mu-\theta)E(X-\mu) + (\mu-\theta)^2$ and take the derivative with respect to θ. Perhaps less intuitively, the median is the arg min of $\theta \mapsto E|X-\theta|$, due to this well known fact.

Proposition 2.2.2. *Let $X \sim F$, where F is a distribution with a density in a neighbourhood of $f(\theta_0)$, θ_0 being the median of F. Then $\theta_0 = \arg\min E(|X-\theta|)$.*

Proof. Observe that

$$\begin{aligned} E(|X-\theta|) &= \int_\theta^\infty (x-\theta)dF(x) + \int_{-\infty}^\theta (\theta - x)dF(x), \\ &= \int_\theta^\infty x dF(x) - \int_{-\infty}^\theta x dF(x) + \theta(2F(\theta)-1). \end{aligned}$$

Since $\frac{d}{d\theta}\int_\theta^\infty x dF(x) = -\frac{d}{d\theta}\int_{-\infty}^\theta x dF(x) = \theta f(\theta)$, this function is differentiable at θ_0 with derivative

$$-2\theta f(\theta) + (2F(\theta)-1) + 2\theta f(\theta) = (2F(\theta)-1),$$

consequently $F(\theta_0) = \frac{1}{2}$ as claimed. □

Define $m_\theta = |\cdot - \theta|$. As above, θ_0 is the true median in the following theorem. The content of this theorem is well known, but this proof is the work of the author.

Theorem 2.2.3. *Let $\widehat{\theta}$ be the sample median obtained from $X_1, ..., X_n$ i.i.d. copies from a distribution F with a non-zero density in the neighbourhood of the median. In this case,*

$$n^{\frac{1}{2}}(\widehat{\theta} - \theta_0) \xrightarrow{d} N(0, (2f(\theta_0)^{-2}).$$

Proof. For simplicity we assume, throughout this proof, that $\theta > \theta_0$. By decomposing the sum of $P_n m_\theta$ as in Proposition 2.2.2, we obtain

$$\frac{1}{n}\sum_{i=1}^{n}|X_i-\theta| = \frac{1}{n}\left[\sum_{X_i>\theta}X_i - \sum_{X_i<\theta}X_i + \theta(2\#\{\theta > X_i\} - n)\right].$$

Define lower, middle and upper random variables as follows:

$$\begin{aligned} L &= \#\{\theta_0 > X_i\}, \\ M &= \#\{\theta > X_i > \theta_0\}, \\ U &= \#\{X_i > \theta\}. \end{aligned}$$

We can understand (L, M, U) as an n-multinomial vector with cell probabilities $F(\theta_0)$, $F(\theta) - F(\theta_0)$, and $1 - F(\theta)$. Take note of the following, Taylor-derived observation: When $\theta \approx \theta_0$, the cell probability of M is approximately $f(\theta_0)(\theta - \theta_0)$. By simple reasoning, $Y_n = P_n m_\theta - P_n m_{\theta_0}$ can be identified as

$$Y_n = 2\sum_{\theta>X_i>\theta_0}(\theta - X_{k_i}) + (\theta-\theta_0)(2L-n),$$

where k_i are the indices of those X_j satisfying $\theta > X_j > \theta_0$. Then X_{k_j} is distributed according to the density $f(x)1_{[\theta_0,\theta]}(F(\theta) - F(\theta_0))^{-1}$, which equals $(\theta-\theta_0)^{-1}1_{[\theta_0,\theta]}$ when $\theta_0 \approx \theta$ by Taylor expansion. Thus $\theta - X_j \sim \mathcal{U}(\theta_0, \theta)$. Denote these variables $(\theta-\theta_0)V_j$, and use this to rewrite $\frac{1}{n}Y_n$ as

$$\frac{1}{n}Y_n = \frac{1}{n}2(\theta-\theta_0)\sum_{i=1}^{M}V_j + \frac{1}{n}2(\theta-\theta_0)(L-0.5n).$$

Since $EM = nf(\theta_0)(\theta-\theta_0)$ and $EV_i = \frac{1}{2}$, we get

$$E[\frac{1}{n}2(\theta-\theta_0)\sum_{i=1}^{M}V_i] = f(\theta_0)(\theta-\theta_0)^2.$$

An application of the law of total variance gives us

$$\begin{aligned} \mathrm{Var}(\sum_{i=1}^{M}V_i) &= E(\mathrm{Var}(\sum_{i=1}^{M}V_i)|M) + \mathrm{Var}(E(\sum_{i=1}^{M}V_i)|M) \\ &= nf(\theta_0)(\theta-\theta_0)12^{-1} + 4^{-1}nf(\theta_0)(\theta-\theta_0)(1-f(\theta_0)(\theta-\theta_0)) \\ &= 3^{-1}nf(\theta_0)(\theta-\theta_0). \end{aligned}$$

Where the higher order terms are discarded. Hence

$$\mathrm{Var}[\frac{1}{n}2(\theta-\theta_0)\sum_{i=1}^{M}V_i] = \frac{4}{3n}f(\theta_0)(\theta-\theta_0)^3.$$

The term $\frac{1}{n}2(\theta-\theta_0)(L-0.5n)$ is easily seen to be normally distributed in the limit:

$$\frac{1}{n}2(\theta-\theta_0)(L-0.5n) \sim N(0, n^{-1}(\theta-\theta_0)^2).$$

Now we find the covariance between $X = \frac{1}{n}2(\theta-\theta_0)\sum_{i=1}^{M}V_j$ and $Y = \frac{1}{n}2(\theta-\theta_0)(L-0.5n)$ by using the law of total covariance,

$$\mathrm{Cov}(X,Y) = E(\mathrm{Cov}(\mathrm{X},\mathrm{Y} \mid \mathrm{M})) + \mathrm{Cov}(\mathrm{E}(\mathrm{X} \mid \mathrm{M}), \mathrm{Cov}(\mathrm{E}(\mathrm{Y} \mid \mathrm{M})).$$

Here $E(\mathrm{Cov}(\mathrm{X},\mathrm{Y} \mid \mathrm{M})) = 0$, as X and Y are conditionally independent given M. Clearly, $E(X \mid M) = \frac{1}{n}(\theta-\theta_0)M$. Since

$$\begin{aligned} E(Y \mid M) &= \frac{1}{n}2(\theta-\theta_0)(\frac{F(\theta_0)}{1-(F(\theta)-F(\theta_0))})(n-M)-0.5n) \\ &= \frac{1}{n}2(\theta-\theta_0)(\frac{F(\theta_0)}{1-f(\theta_0)(\theta-\theta_0)})(n-M)-0.5n), \end{aligned}$$

and

$$\mathrm{Var}M \approx nf(\theta_0)(\theta-\theta_0)(1-f(\theta_0)(\theta-\theta_0)),$$

we obtain

$$\begin{aligned} \mathrm{Cov}(X,Y) &= -\frac{2}{n^2}(\theta-\theta_0)^2F(\theta_0)\mathrm{Var}M, \\ &\approx -\frac{2}{n}(\theta-\theta_0)^3F(\theta_0)f(\theta_0). \end{aligned}$$

Notice that the variance of X is of higher order than the variance of Y. Since the covariance is equally negligible, we obtain

$$\frac{1}{n}Y_n(\theta) \approx f(\theta_0)(\theta-\theta_0)^2 + n^{-\frac{1}{2}}(\theta-\theta_0)Z,$$

where $Z \sim N(0,1)$ is the same across all θs sufficiently close to θ_0. The case when $\theta < \theta_0$ is similar and is omitted.

Now make the substitution $t = |\theta-\theta_0|$ and differentiate $\frac{1}{n}Y_n$ with respect to t. This yields $2f(\theta_0)t + \frac{1}{2}n^{-\frac{1}{2}}Z$, which has its root at $n^{\frac{1}{2}}t = -\frac{1}{2f(\theta_0)}Z$. Hence

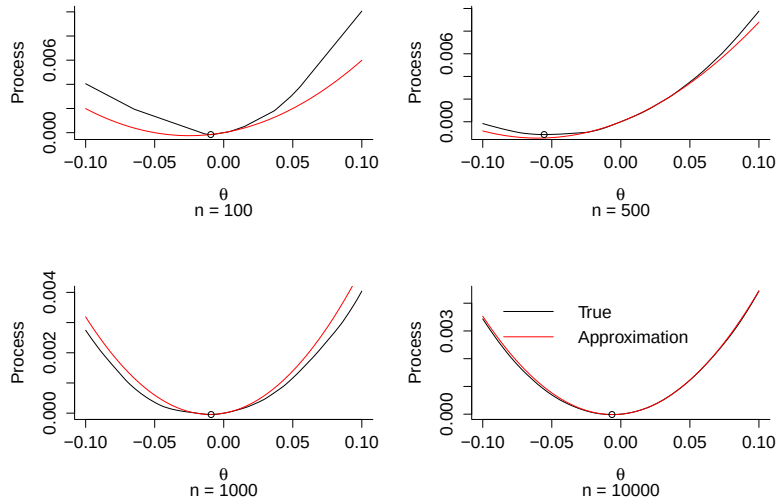

Figure 2.2.1: Simulated example of the true median process $P_n(m_\theta - m_{\theta_0})$ along with the approximation in Theorem 2.2.3. We simulate $n = 100, 500, 1000, 10000$ observations from $N(0,1)$ and calculate the approximation and the true process:

$n^{\frac{1}{2}}(\widehat{\theta} - \theta) \sim N(0, (2f(\theta))^{-2})$, as claimed. □

Importantly, the Z in the previous proposition is constant across all θ — there is one dominating source of randomness, and it doesn't change with θ. This happens because we manage to decompose the process into two parts, one of which has higher order variance than the other, where the dominating part depends linearly on θ. The kind of decomposition we arrived at in this theorem is impossible for cube root asymptotics: The Brownian motions that typically appear are witnesses to this fact, the "dominating randomness" does not depend linearly on θ in this case.

A slight modification of the process above turns it into an "argmax"-process: $-f(\theta_0)(\theta-\theta_0)^2+n^{-\frac{1}{2}}(\theta-\theta_0)Z$. This is a special case of the more general type of expression $U_n(\theta) = -c_1(\theta-\theta_0)^2+c_2n^{-\frac{1}{2}}(\theta-\theta_0)Z_\theta$, where $c_1 > 0, c_2$ are constants and Z_θ is an asymptotically normally distributed variable which might depend on θ. For $t = (\theta - \theta_0)$ to maximise U_n, it has to strike a balance between the negative contribution of c_1t^2 and the positive contribution of $c_2n^{-\frac{1}{2}}(\theta - \theta_0)Z_\theta$. When is this likely to happen? If $|\theta-\theta_0|$ gets too large, c_1 will make the value too small; if $|\theta-\theta_0|$ is too small, there will not be enough positive contribution from $c_2n^{-\frac{1}{2}}|Z_\theta|$. When $(\theta - \theta_0) = tn^{-\frac{1}{2}}$ is of the order $n^{-\frac{1}{2}}$, the contribution of each side equalises: $U_n(tn^{-\frac{1}{2}}) = n^{-1}\left(c_1t^2 - c_2tZ_{tn^{-\frac{1}{2}}+\theta_0}\right)$, but any other choice of α in $n^{-\alpha}$ will put too much weight on either side. We will see this rate intuition at work for cube root asymptotics soon.

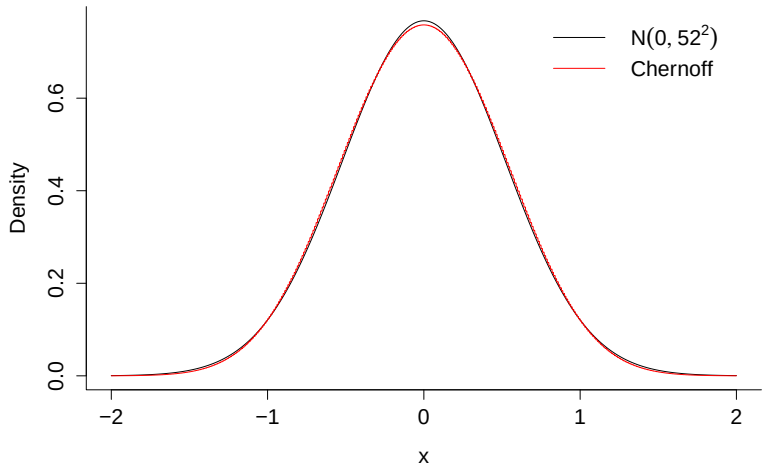

Figure 2.2.2: The $N(0, .52^2)$-density and Chernoff's density obtained through simulations.

In Chernoff (1964), an estimator of the mode was introduced. Let F be a unimodal, smooth density on $\mathbb{R}$, and let $\alpha > 0$ be given. Let P_n be the empirical distribution of $X_1, X_2, ..., X_n \overset{i.i.d.}{\sim} F$, as usual. Chernoff's estimator is then defined as $\widehat{\theta} = \arg\max_{x \in \mathbb{R}} P_n[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha]$. The interpretation is simple: For each $x \in \mathbb{R}$, the interval $[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha]$ contains a fixed number of observations from P_n, and $\widehat{\theta}$ is the centre of the interval of this form which contains the highest number of observations. If α is small enough and F is symmetric around its mode, $x_0 = \arg\max_{x \in \mathbb{R}} P[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha]$ is the mode F. In this case, $f(x - \frac{1}{2}\alpha) = f(x + \frac{1}{2}\alpha)$. We assume that f' exists. Notice that $P[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha]$ is smooth in α, but $P_n[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha]$ is not.

Now we present an heuristic proof of the limiting distribution of Chernoff's mode estimator, which is copied from Chernoff (1964). For a formal derivation, see van der Vaart and Wellner (1996, example 3.2.13). We will make use of the distribution of

$$\arg\max_{x \in \mathbb{R}} \left[W(x) - x^2 \right], \tag{2.2.1}$$

where W is a two-sided Brownian motion originating from 0. This will be denoted Z and is called Chernoff's distribution in the literature (Groeneboom and Wellner, 2001), the name deriving from its first appearance in Chernoff (1964). We will see this distribution several times throughout the course of this thesis. It is symmetric around 0 and is bell-curved. It is reasonably well approximated by $N(0, 0.52^2)$, see Figure on this page.

We will typically encounter distributions of the form $\arg\max_{x\in\mathbb{R}}[aW(x)-bx^2]$. An argument based on the concept of Brownian scaling Billingsley (2008, p. 504): For any $c>0$, $c^{-1}W(c^2x)=W(x)$. The following trick was first observed by Chernoff (1964).

Proposition 2.2.4. *Let Z be Chernoff's distribution, and $Y=\arg\max_{x\in\mathbb{R}}[aW(x)-bx^2]$. In that case, Y and $(\frac{b}{a})^{\frac{2}{3}}Z$ have the same distribution.*

Proof. We will find a c such that z maximises $W(x)-x^2$ if and only if cz maximises $aW(x)-bx^2$. We can do this by making $aW(xc)-c^2x^2\propto W(x)-x^2$. By Brownian scaling $aW(xc)=ac^{\frac{1}{2}}W(x)-bx^2c^2$. The proportionality requirement is fulfilled when $ac^{\frac{1}{2}}=bc^2$, which has solution $c=(\frac{b}{a})^{\frac{2}{3}}$. □

we will also make use of one the heuristic interpretations of a Brownian motion (see Ross (2014, chapter 10)). Let X_i be distributed according to

$$\begin{aligned} P(X_i=-1) &= \frac{1}{2},\\ P(X_i=1) &= \frac{1}{2}. \end{aligned}$$

This distribution is called the *Rademacher distribution.* Define a process,

$$X(t)=\Delta x\sum_{i=1}^{\lfloor t/\Delta t\rfloor}X_i,$$

where Δt is the time increment and Δx is the space increment. Let $\Delta x=\sigma\sqrt{\Delta t}$ for some σ, and let $\Delta t\to 0$. The resulting processes exists and is a Brownian motion with standard deviation σ.

Theorem 2.2.5. *Assume the above conditions, and let $\widehat{\theta}$ be Chernoff's estimator, and θ_0 be the mode. Its limiting distribution is given by*

$$n^{\frac{1}{3}}(\widehat{\theta}-\theta_0)\xrightarrow{d}\tau^{\frac{1}{3}}Z,$$

where $\tau=\frac{8f(x_0+\frac{1}{2}\alpha)}{[f'(x_0-\frac{1}{2}\alpha)-f'(x_0+\frac{1}{2}\alpha)]^2}$.

Proof. Denote

$$Z_n=P_n\left[x-\frac{1}{2}\alpha,x+\frac{1}{2}\alpha\right]-P_n\left[x_0-\frac{1}{2}\alpha,x_0+\frac{1}{2}\alpha\right].$$

We decompose this into

$$Z_n = n^{-\frac{1}{2}} Y_n + u,$$

where

$$n^{-\frac{1}{2}} Y_n = (P_n - P)\left[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha\right] - (P_n - P)\left[x_0 - \frac{1}{2}\alpha, x_0 + \frac{1}{2}\alpha\right]$$

and

$$u = P\left[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha\right] - P\left[x_0 - \frac{1}{2}\alpha, x_0 + \frac{1}{2}\alpha\right].$$

Here u is the *actual deviation*, as seen from the true P, while $n^{-\frac{1}{2}} Y_n$ represents the *random deviation*. We can approximate u by a second order Taylor expansion:

$$\begin{aligned} & P\left[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha\right] - P\left[x_0 - \frac{1}{2}\alpha, x_0 + \frac{1}{2}\alpha\right] \\ = \; & [F(x + \frac{1}{2}\alpha) - F(x_0 + \frac{1}{2}\alpha)] - [F(x - \frac{1}{2}\alpha) - F(x_0 - \frac{1}{2}\alpha)] \\ \approx \; & -\frac{1}{2}(f'(x_0 - \frac{1}{2}\alpha) - f'(x_0 + \frac{1}{2}\alpha))(x - x_0)^2, \end{aligned}$$

where we use that $f(x_0 + a) - f(x_0 - a) = 0$, which makes the first order terms cancel. Now we consider

$$n^{\frac{1}{2}} Y_n = n\,(P_n - P)\left[x - \frac{1}{2}\alpha, x + \frac{1}{2}\alpha\right] - n\,(P_n - P)\left[x_0 - \frac{1}{2}\alpha, x_0 + \frac{1}{2}\alpha\right].$$

This process can be rewritten in two ways, depending on whether $x \leq x_0$ or $x > x_0$. If $x \leq x_0$,

$$n^{\frac{1}{2}} Y_n(x) = n\,(P_n - P)\left[x + \frac{1}{2}\alpha, x_0 + \frac{1}{2}\alpha\right] - n\,(P_n - P)\left[x - \frac{1}{2}\alpha, x_0 - \frac{1}{2}\alpha\right],$$

and if $x > x_0$,

$$n^{\frac{1}{2}} Y_n(x) \;=\; n\,(P_n - P)\left[x_0 + \frac{1}{2}\alpha, x + \frac{1}{2}\alpha\right] - n\,(P_n - P)\left[x_0 - \frac{1}{2}\alpha, x - \frac{1}{2}\alpha\right].$$

We assume $x > x_0$ and $x \approx x_0$ from now on, and put $t = (x - x_0)$. This process counts the deviation from expectation in $\left[x_0 + \frac{1}{2}\alpha, x + \frac{1}{2}\alpha\right]$ and subtracts the deviation from expectation in the disjoint interval $\left[x_0 - \frac{1}{2}\alpha, x_0 - \frac{1}{2}\alpha\right]$. These

counts are approximately independent, and if we increase x a little bit, there is an equal probability of adding one as subtracting one, provided the distribution is approximately symmetric close to the mode. The expected value of each interval is 0 while each of their variances are approximately $n(x-x_0)f(x_0+a) = n(x-x_0)f(x_0-a)$ by a first order Taylor expansion:

$$\begin{aligned}
\mathrm{Var} n^{\frac{1}{2}} Y_n(t+x_0) &= \mathrm{Var}\left[\sum_{i=1}^{n}\left(1_{\left[x_0+\frac{1}{2}\alpha, x+\frac{1}{2}\alpha\right]}(X_i) - 1_{\left[x_0-\frac{1}{2}\alpha, x-\frac{1}{2}\alpha\right]}(X_i)\right)\right] \\
&= P\left[x_0+\frac{1}{2}\alpha, x+\frac{1}{2}\alpha\right] + P\left[x_0-\frac{1}{2}\alpha, x-\frac{1}{2}\alpha\right] \\
&\approx 2ntf(x_0-\frac{1}{2}\alpha), \qquad (2.2.2)
\end{aligned}$$

Thus the process Y_n looks like it tends to a two-sided Brownian motion with variance $2f(x_0+\frac{1}{2}\alpha) = 2f(x_0-\frac{1}{2}\alpha)$ per unit t. From this we find that $Z_n \approx n^{-\frac{1}{2}}\sqrt{2f(x_0+a)}W(t) - \frac{1}{2}Vt^2$, where $V = f'(x_0-a) - f'(x_0+a)$. Furthermore, $\widehat{x} \approx \arg\max_t Z_n$, where $W(t)$ is a standard two-sided Brownian motion starting in 0. Using the trick in the previous proposition we obtain

$$c = n^{-\frac{1}{3}}\left(\frac{8f(x_0+a)}{V^2}\right)^{\frac{1}{3}}.$$

It follows that $n^{\frac{1}{3}}(\widehat{x}-x_0) \xrightarrow{d} \tau^{\frac{1}{3}}Z$, where $\tau = \frac{8f(x_0+a)}{V^2}$. □

Comparing this work to that of the median, there are two important differences. First, there is no unique source of randomness across all θs, instead we must take the maximum over a Brownian motion with parabolic drift. Second, the argument for $n^{-\frac{1}{2}}$ rate doesn't fall through. This is because the standard deviation of the random part $n^{-\frac{1}{2}}\sqrt{2f(x_0+a)}W(y)$ is of too small order, only $\sqrt{\theta-\theta_0}$. More generally, assume the limit process has the form $c_1(\theta-\theta_0)^2 + n^{-\frac{1}{2}}c_2\sqrt{\theta-\theta_0}Z_\theta$ for some constants $c_1 < 0, c_2$. In order to carry through the equalisation mentioned below Theorem on page 13 on the median, $\theta-\theta_0 = tn^{-\frac{1}{3}}$ must be chosen. From this we get $Z_n(tn^{-\frac{1}{3}}) \approx n^{-\frac{2}{3}}(c_1t^2 + c_2\sqrt{t}Z_\theta)$. This is the intuition behind cube root asymptotics offered by Kim & Pollard (1990). It happens when the first order terms of the variance in the random deviation part of $P_n(m_\theta - m_{\theta_0})$ will not cancel, as in equation 2.2.2.

2.2.3 Consistency

In the Donsker and Glivenko-Cantelli results, we were concerned with uniform convergence over the whole space: E.g., $n^{-\frac{1}{2}}(F_n - F) \xrightarrow{d} \mathbb{G}$ concerns the uniform convergence over $\mathbb{R}$. In our case, we're not interested in uniform convergence per se, only the continuity of the argmax functional: When $M_n \to M$ uniformly, it seems intuitively clear that $\arg\max M_n \xrightarrow{d} \arg\max M$, but we only require that $M_n \to M$ uniformly on compacta. In our applications, $M_n(\theta) = P_n m_\theta$ and $M(\theta) = P m_\theta$ for some criterion function m_θ.

The following result, a slightly modified version of van der Vaart and Wellner (1996, corollary 3.2.3, p. 287), is a general theorem for proving consistency of M-estimators. The only difficult condition is the Glivenko-Cantelli condition $||P_n m_\theta - P m_\theta|| \xrightarrow{P} 0$, which we use the bracketing entropy machinery to establish.

Theorem 2.2.6 (Consistency theorem)**.** *Let M_n be a stochastic process index by a metric space Θ, and let $M : \theta \to \mathbb{R}$ be a deterministic function. If $||M_n - M|| \xrightarrow{P} 0$ and θ_0 is well-separated, then any sequence $\widehat{\theta}_n$ which nearly maximises $\theta \mapsto M_n(\theta)$ for any n will be consistent for θ_0.*

The condition that $\widehat{\theta}_n$ nearly maximises the map $\theta \to M_n(\theta)$ is understood to mean that $\widehat{\theta}_n \geq \arg\max_\theta M_n(\theta) - o_p(1)$, where $\theta_0 = \arg\max_\theta M(\theta)$, which we assume is identified. That is, we assume θ_0 is a unique, global maximiser of $M(\theta)$. Also, θ_0 is well-separated (van der Vaart, 2000, p. 60) if it is the unique global maximum and it can't be approximated outside its neighbourhoods: For all $\epsilon > 0$, $\sup_{d(\theta,\theta_0)\geq\epsilon} M(\theta) < M(\theta_0)$. In one dimension, this condition can be broken if there are horizontal asymptotes in $M(\theta)$. Notice that the condition is satisfied on compact neighbourhoods of θ_0 whenever M is continuous. This is because M attains its supremum on any compact set K, hence $\sup_{K\cap d(\theta,\theta_0)\geq\epsilon} M(\theta) = \max_{K\cap d(\theta,\theta_0)\geq\epsilon} M(\theta) < M(\theta_0)$, since θ_0 is the unique maximum.

2.2.4 The rate theorem

The following theorem is the most important technical tool of this thesis, and is almost exactly the same as Theorem 3.2.10 from van der Vaart and Wellner (1996, p. 293). A similar result which only covers $n^{\frac{1}{3}}$-convergence can be found in Kim and Pollard (1990). We will have need of the additional generality of this result in Chapter 4 on histograms.

In order to make all the uniform Lindeberg central limit theorems involved in this theorem work, we require the local bracketing entropy integral to be finite,

$$\int_0^\infty \sup_{\delta<\delta_0} \sqrt{\log N_{[]}(\epsilon||M_\delta||_2, \mathcal{M}_\delta, L_2(P)}d\epsilon < \infty, \tag{2.2.3}$$

where M_δ is an envelope for F_δ, and $\mathcal{M}_\delta = \{m_\theta - m_{\theta_0} \mid d(\theta, \theta_0) \leq \delta\}$. We will not deal with this condition in any detail, as it is not very enlightening and quite tedious. This condition can be replaced with a uniform entropy integral condition, which can be verified by establishing bounds on the Vapnik-Chervonenkis dimension of $\mathcal{M}_\delta$ (see e.g. van der Vaart and Wellner (1996, chapter 2.6) for definitions and results about VC dimension). we will not make use of that approach here, but will use concepts from Vapnik-Chervonenkis theory in Section 4.4.

The following is theorem 3.2.10 from van der Vaart and Wellner (1996, p. 297), and is the main technical tool for rigorously proving convergence of M-estimators. We will use it to derive the limit distributions of irregular histograms in Chapter 4.

rate theorem. *For each θ in an open subset $U \subseteq \mathbb{R}^n$, let m_θ be a measurable function such that $\theta \mapsto Pm_\theta$ is twice continuously differentiable at its maximum θ_0, with non-singular Hessian (or information matrix) V. Let the bracketing entropy integral condition (2.2.3) hold. Assume there is a continuous function ϕ such that $\phi^2(\delta) \geq P^\star M_\delta^2$ with $\delta \mapsto \phi(\delta)/\delta^\alpha$ decreasing for some $\alpha < 2$. Assume the following Lindeberg condition is met: For every $\eta > 0$,*

$$\lim_{\delta \searrow 0} \frac{P^\star M_\delta^2\{M_\delta > \eta\delta^{-2}\phi^2(\delta)\}}{\phi^2(\delta)} = 0. \tag{2.2.4}$$

We also require the uniformity condition

$$\lim_{\epsilon \searrow 0} \limsup_{\delta \searrow 0} \sup_{||h-g|| < \epsilon} \frac{P(m_{\theta_0+\delta g} - m_{\theta_0+\delta h})^2}{\phi^2(\delta)} = 0, \tag{2.2.5}$$

when $\min(||h||, ||g||) \leq K$ for all $K \in \mathbb{N}$. Furthermore,

$$\lim_{\delta \searrow 0} \frac{P(m_{\theta_0+\delta g} - m_{\theta_0+\delta h})^2}{\phi^2(\delta)} = E(G(g) - G(h))^2 \tag{2.2.6}$$

for a zero-mean non-degenerate Gaussian process G such that $G(g) = G(h)$ almost surely if and only if $h = g$. Then there exists a version of G with bounded, uniformly continuous sample paths on compacta. Define r_n as the solution to $r_n^2\phi(r_n^{-1}) = \sqrt{n}$. Then the rescaled process $h \mapsto r_n^2(P_n m_{\theta_0+r_n h} - P_n m_{\theta_0})$ converges weakly to $G(h) + \frac{1}{2}h^T V h$. If $\widehat{\theta}_n$ nearly maximises the map $\theta \to P_n m_\theta$ for every n and converges in outer probability to θ_0, the sequence $r_n(\widehat{\theta}_n - \theta_0)$ converges in

distribution to the unique maximiser $\widehat{h}$ *of the process* $h \mapsto G(h) + \frac{1}{2}h^T V h$.

In our applications, the uniformity and Gaussian conditions are easy to verify. The condition that "$\widehat{\theta}_n$ nearly maximises the map $\theta \to P_n m_\theta$ for every n and converges in outer probability to θ_0", is understood to mean to that $\widehat{\theta}_n$ is consistent for θ_0 and $\widehat{\theta}_n \geq \arg\max_\theta P_n m_\theta - o_p(1)$.

2.3 Least median squares regression

Consider the ordinary linear regression model

$$Y = X^T\beta + \epsilon,\ \epsilon \sim N(0, \sigma^2),$$

where the samples are independent. The maximum likelihood estimate of this model is given by the ordinary least squares (OLS) solution, namely

$$\arg\min_\beta \frac{1}{n}\sum_{i=1}^{n}(X_i^T\beta - Y_i)^2.$$

While this estimator is efficient under model conditions with no contaminated data, it is very sensitive even to single outliers. Notice the sum involved in the least squares solution: We minimise the mean of the squared residuals $r_i^2 = (X_i^T\beta - Y_i)^2$. It is well known that the median is far more robust than the mean as an estimator of the centre in a symmetric distribution, for instance it has a breakdown point of 50%. This knowledge gave rise to the least median squares estimator (LMS) (Rousseeuw, 1984):

$$\begin{aligned}\widehat{\beta} &= \arg\min_\beta \text{med}((X^T\beta - Y)^2) \\ &= \arg\min_\beta \text{med}|X^T\beta - Y|.\end{aligned}$$

The equality between the first and second line follows from the monotonicity of $x \mapsto x^2$ when $x \geq 0$. This estimator is extremely robust to outliers in both the x and y direction. Similarly to the median, it has a breakdown point of 50%, which means that a data set (Y_i, X_i) can contain up to 50% percent contamination without the regressor being affected by it (see Section 3.5). This is clearly the upper bound for any reasonable estimator. Nonetheless, it pays for this robustness by being very inefficient. Its cube root asymptotics were derived both in Rousseeuw and Leroy (2005) and in Kim and Pollard (1990). The

estimator is available in `R` through the function `lmsreg` in the built-in package `MASS`.

We have not succeeded in isolating the conditions under which the LMS is consistent, but it appears likely that it is consistent whenever the covariates are sufficiently nicely distributed and the error terms are symmetric, independent of each other and X_i, and satisfying $\text{med}(\epsilon_i) = 0$.

Another idea for robustifying the OLS is to use $\arg\min_\beta \frac{1}{n}\sum_{i=1}^n |X_i^T\beta - Y_i|$, which is known as L_1-regression or least absolute deviations (LAD) regression. In one sense, this is the generalisation of the median to the regression setting, recall Proposition 2.2.2 on page 13. It is the maximum likelihood estimator under the assumption that the ϵs are Laplace distributed. While this estimator is more robust than OLS in the sense that it doesn't put larger weight on large residuals and smaller weight on small residuals, its breakdown point is equally bad. Modulo regularity conditions on the covariates, LAD is consistent under the assumption that $\text{med}(\epsilon_i) = 0$ and the ϵs being i.i.d. with a positive density in the neighbourhood of 0 (Pollard, 1991).

Properties

The LMS estimator has some nice properties, described in depth in the aforementioned monograph (Rousseeuw and Leroy, 2005, p. 116-117), including several desirable equivariance properties.

Definition 2.3.1. Let $\left\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R} \mid i = 1, ..., n\right\}$ be observed data and $T_n : (\mathbb{R}^d \times \mathbb{R})^n \to \mathbb{R}^{d+1}$ be a regression estimator. We define the following properties:

1.) *Regression equivariance:* T_n satisfies this property if

$$T_n(\{(x_i, y_i + v^T x_i)\}) = T_n(\{(x_i, y_i)\}) + v,$$

whenever $v \in \mathbb{R}^d$.

2.) *Affine equivariance:* This is satisfied when

$$T_n(\{(Ax_i, y_i)\}) = (A^T)^{-1} T_n(\{(x_i, y_i)\}),$$

for any non-singular transformation A of the covariates.

3.) *Scale equivariance:* The estimator is said to be scale equivariant if

$$T_n(\{x_i, cy_i\}) = cT_n\{(x_i, y_i)\}.$$

Affine equivariance implies that the estimator is independent of the choice of coordinate system, which is clearly desirable. Scale invariance also appears important to have. The regression equivariance doesn't look as intuitively desirable. To explain what's going on, take a look at the LS estimator, $\widehat{\beta} = \arg\min_\beta(\beta^T x_i - y_i)^2$. When $v \in \mathbb{R}^d$, $\arg\min_\beta(\beta^T x_i - (y_i + v^T x_i)^2 = \arg\min_\beta((\beta^T - v^T)x_i - y_i)^2$, which equals $\widehat{\beta} + v$. Regression equivariant estimators have the property that one can assume, without loss of generality, that $\widehat{\beta} = 0$.

There are reasonable, highly robust estimators without some of these properties. An example is the repeated median estimator of Siegel (1982), another estimator with 50% breakdown point. Let d be the covariate dimension and $\left\{(x_i, y_i) \in \mathbb{R}^d \times \mathbb{R} \mid i = 1, ..., n\right\}$ be observed data. For any selection of $d+1$ numbers $i_1, i_2..., i_{d+1}$ between 1 and n, the system $(1, x_{i_1}, x_{i_2}, ..., x_{i_d})^T\beta = (y_{i_1}, y_{i_2}, ..., y_{i_d})^T$ has a unique solution. This follows from the ubiquitous full rank condition on the covariates. Denote this solution $\beta(i_1, i_2, ..., i_d)$, and let $\beta_j(i_1, i_2, ..., i_d)$ be its jth coordinate. Define the repeated median as follows:

$$\widehat{\beta}_j = \text{med}_{i_1}(\text{med}_{i_2}(....\text{med}_{i_d}(\{\beta_j(i_1, i_2, ..., i_d)\}))...)$$

When $d = 1$, a pair of tuples (x_j, y_j) and (x_i, y_i) yield the parameter estimates

$$\begin{aligned} \beta_0^{(ij)} &= \frac{x_j y_i - x_i y_j}{x_j - x_i}, \\ \beta_1^{(ij)} &= \frac{y_j - y_i}{x_j - x_i}. \end{aligned}$$

The desired estimates is

$$\widehat{\beta}_0 = \text{med}\{\text{med}\{\beta_0^{(ij)} \mid j = 1, ..., n\} \mid j \neq i\},$$

$$\widehat{\beta}_1 = \text{med}\{\text{med}\{\beta_1^{(ij)} \mid j = 1, ..., n\} \mid j \neq i\}.$$

This estimator is not affine equivariant. It is not a very efficient estimator at standard normal error conditions, with efficiency at $4/\pi^2 \approx 40.5\%$ (Hössjer et al., 1994). (Though it compares nicely to the LMS, with an efficiency of 0!) A related estimator is the Theil-Sen estimator, where the median isn't repeated. In dimension two, the slope estimate is $\text{med}\left\{\frac{y_j - y_i}{x_j - x_i} \mid i < j\right\}$ and the intercept estimate is $\text{med}\left\{\frac{x_j y_i - x_i y_j}{x_j - x_i} \mid i < j\right\}$. It is more efficient (90.5%) than the repeated median estimator, but has a lower breakdown point at 29%, (ibid.).

The Theil-Sen estimator isn't affine equivariant either. The gist of the matter, for both estimators, is that the estimator runs independently on each coordinate, but the runs must be "coordinated" in order to make the A^T commute with the median.

The LMS estimator also has the *exact fit* property: Whenever $\lceil \frac{n}{2} \rceil$ observations lie on a straight line $\theta^T x$, the LMS estimate is $\widehat{\beta} = \theta$. This property is not shared by e.g. OLS regression, but it is not clear whether its desirable in the first place. This property is a straight forward consequence of the geometrical interpretation of the LMS: It attempts to find the mid-line of the tube of smallest radius which contains at least half of the observations Rousseeuw and Leroy (2005, p. 24). The geometry is arguably easier to understand than OLS geometry. Nevertheless, it makes the estimator sensitive to small perturbations in data values, as described in Hettmansperger and Sheather (1992).

Calculation

If there are no covariates, the regression model reduces to a location model $Y_i = \theta + \epsilon_i$. In this special case, it is easy to describe its LMS estimator. Now the objective function reduces to $f(\theta) = \text{med}(|y_i - \theta|)$. The next proposition was first observed Steele and Steiger (1986) in a more general. This particular proof is a simplification of theirs to $d = 2$. In it we use the "high median" convention: When faced with an even number of observations, $n = 2k$, we define $\text{med}\{x_i\} = x_{(k)}$.

Proposition 2.3.2. *The LMS estimate of θ in $Y = \theta + \epsilon$, where med$(\epsilon) = 0$, is given by*

$$\widehat{\theta} = \min_{A \in \mathcal{A}} \frac{1}{2}(\max A + \min A),$$

where $\mathcal{A} = \left\{ S \subset \{y_1, ..., y_2\} \mid \#S = \lfloor \frac{n}{2} \rfloor + 1 \right\}$. In addition,

$$f(\widehat{\theta}) = \min_{A \in \mathcal{A}} (\max A - \min A).$$

Proof. We first show that the local minima of $f(\theta)$ is on the form

$$(\max A - \min A)_{A \in \mathcal{A}}.$$

Let θ be arbitrary, and assume that $f(\theta)$ is on the form above. The value of $f(\theta)$ is obtained by taking the $(\lfloor \frac{n}{2} \rfloor + 1)$st element of the list $\{|y_i - \theta| : i = 1, ..., n\}$, hence a slight tilt of θ to the left will make the distance to the rightmost element slightly larger, increasing the value of $f(\theta)$; the same argument works for a

tilt to the right, hence $f(\theta)$ is a local minimum. Conversely, let A be the $(\lfloor \frac{n}{2} \rfloor + 1)$-element head of $\{|y_i - \theta| : i = 1, ..., n\}$ associated with θ and assume $|\max A - \theta| > |\min A - \theta|$. If we slide θ towards the right, we will reach a point θ' where $|\max A - \theta'| = |\min A - \theta'|$. Here $|\max A - \theta'| < |\max A - \theta|$, showing that $f(\theta)$ is not a local minimum. □

This is reminiscent of the *shorth* estimator of the centre in a symmetric distribution (Andrews and Hampel, 2015), whose cube root asymptotics is rigorously derived in Kim and Pollard (1990). This estimator is defined as $\frac{1}{\lfloor \frac{n}{2} \rfloor + 1} \sum_{y \in A} y$, the mean of the shortest interval containing half of the observations. The computation of these estimators is simple: We order the observations, then check the distances $x_{i+\lfloor \frac{n}{2} \rfloor + 1} - x_i$ for $i = 1, ..., \lfloor \frac{n}{2} \rfloor$, and finally compute the midrange (mean) in order to get the LMS estimate (shorth estimate). Together this takes $O(n \log n)$ time. The "weirdness" of this explicit solution form casts doubt on whether the LMS is such a good idea after all. Consider the following example, inspired by Hettmansperger and Sheather (1992), which shows that the LMS behaves unpredictably when presented with slightly manipulated data. They called this phenomenon "local instability", and observed it in real data sets.

Example 2.3.3. Let $y = (1, 3, 4, 6.0001, 10)$ be the set observations. The 3-ary set attaining the smallest midrange is $(1, 3, 4)$, with midrange 2.5. Assume that $y' = (1, 3, 4, 5.999, 10)$ is observed instead. The 3-ary set attaining the smallest midrange has changed into $(3, 4, 5.999)$, with a midrange approximately equal to 4.5. Clearly, the same criticism applies to the shorth estimator.

Anomalies like this, which stem from discontinuity, is a burden we have to bear when dealing with cube root asymptotics. This kind of jump in estimates only happens in OLS when the covariates are multicollinear. We will discuss a phenomenon like this in Section 4.7 as well.

To compute $\widehat{\beta}$ in a regression setting one would have to use some more advanced combinatorial optimisation. The problem is computationally hard, but has received the attention of the computational geometers Edelsbrunner and Souvaine 1990, where a $O(n^2)$-algorithm is given for the case of one covariate. In practice, approximation algorithms are used, see Rousseeuw and Leroy (2005).

Rousseeuw and Leroy (2005, chapter 6) proposes to use this method to identify outliers, remove them, and perform an ordinary LS regression in order to get confidence intervals and perform hypothesis tests afterwards. As the authors say on p. 229,

> Many diagnostics are based on the residuals resulting from LS. However, this starting point may lead to useless results because of the following reason. By definition, LS tries to avoid large residuals. Consequently, one outlying case may cause a poor fit for the majority of the data because the LS estimator tries to accommodate this case at the expense of the remaining observations. Therefore, an outlier may have a small LS residual, especially when it is a leverage point [...]. As a consequence, diagnostics based on LS residuals often fail to reveal such points.

Finally, we mention that the LMS is not the method of choice for performing robust regression analysis. There are a myriad of different procedures for robust regression, most of which are both easier to compute and far more efficient, see e.g. Maronna et al. (2006, chapter 4,5).

2.4 Binary decision trees

A binary decision tree is a step function of the form $g(x; \beta_l, \beta_u, d) = \beta_l 1_{[x \leq d]} + \beta_u 1_{[x>d]}$, $x \in \mathbb{R}$. Let $(y_i, x_i) \in \mathbb{R}^2$ be observations from some regression model $y_i = f(x_i) + \epsilon_i$, where $f = E(Y \mid x)$ is a given function. Now we wish to approximate f by means of a binary decision tree. We consider the situation when the covariates X come from a density p_X, hence we can talk about the joint distribution of (Y, X). As indicated, we wish to approximate a regression function $E(Y \| X = x) = f(x)$ by a step function

$$g(x; \beta_l, \beta_u, d) = \beta_l \{x \leq d\} + \beta_u \{x > d\}.$$

In machine learning, this procedure is typically iterated (Hastie et al., 2005, chapter 9), an application which will not be discussed here. Given the true distribution P of (Y, X), we can talk about the "least false values" in the mean square error sense:

$$(\beta_l^0, \beta_u^0, d^0) = \underset{(\beta_l, \beta_u, d)}{\arg\min} P\left[(Y - [\beta_l \{X \leq d\} + \beta_u \{X > d\}])^2\right],$$

which gives rise to the least sum of squares estimator,

$$
\begin{aligned}
(\widehat{\beta_l}, \widehat{\beta_u}, \widehat{d}) &= \arg\min_{\beta_l,\beta_u,d} P_n \left[(Y - [\beta_l \{X \leq d\} + \beta_u \{X > d\}])^2 \right], \\
&\quad \arg\min_{\beta_l,\beta_u,d} \sum_{i=1}^{n} \left[(Y_i - [\beta_l \{X_i \leq d\} + \beta_u \{X_i > d\}])^2 \right].
\end{aligned}
$$

Now we must make the distinction between two *very* different problems, only one of which is covered here. It concerns the shape of the true f.

(C1) f is on the form g, or reasonably close in the sense that it actually has a jump discontinuity at d^0,

(C2) f is not on the form g, specifically, it is differentiable at d^0.

Modelling with (C2) satisfied is like using histograms in a regression setting, and is called *split-point analysis* by Banerjee and McKeague (2007). They proved the following:

Theorem 2.4.1. *Under conditions (A1)-(A5) of Banerjee & McKeague, including (C2) above,*

$$
n^{\frac{1}{3}}(\widehat{\beta_l} - \beta_l^0, \widehat{\beta_u} - \beta_u^0, \widehat{d_n} - d^0) \xrightarrow{d} (c_1, c_2, 1) \arg\max_h \left[Vh^2 + aW(h) \right],
$$

where W is a standard two-sided Brownian motion and V and a are real constants depending on $p_X(d^0)$, $f'(d^0)$ $(\beta_l^0, \beta_u^0, d^0)$ and $P(d^0)$.

While the authors only found the limiting distribution for one split point, it is probably straightforward to extend it to any number of split points. This would give us the asymptotic theory for a particular form of histogram regression with data driven partitioning rules (see Nobel et al. (1996)). Peculiarly, the value of $(\widehat{\beta_l} - \beta_l^0, \widehat{\beta_u} - \beta_u^0, \widehat{d_n} - d^0)$ depends only one random variable, namely $Z = \arg\max_h Vh^2 + aW(t)$. We will obtain an analogy of Theorem 2.4.1 for irregular histograms on page 107, where the same curious "single random" Z appears. There aren't many obvious applications for these binary decision trees, but Banerjee and McKeague (2007) applied it on an environmental problem concerning the Everglades National Park in Florida. The convergence rate of the split points was first observed by Büchlmann and Yu (2002), in a paper analysing the effect of *bootstrap aggregation* on decision trees, a variance reduction technique. Interestingly, they erred on the convergence rate of the levels ($\widehat{\beta_l}$ and $\widehat{\beta_u}$), believing they converged at the ordinary $n^{\frac{1}{2}}$-rate instead. This can serve as a warning on how hard it can be to get these things right.

On the other hand, if (C1) is satisfied, we obtain n asymptotics for d and $n^{\frac{1}{2}}$-asymptotics for β_l and β_u, see e.g. Kosorok (2007, section 14.5.1). This happens as we attempt to isolate *real* change-points in the underlying regression model. Note that the rate theorem can't be used in this case, as the smoothness condition on Pm_θ isn't satisfied.

This phenomenon of $n^{\frac{1}{3}}$-convergence for smooth underlying distributions and $n^{\frac{1}{2}}/n$-convergence for appropriately discontinuous underlying distributions is extremely likely to happen for irregular histograms (see Chapter 4) as well. Indeed, our discussion of histograms in that chapter will only concern underlying Fs satisfying the analogue of (C2).

2.5 Resampling

2.5.1 Bootstrap

Let $X_1, \ldots, X_n \overset{i.i.d.}{\sim} P$ and $R_n(X_1, \ldots, X_n; P)$ be a functional of both the data and the probability. Such a functional is called a *root*, after Beran (1983). The most prominent example of R_n is $\sqrt{n}(\widehat{\theta}_n - \theta_0(P))$, where $\widehat{\theta}_n$ is an estimator of $\theta_0(P)$ depending only on $X_1, \ldots, X_n$, for instance an M-estimator or a more general statistical functional. Whenever P is unknown, it is of interest to approximate R_ns distribution in order to construct confidence intervals, perform hypothesis tests, etc. A sensible approach is to approximate P with a known distribution Q_n, usually (if not always) data dependent, and calculate the distribution of $R_n(\widehat{X}_1, \ldots, \widehat{X}_n; Q_n)$, where $\widehat{X}_1, \ldots, \widehat{X}_n \overset{i.i.d.}{\sim} Q_n$ This can sometimes be done analytically, but is typically done numerically through Monte Carlo methods. Some possible choices for Q_n are

1. P_n, the empirical measure. This leads to the ordinary non-parametric bootstrap (Efron, 1979).

2. $Q_n = \widetilde{P}_n$, a smoothed version of P_n, which yields the smoothed bootstrap.

3. Use a parametric distribution with plug-in parameters. From this we get the parametric bootstrap.

The non-parametric bootstrap is the most commonly used of these procedures, and has been thoroughly researched. For a reference, see Shao and Tu (2012). Still, the smoothed bootstrap hasn't received that much attention. In recent

years, there has been done some work on the smoothed bootstrap in the context of cube root asymptotics. This research is motivated by the fact that the ordinary bootstrap isn't consistent in general in cube root asymptotics.

Let $H_n(P)$ be the distribution function of $R_n(X_1, \ldots, X_n; P)$, and assume it converges in distribution to some distribution function $H(P)$. Likewise, let $H_n(Q_n)$ be the distribution function of $R_n(\widehat{X}_1, \ldots, \widehat{X}_n; Q_n)$. Our goal is to estimate $H_n(P)$ by $H_n(Q_n)$, and a necessary condition for this to be reasonable is that $H_n(Q_n)$ is "consistent". Since the bootstrap attempts to approximate a unique distribution conditionally, we say that the bootstrap is consistent if $H_n(Q_n) \xrightarrow{d} H(P)$ conditionally on $X_1, X_2, ... \overset{i.i.d.}{\sim} P$ for almost every such sequence of observations. Hence, if $H(P)$ is assumed to continuous, which is usually the case, the bootstrap is consistent if

$$\sup_x |H_n(Q_n)(x) - H(P)(x)| \to 0$$

almost surely (van der Vaart, 2000, lemma 2.11).

The following basic result on the bootstrap is perhaps the most important one. Here $\mu(P)$ is the mean of P, while $\sigma^2(P)$ is its variance.

Theorem 2.5.1. *Assume P has finite variance. Let P_n be a sequence of distributions such that $P_n \xrightarrow{d} P$. In addition, assume both $\mu(P_n) \to \mu(P)$ and $\sigma^2(P_n) \to \sigma^2(P)$. Then the bootstrap is consistent for the root $H_n(X_1, ..., X_n; P) = \sqrt{n}(\overline{X} - \mu(P))$, where $H(P) \sim N(0, \sigma^2(P))$.*

For a proof, see e.g. Politis et al. (1999, proposition 1.3). The finite variance condition is not only needed to assure asymptotic normality. In fact, for some cases of X_is with infinite variance, the bootstrap isn't only not consistent, but has no deterministic limit in probability (Athreya, 1987). This theorem generalises to the case of smooth functionals as well (Shao and Tu, 2012, theorem 3.6).

The known cases of estimators with cube root asymptotics are in a sense non-smooth, and also the most famous case of n-asymptotics, the maximum of the uniform distribution. The nonparametric bootstrap is inconsistent for the maximum in a uniform distribution, which we now show.

Example 2.5.2. (From Knight (1989)) Let $X_1, ..., X_n \overset{i.i.d.}{\sim} \mathcal{U}(0, \theta)$, and denote the order statistics $X_{(1)} < X_{(2)} < ... < X_{(n)}$. We wish to estimate the θ by its MLE, $\widehat{\theta}_n = X_{(n)}$. It is well known that $H_n(P) = n\frac{\theta - X_{(n)}}{\theta} \xrightarrow{d} \exp(1)$, and we take this as our root, with bootstrap variant $H_n(P_n) = n\frac{X_{(n)} - X^*_{(n)}}{X_{(n)}}$. First we find the

unconditional distribution. Recall the distribution for the maximum of a sample $Z_1, ...Z_n$: $P(Z_{(n)} \leq x) = P(Z_1 \leq x, Z_2 \leq x, ..., Z_n \leq x) = P(Z_1 \leq x)^n$, which we can use to find

$$\begin{aligned} P(X^*_{(n)} = X_{(n-i)}) &= P(X^*_{(n)} \leq X_{(n-i)}) - P(X^*_{(n)} < X_{(n-i)}) \\ &= (1 - \frac{i}{n})^n - (1 - \frac{i+1}{n})^n \\ &\approx e^{-i} - e^{-(i+1)}. \end{aligned}$$

Also, it is assumed known that

$$(0, n(X_{(n)} - X_{(n-1)})/X_{(n)}, n(X_{(n)} - X_{(n-2)})/X_{(n)}, ...) \xrightarrow{d} (U_1, U_2, U_3, ...)$$

where $U_1 = 0$, $U_i = \sum_{j=2}^{i} V_j$, with $V_1, V_2, ...$ i.i.d. standard exponentials. Thus H^*_n converges unconditionally to the mixture $\sum_{i=1}^{\infty} \xi_i U_i$, where ξ is an infinite multinomial vector with cell probabilities $p_i = e^{-i} - e^{-(i+1)}$.

Conditioning on $X_1, ..., X_n$, we get $R_n(P_n) = \sum_{i=0}^{n} \pi_i n \frac{X_{(n)} - X_{(n-i)}}{X_{(n)}}$, where π_i is a multinomial vector with cell probabilities $(1 - \frac{i}{n})^n - (1 - \frac{i+1}{n})^n$. Note that $(1 - \frac{i-1}{n})^n - (1 - \frac{i}{n})^n \nearrow 1 - e^{-1} \approx 0.632$ when $i = 0$. Since $X_{(n)} - X_{(n-0)} = 0$, the bootstrap distribution will always have a point mass at 0, sharply discordant with our wish for consistency. In fact, it has no limit distribution with probability 1, as both $P(\limsup_{n\to\infty} n \frac{X_{(n)} - X_{(n-i)}}{X_{(n)}} = \infty) = 1$ and $P(\liminf_{n\to\infty} n \frac{X_{(n)} - X_{(n-i)}}{X_{(n)}} = 0) = 1$, see Bickel and Freedman (1981).

On the other hand, the parametric bootstrap is easily seen to be consistent. Conditioned on $X_1, X_2, ...$,

$$\begin{aligned} P(n \frac{X_{(n)} - X^*_{(n)}}{X_{(n)}} \leq x) &= P(X^*_{(n)}/X_{(n)} \geq 1 - \frac{x}{n}), \\ &= 1 - (1 - \frac{x}{n})^n, \\ &\to 1 - e^{-x}. \end{aligned}$$

As we can see, not only is the parametric bootstrap consistent, but we have the identity $H_n(Q_n) = H_n(P)$ for every n.

An important task is to identify necessary and / or sufficient conditions for the non-parametric bootstrap to be consistent. This is a difficult task which

goes far beyond the scope of this thesis, but we will supply some heuristics which indicates when we would expect the bootstrap to be consistent.

> [...] The bootstrap is not foolproof, even for statistics whose asymptotic distribution is normal. Asymptotic optimality, or even consistency, of the bootstrap estimate $H_n(\widehat{P}_n)$ is not to be expected unless $H_n(P)$ depends smoothly upon P.
>
> - Rudolph Beran in Beran (1982)

In accordance with this quote, Bickel and Freedman (1981) propose the following heuristics for when we can expect consistency of the bootstrap,

1. Uniform convergence of $H_n(Q_n)$ to $H(P)$ over all Q_ns in a shrinking neighbourhood around P;

2. $H_n(P)$ depends smoothly on P.

In the previous example $H_n(P)$ does not depend smoothly on P whenever the Ps are discrete. In cube root asymptotics, the smoothness condition is usually not satisfied, This is clearly seen in Section 4.5.2 on the histograms, where a slight tilt in P, *even if P is constrained to be smooth*, can give a different rate of convergence (from $n^{\frac{1}{3}}$ to $n^{\frac{1}{2}}$). If P isn't constrained to be smooth, the rate of convergence can even change from $n^{\frac{1}{3}}$ to n! The basic problem appears to be that $H_n(Q_n)$ does not emulate $H_n(P)$ well whenever Q_n is non-smooth, which is clearly the case when $Q_n = P_n$. This has lead to some work on the *smoothed bootstrap* in the context of cube root asymptotics. Kosorok (2008) proved that the non-parametric bootstrap is inconsistent for Grenander's estimator, but developed a consistent variant of the smoothed bootstrap. Abrevaya and Huang (2005) attempted to show that the ordinary bootstrap fails for Manski's estimator. However, as discussed in the paper of Sen et al. (2010) on bootstrapping Grenander's estimator, their result is likely erroneous: Their result strongly indicate that the bootstrap distributions of cube root estimators have no weak limit almost surely, contradicting the main result of the aforementioned paper. Léger and MacGibbon (2006) made an attempt at a general theorem for proving consistency and inconsistency of bootstrap variants for cube root asymptotics. Seijo and Sen (2011) developed a variant of the smoothed bootstrap for Manski's estimator, which appears to work very well.

2.5.2 Subsampling and m-out-of-n bootstrap

Subsampling is an alternative to the bootstrap procedure which is consistent in great generality, but usually requires larger sample sizes in order to work properly. it is also more computationally expensive, as we have to establish a certain nuisance parameter, the block size m. Let $X_1, X_2, X_3, ... \overset{i.i.d.}{\sim} F$ as usual, and let $\widehat{\theta}_n$ be a statistic of interest. Let R_n be $n^\alpha(\widehat{\theta}_n - \theta_0)$, for some $\alpha > 0$, and H_n be its distribution function. We assume that $H_n \xrightarrow{d} H$ as $n \to \infty$ for some continuous limit distribution H.

Choose a block size $b < n$. Define $N_n = \binom{n}{b}$ and let $\widehat{\theta}_{n,b,i}$ be the version of $\widehat{\theta}$ based on the ith sample from the $\binom{n}{b}$ subsets of $X_1, X_2, ..., X_n$ with cardinality b. Define the distribution function $L_n(x)$ by

$$L_n(x) = N_n^{-1} \sum_{i=1}^{N_n} \{b^\alpha(\widehat{\theta}_{n,b,i} - \widehat{\theta}_n) \leq x\},$$

where $\{\cdot\}$ is the characteristic function. Then $L_{n,b}$ is the *subsample distribution* based on $X_1, X_2, ..., X_n$. Unlike the bootstrap, the subsample doesn't attempt to find the distribution $n^\alpha(\widehat{\theta}_n - \theta_0)$, but rather $b^\alpha(\widehat{\theta}_b - \theta_0)$, which gives it an additional source of variance. It depends on the fact that sampling b elements from $X_1, X_2, ..., X_n$ behaves like sampling from the real underlying distribution F, *regardless of the features of* F, provided only that $bn^{-1} \to 0$ as $b \to \infty$.

The following is a variant of Politis and Romano (1994, theorem 2.1). The proof is simple and illuminating, so we provide it in full.

Theorem 2.5.3. *Let b_n be a sequence of block sizes satisfying $n^{-1}b_n \to 0$ as $n \to \infty$, and assume that $H_n(P) \xrightarrow{d} H(P)$, with $H(P)$ continuous. Then*

$$\sup_x |L_n(x) - H(x)| \xrightarrow{p} 0.$$

Proof. Our first observation is

$$\{b^\alpha(\widehat{\theta}_{n,b,i} - \widehat{\theta}_n) \leq x\} \quad = \quad \{b^\alpha(\widehat{\theta}_{n,b,i} - \theta_0) - b^\alpha(\widehat{\theta}_n - \theta_0) \leq x\}.$$

Since the rate of convergence is n^α, we have $b^\alpha(\widehat{\theta}_n - \theta_0) \xrightarrow{p} 0$. Define the random function

$$U_n(x) = N_n^{-1} \sum_{i=1}^{N_n} \{b^\alpha(\widehat{\theta}_{n,b,i} - \theta_0) \leq x\}.$$

Let $\epsilon > 0$ and E_n be the event that $|b^\alpha(\widehat{\theta}_n - \theta_0)| < \epsilon$. Then we obtain the following,

$$U_n(x-\epsilon)1_{E_n} \leq L_n(x)1_{E_n} \leq U_n(x+\epsilon).$$

If $U_n(x) \xrightarrow{p} H(x)$, we can pass ϵ to 0 and obtain $L_n(x) \xrightarrow{p} H(x)$ as well, hence it suffices to show that $U_n(x) \xrightarrow{p} H(x)$. Clearly $E(U_n(x)) = H_n(x)$, and we only need to show that $\mathrm{Var}U_n(x) \to 0$. We can do this by an application of the Rao-Blackwell theorem (Shao, 2007, 2.5): When T is a sufficient statistic for $\mathcal{P}$ and W is an unbiased estimator of a statistic $\tau(P)$, then $W' = E(W|T)$ has less variance than W for every $P \in \mathcal{P}$. Now define

$$U_n^{'} = m^{-1} \sum_{i=0}^{m-1} \{b^\alpha(\widehat{\theta}_{m,i} - \theta_0) \leq x\},$$

where $m = \lfloor n/b \rfloor$ and $\widehat{\theta}_{m,i}$ is $\widehat{\theta}$ calculated from sequence $X_{mb+1}, X_{mb+2}, ..., X_{(m+1)b}$. This is a mean of m identically distributed random variables with variance bounded by 1, hence $\mathrm{Var}L_n^{'} \to 0$. Also, $EL_n^{'} = H(x)$. Since the order of the observations $X_1, X_2, ...$ contains no information about $H_n(x)$, the Rao-Blackwell theorem gives us

$$\mathrm{Var}E(L_n^{'} \mid X_{(1)}, X_{(2)}, ...) \leq \mathrm{Var}L_n^{'}.$$

Seeing as $E(L_n^{'} \mid X_{(1)}, X_{(2)}, ...) = U_n$, the subsample converges pointwise in probability. By Lemma 2.11 in van der Vaart (2000), the convergence is uniform. □

The subsample has been applied on Manski's estimator (Chapter 3) by Delgado et al. (2001), who obtained good results. In ordinary, smooth statistics, the subsample typically behaves worse than the non-parametric bootstrap: While the bootstrap has an error of order $o(n^{-\frac{1}{2}})$, the subsample bootstrap has an error of order $O(n^{-\frac{1}{3}})$, which is substantially worse (Politis and Romano, 1994). Here the error refers to $|L_n(t) - H_n(t)|$.

A similar method is the m-out-of-n bootstrap (Lee and Pun, 2006), a variant of the subsample with block size m where the sampling is done *with* replacement. Under extremely general conditions both the m-out-of-n bootstrap and m-subsampling are consistent provided $\frac{m}{n} \to 0$ as $n \to \infty$. Yet this doesn't give us much guidance in how to select the m in practice. In applications, this choice is made by doing even more resampling.

Say we're interested in obtaining a confidence interval with level α for an

estimator $\widehat{\theta}$. For each choice of m, we can approximate this interval by $[\frac{\frac{\alpha}{2}_m}{n^{\frac{1}{3}}} + \widehat{\theta}, \frac{(1-\frac{\alpha}{2})_m}{n^{\frac{1}{3}}} + \widehat{\theta}]$, where α_m is the α-quantile of $m^{\frac{1}{3}}(\widehat{\theta}^*_m - \widehat{\theta})$. All confidence intervals will not have level α, but rather level $h_\alpha(m)$ for some unknown function h_α dependent on the data generating mechanism. Now we wish to find the m such that $|h_\alpha(m) - \alpha|$ is minimised. Since h_α is unknown in general, we will have to estimate it with $\widehat{h}$. Delgado et al. (2001) propose the *calibration method,* where they use the ordinary, non-parametric bootstrap for this purpose. It is described in Algorithm 2.1.

Algorithm 2.1 The calibration method of Delgado et al. (2001).

1. Select lower and upper bounds for m, called l and u respectively, and minimum step s. We intend to check subsamples of sizes $l, l+s, \ldots, u-s, u$.
2. For $k = 1, ...K$, generate n bootstrap samples $X^*_{1k}, X^*_{2l}, \ldots, X^*_{nk}$. For each applicable subsample size m, put $H(m,k) = 1$ if $\widehat{\theta}$ is in the m-subsampling confidence interval based on these observations, and 0 otherwise.
3. Put $\widehat{h}(m) = \frac{1}{K}\sum_{k=1}^{K} H(m,k)$
4. Minimise $|\widehat{h}(m) - \alpha|$.

Chapter 3

Manski's maximum score estimator

Sometimes in football you have to score goals.

- Thierry Henry, former Arsenal and France striker

Manski's maximum score estimator is a discontinuous M-estimator with cube root asymptotics. In Section 1 we supply the definition of the estimator along with some discussion about what it does. In Section 2 we provide some characterisations of the associated optimisation problem, prove that it is NP-hard, and discuss some related estimators. Section 3 is devoted to its asymptotics, with special emphasis on the conditions leading to consistency. There are no proofs in this Section. Section 4 is devoted to algorithms for its computation. We supply a reformulation of the only workable exact algorithm in the statistical literature, and provide a new exact algorithm (a complete enumeration) for the computation of the entire solution sets when the covariate dimension is $d \leq 2$. Robustness is the theme of Section 5, where we find the breakdown point of the estimator in $d = 1$, and make progress towards finding it in $d > 1$. Finally, we do some simulations to asses the estimator's model robustness and outlier robustness under some different settings, replicating a result of Horowitz (1992).

3.1 Overview

A linear binary response model is a regression model $Y_i = 1_{X_i^T\beta+\epsilon_i\geq 0}$, where the X_is are covariates and ϵ_is are random variables not necessarily independent of the X_is, but independent of each other. Several parametric binary choice

models exists, of which the most famous are the logit and probit models. The logit model arises from taking ϵ_i i.i.d. standard logistic ($F(x) = \frac{\exp(x)}{1+\exp(x)}$), while the probit model arises from taking ϵ_i i.i.d. standard normal.

Fact 3.1.1. *The logit and probit models are the binary choice models where* $\epsilon_i \overset{i.i.d.}{\sim}$ *Logistic*$(0,1)$ *and* $\epsilon_i \overset{i.i.d.}{\sim} N(0,1)$ *respectively.*

Proof. Assume $\epsilon_i \overset{i.i.d.}{\sim}$ Logistic$(0,1)$, and let F be its cdf. Then $Y_i = 1$ if and only if $X_i^T\beta \geq -\epsilon_i$, which since F is symmetric has probability $F(X_i^T\beta)$. The same argument works for the probit model. □

While these two methods are very popular (389,000 results on Google Scholar for "logit" in august 2015, 266,000 for "probit"), they have rather strict distributional assumptions. Most importantly, they exclude heteroskedasticity and react poorly to non-symmetric error distributions.

We will consider a conditional median variant of the binary response model. This problem can be viewed as an instance of quantile regression with missing data. Let the underlying model be $Z_i = X_i^T\beta + \epsilon_i$, where the conditional median is required to be 0: $\text{med}(\epsilon_i|X_i) = 0$. We don't assume independence of the ϵ_is, but we observe only the tuple $(1_{[W_i \geq 0]}, X_i)$. Hence there is a tremendous information loss involved in this model, an information loss so severe that $\sqrt{n}$-consistent estimation under general conditions is impossible, as shown in Chamberlain (1986).

Manski (1975, 1985) proposed a semiparametric estimator of the binary choice model which is consistent provided only $\text{med}(\epsilon_i|X_i) = 0$ (in addition to several regularity conditions discussed in Section 3.3).

$$\begin{aligned} \widehat{\beta} &= \arg\max \left[\sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{X_i^T\beta \geq 0} \right], \qquad (3.1.1) \\ &= \arg\max \left[\sum_{i=1}^{n} \left(Y_i 1_{X_i^T\beta \geq 0} + (1 - Y_i) 1_{X_i^T\beta < 0} \right) \right]. \end{aligned}$$

Notice that the estimator maximises the number of correct predictions, which makes it unusually easy to interpret. The function

$$m(\beta) = \sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{X_i^T\beta \geq 0} \qquad (3.1.2)$$

will sometimes be called *Manski's objective*.

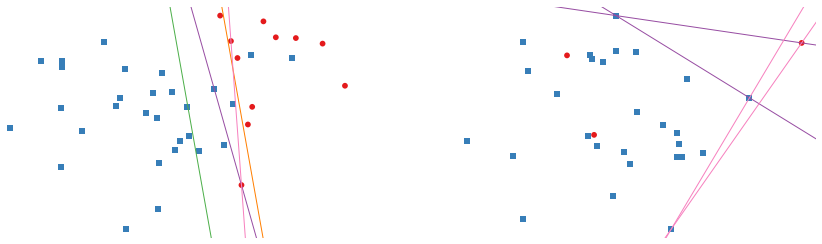

Figure 3.1.1: (left) Generated data according to the probit model, with $\beta_1 = -1$, $\beta_2 = -\frac{1}{2}$, $n = 40$. (Also, $\beta_0 \overset{def}{=} 1$.) Here the green line is the true discriminating line, the orange line is the logistic line (from maximum likelihood), and the pink and purple lines are lines arising from Manski's estimator. Both Manski lines misclassify three points, while the logistic line misclassifies four points. (right) Another simulated data set ($n = 30$, from the same model) with Manski lines in purple and pink. Notice that both pink lines pass through a single blue point and form a "double wedge". Every line in between the bounding lines misclassify two observations, and likewise for the purple lines.

The set $\{x \mid x^T\beta \geq 0\}$ is a half-plane in $\mathbb{R}^d$. Colour the points where $Y_i = 1$ red, and the other points blue, and classify every point in the hyperplane $\{x \mid x^T\beta \geq 0\}$ as red, the others as blue. Then Manski's estimator finds the half-planes which classifies correctly the largest number of observed points. In Figure 3.1.1 we illustrate this by plotting "discriminating lines" of the form $x^T\beta = 0$. If our objective is to obtain good discriminating lines, as compared to obtaining parameter estimates for β, the setting changes from that of Manski's estimator to that of *linear discrimination*. A discriminator based on (3.1.1) is discussed in e.g. Devroye et al. (2013, Section 4.5).

For identifiability, we can postulate $||\beta|| = 1$, where $||\cdot||$ is the Euclidean distance. Such a move is required as we have not imposed any scale on the residuals ϵ_i. Another variant is to impose $\beta_0 = -1$, $\beta_0 = 0$ or $\beta_0 = 1$. Most of our discussion will concern the case when $\beta_0 = 1$, which is easier to analyse than $||\beta|| = 1$. If we know all the solutions for $\beta_0 = -1, 0, 1$, it is easy to find the solutions corresponding to $||\beta|| = 1$.

If the assumptions of the probit model or logit model are close to being true, to use this estimator would be a very bad idea. Not only is it very inefficient, its limiting distribution is intractable, and requires resampling. But this resampling creates a huge computational burden, and behaves poorly when it comes to coverage and size of confidence intervals, etc. Additionally, the theory for the logit and probit is very well worked out, and its standard asymptotics under

maximum likelihood makes it possible to use a wealth of different procedures for model selection. For instance the likelihood ratio test, AIC, BIC and FIC (Claeskens and Hjort, 2008). Also, Manski's estimator gives information about the βs only. If we want to calculate functionals involving the error term ϵ in some way, like estimating the probability on whether a subject with covariates X' will have $Y' \geq 0$, we will require a probability model F for the error terms. Since we allow for arbitrary heteroskedasticity and general "wildness" in the errors, it is impossible to consistently estimate the probability distributions of the errors.

A slight generalisation of the estimator allows non-zero weights w_i on the observations,

$$\widehat{\beta} \;=\; \arg\max \left[\sum_{i=1}^{n} w_i (Y_i - \frac{1}{2}) 1_{\beta^T X_i \geq 0} \right], \tag{3.1.3}$$

a variant we will devise an algorithm to solve. It is useful to have an algorithm for the weighted variant for practical reasons, in particular when performing the bootstrap, as our algorithm will not handle observations that aren't in general position: It is easier to treat two identical observations as one observation with twice the weight. Also, the algorithm is just as fast for the weighted variant as for the non-weighted, and not much more difficult to implement.

In this entire chapter we assume that the covariates are in general position: There are no parallel lines among the covariates, and there is no index i with $\overline{x_i} = \overline{0}$.

Recall the equivariance properties from Section 2.3. There we defined regression, affine and scale equivariance for regression estimators. The regression and scale equivariance concepts don't make sense in the context of the binary response model, as they involve continuous transformations of the responses y_i. Still, Manski's estimator does satisfy the affine equivariance property.

Proposition 3.1.2. *Manski's estimator is affine equivariant.*

Proof. Let $\widehat{\beta} = \arg\max \left[\sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{\beta^T X_i \geq 0} \right]$ be the set of solutions to Manski's objective function. The only place X_i is involved in $\sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{\beta^T X_i \geq 0}$ is in $1_{\beta^T X_i \geq 0}$. Clearly, each of these characteristic functions stay the same when substituting AX_i for X_i and $(A^{-1})^T \beta$ for β, showing that $\sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{((A^{-1})^T \widehat{b})^T A X_i \geq 0} = \sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{\widehat{b}^T X_i \geq 0}$ whenever $\widehat{b} \in \widehat{\beta}$. Since A is invertible, this can be done the other way around as well. This proves that $(A^{-1})^T \widehat{\beta}$ is the solution set of the transformed $\sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{\beta^T A X_i \geq 0}$. □

3.2 Characterisations

It is easier to understand the estimator through two equivalent characterisations, one algebraic and one geometric. Also of interest is its relation to Tukey's concept of *location depth,* a generalisation of the concept of ranks to higher dimensions.

3.2.1 Algebraic formulation

Let A be an $n \times d$-matrix, and $b \in \mathbb{R}^n$. We say that the system of linear inequalities $Ax \leq b$ is *feasible* if there exists a solution to it; if there is no solution, it is called *infeasible.* These terms have the same meaning whenever C is a collection of linear inequalities. Given an infeasible linear system, it is often of interest to find a maximal subsystem which is feasible, that is, a feasible collection C of inequalities from $Ax \leq b$ of cardinality k such that there exists no feasible collection of cardinality greater than k. This problem is of importance in linear programming, as the constraints in such problems easily can turn out to be inconsistent.

Definition 3.2.1. Given a system $Ax \leq b$ of linear inequalities, MAX-FLS (maximal feasible linear subsystem) is the problem of described above. Its corresponding decision problem, FLS(k), decides whether there is a feasible linear subsystem of cardinality k.

Now we will show that solution sets for Manski's estimator 3.1.1 are exactly the solution sets to a certain MAX-FLS problem. For this, define $X_{i0} = 1$ for every i.

Proposition 3.2.2. *Let $X_1, \ldots, X_n \in \mathbb{R}^d$ and $Y_1, \ldots, Y_n \in \{0, 1\}$. Let X^* be the $(n, d+1)-$matrix with elements*

$$x^*_{ij} = \begin{cases} X_{i(j-1)} & \text{when } Y_i = 1, \\ -X_{i(j-1)} & \text{when } Y_i = 0. \end{cases} \tag{3.2.1}$$

Then β^* is an inner solution to 3.1.1 if and only if β^* is a solution to the MAX-FLS of $X^*\beta < 0$.

Proof. In 3.1.1 we wish to maximise the amount of inequalities of the form

$$\begin{aligned} -\beta_0 - \beta_1 X_{i1} ... - \beta_d X_{id} \leq 0 \quad \text{when} \quad Y_i = 1, \\ \beta_0 + \beta_1 X_{i1} + ... + \beta_d X_{id} < 0 \quad \text{when} \quad Y_i = 0, \end{aligned} \tag{3.2.2}$$

that are simultaneously satisfied. Intuitively, one satisfied inequality corresponds to one correct prediction, and we wish to maximise the amount of correct predictions. When β^* is an inner solution to such a set of inequalities, we can substitute any instance of $\leq$ for $<$. Since X^* is the representation of this collection of inequalities in matrix form, we are done. □

In the sequel we will use the representation $X^*\beta \leq 0$ instead of $X^*\beta < 0$. This adds more solutions at the boundaries, but doing the bookkeeping on whether a certain inequality is strict or not is messy. Doing this will not cause any problems in practice. If we want to use the spherical scaling $||\beta|| = 1$, we would intersect the solution set with S^d. If we want to use the more restrictive scaling $\beta = -1, 0, 1$, we will instead use a different matrix (with dimensions $d \times n$): Namely X^* such that

$$x^*_{ij} = \begin{cases} X_{ij}, & \text{when } Y_i = 1, \\ -X_{ij} & \text{when } Y_i = 0. \end{cases}$$

Notice that we have dropped the intercept. Now $\beta = -1$ corresponds to $X^*\beta \leq 1$, $\beta = 0$ corresponds to $X^*\beta \leq 0$ and $\beta = 1$ corresponds to $X^*\beta \leq -1$.

Perhaps the most striking about this characterisation is the fact that solutions are never unique, provided the observations are in general position. This is problematic for several reasons. First, it makes resampling difficult. It is not clear which point we will choose as a centre in the bootstrap or which points to choose from the resampled solutions. This problem is even larger than it looks, because small resampling sizes will often give unbounded solution regions, and we will have to use small samples in order to make the m-out-of-n bootstrap work. Second, it is possible for qualitatively very different points to be maximal.

3.2.2 Geometry

Building on results in the previous section, we present the basic geometry of the problem. Recall the equations (3.2.2),

$$-\beta_0 - \beta_1 X_{i1} - ... - \beta_d X_{id} \leq 0 \quad \text{when} \quad Y_i = 1,$$
$$-\beta_0 + \beta_1 X_{i1} + ... + \beta_d X_{id} < -0 \quad \text{when} \quad Y_i = 0.$$

These describe a collection of affine hyperplanes. Define H_i as the halfspace corresponding to i-th observation. Then we have a geometric analogue to Proposition 3.2.2, namely:

Proposition 3.2.3. *A point β is a solution to Manski's objective (3.1.3) if and only if there is a k and distinct indices $i_1, ..., i_k$ such that $\beta \in \bigcap_{i=1}^{k} H_{i_k}$ and k is maximal in the following sense: Whenever $j_1, ...,\ j_k,\ j_{k+1}$is are distinct indices, $\bigcap_{i=1}^{k+1} H_{i_k}$ is empty.*

The kind of set described in Proposition (3.2.3) will variably be called a *maximal non-empty intersection* or a solution polytope. When we include weights $w = (w_1, ..., w_n)$, the analogue is *w-maximal non-empty intersection.* A set like this is a convex polytope, potentially unbounded. Note that we are *not* looking for a convex polytope with a maximal number of bounding cells. For it is straight forward to construct a convex polytope with more bounding cells than any other convex polytope which isn't a maximal non-empty intersection in our sense. The entire solution set is a disjoint union of such sets (that is, disjoint modulo the boundaries of the planes), which might be many: There is no guarantee of having a unique, connected solution set even when n is large. We provide an example of a solution set consisting of three polygons in Figure 3.2.1. Frequently we will talk about coloured halplanes, where a blue half-plane points downwards while a red half-plane points upwards. The intuition behind this colouring should be clear from Figure 3.2.1.

Notice that the estimator is very dependent on the values of the covariates. Certainly far more than, for instance, logistic regression, which only demands the ordinary Lindeberg conditions on the covariates distribution G. In our case, it will often be very difficult to get precise estimates when the covariates aren't distributed "nicely" enough relatively to the actual parameter values, an issue we will discuss further in Section (3.6).

Number of faces in an arrangement

The world of Manski's estimator in dimension d contains n coloured hyperplanes in $\mathbb{R}^d$, and a matter of interest is the d-polytopes bounded by these coloured hyperplanes. Modulo the colours, this is a fundamental object of study in discrete

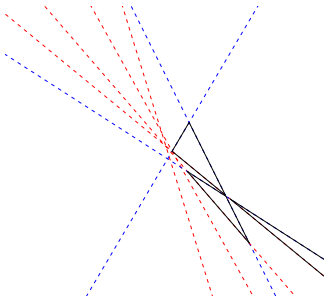

Figure 3.2.1: Random lines in the plane. The black lines delineate the solution sets. The red lines represents half-planes pointing upwards, while the blue lines represent half-planes pointing downwards. All points in the solution set lies in the intersection of exactly 5 half-planes.

and computational geometry (Matoušek, 2002, chapter 6). Proposition 6.1.1 from that book is illuminating. The faces of an arrangement are the minimal d-polytopes, polytopes that can't be made smaller by cutting it with yet another hyperplane from the arrangement.

Theorem 3.2.4. *The number of faces in a simple arrangement of n hyperplanes $\mathbb{R}^d$ (an arrangement where all hyperplanes are in general position) equals*

$$\Phi_d(n) = \sum_{i=1}^{d} \binom{n}{i}.$$

The kind of polytopes we are interested in result from the intersection of half-planes, and there will necessarily be fewer of them. It would be interesting to find an analogue of this theorem for coloured hyperplanes, though we would most likely have to be content with upper and lower bounds. An interesting corollary concerns resampling,

Corollary 3.2.5. *For any resampling procedure, the maximal amount of distinct possible solution sets is bounded above by $\sum_{i=1}^{d} \binom{n}{i}$.*

This is in acute contrast with smooth estimators. As an example, take the mean of $X_i \overset{i.i.d.}{\sim} F$, $i = 1, ..., n$, where F has a density. Then the non-parametric bootstrap has $\binom{2n-1}{n}$ distinct solutions, the subsampler has $\binom{n}{m}$ and m-out-of-n has $\binom{n+m-1}{m}$. This alone could give an indication of why the bootstrap tends to fail for estimators with cube root asymptotics.

For coloured hyperplanes and $d = 1$, the upper bound in Theorem 3.2.4 can be improved to$\lceil \frac{n}{2} \rceil$, which is sharp: Let the first element be red, the second blue, third red, etc. The lower bound on the number of solution sets is 1, which happens whenever the points are perfectly separated by a line (all red points to the left, all blue points to the right). This arrangement appears when the variance of ϵ_i is too small to push elements over to the wrong side.

Selecting a solution

It has been shown that the solution sets are unions of convex polytopes. Clearly, we need a reasonable procedure to select a single solution from these. One possibility is to select the *centroid* of the polytope with the smallest area. This is not possible when all the solution polytopes are unbounded, however. Another option is to select the vertex with smallest Euclidean norm, which is what we will do.

Another option is to use the *optimal separating hyperplanes* from support vector machines (Hastie et al., 2005, p. 132).

3.2.3 Location depth

Given unidimensional data $x_1, \ldots, x_n$, we define the n-th order statistic as the n-th element in the sorted list $(x_{i_1}, x_{i_2}, \ldots, x_{i_n})$. The rank $R(x_j)$ of a point x_j is its index in the preceding list. In 1975, Tukey considered a slight modification of this concept. He defined what is now called the location depth, in one dimension, as the minimum of $R(x_j)$ and $n - R(x_j) + 1$. Hence values far to the left and far to the right have small location depths, while those in the middle — deep in the data — have large location depths. Furthermore, it is clear that the median has depth $\approx \frac{n}{2}$, the quartiles have depth $\approx \frac{n}{4}$, etc.

Going further, this concept was generalised to dimensions bigger than one. Here the concept of ranks usually doesn't make sense, as the available orders, like the lexicographic ordering, don't correspond to anything of interest.

Definition 3.2.6. Let $A = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ for some $d > 0$, and let $y \in \mathbb{R}^d$. The *location depth of y relative to A* is the minimal $k \in \mathbb{N}$ such that there exists closed half-plane H_p, with y at its boundary, satisfying $H_p \cap A = k$.

it is easy to see that this is a generalisation of the unidimensional depth described in the first paragraph. An equivalent definition of the location depth, used in e.g. Donoho and Gasko (1992), is

$$D(y; A) = \min_{||u||=1} \#\{u^T x_i \geq u^T y\}.$$

Here $u^T x$ is the projection on the direction u. These definitions are equivalent, as $\{x \mid u^T x \geq u^T y\}$ defines a half-plane with y on the boundary. While Tukey was motivated by picturing bidimensional data through drawing of location depth contour lines, Donoho and Gasko (ibid.) were interested in the robustness properties of the *Tukey median*. It is defined as the point in A which has the highest depth, hence it is a natural generalisation of the median to higher dimensions. They showed that its breakdown point is bounded below by $\frac{1}{1+d}$, where d is the dimension of the underlying space. The concept of location depth has other applications as well, such as in the construction of bootstrap regions (Yeh and Singh, 1997). It can be used to generalise the empirical cumulative distribution function, and also in to generalise L-statistics: For instance, the analogue of the *midhinge* (mean of the quartiles) is the mean of all points with depth $\frac{n}{4}$; the α-trimmed mean is the mean of all points with the $(1-\alpha)\%$ largest depths.

Now we show that computing the location depth can be reduced to MAX-FLS. Let $x_i \in A \subset \mathbb{R}^d$ be data points and consider finding $D(x_j; A)$, where $x_j \in A$. Take note of that $\min_{||u||=1} \#\{u^T x_i \geq u^T y\} = \max_{||u||=1} \#\{u^T(x_i - y) < 0\}$. For any sequence of $i = i_1, ..., i_k$, $u^T(x_i - y) < 0$ for every such i if and only if the linear inequality system $Zu < \mathbf{0}$ is feasible, where $Z_{ij} = x_{ij} - y_j$ and $\mathbf{0}$ is a k-ary vector of zeros. Hence it is an instance of MAX-FLS with strict inequalities. One difference between this problem and Manski's estimator is the focus: In the Manski setting, we are interested primarily in the solution set $\{u \mid \#\{u^T x_i < u^T y\} = \max_{||u||=1} \#\{u^T x_i < u^T y\}\}$, while this is not of interest in the location depth setting. This has some consequences when it comes to algorithms. Since we are interested in knowing about qualitatively different solutions u for Manski's estimator, but not for the location depth, we need not be as considerate in finding every possible solution in the latter case. This will potentially save time and space. Still, this could be part of an explanation why statistics and methods based on the location depth aren't more popular, as the NP-completeness of the underlying problem guarantees that everything runs slowly.

3.2.4 Deepest regression

In this section we discuss the deepest regression estimator of Rousseeuw and Hubert (1999*b*), which is strongly related to the location depth. This particular

estimator deserves to be included here, as it is very similar to Manski's estimator in several regards. In covariate dimension one, it has similar geometry and interpretation as Manski's estimator, and part of the algorithm for its computation, discussed in Section 4.4, is almost exactly the same. Furthermore, it works under similar conditions, most importantly $\text{med}(\epsilon \mid x) = 0$ with arbitrary heteroskedasticity. Luckily, this estimator is $\sqrt{n}$-consistent.

Let the data generating mechanism be $Y = \beta^T X + \epsilon$, where $\text{med}(\epsilon \mid x) = 0$, and let $(y_i, x_i) \in A$ be the data. For a regression line α, define the ith residual as $r_i = y_i - (\alpha^T x_i)$. Now we define *unfitness.* A regression line β is *unfit* to the data whenever there is a z such that every residual has the same sign on each side of z: $r_i < 0$ when $\alpha^T x_i < z$ and $r_i > 0$ when $\alpha^T x_i > z$, or, $r_i > 0$ when $\alpha^T x_i < z$ and $r_i < 0$ when $\alpha^T x_i > z$. Now define $\text{depth}(\alpha; A)$ as the minimal amount of points to be removed in order to make α unfit. This is the minimum number of points one would have to cross in order to make the hyperplane vertical, which "liberates" the line. Figure (3.2.2) shows two OLS lines with quite different depths compared to $|A|$. A regression line with large depth could be considered *balanced*, in the sense that it is not easy to tilt it far in any direction without crossing many observations. This concept gives rise to the method of deepest regression. A line α is a deepest regression estimator if it solves $\arg\max_\alpha \text{depth}(\alpha; A)$. The resulting estimator is robust, with an asymptotic breakdown point of $\frac{1}{3}$. Its downsides are computational difficulties, in particular for dimensions greater than 2, and an intractable limit distribution. Compared to LAD (least absolute deviations) regression, this method is less efficient, reaching 87% efficiency for the slope and 82% for the intercept whenever the data is binormally distributed, see van Aelst and Rousseeuw (2000). It has to my knowledge not been made any attempt at generalizing L-statistics with respect to this depth.

3.3 Asymptotics

Consistency

In 1985, Manski proved the strong consistency of his estimator under three assumptions. Since Manski's estimator isn't uniquely defined, it requires a more general form of strong consistency than usual. Let $\widehat{B_n}$ be the solution set. By strong consistency we mean the following,

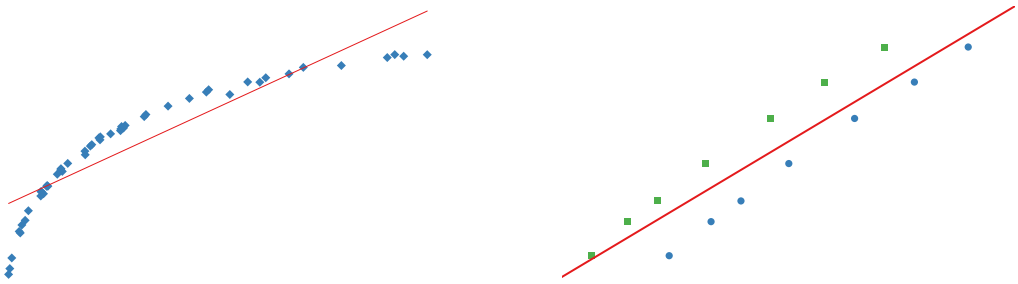

Figure 3.2.2: Examples of regression lines obtained by OLS regression with low and high depth, respectively. On the plot to the left, we would only have to remove five out of fifty points in order to "liberate" the regression line and make it unfit to the data. On the other hand, the line on right is almost perfectly fit: There are seven points both below and above that are well inter spaced, and there is no way to liberate the line without removing at least five (of fourteen) of them. Notice that, if the line had been placed above the left-most green point, and below the rightmost blue point, we would have had to remove seven points instead.

$$P(\lim_{n\to\infty} \sup_{\widehat{\beta}_n \in \widehat{B}_n} ||\widehat{\beta}_n - \beta|| = 0) = 1,$$

where the supremum is needed since $\widehat{B}_n$ is a set. Manski's (ibid.) assumptions are:

1. The support of F_x is not contained in any proper subspace of $\mathbb{R}^d$.

2. Both $Y = 1$ and $Y = 0$ is possible for any X: $0 < P(Y > 0 \mid X) < 1$ with probability 1.

3. There is a $k \in \{1, ..., d\}$ with $\beta_k \neq 0$ such that for almost any value of $(x_1, x_2, ..., x_{k-1}, x_{k+1}, ..., x_d)$, the distribution of X_k conditioned on x_{-k} has a positive density on $\mathbb{R}$.

In Section (3.6) it will be made clear that some form of assumption 3) is required. Assumption 1) is a full rank condition, necessary for identification. Assumption 2) is probably not required, and we will perform simulations where 2) is false in Section (3.6).

The limit distribution

The following limit distribution is of no practical interest. It is very time consuming to calculate it, and it is analytically intractable. The exception is with one covariate in the $\beta_0 = 1$ setting, where we yet again get a rescaled Chernoff's distribution.

It was Kim and Pollard (1990) who were the first to find the limit distribution of Manski's estimator. They found the distribution under the "spherical" identifiability condition $||\beta|| = 1$. Instead, we will supply the limiting distribution when $\beta_0 = 1$, ignoring the 10 conditions for it to work. This theorem is taken from Abrevaya and Huang (2005, theorem 4).

Theorem 3.3.1. *Let assumptions MS1-MS10 (in Abrevaya et al.) hold. Then*

$$n^{\frac{1}{3}}(\beta_n - \beta_0) \xrightarrow{d} \arg\max_h \left[\frac{1}{2}h^T V h + W(h)\right],$$

where W is a mean zero Gaussian process with covariance kernel

$$H(u,v) = E[\min(|X^T u|, |X^T v|)1_{[signX^T u = signX^T u]}g(-X^T\beta|X)],$$

where g is the density of X.

3.4 Algorithms and complexity

Following Florios and Skouras (2008), we wish to find exact solutions to maximisation problems of the form

$$\arg\max_{\beta\in B} \sum_{i=1}^{n} w_i(Y_i - \frac{1}{2})1_{[X_i^T\beta \geq 0]}, \tag{3.4.1}$$

where w_i are positive weights. It has already been said that this problem isn't identified — if $a > 0$ and $\widehat{\beta}$ is a solution, $a\widehat{\beta}$ is also a solution. In order to fix the scales, two approaches are used: 1.) If the intercept is believed to be positive, put $\beta_0 = 1$. Likewise, if the intercept is believed to be negative, put $\beta_0 = -1$. On the other hand, if we believe than $\beta_0 = 0$, the scale will not be fixed, and we will have to use a method like the next. 2.) Add the constraint $||\beta|| = 1$, where $||\cdot||$ is the Euclidean norm in $\mathbb{R}^d$. Clearly, these approaches are not equivalent. For the constraint $||\beta|| = 1$ allows for more solutions that $\beta_0 = 1$ does, while a solution set with the $||\beta|| = 1$ constraint (where $\beta_0 > 0$) easily can be transformed into a solution set for the $\beta_0 = 1$ condition by dividing the

solution by β_0. Also, if we solve the arg max for $\beta_0 = 0, 1, -1$, we can combine the solution sets separately into a solution set consistent with $||\beta|| = 1$.

3.4.1 Computational complexity

The computational complexity of Manski's estimator has not been considered in the statistical literature before, but it is known when formulated as MAX-FLS. The general problem is *NP*-complete, which is quite easy to prove.

Definition 3.4.1 (Complexity classes)**.** A decision problem is a function with boolean output. P is the class of all decision problems computable in polynomial time. NP is the class of all decision problems that are polynomially verifiable. That is, given a solution certificate which is polynomial in the size of the input, it takes polynomial time to verify it. A polynomial time reduction between decision problems A and B is a polynomial time program p which translates an instance of A into an instance of B. A problem A is NP-hard if any $B \in NP$ can be polynomial time reduced to A. A problem is NP-complete if it is in NP and is NP-hard.

These definitions are not completely spelled out. For more information about computational complexity theory, see e.g. Papadimitriou (2003). The main point of discussing NP-completeness is roughly that NP-complete problems can't be solved fast unless $P = NP$. It is generally assumed that $P \neq NP$, but proving it is considered among the greatest unsolved problems of mathematics. As can be seen from the definition, the theory doesn't apply to optimisation problems, but only to decision problems. When given an maximisation problem with objective function M, its decision problem variant is "is there an x such that $M(x) \geq y$?".

Now we will prove that the decision problem variant of MAX-FLS is *NP*-complete. This has been proved in Amaldi and Kann (1995) by reducing the NP-complete problem EXACT 3-COVER. We use a different approach, namely a reduction of MAX-2-SAT to MAX-FLS. We need some definitions. Let ϕ be a propositional formula with variables $x_1, ..., x_m$. We say that ϕ is in *conjunctive normal form (CFN)*, if it has the form $\phi_1 \wedge \phi_2 \wedge \ldots \wedge \phi_n$ for some *clauses* $\phi_i = a_{i1} \vee a_{i2} \vee ... \vee a_{im}$, where a_{ij} are *literals*. A propositional formula a is a literal if it has the form $a = \neg x$ or $a = x$ for a variable x. It is known that every propositional formula can be written in conjunctive normal form Mendelson (2009, p. 22). We say that a propositional formula is *satisfiable* if there exists an assignment of the variables $x_1, \ldots, x_k$ that makes it true, e.g. $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_2)$ is satisfiable:

It is true when both $x_1$and x_2 are true. An unsatisfiable formula is a contradiction — it is always false. A formula ϕ is in *2-CNF* if every clause contains at most two literals.

Definition 3.4.2. (MAX 2-SAT). Let ϕ be a formula in 2-conjunctive normal form (2-CNF). MAX 2-SAT is the problem of determining the maximal number of clauses which can be simultaneously satisfied. The corresponding decision problem is: Given a $k \in \mathbb{N}$, is there a collection of k clauses that are simultaneously satisfiable?

It is known that MAX 2-SAT(k) is NP-complete Garey et al. (1976). This can be proved through a reduction of 3-SAT, the problem of establishing whether a 3-CNF formula is satisfiable, which i*s* one of the core NP-complete problems Papadimitriou (2003, chap. 9).

Proposition 3.4.3. *The decision problem MAX-FLS is NP-complete.*

Proof. We show that MAX 2-SAT can be polynomial-time reduced to MAX-FLS. Let m be the number of variables in ϕ and n be its number of clauses. Also, define a_{ij} as the "sign" of the j-th variable in the i-th clause: For example, if the i-th clause is $(x_5 \wedge \neg x_9)$, then $a_{i5} = 1$, $a_{i9} = -1$ and $a_{ij} = 0$ for all $j \neq 5, 9$. Let A be the matrix with a_{ij} as its (i, j)-th element and let x be an m-dimensional vector having only 0 and 1 as elements. Then there are k satisfiable clauses if and only if there is a subsystem A' with k rows such that $A'x \geq \mathbf{1}$, where $\mathbf{1}$ is the appropriately sized vector with only ones. Now add the constraints $x_i = 0$ and $x_i = 1$, each $n + 1$ times. (These can be implemented as two inequalities of the form $\leq$ and $\geq$). Then we effectively force the x-vector to be integral by adding an additional $4m(n+1)$ constraints. The resulting system is ready to be solved by a MAX-FLS(k) solver, and its solution is clearly the same as for MAX-SAT(k). Since the input size is polynomial in the input size of MAX 2-SAT, the problem is NP-hard. MAX-FLS(k) is in NP: Given a solution, it can be verified in polynomial time simply by checking each inequality. □

Notice that, while MAX-FLS is NP-hard, it is only so when the number of inequality constraints and variables are considered at the same time. Soon we will describe an algorithm which runs in $O(n^d \log(n))$ time, where n is the number of constraints and d is the number of variables. This translates into n observations and covariate dimension d in our setting, and since we think of the dimension d as something constant (or a parameter), the algorithm could well be understood as polynomial in d — which is considered "fast" in this context.

But outside this specific problem, the amount of variables (that is d) is not necessarily constant.

Due to its connection with linear programming, there has been devoted some resources to finding algorithms for this problem. Many of those are described in Chinneck (2007, chap. 7), and the most prominent is described in the next section.

3.4.2 Earlier work

Pinkse (1993) described the first exact algorithm for Manski's estimator. He proposed to iteratively check the function value on every corner of the candidate solution polytopes. There are $\binom{n}{d} \approx (d!)^{-1}n^d$ possible corners to check, and it takes around $O(d^3)$ time to invert a matrix. (This is done in order to obtain the coordinates of the corners.) In addition, finding the value of the objective function is $O(nd)$, giving an algorithm with time complexity $O(n^{d+1})$. Pinkse also discussed a variety of approximation algorithms. Manski and Thompson (1986) introduced the approximate "great circle search" algorithm, but this algorithm does not perform very well, see the computational results of Florios and Skouras (2008). Florios and Skouras (ibid.) were the first to propose a workable exact algorithm for the weighted Manski's objective 3.4.1 in the statistical literature, an algorithm based on mixed integer programming (see eg Conforti et al., 2014). A mixed integer program is a linear program where some of the variables can be constrained to be integers, and most natural combinatorial optimisation problems can be represented as an integer program. We present a slightly modified variant of their program, taken from Chinneck (2007, section 7.1.1):

$$\min \sum_{i=1}^{n} w_i z_i \qquad \text{such that}$$

$$\begin{aligned} -x_i\beta &\leq 1 + Mz_i, \text{ when } Y_i = 1, \\ x_i\beta &< -1 + Mz_i, \text{ when } Y_i = 0, \\ & z_i \in \{0, 1\}, \\ & \beta \in \mathbb{R}^d. \end{aligned}$$

The constraints run over every $i = 1, ..., n$, and M is a large constant. In order to see that this is the right formulation, take a subset $z_{i_1}, z_{i_2}, ..., z_{i_k}$ and

$\beta_1, \beta_2, ...$ which satisfies it. An equation of the form $-x_i\beta \leq 1 + Mz_i$ will have $z_i = 1$ iff $-x_i\beta \leq 1$ is unsatisfied; the same goes for $x_i\beta \leq -1 + Mz_i$. The objective $\min \sum_{i=1}^{n} w_i z_i$ assures that we include the minimal weighted number of unsatisfied constraints, yielding a solution to the maximal problem. The role of M is to loosen the constraint so much that they are always satisfied. This can be chosen to be $Bd \max_{i,j} x_{ij}$, where B is a reasonable upper bound on the absolute value of the β_is, while $\max_{i,j} x_{ij}$ is the maximal recorded covariate value in any index.

We implement this program `R` through the use of `Rglpk` (Theussl et al., 2015), an **R** interface to the **G**NU **l**inear **p**rogramming **k**it. The computation speed is slow, which is not surprising, as mixed integer programming in general is NP-hard. Smart algorithms for solving them have been well studied: The main algorithm is called "Branch-and-cut", which operates by sequentially relaxing the integer program into ordinary linear programs, which can be solved quickly through the simplex method. For a thorough introduction, see Conforti et al. (2014). In this thesis, our attention is restricted to $d = 1$ and $d = 2$, and it might be that the integer programming formulation is faster than the complete enumeration algorithm when d is sufficiently large.

An important fact is the following inapproximability result of Amaldi and Kann (1994, 1995), which warns us that it may be futile to search for a general purpose approximation algorithm for Manski's estimator. Let M be the real maximum and M' be an approximate maximum. We say that a maximisation problem is d-approximable if $\frac{M}{M'} \leq d$ for every instance of the problem.

Theorem 3.4.4. *MAX-FLS is 2-approximable, but not approximable within any factor unless $P = NP$. (That is, MAX-FLS is APX-complete with approximation factor bounded by 2 (Wegener, 2005, chapter 8)).*

In the next section we describe an $O(n^d \log n)$ enumeration algorithm for finding the entire solution set.

3.4.3 An enumeration algorithm

The algorithms of this section were invented and analysed by me. After the work on them was finished, it was discovered that an algorithm with runtime $O(n^d \log n)$ was already described in Johnson and Preparata (1977). Still, this section describes the first reasonably fast algorithm for the computation of Manski's algorithm in the *statistical* literature. First we will explore the case $d = 1$, which corresponds to an intercept and one covariate. The focus is on the

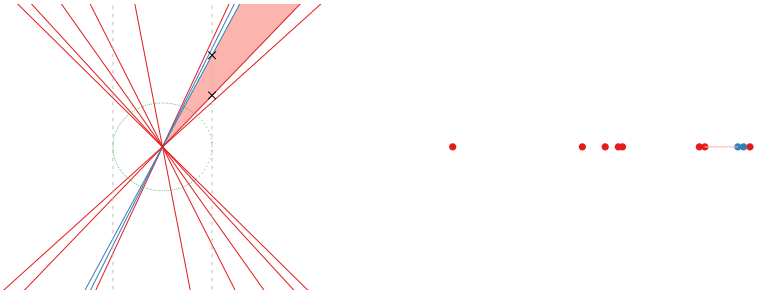

Figure 3.4.1: (left) Geometry of the problem in covariate dimension one. The blue lines signify half-planes pointing downwards, while the red lines point upward. In this example, all half-planes have weights equal to one, and the unique solution set is the pink area, which is the intersection of nine half-planes. The only half-plane not included in the intersection is the red half-plane with the steepest slope. (right) Points on the line corresponding to the rays left of $\beta_0 = 0$. The purple segment is the solution set intersected with the line $\beta_0 = 1$.

identification of the whole solution set, unscaled. This is done by identifying solutions corresponding to $\beta = -1, 0, 1$, rescale them to the unit circle, and expand them along rays. This effectively reduces the problem to a one-dimensional one.

As already mentioned, we want to find the w-maximal non-empty intersections between half-planes H_i. Such a set is always the area between two rays on the same side of $x = 0$.

Figure (3.4.1) illustrates the setting, and points to an important fact: When we have found a solution set for $\beta = 1$, it can be expanded to the whole solution set, by simply identifying the lines that give rise to the solutions. Hence we can reduce the problem to finding w-maximal non-empty intersections of intervals, first to the left of $\beta_0 = 0$, then to the right of $\beta_0 = 0$. In the following we impose the constraint $\beta_0 = 1$.

Definition 3.4.5. Let (y_i, x_i), $i = 1, ..., n$ be observations from a binary response model with $d = 1$. The colour of a point $-\frac{1}{x_i}$ is defined by

$$\text{col}(-\frac{1}{x_i}) = \begin{cases} \text{red} & \text{when } (x_i > 0 \text{ and } y_i = 1) \text{ or } (x_i < 0 \text{ and } y_i = 0), \\ \text{blue} & \text{when } (x_i < 0 \text{ and } y_i = 1) \text{ or } (x_i > 0 \text{ and } y_i = 0). \end{cases}$$

We make use of the function col any more, but simple call points red or blue

depending on which of the two conditions is satisfied. As for the motivation for this colouring, recall the algebraic characterisation of Manski's objective in Proposition 3.2.2. When formulated for $d = 1$, we get

$$\begin{aligned} -\beta_1 x_i \leq 1 \quad &\text{when} \quad Y_i = 1, \\ \beta_1 x_i < -1 \quad &\text{when} \quad Y_i = 0. \end{aligned} \tag{3.4.2}$$

If $x_i > 0$, the these inequalities are equivalent to

$$\begin{aligned} \beta_1 \geq -\frac{1}{x_i} \quad &\text{when} \quad Y_i = 1, \\ \beta_1 < -\frac{1}{x_i} \quad &\text{when} \quad Y_i = 0, \end{aligned}$$

while $x_i < 0$ gives

$$\begin{aligned} \beta_1 \leq -\frac{1}{x_i} \quad &\text{when} \quad Y_i = 1, \\ \beta_1 > -\frac{1}{x_i} \quad &\text{when} \quad Y_i = 0. \end{aligned}$$

Accordingly a point is $-\frac{1}{x_i}$ is blue when its corresponding equation is satisfied for every $\beta_1 < -\frac{1}{x_i}$, and similarly for red points. In the following proposition, we denote Manski's objective (3.1.2)

$$m(z) = \sum_{i=1}^{n} (Y_i - \frac{1}{2}) 1_{1+zx_i \geq 0}.$$

Proposition 3.4.6. *Denote $z_i = -\frac{1}{x_i}$, let $z \in \mathbb{R}$ and define*

$$\begin{aligned} r_l(z) &= \textit{number of red points to the left of } z, \\ b_r(z) &= \textit{number of blue points to the right of } z, \end{aligned}$$

both quantities being inclusive. Then $m(z)$ equals $r_l(z) + b_r(z)$.

Proof. The objective function $m(z)$ counts the number of satisfied inequalities in (3.4.2). By the reasoning above, the inequality belonging to a blue point z_i is satisfied by z iff z_i is to the right of z. Similarly, an inequality belonging to a red point z_j is satisfied iff z_j is to the left z. The result follows. □

We can use this proposition to identify the solution sets. Recall from the

geometric characterisation (Lemma 3.2.3) that the solution sets are maximal disjoint convex polytopes, which in $\mathbb{R}$ means disjoint intervals. If $[z_i, z_j]$ is a maximal bounded solution interval, z_i must be red and z_j blue. For if z_i is blue, either there is red point z_l to its left with $m(z_l) = m(z_i)$, or there is no point at all, which implies $m(z) = m(z_i)$ for every $z < z_i$. This contradicts the maximality of $[z_i, z_j]$.

Proposition 3.4.7. *Assume the list $z_i = -\frac{1}{x_i}$ is sorted, and also add two non-coloured points $z_0 = -\infty$ and $z_{n+1} = \infty$. A solution interval is of the form $[z_i, z_{i+1}]$, where z_i is either red or $-\infty$ and z_{i+1} is either blue or ∞.*

Proof. it is impossible to have three adjacent z_i, z_{i+1}, z_{i+2} with $m(z_i) = m(z_{i+1}) = m(z_{i+2})$: For if z_{i+1} is blue, $m(z_i) > m(z_{i+2})$, and if z_{i+1} is red, $m(z_i) < m(z_{i+2})$. □

What happens if $\beta_0 = -1$ instead? Then all the points swap colours and signs, and the same argument applies. Now we're ready to describe an algorithm for computing the estimator. Essentially we sort the elements $z_i = -\frac{1}{x_i}$, find their colour, and sequentially calculate $r_l(z_i) + b_r(z_i)$, starting from the left. Using this algorithm, we only have to visit each point twice, because

$$\begin{aligned} r_l(z_{i+1}) &= r_l(z_i) + 1_{col(z_{i+1})=\text{red}}, \\ b_r(z_{i+1}) &= b_r(z_i) - 1_{col(z_i)=\text{blue}}, \end{aligned}$$

and both $r_l(z_1)$ and $b_l(z_1)$ can be calculated in by visiting every point one time. The output is the list of values $r_l(z_i)$ and $b_l(z_i)$ for every z_i, and the solution sets can then be identified by Proposition 3.4.7.

Algorithm 3.1 RedBlue

Input: a list of colored z_is
Output: $r_l(z_i)$ and $b_r(z_i)$ for each z_is

```
1: procedure REDBLUE(points)
2:     sort the points by value
3:     Calculate r_l(z_1)
4:     for i = 1; i ≤ n; i++ do
5:         r_l(z_{i+1}) = r_l(z_i) + 1_{col(z_{i+1})=red}
6:         b_r(z_{i+1}) = b_r(z_i) − 1_{col(z_i)=blue}
7:     end for
8: end procedure
```

Proposition 3.4.8. *The procedure RedBlue in Algorithm 1 is correct. Its asymptotic runtime is* $\Theta(n \log n)$.[1].

Proof. As the algorithm is correct by the preceding discussion, we will calculate the runtime. As for the runtime, it is well known that sorting is $O(n \log n)$. The amount of arithmetic operations done is $O(n)$ for the first element and 4 for each of the other elements. Therefore the algorithm is $O(n \log n)+O(n) = O(n \log n)$. As for the lower bound, it is also known that comparison-based sort has the asymptotic lower bound $\Omega(n \log n)$ (Dasgupta et al., 2006, p. 58). Knowing the solution to this problem one can clearly find the sorted list in $O(n \log n)$ operations, by sorting on the $r_l(z_i)$, and then the $b_r(z_i)$. From this we arrive at the lower bound. □

Clearly, the algorithm can easily be extended to work for the weighted m as well. If we maximise for both $\beta_0 = 1$ and $\beta_0 = -1$ simultaneously, this algorithm is the same as the algorithm *rdepth* for the calculation of the regression depth in Rousseeuw and Hubert (1999*b*), except they are minimising instead of maximising. To maximise for $\beta_0 = -1$, we'd substitute $r_l(z)$ for $r_r(z)$ and $b_r(z)$ for $b_l(z)$, where r_r and b_l have the obvious meanings. Note that if the elements are pre-sorted, this is an $O(n)$-algorithm.

This algorithm has been implemented in `C++`, with a link to `R` through `Rcpp` (Eddelbuettel and François, 2011) and runs very fast, comparable to `sort` in `R`. A comparison is provided in Figure (3.4.2).

If we are only interested in finding one solution, not the entire solution set, it might be possible to improve the runtime of this algorithm by using an approach similar to that of the Quickselect algorithm: The reason why our algorithm isn't linear time is because we presort the entire list, something which is not necessary when we only need one maximum. By using the concept of colour depth from Section (3.5), one can attempt to create a sophisticated branch-and-bound algorithm in order to prune the computation tree of QuickSort. It is unknown whether such an idea would work out or not.

Two dimensions

An approach to solving this problem is to iteratively apply the RedBlue algorithm: For each half-plane H_i with boundary L_i, we can calculate the value of $m(\beta_1, \beta_2)$ (Manski's objective, (3.1.2)) for every $\beta_1, \beta_2 \in L_i$ by using the RedBlue algorithm. This is done by calculating the intersection point $z_j = L_i \cap L_j$

[1]This means that the algorithm is both $O(n \log n)$ and $\Omega(n \log n)$. That is, it is asymptotically bounded both above and below by $n \log n$.

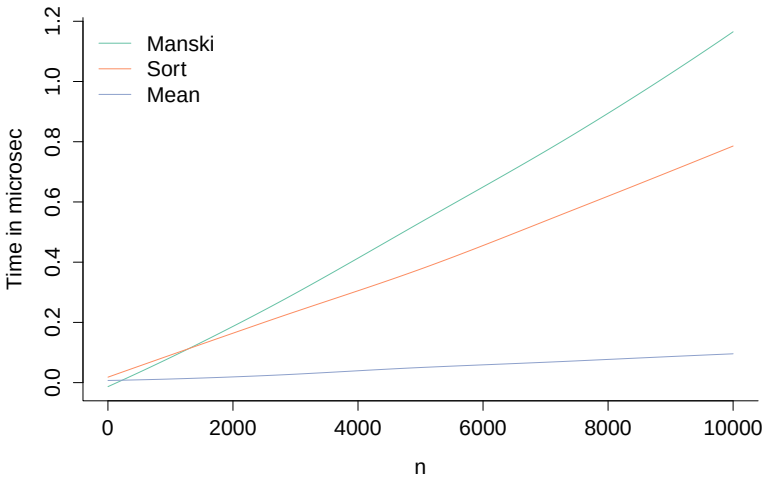

Figure 3.4.2: The speed of some `R` procedures, in microseconds. The speeds are measured by `microbenchmark` Mersmann (2014), on a Windows 10, Intel Core i7-4790 3.60 GHz, `R` version "Pumpkin Helmet". Curiously, RedBlue is faster than sorting for small ns, something which can only be explained by a bad implementation of `sort` in `R`. The performance of `mean` is plotted for comparisons sake. The timings were obtained through the package `microbenchmark` and the smoothing is done by the package `locfit`.

for every $j \neq i$, and decide on whether z_i is blue or red. We walk along L_i from right to left. When we encounter another line L_j, we observe whether the half-plane H_i covers the ground we've already passed or not, if it is covered, z_i is coloured red; if it doesn't, it is coloured blue. Be aware that "H_i covers the ground we've already passed" is *not* the same as "H_i is red", where the colour of hyperplanes are defined as in Section (3.2.2). It depends on the relation between the slopes of L_i and L_j in addition to the colour, as illustrated in the figure. Let s_i be the slope of L_i and s_j be the slope of L_j. If $s_j < s_i$, let z_j be coloured oppositely of H_i, and if $s_i < s_j$, keep its colour. How this works is illustrated in Figure (3.4.3).

If we do this for every line L_i, we get a table of $m(z_j^i)$ for $j \neq i$. The maxima of all these will come in pairs corresponding to intervals on some L_i, hence they will form line segments. These line segments will, by the geometric characterisation, form convex polygons in the plane, and the solution set is the union of these polygons. Running through the procedure, we visit n lines and check $n-1$ intersections for each of them. Since RedBlue runs in $O(n \log n)$ time, the runtime of this two dimensional algorithm is $O(n^2 \log n)$. It has been implemented in `C++`, with a link to `R` through `Rcpp`, just like the RedBlue algorithm. The code is in Appendix (B).

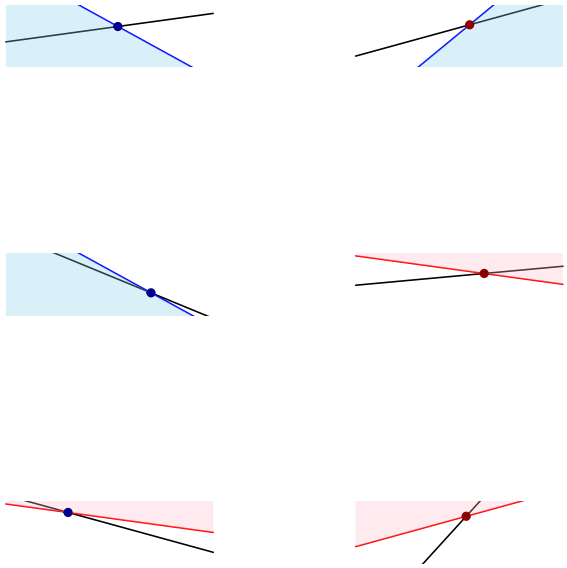

Figure 3.4.3: Different angles between L_i(black) and L_j (coloured). The colour of the point is used in the RedBlue algorithm.

Higher dimensions

We can recursively apply the same idea in covariate dimensions $d > 2$. Assuming we can solve MAX-FLS in $d-1$ dimensions in $n^{d-1}\log n$ time, we can iterate through all hyperplanes $1+X_i^T\beta$ in order to find the optimal $(d-1)$-dimensional facets there. As an example, consider $d = 3$. In this case, the solution sets are polyhedra. In order to find the w-maximal polyhedra, we identify the w-maximal faces. For a single plane $1 + X_i^T\beta$, we cam find its optimal faces by running the 2-dimensional algorithm on the arrangement arising from $1 + X_i^T\beta$ intersected with $\left\{1 + X_j^T\beta\right\}_{j\neq i}$. If we do this for each i, we will uncover the maximal faces, from which we can construct the maximal polytopes. This procedure is $O(n^3 \log n)$, as it runs through n different 2-dimensional algorithms with run-time $O(n^2 \log n)$. This can clearly be generalised to arbitrary dimensions, yielding the general run time $O(n^d \log n)$. Alas, we have not found time to implement the algorithm for $d > 2$.

3.5 Robustness

The robustness of Manski's estimator has not been studied before. As has been alluded to in Section 3.2.4, the geometry of Manski's estimator is quite similar to that of the deepest regression estimator (Rousseeuw and Hubert, 1999*b*). We will borrow, and change slightly, concepts from the literature on that estimator. A particularly important concept is the generalisation of the concept of *undirected depth* to our "coloured" domain.

Why study robustness? The reason why robust estimators are important is that basic, convenient assumptions might not hold. Typically, robustness is studied in the parametric context, with the archetypical example being the estimation of the expectation μ when we *think* that $X_1, ..., X_n \overset{i.i.d.}{\sim} N(\mu, 1)$. It is well known that the sample mean is the maximum likelihood estimator, which is efficient, easy to compute, easy to interpret, and incidentally very easy to find the distribution of. Despite these qualities, it is very sensitive to *single outliers* in the following sense: Given n "good" observations $X_1, ..., X_n$ from $N(\mu, 1)$, we can add another X_{n+1} and set its value to x. Denote $\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ and $\overline{X}' = \frac{1}{n+1}\sum_{i=1}^{n} X_i + \frac{1}{n+1}X_{n+1}$. Clearly, $\frac{1}{n+1}\sum_{i=1}^{n} X_i \approx \frac{1}{n}\sum_{i=1}^{n} X_i$, but by sending x off to infinity, we can make the "bad" mean arbitrarily large by adding just one outlier. Generally, if T is a statistical functional, this kind of analysis can be formalized, asymptotically, by the concept of *influence curves.* Also, in the finite sample domain, we have the far less popular concept of sensitivity curve.

Definition 3.5.1. (Maronna et al., 2006, section 3.1)Let T be a statistical functional, F be a distribution and δ_x be the unit mass on x. The influence curve of T at x is $IF(x; F) = \lim_{\epsilon \searrow 0} \frac{T((1-\epsilon)F + \epsilon\delta_x) - T(F)}{\epsilon}$. If F_n is the empirical cdf obtained from n observations, the sensitivity curve is $SF_n(x; F_n) = (n+1)T(\frac{n}{n+1}F_n + \frac{1}{n+1}\delta_x) - T(F_n)$.

The influence curve tells us about the behaviour of T when F is infinitesimally contaminated by δ_x. The definition looks strikingly much like that of the ordinary derivative, and it can often be used as such in Taylor expansions of statistical functionals, see e.g. van der Vaart (2000, chapter 20). We say that an estimator is robust against single outliers if it has a bounded influence curve. Examples of estimators with bounded influence curve for μ is the α-trimmed mean and the median. We will not discuss the influence curve more. For more information about the influence curve and related concepts, see any book on robust statistics, for instance Huber (1981) and Maronna et al. (2006).

The influence curve does not quantify resistance to multiple outliers. There

are several measures that do this, but we will focus on the breakdown point, introduced in Donoho and Huber (1983).

Definition 3.5.2 (Breakdown point)**.** Let T be a statistical functional and F be a distribution. The (asymptotic) *breakdown point* is defined by $\epsilon^*(T,F) = \inf\{\epsilon \mid \sup_G ||T((1-\epsilon)F + \epsilon G) - T(F)|| = \infty\}$.

The breakdown point also has a finite sample analogue, the finite sample breakdown point $\epsilon_f^*(T, F_n)$. This can be defined in two almost equal ways: Either the minimal number of observations which has to be *replaced* in order to make the estimator diverge, or the number of observation which has to be *added* in order to make the estimator diverge. Following Rousseeuw and Leroy (2005), we use the first definition. Their justification is that this is how most outliers actually arise — originally good observations are contaminated by some source, for instance clerical errors in the entry of data. Another, more practical reason is that this definition is easier to work with in this particular problem.

This concept of finite sample breakdown point is, non-shockingly, often sample dependent, though this is not always the case. For instance the least median of squares estimator has a finite sample breakdown point of $\frac{\lceil n/2 \rceil + d - 1}{n}$ independent of the sample (Rousseeuw and Leroy, 2005). The breakdown point of Manski's estimator is extremely sample dependent, as we will see soon. Everything in the two next section is our original work.

3.5.1 Breakdown in one dimension

Recall that the setting of Manski's estimator in one dimension is a line with several blue and red points. If our observations are $x_1, ...x_n$ and $y_1, ..., y_n$, the red points are the points $-\frac{1}{x_i}$ where either both $y_i = 1$ and $x_i > 0$ or both $y_i = 0$ and $x_i < 0$. The rest of the points are blue. Given such a set of coloured points on the line, we can find the *colour depth* of a point.

Definition 3.5.3 (Colour depth)**.** Let $z \in \mathbb{R}$. Define $r_l(z), r_r(z), b_l(z), b_r(z)$ as the amount of red points to the left of z, red points to the right of z, blue points to the left of z and blue points to the right of z, respectively. All of this is inclusive. We define the colour depth of z, denoted $\mathfrak{C}(z) = \min\{r_l(z) - b_l(z), b_r(z) - r_r(z)\}$. Also let $\mathfrak{M} = \max_{z\in\mathbb{R}} \mathfrak{C}(z)$ be the *maximal colour depth*. If $\mathfrak{C}(z) = \mathfrak{M}$, z is a *witness* to the maximal colour depth.

The following lemma indicates why this is an important concept: To maximise the colour depth is equivalent to the maximisation of Manski's objective function (3.1.2), which we denote m.

Lemma 3.5.4. *A point $z \in \mathbb{R}$ is a solution to Manski's objective if and only if it is of maximal colour depth.*

Proof. Assume z is of non-maximal colour depth, and assume wlog there is a point z' of greater colour depth to its right. Now count of blue (b) and red (r) points between z and z'. We have three cases: If $r > b$, then $m(z) = b_r(z) + r_l(z) = b + b_r(z') + r_l(z) < r + b_r(z') + r_l(z) = m(z')$. If $r = b$, both $r_l(z') - b_l(z') = r_l(z) - b_l(z)$ and $b_r(z') - r_r(z') = b_r(z) - r_r(z)$ and z' hasn't greater colour depth than z. Lastly, if $r < b$, then $r_l(z') - b_l(z') < r_l(z) - b_l(z)$ and $b_r(z') - r_r(z') < b_r(z) - r_r(z)$, again contradicting that z' has greater colour depth. From this both claims follows: The first implication ($\Rightarrow$) is proved contrapositively, while the second ($\Leftarrow$) by assuming another point is maximal and reaching a contradiction. □

In order to make the estimator break down, we will have to add and remove strategically chosen lines such that 1.) every point z of maximal colour depth loses that status and 2.) there is a new set of points with maximal colour depth placed arbitrarily far out.

Theorem 3.5.5. *The finite sample breakdown point of Manski's estimator with $d = 1$ is $\lceil \frac{\mathfrak{M}}{2} \rceil / n$.*

Proof. We begin with $\epsilon^* \leq \lceil \frac{\mathfrak{M}}{2} \rceil / n$. Let z have maximal colour depth, and assume wlog the minimal weight is on its right. In that case, there are $\mathfrak{M}$ surplus blues on its right. Now change the colour of these $\lceil \frac{\mathfrak{M}}{2} \rceil$ points, starting from the rightmost one. Let z' be the rightmost of the red points, with $m(z') = r_l(z) + r_r(z) + \lceil \frac{\mathfrak{M}}{2} \rceil$. Also, $m(z) = r_l(z) + b_r(z) - \lceil \frac{\mathfrak{M}}{2} \rceil$. But since $r_r(z) + \frac{\mathfrak{M}}{2} = b_r(z) - \frac{\mathfrak{M}}{2}$ by definition, the rightmost point is a solution and we have attained a breakdown.

Now assume $\epsilon^* < \lceil \frac{\mathfrak{M}}{2} \rceil / n$, and let the points be reconfigured such that a breakdown has been attained in less than $\lceil \frac{\mathfrak{M}}{2} \rceil$ steps. Without loss of generality, assume that the rightmost element of the new configuration is red. Since this point is a solution, it has maximal colour depth. Furthermore, the maximal colour depth is equal to zero, as there are 0 points to the right of it. Let z be any other point, which automatically satisfies $\mathfrak{C}(z) \leq 0$. If z is a witness to the the maximal colour depth of the original configuration, we must be able to recover this number by replacing $\lceil \frac{\mathfrak{M}}{2} \rceil$ points. Since an element can increase its maximal colour depth with at most two for each rearrangement of a single point, we have $\mathfrak{C}(z) < 2\lceil \frac{\mathfrak{M}}{2} \rceil$. If $\mathfrak{M}$ is even, this is contrary to hypothesis. If $\mathfrak{M}$ is odd,

notice that since $\lfloor \frac{\mathfrak{M}}{2} \rfloor = \lceil \frac{\mathfrak{M}}{2} \rceil + 1$, we would have to be able to recover $\mathfrak{C}(z)$ in at least $2\lfloor \frac{\mathfrak{M}}{2} \rfloor$. But this gives $\mathfrak{C}(z) \leq 2\lfloor \frac{\mathfrak{M}}{2} \rfloor < \mathfrak{M}$. □

A special case occurs when $\mathfrak{M} = 0$, which gives a breakdown point of 0. This is not an error. When the $\mathfrak{M}$ is 0, the solution set is unbounded either to the right or to the left, hence it has already broke down. To see why, assume we have a configuration which has $\mathfrak{M}$ =0. Choice a maximal point z and assume its minimal colour depth occurs to its right. Clearly, there are equally many, say k, blues and reds on that side. If the rightmost point is red, the solution set is unbounded, so assume it is not. If the rightmost point is blue, choose the red point closest to it, named z'. Then $m(z') \geq r_l(z) + k + 1$, as it covers all the red points z covers, all the red points to its right and at least one blue point. But this contradicts the maximality of z, as $m(z) = r_l(z) + k$. This happens with non-zero probability in familiar settings when $n = 100$, as we can see in Figure 3.5.2 on page 66.

Remark 3.5.6. Since solutions aren't unique, it is not entirely clear how to define the breakdown point. Two different choices can be made. We can think of the estimator as broken down whenever there is a solution with unbounded absolute value, which is our choice here. On the other hand, we could also think of the estimator as broken when there are no "small" solutions at all: If for all $n \in \mathbb{N}$ there are some points such that the absolute value of the new configuration's minimal solution is greater than n. In order to make the second definition work, the breakdown point would be $(\lfloor \frac{\mathfrak{M}}{2} \rfloor + 1)/n$ instead. We will use the first definition as it makes proofs slightly easier: When replacing points, we will only have to think about switching colours, not changing coordinates.

From Lemma 3.5.5 we get the following rough upper bound on the asymptotic breakdown point.

Corollary 3.5.7. *The breakdown point of Manski's estimator with $d = 1$ is bounded above by $1/4$.*

Proof. The best possible value of $\mathfrak{M}$ is $n/2$, which is attained iff the points are perfectly separated into a set of red elements only on the left and a set blue elements on the right. Pass n to the limit in order to get the result. □

Given covariates $X_i \sim G$ and an error distribution F, we can calculate the asymptotic analogues of $r_l(\widehat{\beta}), b_r(\widehat{\beta})$ etc. through substituting β for $\widehat{\beta}$ and then calculating their exact distributions. If we denote the exact distributions $r_l^\infty(\beta), b_r^\infty(\beta)$ etc., we see that

$$
\begin{aligned}
r_l^\infty(\beta) &= P(Y=0, X<0, -\frac{1}{X} < \beta) + P(Y=1, X \geq 0, -\frac{1}{X} < \beta), \\
b_l^\infty(\beta) &= P(Y=1, X<0, -\frac{1}{X} < \beta) + P(Y=0, X \geq 0, -\frac{1}{X} < \beta), \\
b_r^\infty(\beta) &= P(Y=1, X<0, -\frac{1}{X} > \beta) + P(Y=0, X \geq 0, -\frac{1}{X} > \beta), \\
r_r^\infty(\beta) &= P(Y=0, X<0, -\frac{1}{X} > \beta) + P(Y=1, X \geq 0, -\frac{1}{X} > \beta).
\end{aligned}
$$

These equations follow directly from the definitions of blue and red points. As corollary of this observation and the preceding proposition, we get

Corollary 3.5.8. *The breakdown point of Manski's estimator ($d = 1$) is*

$$\mathcal{B} = \frac{1}{2} \min \left(r_l^\infty(\beta) - b_l^\infty(\beta), b_r^\infty(\beta) - r_r^\infty(\beta) \right)$$

Now we provide some illustrations of these results. Using numerical integration, it is not hard to calculate the asymptotic breakdown points, which we do for a handful of covariate and error distributions in Figure 3.5.1 on the next page. In Figure 3.5.2 on page 66 we show simulated maximal colour depths for the case of standard logistic errors and normal covariates, a case close to ordinary logistic regression.

Contrary to most estimators for linear regression, this estimator has the bad properties of being both sample dependent (in the finite sample case) and dependent on the data generating mechanism.

3.5.2 Breakdown in several dimensions

We extend to concept of colour depth to $d > 1$. The challenge is to find the right generalisation, as there is no suitable total order in higher dimensions. We rectify this by using projections. In the following definition, $\mathcal{H}$ is the set of half-planes induced by the data set $(y_1, x_1), (y_2, x_2), ..., (y_n, x_n)$.

Definition 3.5.9. Let l be a line, and define the colour depth of a line l as the maximal colour depth of the configuration induced by $\{H \cap l \mid H \in \mathcal{H}\}$. Let $p \in \mathbb{R}^d$, and define L_p to be the set of all directed lines passing through p. Then $\mathfrak{C}(p)$ is $\min_{l \in L_p} \{\mathfrak{C}(l)\}$; the minimal colour depth (at p) of all lines passing through p. The maximal directed colour depth is $\mathfrak{M} = \max_{p \in \mathbb{R}^d} \mathfrak{C}(p)$.

This is the "coloured" variant of the notion of *undirected depth*, used in the analysis of the deepest regression estimator Rousseeuw and Hubert (1999*a*).

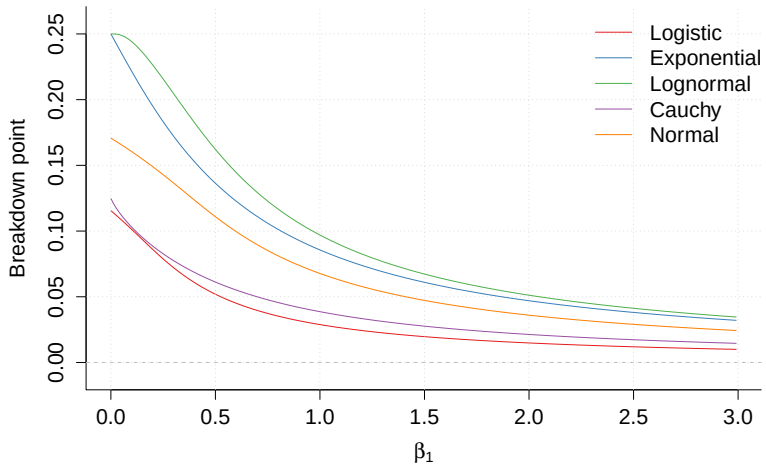

Figure 3.5.1: Breakdown point plot for several different error distributions. The exponential and log normal errors have been normalized to have median equal to 0. (Subtract log(2) and 1 respectively). The corresponding covariate distribution are logistic, normal, normal, Cauchy and normal, respectively. When the errors are exponential, the breakdown point is nearly $1/4$ when β is small, which indicates near-perfect separation. This comes as no surprise; the exponential distribution has a median of $\log(2)$, hence the error-term accounts is $1 - \log(2)$ at minimum, a value it is difficult to overcome for βX when β is small; this reasoning makes it clear that the breakdown point converges to 0.25 when $\beta \to 0$ for both the log normal and exponential errors. Similar considerations explain the other features of the plot, for instance that Cauchy/Cauchy has a smaller breakdown points than Normal/Normal.

The concept is deeper and more difficult to analyse. The following is a natural generalisation of Lemma 3.5.4.

Conjecture 3.5.10. *A point has p is of maximal colour depth iff it is a solution to Manski's objective.*

Given n points, which configuration of coloured planes would give the highest breakdown point? It appears likely that this is a *d-simplex* where each face has $\frac{n}{d+1}$ planes of the same colour stacked on the top of one another. A $d-$simplex is d-polytope with the smallest possible number of faces, namely $d+1$. Assume $n = (d+1)k$ for a k, and observe that the maximal colour depth of this configuration is k. Furthermore, its breakdown is easily seen to be $\frac{k}{2}$; one would need to replace half of the planes adjacent to one of the faces with planes of the opposite colour. This gives rise to the following conjecture, which generalises Corollary 3.5.7.

Conjecture 3.5.11. *The breakdown point of Manski's estimator in covariate dimension d is bounded above by $\frac{1}{2(1+d)}$.*

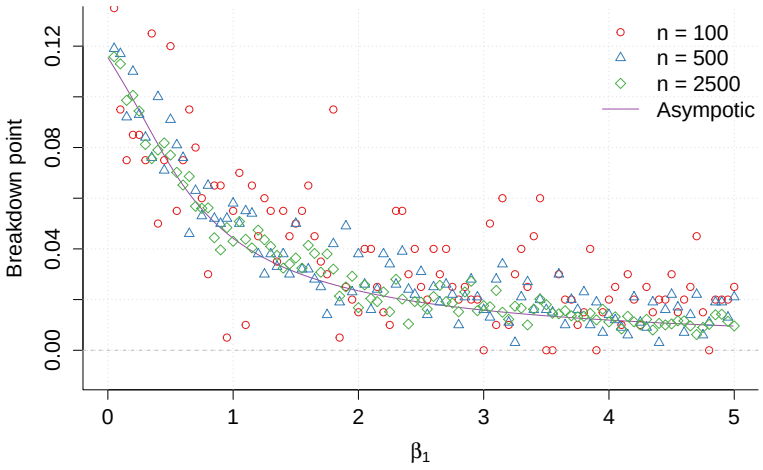

Figure 3.5.2: Simulated breakdown points together with asymptotic breakdown points for standard normally distributed covariates and logistic error distribution.

The generalisation of Theorem 3.5.5 is likely false. The reason is that a point of maximal colour depth can have "independent" witnessing lines, in the sense that swapping the colours of $\mathfrak{M}/2$ points on one witnessing line doesn't necessarily make every other witnessing line get its colour depth pushed down to 0. However, that the finite sample breakdown point is bounded *below* by $\mathfrak{M}/2$ is probably easy to show.

Conjecture 3.5.12. *The finite sample breakdown point of Manski's estimator in covariate dimension d is bounded below by $\mathfrak{M}/2$.*

The breakdown point will probably be very close to this in any case. It seems difficult to analyse how and when the phenomenon above occurs, and how bad it could be. We leave attempted proofs or disproofs of these conjectures to another time, but we're not entirely done with the topic of robustness, as empirical robustness studies are included in the next section.

3.6 Illustrations and simulations

3.6.1 The role of the covariates' distribution

Manski's estimator (in $d = 1$) has large problems estimating values of β_1 that are close to 0. Essentially, this is because the candidate intervals $< -\frac{1}{x_i}, -\frac{1}{x_j} >$ require some very large x_is in order to get close to 0. Assume $X_i \overset{i.i.d.}{\sim} N(0,1)$, $Y = 1_{[1+0\cdot X_i+\epsilon_i \geq 0]}$, and for simplicity that the optimal set is $< -\frac{1}{x_i}, -\frac{1}{x_j} >$,

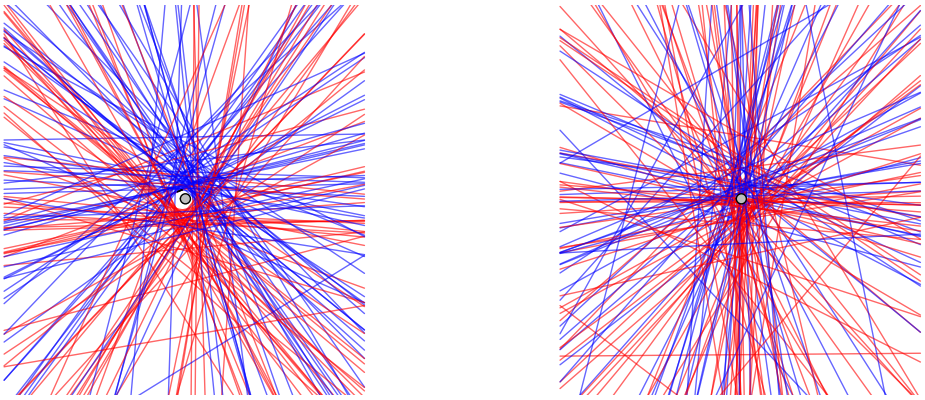

Figure 3.6.1: Line plot for the model $Y_i = 1_{[1+0.05X_{1i}+0.05X_{2i}+\epsilon_i]}$, where $\epsilon_i \sim N(0,1)$. The true betas are grey dots. (left) Standard normal covariates, which need very large n in order to make the gap small, (right) Cauchy distribution covariates, which have no problem closing the gap.

with $x_i = \max X_i$. Say we want an n such that the lower bound $0 > -\frac{1}{x_i} > -0.20$ holds with probability at least 0.5. This requires at least one observation greater than 5, and this happens with probability greater than 0.5 for $n \approx 2418000$. If the X_is are Cauchy, the lower bound is $n = 11$, an enormous difference.[2] Note that the estimator will, correctly, find an optimal set covering 0, but it will take an absurdly large n to get this interval small. This also shows that a form of assumption 2c (at the beginning of Section 3.3) is required for consistency: If the covariates were uniformly distributed on $[-a, a]$, for instance, the infimal modulus of a candidate solution is $\frac{1}{a}$, which will not be enough to assure consistency whenever $\beta_1 < \frac{1}{a}$.

As seen in Figure 3.6.1, there are similar problems in covariate dimensions greater than 1. For a line in the dual plane to come close to a $\beta \sim (0,0)$, it will require an intercept close to 0, which can happen only if x_2 is very large. This reasoning clearly extends to arbitrary dimensions.

[2]For both these observations we use that the distribution of the maximum is $F(x)^n$, hence $P(\max X_i > a) = 1 - F(a)^n = 0.5$. The values can then be found using logarithms and `R`.

3.6.2 Horowitz' distributions

Since Manski's estimator has cube root asymptotics, we expect it to perform very badly in the MSE-sense versus probit or logit whenever the assumptions on probit and logit are almost true. We simulate the MSE for several choices of covariate and error distributions when $d = 2$. Also, we study the relative robustness properties under the ϵ-contamination model. For all simulations in this section we select the first computed vertex of the solution set as an estimate. We compute the ML estimator for the logistic regression model (logit), so that we have something to compare the results to. Due to the very high variance of Manski's estimator, its only fair shot at overpowering the logit is for its bias to be much smaller. When n is very small, the variance will dominate the bias. Still, we report the MSE, variance and bias for completeness.

First we consider the the setups L, U, T_3 and H from Horowitz (1992), which were also used Delgado et al. Delgado et al. (2001) in their study of subsampling. Here $d = 2$, $X_{1i} \overset{i.i.d.}{\sim} N(0,1)$, $X_{2i} \overset{i.i.d.}{\sim} N(1,1)$, and the errors are distributed as follows:

- L: logistic with mean $\mu = 0$ and variance 1; scale $s = \sqrt{\frac{3}{\pi^2}}$ (i.e. rescaled logistic regression)
- U: Uniform on $[-\sqrt{3}, \sqrt{3}]$; mean 0 and variance 1.
- T_3: t-distributed with $\nu = 3$, normalized to have variance 1,
- H: $\frac{1}{4}(1 + 2z^2 + z^4)v$, where $z = X_{1i} + X_{2i}$ and v is logistic with $\mu = 0$ and $s = s = \sqrt{\frac{3}{\pi^2}}$.

Horowitz (1992, p. 517) works in the rather peculiar setting of $Y = 1_{[X_{1i}+\beta X_{2i} \geq 0]}$, $\beta = 1$, a choice made to please the slow computers at that time. It only requires a small modification of the one-dimensional algorithm to accommodate to this change. While Horowitz compared Manski's estimator, logistic regression and his own smoothed maximum score estimator, we compare Manski's estimator and logistic regression. Initially we calculated the values for probit regression as well, but as they are very similar to logit they are omitted.

The results in Figure 3.1 on the facing page are similar to those of Horowitz, with the exception of logit is MSE for H. Horowitz obtained much higher values here, which might be explained by the fact that he used $N = 1,000$ replications and we used $N = 10,000$. As it turns out, Horowitz' choice of distributions, covariates and βs does not paint of an unbiased picture of the situation. Some

Table 3.1: Mean square errors, variances and biases for Horowitz' choice of error distributions.

		Logit			Manski			
		MSE	Variance	Bias	MSE	Variance	Bias	$\frac{MSE_l}{MSE_m}$
250	L	0.01630	0.01627	0.00621	0.06972	0.06872	-0.03162	0.23379
500		0.00792	0.00791	0.00238	0.04193	0.04121	-0.02679	0.18889
1000		0.00397	0.00397	0.00209	0.02552	0.02524	-0.01676	0.15556
250	U	0.01916	0.01897	0.01381	0.12154	0.12083	-0.02674	0.15764
500		0.00935	0.00926	0.00933	0.07138	0.07093	-0.02123	0.13099
1000		0.00461	0.00455	0.00815	0.04490	0.04479	-0.01058	0.10267
250	T_3	0.01736	0.01058	0.08235	0.05404	0.05096	0.05553	0.32124
500		0.00615	0.00615	-0.00076	0.02689	0.02614	-0.02753	0.22871
1000		0.00311	0.00310	-0.00202	0.01711	0.01685	-0.01612	0.18177
250	H	1.25814	0.38534	0.93424	0.14850	0.14684	0.04071	8.47232
500		0.91018	0.14515	0.87466	0.05876	0.05850	0.01593	15.48979
1000		0.78567	0.06512	0.84885	0.03044	0.03032	0.01093	25.81045

cases of interest are not covered, most notably non-symmetric distributions, extremely heavy-tailed distributions and combinations of heteroskedasticity with these features. we will study these wild distributions in the next section.

3.6.3 Wild distributions

We extend the Monte Carlo study to cover a non-symmetric distribution with support bounded to the left (adjusted standard log normal), an extremely heavy-tailed distribution (the Cauchy distribution), a skewed distribution with support on $(-\infty, \infty)$, the skewed t-distribution ($\nu = 3$,$\gamma = 3$, adjusted to have median equal to 0), and a bimodal equal weight mixture of $N(-1, 1)$ and $N(1, 1)$. In addition, we will consider homoskedastic and heteroskedastic variants of the log-normal, Cauchy and the bimodal mixture. The heteroskedasticity will be of the form $\sqrt{X_{1i}^2 + X_{2i}^2}$ and $\exp(-X_1)$. The results are found in Table 3.2 on the next page. The reduced bias of Manski's estimator is large enough to compensate for its increased variance in some cases, most notably for the heteroskedastic log-normals.

3.6.4 Contaminated data

We perform a small Monte Carlo study on contaminated data. The basic setup is $Y_i = 1_{[1+\beta_1 X_{1i}+\beta_2 X_{2i}+\epsilon_i]}$, where the covariates are standard normal and the ϵ_is are standard logistic. We consider three different forms of contamination:

Table 3.2: Mean square errors, variances and biases for our set of wild distributions, $n = 800$.

		Logit			Manski			
		MSE	Var	Bias	MSE	Var	Bias	$\frac{MSE_l}{MSE_m}$
	Cauchy	0.02319	0.02105	-0.04624	0.06289	0.06202	0.02962	0.36875
	Log-normal	0.052	0.00511	-0.21654	0.03951	0.03943	0.0089	1.31611
	Skew t	0.02917	0.01613	-0.11418	0.18617	0.17794	0.09076	0.15666
	Bimodal	0.01457	0.01403	-0.02305	0.094	0.09032	0.06068	0.15496
$\sqrt{X_{1i}^2 + X_{2i}^2}$	Cauchy	0.10741	0.01509	-0.30383	0.07596	0.07591	0.00661	1.41404
$\exp -X_1$	Cauchy	0.03224	0.01426	-0.13406	0.06229	0.06227	0.00429	0.5175
$\sqrt{X_{1i}^2 + X_{2i}^2}$	Log-normal	0.11836	0.00461	-0.33727	0.04408	0.04308	-0.0316	2.68543
$\exp -X_1$	Log-normal	0.1194	0.00571	-0.33718	0.03803	0.03749	-0.02333	3.13947
$\sqrt{X_{1i}^2 + X_{2i}^2}$	Bimodal	0.03102	0.01626	-0.12147	0.19252	0.18382	0.09329	0.16112
$\exp -X_1$	Bimodal	0.03079	0.01232	-0.13587	0.13821	0.13372	0.06701	0.22274

1. (Y-contamination). The contaminated data comes from the same model, but with the Y_is swapped. The contamination rate is denoted ϵ.

2. (X-contamination). The contaminated data is from the same model, but a random covariate is multiplied by 10. The contamination rate is denoted δ.

3. A combination of these two: A$<$ fraction ϵ of the the data is Y-contaminated and a fraction δ of the data is X-contaminated.

Both kinds of contaminations can be though of as clerical errors; in the second case, a decimal is misplaced, while the first corresponds to pressing the one wrong button out of two. Also, these different forms of contamination can be seen as logistic regression analogues of outliers in the y-direction and outliers in the x-direction from regression analysis, discussed in Rousseeuw and Leroy (2005, introduction).

We will let $(\beta_1, \beta_2) = (1, 1)$, and will only report $\widehat{\beta_1}$ due to readability and the symmetry involved here: There is no systematic difference between $\widehat{\beta_1}$ and $\widehat{\beta_2}$. we will consider $\epsilon = 0, 0.05, 0.1.0.2$ and $\delta = 0, 0.1, 0.5$. The contamination rate of 0 is taken into the analysis in order to have a baseline we can compare with, while the other choices of ϵ and δ are sort of arbitrary. It should be clear that too large values of ϵ, for instance $\epsilon \approx 0.5$, aren't worth studying, especially with the breakdown point of Manski's estimator in mind. We settle on $\epsilon = 0.4$ and expect things to go wrong. We let $n = 800$, typical of mid-large scale logistic setup. Also, the replication count is $N = 10,000$, which might be too small to catch all that is happening. Nonetheless, the results, reported in Table

Table 3.3: Mean square errors, variances and biases for contaminated data.

		Logit			Manski			
ϵ	δ	MSE	Var	Bias	MSE	Var	Bias	$\frac{MSE_L}{MSE_m}$
0	0	0.00897	0.00846	-0.02259	0.04385	0.04274	0.03338	0.20450
	0.1	0.24801	0.01529	-0.48242	0.04568	0.04523	0.02114	5.42964
	0.5	0.7396	0.00031	-0.85982	0.1101	0.09839	-0.10823	6.71742
0.1	0	0.02095	0.01681	-0.06438	0.06347	0.06165	0.04271	0.33009
	0.1	1.02071	0.00132	-1.00964	0.05438	0.05267	0.04135	18.7709
	0.5	0.82658	0.00032	-0.90899	0.07552	0.0748	-0.02681	10.9453
0.2	0	0.04433	0.03462	-0.09857	0.11281	0.10869	0.06418	0.393
	0.1	1.09034	0.00191	-1.04328	0.08905	0.08604	0.05483	12.2446
	0.5	0.90703	0.00045	-0.95214	0.09005	0.08983	0.01485	10.0723
0.4	0	456.735	456.731	0.07046	443725	443629	9.81226	0.00103
	0.1	290.959	288.945	-1.41915	1068.18	1068.15	0.16433	0.27239
	0.5	18.2414	16.6176	-1.27425	1.62646	0.01987	-1.26751	11.2154

3.3, are illustrating. Not surprisingly, things go out of control with $\epsilon = 0.4$. As can be seen, Y-contamination alone has little effect on the bias of the logistic regression estimator, though the variance increases. Hence it is still able to see the signal under these circumstances. Since the variance of logit is very much smaller than the base variance of Manski's estimator, Manski's estimator has to perform very well on the bias side of things in order to out-perform logit MSE-wise. Luckily for Manski's estimator, it does better than logit for any combination where $\epsilon \neq 0, 0.4$, which suggests usefulness as a robust estimator in practice.

Chapter 4

Density estimation on the unit interval

> The continuous differentiable function is losing its preeminence as a paradigm of knowledge and prediction.
>
> - Jean-François Lyotard in *The Postmodern Condition*

As per Lyotard's dictum, we study density estimation on the unit interval by means of step functions. The object of our attention is the *irregular histogram*, the main worked-out example of cube root asymptotics in this thesis. The study of this histogram is easiest when restricted to the unit interval, whence the name of this chapter. We start out by discussing the important class of density estimators known as *kernel density estimators* (KDEs), with special focus on the *Gaussian copula KDE* Jones and Henderson (2007*a*). In Section 2 we define a rather large class of histogram estimators defined through the minimisation of statistical divergences. Our focus will be on the Kullback-Leibler divergence, and the L_2-divergence will also have a role to play. Section 3 is a tiny one, being entirely about regular histograms. Section 4 is devoted to the question of L_1-consistency of the histograms. In Section 5, the longest section of this thesis, we will prove consistency of the irregular histograms introduced in Section 2, under certain conditions, using the theory developed in Chapter 2. In particular we will make use of a $\delta > 0$ in order to uniformly bound the maximised objective function. We will also use the rate theorem in order to establish the cube root rate of convergence for the split points and weights of these histograms. In

Section 6 we discuss algorithms for the computation of irregular histograms given some data $x_1, x_2, ..., x_n$. As was the case with Manski's estimator, the usual tools of gradient descent will not be available, and we will be forced to find other ways to deal with the problem. In the process, we show that the histograms are computable in $O(n^2k)$ time. In Section 7 we describe a frequent anomaly encountered when using these histograms, and a potential remedy. The penultimate section is devoted to an illustration of the histogram on some examples of real data, including an application of the subsampling bootstrap in order to get confidence intervals for the split points. Finally, the last section is devoted to the description and investigation of the *CIC* (Hjort in (Hjort, 2007, p. 33)), a generalisation of AIC to the setting of irregular histograms. It turns out that the bias term is difficult to estimate, but a subsampling approach appears to work fine.

4.1 Kernel density estimators

4.1.1 Introduction

An important class of non-parametric density estimators are the kernel density estimators (KDEs), which we now describe. Let $X_1, \ldots, X_n \sim F$, where F has a strictly positive smooth density f on $\mathbb{R}$. Let K be a function which is symmetric around 0 and positive. The kernel density estimate of f is

$$\widehat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{h} K\left(\frac{x - X_i}{h}\right),$$

where h is the (data dependent) bandwidth. Choices for K include any density function symmetric around 0, e.g the standard normal distribution, the Epanechnikov kernel and the tricube kernel (Hastie et al., 2005, chapter 6.1). As it turns out, in the present setting of strictly positive smooth densities on $\mathbb{R}$, the choice of kernel is unimportant Wand and Jones (1994, p. 28). Note that $\widehat{f}$ is a density of an actual random variable, namely $\sum_{i=1}^{n} \xi_i \frac{1}{h} K(\frac{x-X_i}{h})$, where ξ is a multinomial vector with equal cell probabilities. This makes it very easy to do Monte Carlo simulations, which can be of interest in pre-smoothing and the smoothed bootstrap.

Under *no* conditions on f, the kernel density estimator is L_1-consistent whenever $nh \to \infty$ as $h \to 0$, provided the kernel K has integral 1 and is everywhere positive Devroye and Györfi (1985, chapter 3, theorem 1). We will discuss this concept in the context of histograms in Section 4.4. A widely used criterion

for selecting the bandwidth is to minimise the *integrated mean squared error*, that is $\int E(f(x) - \widehat{f}(x))^2 dx$. As usual, the mean squared error is used mainly due its mathematical tractability, and the interpretation is simple enough. By Fubini's theorem it equals $E\int(f(x) - \hat{f}(x))dx$, which gives rise to its acronym, MISE (mean integrated squared error). Note that $\hat{f}$ is a random function, hence $\widehat{f}(x)$ is the random variable we're taking expectations with regards to. From a given criterion, like the MISE or pointwise MSE, the optimal bandwidth is the bandwidth h_n which minimises the criterion. For these criteria (which are the ones we will consider), the optimal bandwidth can be understood as making a balanced trade-off between variance and bias. Through asymptotics and straightforward calculations we find that the optimal bandwidth has order $n^{-\frac{1}{5}}$ (Wand and Jones, 1994, p. 28):

$$h_n = n^{-\frac{1}{5}} \left[\frac{\int K(x)^2 dx}{(\int x^2 K(x) dx)^2 \int (f''(x))^2 dx} \right]^{\frac{1}{5}}. \tag{4.1.1}$$

Let K be the standard normal density. If we approximate f by a normal density and let $\widehat{\sigma}$ be a consistent estimator of σ, we get the ready to use formula Silverman (1986, p. 47)

$$h_n = 1.059 \widehat{\sigma} n^{-\frac{1}{5}},$$

this is the *normal reference rule* for bandwidth selection.

Several methods are used to compare different density estimation procedures, where by procedure we understand a method which does everything automatically, including the selection of a bandwidth. An example of such a procedure is the normal reference rule for KDEs with Gaussian kernels. Perhaps the most prominent method of comparison is the use of L_2-distances and Hellinger distances, obtained through simulations on a suitable selection of prototype densities.

As it turns out, this sort of density estimator is not appropriate when studying densities on the unit interval. Assume our density looks like the one in the plot: It has high density close to the boundary at 0. Since there is no data to the left of 0, $\widehat{f}(x)$ will tend to underestimate $f(x)$ whenever x is close to 0, while assigning positive mass to the "forbidden region" to the left of 0. This phenomenon is called *boundary bias* and is explained further in Hastie et al. (2005, chap. 6).

There are many methods designed to deal with the boundary bias issue, for a brief review see e.g. Chen (1999). we will describe a few, with speciel emphasis

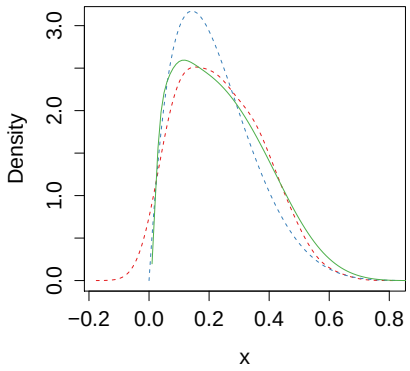

Figure 4.1.1: Illustration of boundary bias. 100 observations were drawn from $\beta(2, 7)$. The blue curve is the true density, while the red curve is a kernel density estimate using the normal reference rule. The green curve is the Gaussian copula kernel estimate described below.

on the Gaussian copula density estimator.

Beta kernels Seemingly, the source of boundary bias is that symmetric kernels can't take into account that there exists regions with no data whatsoever. A reasonable solution is to use a semi-flexible non-symmetric kernel which continuously manipulates the support. Chen's Chen (1999) proposal is to use,

$$\widehat{f}(x) = \frac{1}{n}\sum_{i=1}^{n} K_{x/b+1,(1-x)/b+1}(X_i),$$

where $K_{a,b}$ is the Beta density with shape parameters a, b. This is not a bona fide density, however, for x appears as a *parameter* in the Beta-density, not as an argument. This makes the estimator difficult to interpret and to sample from. Jones and Henderson (2007*b*) explored the possibility of switching these roles. Also, Chen (1999) proposes a superior bias-corrected version, which has proven to be quite popular and exhibits good performance.

Probit-transformed local likelihood Through any quantile function for a strictly positive density on $\mathbb{R}$, we can transform data from $[0, 1]$ to $\mathbb{R}$, apply an ordinary kernel density estimator there, and back-transform the results. A downside is that this method can induce serious bias unless care is taken. Geenens (2014) proposal is to use the probit transform in this manner, but instead of using an ordinary kernel density estimator, he uses a local likelihood approach, which is known to reduce bias (Hjort and Jones, 1996). This estimate is neither

a bona fide density or easy to sample from, but has superior performance.

4.1.2 Gaussian copula kernels

Jones and Henderson (2007*a*) proposed an elegant modification of the beta kernel approach. The idea is to use the conditionals of a bivariate Gaussian copula as kernels. The bivariate Gaussian copula is a random variable of one parameter ρ, defined in the following manner: Let $(Z_1, Z_2) \sim N_2(0, \Sigma)$, with $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$, and put $(X_1, X_2) = (\Phi(Z_1), \Phi(Z_2))$. Its density is given by

$$c(x_1, x_2; \rho) = \frac{1}{\sqrt{1-\rho^2}} \exp\left(-\frac{[\rho^2(\Phi^{-1}(x_1)^2 - 2\rho\Phi^{-1}(x_1)\Phi^{-1}(x_2) + \rho^2(\Phi^{-1}(x_2))^2]}{2(1-\rho^2)} \right).$$

The basic feature of a copula is that the marginals are uniform, but the conditionals can be very rich.

Definition 4.1.1 (Gaussian copula kernel density estimator)**.** Let $\{X_i\}_{i=1}^n \overset{i.i.d.}{\sim} f$ for some density $f : [0,1] \to \mathbb{R}$, $\{Y_i\}_{i=1}^n$ be independent Gaussian copulas, and $h = \sqrt{1-\rho}$ be a (data driven) bandwidth. The Gaussian copula kernel estimator is the mixture $\sum_{i=1}^n \xi_i [Y_i | y_2 = X_i]$, where $\xi = (\xi_1, \xi_2, ..., \xi_n)$ is a multinomial vector with equal cell probabilities. This variable has density

$$\widehat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} c(x, X_i; 1-h^2).$$

For use in pre-smoothing of histograms (Section 4.7), we will need to sample from this estimator. As a first step, we must sample from the conditional distribution $c(x_1 \mid X_2; \rho)$. Recall the formulas for the conditional mean and variance in a bivariate normal (Z_1, Z_2),

$$\begin{aligned} \mu_1|Z_2 &= \mu_1 + \frac{\rho}{\sigma_2^2}(Z_2 - \mu_2), \\ \sigma_1^2|Z_2 &= \sigma_1^2 - \frac{\rho^2}{\sigma_2^2}, \end{aligned}$$

which in our setting translates to $\mu|Z_2 = \rho Z_2$ and $\sigma^2|Z_2 = 1-\rho^2$. Using these formulas together with the fact that the conditional distributions are normal, we sample from $Z_{X_2} \sim N(\rho\Phi^{-1}(X_2), 1-\rho^2)$ and apply Φ, so that $X_1|X_2 \sim \Phi(Z_{X_2})$. It is now straight forward to sample from the Gaussian copula density estimator.

The Gaussian copula KDE has several nice features,

1. It results in a bona fide density,
2. it is easy to interpret and fast to calculate,
3. easy to sample from,
4. has good Hellinger and MISE performance in many different settings.

In this thesis we will use this density estimator whenever nothing else in mentioned, together with a normal reference rule, giving the bandwidth

$$h = \widehat{\sigma}(2\widehat{\mu}^2\widehat{\sigma}^2 + 3(1-\widehat{\sigma}^2))^{-\frac{1}{5}} n^{-\frac{1}{5}},$$

where $\widehat{\sigma}$, $\widehat{\mu}$ are the maximum likelihood estimates of the probit-transformed data (see Jones and Henderson (2007*b*)). A natural extension of the Gaussian copula KDE to the realm of derivatives is

$$\widehat{f}'(x;h) = \frac{1}{n}\sum_{i=1}^{n} \frac{d}{dx} c(x, X_i; 1-h^2).$$

Here the optimal rate for the bandwidth has been reduced from $n^{-\frac{1}{5}}$ to $n^{-\frac{1}{7}}$. This is not surprising, as there is seemingly a larger inherent difficulty involved in the estimating derivatives *vis-à-vis* estimating the density. This manifests itself through a much larger pointwise variance as a function of h. The rate $n^{-\frac{1}{7}}$ is conjectured from a result on ordinary kernel density estimators, see Wand and Jones 1994, section 2.12.

4.2 General histograms

A general histogram on $[0,1]$ is a density of the form

$$h(x) = \sum_{i=1}^{k} \frac{w_i}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i)}(x), \tag{4.2.1}$$

where $a_0 = 0$, $a_k = 1$, $a_i < a_{i+1}$ for each i, and $w_i < 0$ are weights summing to one; that is, $\sum_{i=1}^{k} w_i = 1$. Let $\mathcal{F}$ be a suitable class of densities on the unit interval. The exact nature of $\mathcal{F}$ will vary from problem to problem, and isn't very important right now. Later on, when we derive our limiting distributions, the exact nature of $\mathcal{F}$ *is* important — it contains only smooth densities and no densities with the shape of a histogram. Whenever we have a $P \in \mathcal{F}$, with

associated density f, we can define its *best approximating histogram* through the minimisation of statistical divergences, which we will also call statistical distances. A statistical divergence is defined as follows: Let $\mathcal{P}$ be space of probability measures with common support. If $d : \mathcal{P}^2 \to \mathbb{R}$ is a function such that $d(f, g) \geq 0$ for any $f, g \in \mathcal{P}$ and $d(f, g) = 0$ iff $f = g$.

We will consider two divergences:

1.) Kullback-Leibler divergence. The quantity is defined as $d_{KL}(f, h) = \int f(x) \log \frac{f(x)}{h(x)} dx$. In ordinary parametric statistics, minimising this distance is equivalent to finding the maximum likelihood estimate. It is an information theoretic distance, and is difficult to understand visually. Accordingly, it can be argued that it is inappropriate for histograms. As we will see, it is equivalent to L_2-minimisation in regular histograms, and it is asymptotically equivalent to the minimisation of the Hellinger distance, which is more easily interpretable visually, and is arguably more adequate than the L_2-distance. The minimisation of the Kullback-Leibler divergence is equivalent to the maximisation of $\int f(x) \log h(x) dx = P \log h$. Since $h(x) = \sum_{i=1}^{k} \frac{w_i}{a_{i-1} - a} 1_{[a_{i-1}, a_i)}(x)$, we get the criterion function

$$m_{(a,w)}^{KL} = \sum_{i=1}^{k} \log \frac{w_i}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i)}(x), \tag{4.2.2}$$

and the desired argmax objective

$$Pm_{(a,w)}^{KL} = \sum_{i=1}^{k} \log \frac{w_i}{a_i - a_{i-1}} P[a_{i-1}, a_i).$$

2.) L_2-distance. This is the ordinary L_2-norm, defined by $\int (f(x) - h(x))^2 dx = \int f(x)^2 dx - 2 \int f(x) h(x) + \int h(x)^2 dx$. This is a classical distance measure which is easy to interpret, but it weights small distances too little. For histograms, to minimise the expression above is equivalent to the maximisation of

$$Pm_{(a,w)}^{L_2} = \sum_{i=1}^{k} w_i \frac{(2P[a_{i-1}, a_i) - w_i)}{a_i - a_{i-1}},$$

with underlying criterion function

$$m_{(a,w)}^{L_2} = \sum_{i=1}^{k} \left[\frac{2w_i}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i)}(x) - \frac{w_i^2}{a_i - a_{i-1}} \right], \tag{4.2.3}$$

as can be seen by straight forward calculations. We require the assumption $f \in L_2$ for this to work.

Special mention: The Hellinger distance. The Hellinger distance is defined by $d(f,h) = \frac{1}{2}\int(\sqrt{f(x)}-\sqrt{h(x)})^2 dx$, which behaves asymptotically like Kullback-Leibler. Compared to the Kullback-Leibler divergence, it has the nice property of being an actual metric, and it is easier to understand visually. When taking roots, values below one are pushed upwards, and values above one are pushed downwards. This makes the distance perhaps more appropriate for histograms than L_2, which weight differences when densities are high more than differences when densities are small. To minimise the Hellinger distance is equivalent to maximising

$$\sum_{i=1}^{k} \sqrt{\frac{w_i}{a_i - a_{i-1}}} \int_{a_{i-1}}^{a_i} \sqrt{f(x)} dx,$$

which requires an estimate $\hat{f}(x)$ of $f(x)$, for instance a Gaussian copula kernel density estimate. However, this is hardly an argument against the use of the Hellinger distance, as it is convenient to use kernel density estimates for all sorts of histograms.

Kinds of histograms. Given a statistical distance, we can define three forms of histograms. Their forms are decided by which quantities we allow to vary and which we force to be constant. *Regular histogram* are histograms with a fixed bin width and variable weight, which in this case equals height times the bin width. These are the objects we usually associate with histograms. *Irregular histograms* are usually understood as histograms with variable bin width and variable weight, but we will also consider irregular histograms with fixed weight and variable bin width. We will sometimes call these irregular histograms *quantile histograms.* Figure 4.2.1 illustrates the main graphical differences between these three kinds of histograms, and Table 4.1 compares gives a summary of their pros and cons. In Section 4.7 we do a small simulation study comparing their MISEs and Hellinger distances.

Proposition 4.2.1. *For regular histograms, minimisation of L_2 and Kullback-Leibler divergences are equivalent.*

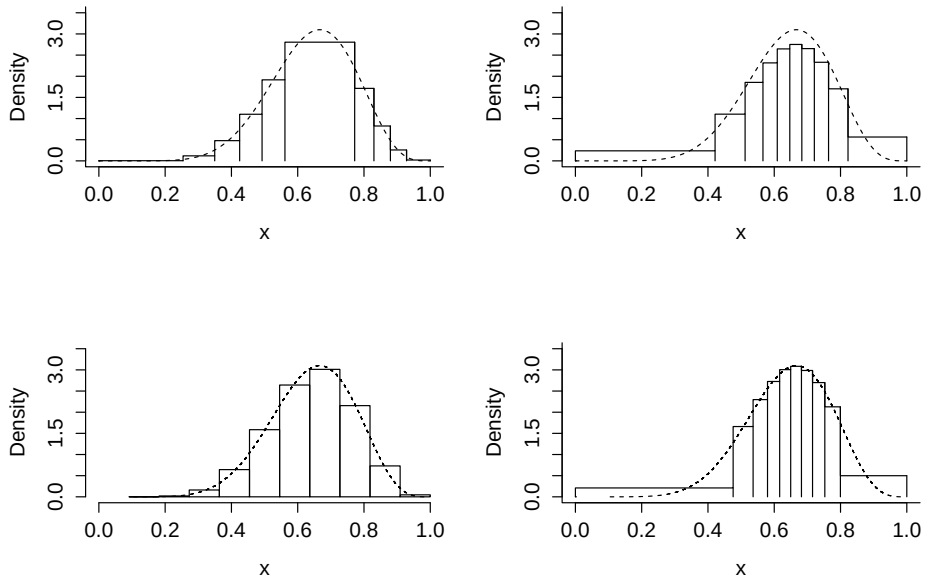

Figure 4.2.1: Three different best approximating histograms and one k-spacing estimator with $k = 10$ and underlying distribution $\beta(9, 5)$ (upper left) Irregular histogram obtained from KL, (upper right) quantile histogram (KL), (lower left) regular histogram, (lower right) the k-spacing estimator described in the next section. Notice the following about the quantile histogram: Since it cannot flexibly manipulate the height of the left-most bar, it has been forced to become too tall. The same goes for the the middle section, since it cannot manipulate the weights, it is forced to underestimate the density, even tough it approximates its shape quite well. The same mechanism forces the bin widths to be small around the high regions, explaining the difference between its looks and that of the irregular histograms to its left.

Table 4.1: Pros and cons of the different kinds of histograms.

	Pros	Cons
Regular	Well known and easy to interpret. L_2 and KL histograms are equivalent, and so is the usual histogram construction. It has smaller variance than the other histogram procedures. The limiting distribution of the weights is multivariate normal, and converges with the rate $n^{\frac{1}{2}}$.	They fail to capture the structure of the data well when there are few bins and relatively large sections of small density, which leads to bias. If a bin contains 0 observations, $d_{KL}(f,h) = \infty$ under the assumption that f is everywhere non-zero.
Irregular	Is a classical histogram in the sense that each bin has weight corresponding to the amount of observations in it. Can match any histogram-shaped density exactly. Has smaller bias than the other forms.	It is costly to construct, yields unstable estimates and converges at a $n^{\frac{1}{3}}$-rate. Every bin will contain at least one observation.
Quantile	Each a_i corresponds to the $\frac{i}{k}$th quantile of distribution $h(x)$ defined by the histogram.	It is costly to construct, yields unstable estimates and converges at a $n^{\frac{1}{3}}$-rate. It is difficult to interpret. Also, it can be seriously biased upwards in sections of small density, often giving a wrong impression of the data. The same kind bias also occurs in sections of high density. (See figure 4.2.1).

Proof. (L_2) Recall that the objective function for minimisation is

$$f(w) = -2\sum_{i=1}^{k} P(a_{i-1}, a_i)w_i + \sum_{i=1}^{k} w_i^2,$$

where $a_0 = 0$, $a_k = 1$, together with the constraint $\sum_{i=1}^{k} w_i = 1$ and $w_i \geq 0$ for all i. Using Lagrange multipliers, with constraint $g = \sum_{i=1}^{k} w_i = 1$, we find that $\frac{\partial(f+g)(w)}{\partial w_i} = -2P(a_{i-1}, a_i) + 2w_i + \lambda$. Leave λ alone on the right hand side and sum over i to get $\lambda = 0$, which yields the unique solution $w_i = P(a_{i-1}, a_i)$. Note that the constraint $w_i \geq 0$ is automatically fulfilled. As for Kullback-Leibler, we use the same method on $f(w) = \sum_{i=1}^{k} P(a_{i-1}, a_i) \log w_i$ to get $\frac{\partial(f+g)(w)}{\partial w_i} = P(a_{i-1}, a_i)\frac{1}{w_i} + \lambda$, which gives $\lambda = -1$ and the maximum $w_i = P(a_{i-1}, a_i)$. □

By using the finite sample analogue $P_n(a_{i-1}, a_i)$ for $P(a_{i-1}, a_i)$, we see that both the L_2 and KL regular histograms correspond to the ordinary regular histograms we know from elementary school. The following corollary tells us that KL and L_2 irregular histograms (with variable weights) are histograms as we know them.

Corollary 4.2.2. *Let P be a probability measure on $[0, 1]$. The optimal weights for both the L_2 and KL irregular histograms are $P([a_{i-1}, a_i))$, where a_i are the optimal split points.*

Proof. Given the split points, the problem reduces to that of the previous proposition. □

We call this, colloquially, the "good histogram property". In addition to making the resulting histograms correspond to ordinary histograms as we know them, the property effectively reduces the dimension of the objective function.

Now that we have defined the best approximating histogram for the underlying distribution F, we will discuss their estimation. This estimation is carried out by using the "plug in estimator" P_n in place of P. Thus

$$(\widehat{a}, \widehat{w}) = \arg\max_{(a,w)} P_n m_{(a,w)},$$

where $m_{(a,w)} = m^{L_2}_{(a,w)}$ if we want to minimise the L_2 distance, and $m_{(a,w)} = m^{KL}_{(a,w)}$ for Kullback-Leibler. For ease of exposition, we always include both a and w in the arguments of m, even tough we are allowed to pre-specify their values, a choice that hopefully will not cause confusion. Note that this is the setting of M-estimation, discussed Section 2.2.

Other divergences that might be considered are the BHHJ divergences Basu et al. (1998), a parametrized family of divergences defined by

$$d_\alpha(f,g) = \int \left[f^{1+\alpha}(z) - (1+\frac{1}{\alpha})g(z)f^\alpha(z) + \frac{1}{\alpha}g^{1+\alpha}(z) \right] dz$$

for $\alpha > 0$ and $d_0(f,g) = d_{KL}(f,g)$. These divergences were developed for outlier robust statistics, and don't seem very relevant for the case at hand. Nonetheless, they will allow us make a finer trade-off between the features of the Kullback-Leibler histograms and L_2-histograms, and would connect the limiting distributions of these two in a unifying manner. Still, the rather similar behaviour of the Kullback-Leibler and L_2-histograms, illustrated in Section 4.7, makes it seem like this isn't worth the effort. Finally, there are other important distances we have not investigated at all, for instance the Kolmogorov distance defined by $d_K(F,G) = \sup_x |F(x) - G(x)|$, and the L_1-distance, $d_{L_1} = \int |f(x) - g(x)|dx$. The Kolmogorov is a global distance measure, and isn't appropriate for density estimation. While the L_1-distance is arguably the most appropriate distance measure between densities, it is difficult to work with. By Scheffé's identity (see Section 4.4 on L_1-consistency), minimising the L_1 criterion is equivalent to the minimisation of the *total variational distance,* $d_{TV}(f,g) = \sup_{B\in\mathscr{B}} |\int_B f(x)dx - \int_B g(x)dx|$, where $\mathscr{B}$ is the Borel σ-algebra on $[0,1]$. This is particularly attractive, as it minimises the maximal error of $|\widehat{P}(A) - P(A)|$ across *every* measurable set A. Furthermore, Devroye and Lugosi, 2012, chapter 6 argue that both the KL- and L_2-distances are fundamentally inappropriate as criteria for density estimation, but their examples of bad properties for these two distances only hold for unbounded intervals. Nevertheless, minimising $d_{TV}(f,g) = \sup_{B\in\mathscr{B}} |\int_B f(x)dx - \int_B g(x)dx|$ is likely not amenable to the techniques described in Section 2.2 on M-estimation.

4.3 Regular histograms

Regular histograms are considerably easier to analyse and understand than irregular histograms, so they are a convenient starting point for our discussion. We will also discuss the so-called k-spacing estimator (Lugosi and Nobel, 1996), a kind of histogram reminiscent of the KL/L_2-histograms with constant weights.

Let $k+1$ be the bin count and $a_i - a_{i-1}$ be the predefined bin widths and F

be an arbitrary distribution supported on $[0,1]$. The estimates of interest are

$$\widehat{\theta}_n = (P_n[0,a_1), P_n[a_1,a_2), ..., P_n[a_k,1]).$$

The vector $n\widehat{\theta}_n$ is multinomially distributed with n trials and cell probabilities $P(a_{i-1},a_i)$. Let $n \to \infty$, and $\sqrt{n}\widehat{\theta}_n$ can be approximated by the multivariate normal distribution with mean $\mu = (P[a_0,a_1), P[a_1,a_2), ..., P[a_k,1])$ and covariance matrix $\Sigma_{ij} = P(a_{i-1},a_i)P(a_{j-1},a_j)$ when $i \neq j$ and $\Sigma_{ii} = P(a_{i-1},a_i)(1-P(a_{i-1},a_i))$ otherwise. Hence $\sqrt{n}(\widehat{\theta}_n - \mu) \xrightarrow{d} N_k(0,\Sigma)$.

k-spacing estimator

A variant of histograms not yet discussed is the density

$$\begin{aligned} h(x) &= \sum_{i=1}^{k} \frac{k^{-1}}{q_i - q_{i-1}} 1_{[q_{i-1},q_i)}(x), \\ &= \sum_{i=1}^{k} \frac{P[q_{i-1},q_i)}{q_i - q_{i-1}} 1_{[q_{i-1},q_i)}(x). \end{aligned}$$

where q_i is the $\frac{i}{k}$th quantile of P. This histogram is sometimes called the k-spacing estimator (Lugosi and Nobel, 1996), and was first investigated by Van Ryzin (1973). While it looks similar to our quantile histograms, it is not the result of KL or L_2minimisation of the histogram objective function in equation 4.2.1. Interestingly, it has constant weights but also satisfies the good histogram property, for $k^{-1} = P_n[q_{i-1},q_i)$ by construction. Let $\widehat{q}_i$ be the $\frac{i}{k}$-th quantile of the empirical distribution P_n, and define $\widehat{h}(x) = \sum_{i=1}^{k} \frac{k^{-1}}{\widehat{q}_i - \widehat{q}_{i-1}} 1_{[\widehat{q}_{i-1},\widehat{q}_i)}(x)$. Now we're interested in the limiting distribution of $(\widehat{q}_1, ..., \widehat{q}_k)$.

Proposition 4.3.1. *Let F be a distribution with density f, $X_1, ..., X_n \overset{i.i.d.}{\sim} F$ and let $q = (q_1, ..., q_{k-1})$, $\widehat{q} = (\widehat{q}_1, ..., \widehat{q}_{k-1})$ be the $\frac{i}{k}$th quantiles and sample quantiles, respectively. Then*

$$\sqrt{n}(\widehat{q} - q) \xrightarrow{d} N(0,\Sigma),$$

where $\Sigma_{ij} = \frac{i(k-j)}{k^2 f(q_{(j)}) f(q_{(i)})}$ for $j \geq i$.

A proof is found in e.g. Babu and Rao (1988, theorem 2.1). We could also use the ideas of Section 2.2, or more specifically the rate theorem 2.2.4 on page 22 to find the limiting covariance structure and the Hessian V. In order to put

quantile estimation into this, we could use the loss function

$$d_p(x,q) = \begin{cases} (1-p)|x-q| & \text{when } x < q, \\ p|x-q| & \text{when } x \geq q. \end{cases}$$

This function has the property that $\arg\min_q Ed_p(X,q) = F^{-1}(p)$ (see e.g. Hao and Naiman (2007, p. 21)). Thus $m_p = \sum_{i=1}^{k} d_{p_i}(\cdot, q_i)$ is a suitable objective function, where p is constrained to be an increasing vector in $(0,1)$. Notice that when $k = 2$ and $p = \frac{1}{2}$, q is the median. The objective function is $d_{\frac{1}{2}}(x;q) \propto |x-q|$, which was used when we established the limiting distribution of the median, recall Theorem 2.2.3 on page 13. It would be interesting to find out whether there is a statistical divergence d such that $\arg\max_q d(h(x;q,k^{-1}), f(x))$ equals the k-spacing estimate.

This method can clearly be extended to choices of non-equally spaced quantiles. This variant of the quantile histogram is probably better than what we obtain through L_2 and KL minimisation. It retains the positive property of quantile histograms, namely that q_i corresponds to the ith quantile of $h(x)$, but disposes of several negative properties: It doesn't have the unstable estimates we soon will describe (in Section 4.7), the split points have $\sqrt{n}$-asymptotics, its limiting distribution is normal. It is computationally very inexpensive to construct, as finding the k quantiles is done in $O(n \log n)$ time. Also, they look roughly the same whenever the L_2/KL histograms have been estimated properly. As the main purpose of histograms is visualisation, this is yet another point in favour of this variant. If the goal of constructing a histogram is compression instead, this approach should also be preferable, as the quantiles themselves are very well understood. Another point in its favour is the fact that L_2/KL quantile histograms can fool us in a subtle way: In figure 4.2.1, notice that the quantile histograms seem to "focus" on the region around the mode. This makes it seem intuitive that the estimator is more accurate in exactly this region. But as we can see from the plot, this is not the case. The regular quantile histogram doesn't suffer from this defect.

4.4 L_1-consistency

In this section we obtain a result on L_1-consistency for the irregular histograms we have discussed. A density estimate $\widehat{f}_n$ is said to be L_1-consistent for f if $\int |\widehat{f}_n(x) - f(x)|dx \to 0$ almost surely. This is sometimes called *strong*

L_1-consistency, where in *weak* L_1-consistency almost sure convergence is replaced with convergence in probability. We only consider L_1-consistency as it is, arguably, the most reasonable criterion of consistency for density estimates, see Devroye and Györfi (1985). Note that L_1 and L_p consistency is not the same: Since $\lambda([0,1]) = 1$, $\int f_n dx \to 0$ if $\int f_n^p dx \to 0$ for any positive f and $p > 2$ by Hölder's inequality (Folland, 1984, p. 178). The converse if false, just take $f_n = x^{-\frac{1}{p}}$.

We show that the irregular KL/L_2-histograms with variable weights are L_1-consistent provided the condition

$$\mathbb{P}[\max_{i=1,\ldots,k-1} P[\widehat{a_{i-1}}, \widehat{a_i}) > \gamma] \overset{a.s}{\to} 0 \text{ for every } \gamma > 0, \tag{4.4.1}$$

is satisfied. This is a relatively simple corollary of Theorem 1 of the delightful paper "Consistency of data-driven histogram methods for density estimation and classification" of Lugosi and Nobel (1996). Their paper builds on previous work by Zhao et al. (1991). In this section, we present simplified variants of their results: The results Lugosi and Nobel are far more general, "living on" $\mathbb{R}^d$ instead of the unit interval. Their proofs of Lemma 1 and Theorem 1 and are simplified for our more restricted setting. We supply some definitions from their paper. Let π be a random partition of $\mathbb{R}^d$, λ be the Lebesgue measure, P_n be the empirical measure of n i.i.d. observations $X_1, ..., X_n$ from P with density f, and define

$$\widehat{f}(x) = \begin{cases} \frac{P_n(\pi[x])}{\lambda(\pi[x])}, & \text{if } \lambda(\pi[x]) < \infty, \\ 0 & \text{otherwise.} \end{cases}$$

where $\pi[x]$ is the element $A \in \pi$ which contains x. In Lugosi and Nobel, the partitions π are any random (data driven) partitioning rules, for instance tree-based rules.

In our setting, $\mathbb{R}^d$ is replaced by $[0,1]$ and π is always a collection of disjoint intervals, which makes things simpler. We briefly describe what π and $\pi[x]$ look like for our collection of histogram variants, where k is fixed: For the irregular KL/L_2-histograms of variable weight, $\pi[x]$ is the set $[\widehat{a_{i-1}}, \widehat{a_i})$ which x belongs to, where $\widehat{a_i}$s are the corresponding split point estimates, and π is the collection of all of these intervals. For the k-spacing histograms, $\pi[x] = [\widehat{q_{i-1}}, \widehat{q_i})$, where $\widehat{q_i}$s are the sample quantiles, and $\pi = \{[\widehat{q_{i-1}}, \widehat{q_i}) \mid i = 1, .., k-1\}$. The regular histogram can have any non-random partition π of k disjoint intervals, and $\pi[x]$ picks which one it belongs to. Finally, the constant weight KL/L_2-histograms don't fit into

this framework, but we will speculate on how to show L_1-consistency for these later on.

We require two technical definitions.

Definition 4.4.1. Let $\mathscr{G}$ be a family of sets in $\mathbb{R}^d$. We define the *shatter coefficient by*

$$\mathcal{S}_n(\mathscr{G}) \quad = \quad \max_{B\subset\mathbb{R}^d,\, |B|=n} (|\,\{B\cap C \mid C\in\mathscr{G}\}\,|)\,.$$

Let $\mathscr{A}$ be a family of partitions of $\mathbb{R}^d$. We define the *growth function of* $\mathscr{A}$ by

$$\Delta_n^*(\mathscr{A}) = \max_{B\subset\mathbb{R}^d,\, |B|=n} |\,\{\{A_1\cap B, A_2\cap B, ...A_r\cap B\} \mid (A_1,A_2,...,A_r)\in\mathscr{A}\}\,|.$$

While the shatter coefficient is a classic of VC-theory, the growth function is specific to this particular paper (Lugosi and Nobel, 1996). Take note of the difference between these two concepts: The shatter coefficients finds the cardinality of a set of sets: It finds the maximal number of distinct subsets of an n-ary set of numbers which $\mathscr{G}$ can isolate. The growth function finds the cardinality of a set of sets of sets, namely the maximal number of distinct *partitions* of any n-ary set in $\mathbb{R}^d$ that $\mathscr{A}$ can isolate. The concept $\Delta_n^*(\mathscr{A})$ is strictly speaking not needed in the sequel. Still we feel it is worth including, just to be true to the paper. Also, as it gives a better feeling for the situation. The following theorem is called the *Vapnik-Chervonenkis inequality* (Vapnik and Chervonenkis, 1971), and a proof can be found in e.g. Devroye et al. (2013, section 12.4).

Theorem 4.4.2. *Let $\mathscr{G}$ be a family of sets, P a probability measure, and P_n the empirical measure of P. For every $n\geq 1$ and $\epsilon>0$,*

$$\mathbb{P}(\sup_{A\in\mathscr{G}} |P_n(A)-P(A)|>\epsilon) = 4\mathcal{S}_{2n}(\mathscr{G})\exp(-n\epsilon^2/8), \tag{4.4.2}$$

where $\mathcal{S}_{2n}(\mathscr{G})$ is the shatter coefficient of $\mathscr{G}$.

The following beautiful lemma is due to Scheffé (1947), with the proof from Devroye and Györfi (1985, theorem 1). Here $\mathscr{B}$ is the Borel σ-algebra on $\mathbb{R}^d$.

Lemma 4.4.3. *For every pair of densities f,g on $\mathbb{R}^d$,*

$$\int |f(x)-g(x)|dx = 2\sup_{B\in\mathscr{B}} |\int_B (f(x)-g(x))\,dx| \tag{4.4.3}$$

where $\mathscr{B}$ is the Borel σ-algebra on $\mathbb{R}^d$. Furthermore, $B = \{x \in \mathbb{R}^d \mid f(x) > g(x)\}$ is a witness to the supremum.

Proof. Let B be as above, and observe that

$$\int |f(x) - g(x)|dx = \int_B (f(x) - g(x))\, dx + \int_{B^c} (g(x) - f(x))\, dx.$$

Since $\int (f(x) - g(x))dx = 0$,

$$\begin{aligned}\int_{B^c} (g(x) - f(x))\, dx &= \int (g(x) - f(x))dx - \int_B (g(x) - f(x)dx \\ &= \int_B (f(x) - g(x))dx,\end{aligned}$$

and $\int |f(x)-g(x)|dx = 2\int_B (f(x) - g(x))\, dx$. Thus 4.4.3 holds with the equality replaced with " $\leq$ ".

Let $A \in \mathscr{B}$ be arbitrary. Then

$$\begin{aligned}|\int_A (f(x) - g(x))dx| &= |\int_{A\cap B} (f(x) - g(x))dx + \int_{A\cap B^c} (f(x) - g(x))dx| \\ &\leq \max\left(\int_{A\cap B} (f(x) - g(x))dx, \int_{A\cap B^c} (g(x) - f(x))dx\right) \\ &\leq \max\left(\int_B (f(x) - g(x))dx, \int_{B^c} (g(x) - f(x))dx\right) \\ &= \frac{1}{2}\int |f(x) - g(x)|dx.\end{aligned}$$

And the result is proved. □

Lugosi and Nobel arrive at the next result (Lemma 1) through a combination of the Vapnik-Chervonenkis inequality and the previous result.

Lemma 4.4.4. *Let $\mathcal{A}$ be a collection of partitions of $\mathbb{R}^d$. For each $n \geq 1$ and $\epsilon > 0$,*

$$P\left[\sup_{\pi\in\mathcal{A}_n} \sum_{A\in\pi} |P_n(A) - P(A)| > \epsilon\right] \leq 4\Delta^*_{2n}(\mathcal{A}_n)2^{m(\mathcal{A}_n)} \exp(-n\epsilon^2/32), \quad (4.4.4)$$

where $m(\mathcal{A}_n) = \max(|\pi| \mid \pi \in \mathcal{A}_n)$.

Proof. We define $f(x;\pi) = P(\pi[x])$ and $f_n(x;\pi) = P_n(\pi[x])$, and let $\mathcal{U}_\pi$ be the set of all sets which can be written as a union of elements from π.

$$\begin{aligned}\sum_{A\in\pi} |P_n(A) - P(A)| &= \int |f_n(x;\pi) - f(x;\pi)|dx \\ &= 2\sup_{A\in\mathcal{U}} |\int_A (f_n(x;\pi) - f(x;\pi))\, dx|,\end{aligned}$$

by Scheffé's theorem. Now define $\mathcal{U}^{\mathcal{A}_n} = \cup_{\pi\in\mathcal{A}}\mathcal{U}_\pi$, and obtain

$$\begin{aligned}\sup_{\pi\in\mathcal{A}_n}\sum_{A\in\pi} |P_n(A) - P(A)| &= 2\sup_{A\in\mathcal{U}^{\mathcal{A}_n}} |\int_U (f_n(x;\pi) - f(x;\pi))\, dx| \\ &= 2\sup_{A\in\mathcal{U}^{\mathcal{A}_n}} |P_n(A) - P(A)|.\end{aligned}$$

By the Vapnik-Chervonenkis inequality,

$$\mathbb{P}(\sup_{A\in\mathcal{U}^{\mathcal{A}_n}} |P_n(A) - P(A)| > \epsilon) = 4\mathcal{S}_{2n}(\mathcal{U}^{\mathcal{A}_n})\exp(-n\epsilon^2/32).$$

Now we need to show that $\mathcal{S}_{2n}(\mathcal{U}^{\mathcal{A}_n}) \leq \Delta^*_{2n}(\mathcal{A}_n)2^{m(\mathcal{A})}$. Let $B \subset \mathbb{R}^d$ have cardinality n, and suppose that $\max|\{B\cap U \mid U \in \mathcal{U}^{\mathcal{A}_n}\}| = r$, with $U_1, U_2, ..., U_r$ being witnesses to this partitioning. We can then assume that each U_i is from a single partition. The number of such U_is is clearly bounded by the maximal number of partitions for n sets multiplied by the maximal cardinality of these partitions. □

The following corollary (Corollary 1 of Lugosi and Nobel (1996)) is a key result, where it is stated without proof.

Corollary 4.4.5. *Let $X_i \overset{i.i.d.}{\sim} P$ be random vectors in $\mathbb{R}^d$, and let $\mathcal{A}_i$, $i = 1, ..., n$ be a sequence of partition families. Assume the conditions*

(a) $n^{-1}m(\mathcal{A}_n) \to 0$, and,

(b) $n^{-1}\log(\Delta^*_n(\mathcal{A}_n)) \to 0$,

hold as $n \to \infty$. Then

$$\sup_{\pi\in\mathcal{A}}\sum_{A\in\pi} |P_n(A) - P(A)| \overset{a.s.}{\to} 0.$$

Proof. We will use the following consequence of Borel-Cantelli lemma: If for any $\epsilon > 0$, $\sum_{i=1}^\infty P(|X_i - X| > \epsilon) < \infty$, then $X_i \overset{a.s.}{\to} X$ (Rosenthal, 2006, p. 59, corollary 5.2.2). By Lemma 4.4.4,

$$\sum_{n=1}^{\infty} E_n = \sum_{n=1}^{\infty} 4\Delta_{2n}^*(\mathcal{A}) 2^{m(\mathcal{A})} \exp(-n\epsilon^2/32),$$

where E_n is the event $\left\{\sup_{\pi \in \mathcal{A}} \sum_{A \in \pi} |P_n(A) - P(A)| > \epsilon\right\}$. By the root test Rudin (1987, p. 200), the radius of convergence R (here ϵ is the variable!) is given by

$$R^{-1} = \limsup_{n \to \infty} (4\Delta_{2n}^*(\mathcal{A}) 2^{m(\mathcal{A})})^{\frac{1}{n}}.$$

Observe that

$$\limsup_{n \to \infty} (4e^{2n} 2^n)^{\frac{1}{n}} = 8e^2,$$

which gives $R = \infty$ in conjunction with a) and b). Hence the power series converges for every $\epsilon > 0$, and the result is proved. □

Now we're ready for the histogram consistency theorem! Lugosi and Nobel proved a more general theorem for $\mathbb{R}^d$ (see their Theorem 1), which we modify for the case of our restricted setting on $[0, 1]$. The following result also uses ideas from their Theorem 4 on the consistency of k-spacing histograms. In the next theorem, $\widehat{a_i}$ are the random split points corresponding to one of the histogram types discussed in the preceding sections.

Theorem 4.4.6. *Let $X_1, ..., X_n \overset{i.i.d.}{\sim} P$, where P is supported on $[0, 1]$ and has a density f. If, as $n \to \infty$ and $k_n \to \infty$ the following conditions hold:*

(a) $n^{-1} k_n \to 0$, and,

(b) $\mathbb{P}[\max_{i=1,...,k-1} P[\widehat{a_{i-1}}, \widehat{a_i}) > \gamma] \overset{a.s}{\to} 0$ for every $\gamma > 0$,

then

$$\int |f(x) - \widehat{f_n}(x)| dx \overset{a.s}{\to} 0.$$

Proof. Define $f_n = P(\pi[x])/\lambda(\pi[x])$, which is the limit histogram for the partition π_n. Here the partition π_n is defined by

$$\pi_n = \left\{[0, \widehat{a_1}), [\widehat{a_1}, \widehat{a_2}), ..., [\widehat{a_{k-1}}, 1]\right\},$$

where the $\widehat{a_i}$s are obtained by some minimum divergence method. By the triangle inequality, it suffices to show that both $\int |\widehat{f_n}(x) - f_n(x)| dx \to 0$ and $\int |f_n(x) - f(x)| dx \to 0$ almost surely. For the first integral,

$$\begin{aligned}\int |\widehat{f}_n(x) - f_n(x)|dx &= \int \sum_{i=1}^{k} |\frac{P(\pi[x]) - P_n(\pi[x])}{\lambda(\pi[x])}|dx \\ &= \sum_{A\in\pi} |P(A) - P_n(A)| \\ &\leq \sup_{\pi\in\mathscr{A}_n} \sum_{A\in\pi} |P(A) - P_n(A)|.\end{aligned}$$

In order to use Corollary 4.4.5 to conclude $\int |\widehat{f}_n(x) - f_n(x)|dx \to 0$, we have to verify condition a) and b) of that corollary. a) Is true since $m(\mathcal{A}) = k_n$ and $k_n n^{-1} \to 0$ by assumption. To verify b) is slightly more involved. For each n, the quantity $\Delta_n^*(\mathcal{A}_n)$ equals the number of ways n points can be partitioned into k_n intervals. It can be shown that $n^{-1}\log(\Delta_n^*(\mathcal{A}_n)) \leq 2h(\frac{k}{n+k}) \to 0$, where h is the binary entropy function $h(x) = -x\log x - (1-x)\log(1-x)$ (see Lugosi and Nobel (1996, proof of theorem 4)).

As for the other integral,

$$\begin{aligned}\int |f_n(x) - f(x)|dx &= \sum_{A\in\pi}\int_A |f(x) - \frac{P(A)}{\lambda(A)}|dx \\ &= \sum_{A\in\pi}(\lambda(A))^{-1}\int_A |f(x)\lambda(A) - P(A)|dx \\ &= \sum_{A\in\pi}(\lambda(A))^{-1}\int_A |f(x)\int_A dy - \int_A f(y)dy|dx \\ &= \sum_{A\in\pi}(\lambda(A))^{-1}\int_A |\int_A (f(x) - f(y))dy|dx.\end{aligned}$$

Now we apply Fubini's theorem together with the fact that $|\int f d\mu| \leq \int |f|d\mu$ to get

$$\int |f_n(x) - f(x)|dx \leq \sum_{A\in\pi}(\lambda(A))^{-1}\int_{A\times A} |f(x) - f(y)|dxdy.$$

Fix an $\epsilon > 0$ and find a $\gamma > 0$ such that $\sup_{x,y\in A}|f(x) - f(y)| < \epsilon$ for any set A with diameter bounded by δ. For an A in π, either A has diameter bounded by γ, whereupon

$$(\lambda(A))^{-1}\int_{A\times A} |f(x) - f(y)|dxdy \leq \epsilon\lambda(A),$$

or A has diameter greater than γ, which implies

$$\begin{aligned}(\lambda(A))^{-1}\int_{A\times A}|f(x)-f(y)|dxdy &\leq (\lambda(A))^{-1}2\int_A f(x)dxdy\\ &\leq 2P(A).\end{aligned}$$

it is time to make use of assumption (b). Let $A_\pi^\star$ be the union of all $A \in \pi$ with diameter greater than γ. Then there are maximally γ^{-1} such intervals, as more would require an interval of length greater than 1. Then $P(A_\pi^\star) \leq \frac{1}{\gamma}\max P(A)$, which is almost surely less than ϵ in the limit (simply chose $\epsilon\gamma$ in assumption (b)). Since $\lambda[0,1] = 1$ and π is a partition, the event

$$\int |f_n(x) - f(x)|dx \leq 3\epsilon$$

happens almost always for any fixed ϵ, and the result follows. □

The content of our modification is, besides the simplification to $[0,1]$, modified conditions for the theorem to hold. These conditions are more in line with the rest of our discussion. The original theorem replaced our conditions (a) and (b) by condition (c):

$$\mathbb{P}[x \mid \operatorname{diam}(\pi[x]) > \gamma] \overset{a.s}{\to} 0, \text{ for every } \gamma > 0,$$

where π is allowed to be an arbitrary partitioning, not only intervals. Also, condition (a) and (b) from Corollary 4.4.5 were included.

It follows immediately that regular histograms with bin widths going to zero are L_1-consistent. The k-spacing density estimates are also consistent, which follows from the fact that $P_n[\widehat{a_{i-1}}, \widehat{a_i}) = k^{-1}$ by construction and $\sup_i |P_n[\widehat{a_{i-1}}, \widehat{a_i}) - P[\widehat{a_{i-1}}, \widehat{a_i})| \overset{a.s.}{\to} 0$ by the Glivenko-Cantelli theorem; hence (b) is satisfied.

Let's consider the KL/L_2-histograms with variable weights. It appears difficult to find simple conditions which makes condition (b) true. In the next section we will prove consistency of the split points for these histograms, and in the process we introduce a $\delta > 0$ which provides bounds on how small the windows can be. It is natural to let $\delta_k \to 0$ when k increases, and analogously, we can provide a sequence $\gamma_k \to 0$ which forces (b) to be true by the same Glivenko-Cantelli argument as above. We arrive at the following proposition:

Proposition 4.4.7. *Let $\widehat{h}(x)$ be the KL/L_2-histogram with variable weights constructed from f. Then $\widehat{h} \overset{L_1}{\to} f$ almost surely, provided the bin widths are kept smaller than γ_k for some $\gamma_k \to 0$.*

Notice that condition (b) is false for several histograms that probably are consistent. For instance, let f be the density given by

$$f(x) = \alpha \begin{cases} (1-x)^2, & \text{when } x \leq 1/2, \\ 1/4, & \text{otherwise.} \end{cases}$$

Here $\alpha = \frac{5}{12}$ is the norming constant. The true split points of the KL/L_2-histograms will be have $\widehat{a}_{k-1} \approx 0.5$, when k is large enough, but should be consistent provided $f(x) = 3(1-x)^2$ is.

While the irregular histograms are as much about the placing of the split points as the fact that they are good histograms (recall the definition following Proposition 4.2.2), Proposition 4.4.7 only makes use of the fact that they are good histograms, and do not say anything about the limit properties of the split points. It appears plausible that condition

$$\mathbb{P}[\max_{i=1,\ldots,k-1} P[\widehat{a_{i-1}}, \widehat{a_i}) > \gamma] \overset{a.s}{\to} 0 \text{ for every } \gamma > 0,$$

(4.4.1) holds when we restrict our attention to densities f which are nowhere locally constant (*i.e.* there is no x,δ and c such that $f((x-\delta, x+\delta)) = c$). Anyhow, since the histograms are consistent for any such choice of γ_k, the question of consistency without γ_ks is not tremendously important. A possible approach to verifying the condition is to 1.) Prove that $\max_{i=1,\ldots,k-1} P[a_{i-1}, a_i) < \gamma$, where a_{i-1}, a_i are the true split points of P when k is sufficiently large, and 2.) Find a probabilistic bound on the difference $\mathbb{P}(|P[\widehat{a_{i-1}}, \widehat{a_i}) - P[a_{i-1}, a_i)| > \epsilon)$. An example of such a bound, albeit in a situation more akin to that of Manski's estimator, can be found in Devroye et al. (2013, Theorem 4.5).

For the KL/L_2-histograms with constant weights, the same reasoning can't be used, since Theorem 4.4.6 only applies to good histograms. One idea to explore is whether the constant weight histograms behave asymptotically like good histograms in some sense. For instance, one could explore the L_1-distance between the variable weight KL histogram evaluated at $\widehat{a}$, and the constant weight KL histogram evaluated at $\widehat{a}$, where $\widehat{a}$ is the split point estimate of KL with constant weights.

$$|\sum_{i=1}^{k} \frac{k^{-1} - P_n[\widehat{a_{i-1}}, \widehat{a_i})}{\widehat{a_i} - \widehat{a_{i-1}}} \log(P_n[\widehat{a_{i-1}}, \widehat{a_i}))| 1_{[\widehat{a_{i-1}}, \widehat{a_i})}(x) dx \leq$$

$$\sum_{i=1}^{k} |\left[k^{-1} - P_n[\widehat{a_{i-1}}, \widehat{a_i})\right] \log(P_n[\widehat{a_{i-1}}, \widehat{a_i}))|$$

Does this converge to 0 when $n^{-1}k \to 0$ and $k \to \infty$? (Or at some slower rate, like $n^{-1}e^k \to 0$?) Another possibility is that $\widehat{a}$ gets asymptotically close to the vector of k^{-1}-quantiles q_k as $k \to \infty$. Finally, these options might fail, and we would need an entirely different consistency proof, or the estimators fail to be consistent.

Finally, recall the regression histograms mentioned in Section 2.4 on page 28 on binary decision trees. Nobel et al. (1996) studied, in a sister paper of Lugosi and Nobel (1996), sufficient conditions for data-driven regression histograms to be L_2-consistent, meaning that $\int (\widehat{f}(x) - f(x))^2 dP(x) \overset{a.s.}{\to} 0$, where f is the real regression function, $\widehat{f}$ is the data-dependent histogram regressor, and P is the distribution of the covariates X.

4.5 Limit distributions

In this section we prove consistency of the split points of our irregular histograms and derive their limit distributions. Everything in this section is new work. (Hjort, 2007, p. 33) provided the explicit formula for the limiting distribution of the irregular KL-histogram with constant weights, which provided the starting point for this thesis.

4.5.1 Asymptotics for the Kullback-Leibler histogram

We focus on the irregular histogram with KL-minimisation. The objective function has the form

$$m_{(a,w)} = \sum_{i=1}^{k} \log \frac{w_i}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i)}(x),$$

where $(a, w) = (a_1, ...a_{k-1}, w_1, ...w_{k-1})$. When appropriate, we will consider $a_0 = 0$ and $a_k = 1$ to be fixed parameters. We already know that the actual minimiser for the histogram with variable weights is of the form

$$\sum_{i=1}^{k} \log(\frac{P([a_{i-1}, a_i))}{a_i - a_{i-1}}) P([a_{i-1}, a_i)).$$

We define the set of feasible solutions and denote it

$$\mathcal{S} = \left\{ (a, w) \mid a_0 = 0 < a_1 < ... < a_{k-1} < a_k = 1; \sum w_i = 1, w_i < 1 \right\}.$$

We impose the following conditions on the underlying distribution F.

(A1) It has an everywhere positive density $f(x)$ on $(0, 1)$,

(A2) it is differentiable at each $x \in (0, 1)$;

(A3) the Hessian V of $Pm_{(a,w)}$ is negative definite;

(A4) there is a unique, well-separated maximiser of the objective function;

(A5) it satisfies the condition $\frac{w_{i+1}}{a_{i+1}-a_i} \neq \frac{w_i}{a_i - a_{i-1}}$ for each i.

Condition A1 and A2 can be weakened, as we strictly speaking only require them to be non-zero and differentiable in neighbourhoods around the points a_i, $i = 1, ..., k-1$. But this relaxed condition is elusive and difficult to verify. Conditions A1 and A2 tell us which setting we're in: Most importantly, we are in a smooth world. A3 is needed in order to make the limiting distribution make sense and handy for making sure the estimates are consistent. A4 is an identifiability condition: Without it there is no unique histogram which minimises the distance between the underlying F and itself, so it is a natural condition. A5 is necessary to ascertain $n^{\frac{1}{3}}$-convergence, in particular for condition 2.2.5 on page 22 to hold. We will also establish a result for an alternative condition to A5, namely

(A5') F satisfies the condition $\frac{w_{i+1}}{a_{i+1}-a_i} = \frac{w_i}{a_i - a_{i-1}}$ for each i,

which curiously makes everything converge at $\sqrt{n}$-rate to a normal distribution!

First we will prove consistency of the estimates $\widehat{a}, \widehat{w}$. Recall the Consistency Theorem on page 21, which states that we only need to show that $\mathcal{F}$ is Glivenko-Cantelli for consistency to hold whenever the maximum is well-separated. We have already assumed that the maximum is well-separated in A3, which seems reasonable, although we will not attempt to prove or find any necessary or sufficient conditions for this condition to hold in this thesis. In order to cope with unboundedness we slightly modify the set of allowed values $\mathcal{S}$ with the additional constraint $a_i - a_{i-1} \geq \delta$, where $\delta > 0$ is arbitrary. We denote an $\mathcal{S}$ modified in such a way by $\mathcal{S}_\delta$ and go on to prove that the KL histogram with constant weights is Glivenko-Cantelli for every fixed k.

Proposition 4.5.1. *The class* $\mathcal{F} = \left\{ m_a = \sum_{i=1}^{k} \log \frac{k^{-1}}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i)} \mid a \in \mathcal{S}_\delta \right\}$ *is*

Glivenko-Cantelli for any choice of δ.

Proof. By theorem 2.1.4 it suffices to show that $N_{[]}(\epsilon, \mathcal{F}, L_1(P)) < \infty$ for any $\epsilon > 0$ sufficiently small. We show this in detail for $k = 2$. We define the following functions $l_{a,\eta}$ and $u_{a,\eta}$ by

$$\begin{aligned} l_{a,\eta}(x) &= -\log(a+\eta)1_{[0,a-\eta)}(x) - \log(1-a+\eta)1_{[a-\eta,1]}(x), \\ u_{a,\eta}(x) &= -\log(a-\eta)1_{[0,a+\eta)}(x) - \log(1-a+\eta)1_{[a+\eta,1]}(x), \end{aligned}$$

when $a < 0.5$, where the signs in the indicator functions are flipped whenever $a \geq 0.5$. We have $l_{a,\eta} \leq m_{a+\xi} \leq u_{a,\eta}$ whenever $\xi < \eta$, which can be verified by inspecting the definition of m_a.

The $L_1(\lambda)$-distance between these brackets is

$$\begin{aligned} & [\log(a+\eta) - \log(a-\eta)]\,(a-\eta) \\ + \; & [\log(1-a+\eta) - \log(a-\eta)]\,2\eta \\ + \; & [\log(1-a+\eta) - \log(a+\eta)]\,(1-a+\eta). \end{aligned}$$

This simplifies to $2\eta|\log(1-a) - \log(a)|$ whenever η is close to 0 (higher order terms are discarded). Since $a \geq \delta$ and $1 - a \geq \delta$ by our previously imposed constraint, the $L_1(\lambda)$-distances are bounded by $\epsilon = 4\eta\log(\delta^{-1})$. Now assume $P \ll \lambda$, and notice that the $L_1(P)$-distance is bounded by $12\eta\log(\delta^{-1})$ whenever δ is sufficiently small. The increased bound is due to the fact that P might have all its mass inbetween $a-\eta$ and $a+\eta$. In conclusion, we need $\frac{\eta^{-1}}{2} = 6\epsilon^{-1}\log(\delta^{-1})$ such brackets, hence the covering number $N_{[]}(\epsilon, \mathcal{F}, L_1(P)) \leq 6\epsilon^{-1}\log(\delta^{-1}) < \infty$ for every $\epsilon > 0$.

We extended this technique, not entirely rigorously, to the case of an arbitrary k. We define k-variate generalisations of $l_{a,\eta}$ and $u_{a,\eta}$ in a similar manner as above: We let $l_{a,\eta}(x) = \sum_{i=1}^{k} \log \frac{k^{-1}}{a_i - a_{i-1} + \eta} 1_{A_i^\eta}$, where $A_1^\eta = [0, a_1 + \eta)$ when $a_1 > a_2 - a_1$ and $A_1^\eta = [0, a_1 - \eta)$ otherwise. In general, $A_i^\eta = [b_{i-1}, b_i)$ where $b_{i-1} = a_{i-1} + \eta$ if $a_i - a_{i-1} > a_{i-1} - a_{i-2}$ and $b_{i-1} = a_{i-1} - \eta$ otherwise. The point is that $l_{a,\eta}$ lies below $m_{a+\overline{\eta}}$ for any vector of deviations $\overline{\eta}$ pointwise bounded by η.

The $L_1(P)$-distance between $l_{a,\eta}$ and $u_{a,\eta}$ is bounded by $4\eta(k+1)\log(\delta^{-1})$ by the same argument as above. We need $O(\eta^{-k+1})$ such brackets. We can partition $[0,1]$ into $\frac{1}{2\eta}$ sets of length η, $\{[0, 2\eta), [2\eta, 4\eta), ...\}$ where each tuple of

midpoints $\alpha_1, ...\alpha_k$ covers every m_a with $a = \alpha_1 + \xi_1, ...\alpha_k + \xi_k$ with $\xi_i < \eta$. It follows that $N_{[]}(\epsilon, \mathcal{F}, L_1(P)) \leq K\epsilon^{-k+1}$, for some K independent of ϵ, and the result follows. □

The preceding bound also works on the L_2 bracketing entropy $N_{[]}(\epsilon, \mathcal{F}, L_2(P))$. For when η is small, the L_2-distance between $l_{a,\eta}$ and $u_{a,\eta}$ is bounded above by $K\eta^2$ for some K, hence they are clearly bounded by $K'\eta$ for some K' as well. We claim without proof that this construction can be extended to the other two types of histograms. The choice of a $\delta > 0$ *may be* unnecessary, as we could conceivable have used features of the distribution P instead. A good choice of δ is of paramount importance in practice, as it avoids the "Dirac catastrophe" described in Section 4.7 on page 122.

Since $\log N_{[]}(\epsilon, \mathcal{F}, L_2(P)) = 0$ when ϵ is sufficiently large, we can bound the bracketing integral by

$$\begin{aligned} J_{[]}(\delta, F, L_2(P)) &= \int_0^\infty \sqrt{\log N_{[]}(\epsilon, \mathcal{F}, L_2(P))} d\epsilon \\ &= \int_0^1 \sqrt{K_1 \log \epsilon} d\epsilon + K_2 \\ &= \frac{1}{2}\sqrt{\pi K_1} + K_2. \end{aligned}$$

Where K_1, K_2 are constants and the value of $\int_0^1 \sqrt{K_1 \log \epsilon} d\epsilon$ was found by the computer algebra system Maxima (2014). From this we can conclude that the histograms are P-Donsker for fixed k.

Recall that $\mathcal{M}_\delta = \left\{ m_{(a,w)} - m_{(a_0,w_0)} \mid d((a,w),(a^0,w^0)) \leq \delta \right\}$, a local variant $\mathcal{F}$. Here d is the pointwise maximum among $|a_i - a_i^0|$. We define an envelope M_δ for this class,

$$M_\delta = \sum_{i=1}^k \log \frac{w_i + \delta}{a_i - a_{i-1} - 2\delta} 1_{B_i} - \log \frac{w_i}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i)},$$

where $B_i = [a_i \pm \delta, a_{i-1} \pm \delta)$ and the sign of δ is chosen so that $M_\delta(x)$ is maximised for each x. By reasoning similar to that used above, it is clear that the local bracketing integral is bounded,

$$\int_0^\infty \sup_{\delta < \delta_0} \sqrt{\log N_{[]}(\epsilon, \mathcal{M}_\delta, L_2(P)} d\epsilon < \infty,$$

which is one of the conditions for the rate theorem (Theorem 2.2.4 on page 22).

Now we're ready for a result that does several things: It describes the shape of

the limiting Gaussian process in the rate theorem, and establishes the uniformity condition (2.2.5) and Gaussian condition (2.2.6) involved in said theorem.

Theorem 4.5.2. *Let $m_{(a,w)} = \sum_{i=1}^{k} \log \frac{w_i}{a_i - a_{i-1}} 1_{I_i}(x)$ be the objective associated with the KL histogram with either variable or constant weights w_j. Denote the least false vectors by a and w, and let f be the density of P. Provided the conditions (A1-A5) are satisfied, the following is true*

$$\lim_{\delta \searrow 0} \frac{P\left(m_{(a,w)+\delta(g,g')} - m_{(a,w)+\delta(h,h')}\right)^2}{\delta} = \sum_{i=1}^{k-1} f(q_i^0)|g_i - h_i|\beta_i^2,$$

where

$$\beta_i = \log \frac{w_{i+1}}{a_{i+1} - a_i} - \log \frac{w_i}{a_i - a_{i-1}}.$$

Thus condition (2.2.5) is satisfied for $\phi(\delta) = \sqrt{\delta}$. Furthermore, condition (2.2.6)is satisfied with

$$G(h) = \sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \log \frac{w_{i+1}}{a_{i+1} - a_i} - \log \frac{w_i}{a_i - a_{i-1}} \right| W_i(h_i) \tag{4.5.1}$$

where W_i are independent standard Brownian motions.

Proof. First we extend, for convenience's sake, h, g with $h_0, h_k, g_0, h_k = 0$. To calculate $\lim_{\delta \searrow 0} \frac{1}{\delta} \mathbb{P}\left(m_{(a^0,w^0)+\delta(h,h')} - m_{(a^0,w^0)+\delta(g,g')}\right)^2$ we will find

$$\frac{1}{\delta} P\left[m_{(a^0,w^0)+\delta(h,h')} m_{(a^0,w^0)+\delta(g,g')}\right]$$

first, making use of the fact that δ is arbitrarily small. We define two handy elements, namely

$$S_i^f(\delta) = [a_{i-1} + \delta f_{i-1}, a_i + \delta f_i),$$

which is the δf-extended version of $[a_{i-1}, a_i)$, and

$$\begin{aligned} C_i^f(\delta) &= \log(w_i + \delta f_i') - \log((a_i + \delta f_i) - (a_{i-1} - \delta f_{i-1})) \\ &\approx \log \frac{w_i}{a_i - a_{i-1}} + \frac{\delta f'_i}{w_i} - \frac{\delta(f_i - f_{i-1})}{a_i - a_{i-1}}, \end{aligned} \tag{4.5.2}$$

The $\delta(f, f')$-extended variant of $(\log w_i - \log a_i - a_{i-1})$. Here the approximation follows from a first order Taylor expansion and the fact that δ is small. Then

we have

$$\begin{aligned} & P\left[m_{(a^0,w^0)+\delta(h,h')}m_{(a^0,w^0)+\delta(g,g')}\right] \\ = & \sum_{i=1}^{k}\sum_{j=1}^{k} C_i^h(\delta)C_j^g(\delta)P(S_i^h(\delta)\cap S_j^g(\delta)), \end{aligned}$$

by the definitions of the$C$s and Ss, for any choice of g, h.

We first consider the case $g \neq h$. Note that when δ is small enough $S_i^g(\delta) \cap S_j^h(\delta) = \emptyset$ for all i, j except $i = j$, and $i = j+1, j = i+1$. Also, $S_i^h(\delta) \cap S_j^h(\delta) = S_i^g(\delta) \cap S_j^g(\delta) = \emptyset$ for all $i \neq j$. Using this fact, we calculate the term $\mathbb{P}\left[m_{(a,w)+\delta(h,h')}m_{(a,w)+\delta(g,g')}\right] = A_1 + A_2 + A_3$, where

$$\begin{aligned} A_1 &= \frac{1}{\delta}\sum_{i=1}^{k} C_{i+1}^g(\delta)C_i^h(\delta)P(S_{i+1}^g(\delta)\cap S_i^h(\delta)), \\ A_2 &= \frac{1}{\delta}\sum_{i=1}^{k} C_i^g(\delta)C_{i+1}^h(\delta)P(S_i^g(\delta)\cap S_{i+1}^h(\delta)), \\ A_3 &= \frac{1}{\delta}\sum_{i=1}^{k} C_i^g(\delta)C_i^h(\delta)P(S_i^g(\delta)\cap S_i^h(\delta)). \end{aligned}$$

The probabilities in the first sum are easily found to be $P[a_i + \delta g_i, a_i + \delta h_i) = f(a_i)\delta(h_i - g_i)$ if $h_i \geq g_i$, or 0 if $h_i < g_i$, while for the second sum we get $f(a_i)\delta(g_i - h_i)$ if $h_i \leq g_i$ or 0 if $g_i < h_i$. Since these conditions can't be true at the same time, we combine the sums, use (4.5.2), and pass $\delta \to 0$ to get

$$\lim_{\delta \searrow 0}\frac{1}{\delta}(A_1 + A_2) = \sum_{i=1}^{k} f(a_i)\log\frac{w_i}{a_i - a_{i-1}}\log\frac{w_{i+1}}{a_{i+1} - a_i}|h_i - g_i|. \tag{4.5.3}$$

Similarly, the third sum is found to be

$$\begin{aligned} A_3 = \sum_{i=1}^{k}[C_i^f(\delta)C_i^g(\delta)(P([a_{i-1}, a_i)) \\ + \delta\left[f(a_i)\min(g_i, h_i) - f(a_{i-1})\max(g_{i-1}, h_{i-1})\right])]. \end{aligned}$$

Now we consider the case when $h = g$. In this case the cross terms corresponding to A_1 and A_2 equals 0 and we're left with the natural analogues of A_3,

$$\begin{aligned} A_4 &= \sum C_i^g(\delta)C_i^g(\delta)(P([a_{i-1},a_i)) + \delta\left[f(a_i)g_i - f(a_{i-1})g_{i-1}\right]), \\ A_5 &= \sum C_i^h(\delta)C_i^h(\delta)(P([a_{i-1},a_i)) + \delta\left[f(a_i)h_i - f(a_{i-1})h_{i-1}\right]). \end{aligned}$$

Notice that $C_i^g(\delta)C_i^h(\delta) = \left(\log\frac{w_i}{a_i-a_{i-1}}\right)^2 + O(\delta)$, for any h, g. Also, $x + y - 2\min(x,y) = 2\max(x,y) - x - y = |x-y|$, which can be applied on the sum

$$\begin{aligned} &[f(a_i)g_i - f(a_{i-1})g_{i-1}] + [f(a_i)h_i - f(a_{i-1})h_{i-1}] \\ &-2\left[f(a_i)\min(g_i,h_i) - f(a_{i-1})\max(g_{i-1},h_{i-1})\right] \end{aligned}$$

for each i to get $|h_i - g_i|f(a_i) + |h_{i-1} - g_{i-1}|f(a_{i-1})$, which we denote K_i. The other term to the left in A_3, A_4, A_5 is $P([a_{i-1}, a_i))$; when these terms are multiplied with the $\left(\log\frac{w_i}{a_i-a_{i-1}}\right)^2$, they cancel each other out. Furthermore, the terms $O(\delta)$ terms in $C_i^g(\delta)C_i^h(\delta) + C_i^h(\delta)C_i^h(\delta) + C_i^g(\delta)C_i^g(\delta)$ also cancel each other, and we're left with only $O(\delta^2)$ terms, which will get sent off to zero. Hence we get

$$\begin{aligned} &\lim_{\delta\searrow 0}\frac{1}{\delta}\left[A_4 + A_5 - 2A_3)\right] \\ =\ &\lim_{\delta\searrow 0}\left(\frac{1}{\delta}\sum_{i=1}^{k}[\left(\log\frac{w_i}{a_i-a_{i-1}}\right)^2 + O(\delta)]\delta K_i\right) \qquad (4.5.4) \\ =\ &\sum_{i=1}^{k} K_i\left(\log\frac{w_i}{a_i-a_{i-1}}\right)^2 \\ =\ &\sum_{i=1}^{k-1}|h_i-g_i|f(a_i)\left(\left(\log\frac{w_{i+1}}{a_{i+1}-a_i}\right)^2 + \left(\frac{w_i}{a_i-a_{i-1}}\right)^2\right). \end{aligned}$$

Now we take the difference $\lim_{\delta\searrow 0}\frac{1}{\delta}\left[A_4 + A_5 - 2A_3)\right] - \lim_{\delta\searrow 0}\frac{1}{\delta}2(A_1 + A_2)$, which equals

$$\sum_{i=1}^{k-1} f(a_i)|g_i - h_i|\left(\log\frac{w_{i+1}}{a_{i+1}-a_i} - \log\frac{w_i}{a_i-a_{i-1}}\right)^2.$$

It remains to show that this corresponds to $E(G(h) - G(g))^2$ for the Gaussian process described above. Its covariance kernel is given by

$$\sum_{i=1}^{k-1}\min(g_i,h_i)f(a_i)\left(\log\frac{w_{i+1}}{a_{i+1}-a_i} - \log\frac{w_i}{a_i-a_{i-1}}\right)^2,$$

by the property of Brownian motions that $\mathrm{Cov}(B(t), B(s)) = \sigma^2 \min(t, s)$ whenever B is a Brownian motion such that $B(t)$ has variance σ^2. From this it follows that

$$\begin{aligned} E(G(h) - G(g))^2 &= \mathrm{Var}G(h) + \mathrm{Var}G(g) - 2\mathrm{Cov}(G(g), G(h)) \\ &= \sum_{i=1}^{k} f(a_i)\,|\beta|\,(h_i + g_i - 2\min(h_i, g_i)) \\ &= \sum_{i=1}^{k-1} f(a_i)\beta^2 |h_i - g_i|. \end{aligned}$$

Notice that h' and g', the free variables corresponding the weights w_i, play no active role in the proof. Their role in the limit distribution is solely confined to the Hessian or information matrix V, considered in the next theorem. □

Remark 4.5.3. Notice the role of A5: It makes sure that

$$\sum_{i=1}^{k-1} f(q_i^0)|g_i - h_i| \left(\log \frac{w_{i+1}}{a_{i+1} - a_i} - \log \frac{w_i}{a_i - a_{i-1}} \right)^2$$

is non-zero for every choice of h and g. Assumption A4 is clearly necessary, as we can see in this example.

Example 4.5.4. (On assumption A4). Consider the case of $Beta(\alpha, \alpha)$, constant weights and $k = 2$. These distributions are symmetric, are unimodal whenever $\alpha \geq 1$ and have two "arms" on each side when $\alpha < 1$. Not surprisingly, there are two best approximating histograms when $\alpha < 1$ is sufficiently close to 0. For instance, when $\alpha = \frac{1}{10}$, there is one maximum in $a \approx 0.99995$ and one in $1 - a$. As can be seen in the Figure 4.5.1, not every Beta(α, α) with $\alpha < 1$ has this feature.

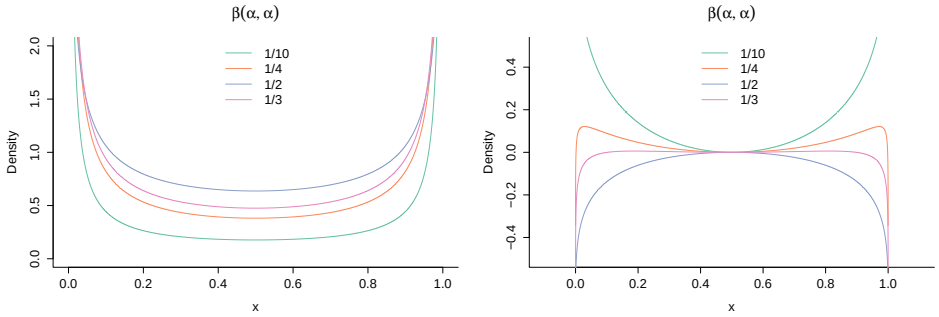

Figure 4.5.1: (left) Densities for Beta(α, α) for several choices of α. (right) The corresponding objective functions.

We proceed to find the information matrix V: Recall from the rate theorem on page 22 that the limiting distribution of $n^{\frac{1}{3}}(\widehat{(a,w)} - (a,w))$ us on the form $\arg\max_h(\frac{1}{1}h^T V g + G(h))$, where $G(h)$ is found in the previous theorem and V is the Hessian matrix of $Pm_{(a,w)}$ with respect to the parameter vector (a,w). The next proposition provides the elements of V for both the case of constant and variable weights.

Proposition 4.5.5. *The information matrix V has elements*

$$\begin{aligned} V_{i,i+1} &= \frac{f(a_i)+f(a_{i+1})}{a_{i+1}-a_i} - \frac{[F(a_{i+1})-F(a_i)]}{(a_{i+1}-a_i)^2}, \\ V_{i,i} &= f'(a_i)(\log\frac{w_i}{a_i-a_{i-1}} - \log\frac{w_{i+1}}{a_{i+1}-a_i}) \\ &\quad -2f(a_i)\left(\frac{1}{a_{i+1}-a_i}+\frac{1}{a_i-a_{i-1}}\right) \\ &\quad +\frac{[F(a_{i+1})-F(a_i)]}{(a_{i+1}-a_i)^2} + \frac{[F(a_i)-F(a_{i-1})]}{(a_i-a_{i-1})^2}, \end{aligned}$$

whenever i is less than k (these correspond to the split points). For $i \neq k-1$, we have

$$\begin{aligned} V_{i,i+k-1} &= \frac{f(a_i)}{w_i}, \\ V_{i-1,i+k-1} &= \frac{f(a_{i-1})}{w_i}, \\ V_{k-1,i+k-1} &= \frac{f(a_{k-1})}{1-w_k}, \end{aligned}$$

while $i = k-1$ gives us

$$\begin{aligned} V_{k-1,2k-2} &= \frac{f(a_i)}{w_i} + \frac{f(a_{k-1})}{w_k}, \\ V_{k-2,2k-2} &= \frac{f(a_{i-1})}{w_i}. \end{aligned}$$

Here the conventions $a_0 = 0$ and $a_k = 1$ are used. All the other elements are 0.

Proof. For the elements only involving the split points, observe that the only part of the sum involving a_i is

$$v = [F(a_i) - F(a_{i-1})] (\log w_i - \log (a_i - a_{i-1})) \\ + [F(a_{i+1}) - F(a_i)] (\log w_{i+1} - \log (a_{i+1} - a_i)).$$

Differentiation gives us

$$\begin{aligned} \frac{dv}{da_i} &= f(a_i) (\log w_i - \log (a_i - a_{i-1})) - \frac{[F(a_i) - F(a_{i-1})]}{a_i - a_{i-1}} \\ &\quad -f(a_i) (\log w_{i+1} - \log (a_{i+1} - a_i)) + \frac{[F(a_{i+1}) - F(a_i)]}{a_{i+1} - a_i}. \end{aligned}$$

Hence

$$\frac{dv}{da_i a_{i+1}} = \frac{f(a_i) + f(a_{i+1})}{a_{i+1} - a_i} - \frac{[F(a_{i+1}) - F(a_i)]}{(a_{i+1} - a_i)^2}$$

while

$$\begin{aligned} \frac{dv}{da_i da_i} &= f'(a_i)(\log w_i - \log w_{i+1} - \log(a_i - a_{i-1}) + \log (a_{i+1} - a_i)) \\ &\quad -2f(a_i) \left(\frac{1}{a_{i+1} - a_i} + \frac{1}{a_i - a_{i-1}} \right), \\ &\quad + \frac{[F(a_{i+1}) - F(a_i)]}{(a_{i+1} - a_i)^2} + \frac{[F(a_i) - F(a_{i-1})]}{(a_i - a_{i-1})^2}. \end{aligned}$$

Concerning i', recall that $w_k = \sum_{i=1}^{k} w_i$, and that w_k only appears in

$$[1 - F(a_{k-1})] (\log w_k - \log (1 - a_{k-1})).$$

This means that w_i also cares about the contribution it gives to the last box, an issue not faced by the a_is. it is clear that all elements involving i' are 0 except

$$\frac{d}{dw_i dw_i} = - \left(\frac{1}{F(a_i) - F(a_{i-1})} + \frac{1}{1 - F(a_{k-1})} \right),$$

where $\frac{1}{1-F(a_{k-1})}$ appears due to the additional contribution from w_k. Also, when $i \neq k - 1$, we have

$$\begin{aligned}\frac{d}{dw_i da_i} &= \frac{f(a_i)}{w_i},\\ \frac{d}{dw_i da_{i-1}} &= \frac{f(a_{i-1})}{w_i},\\ \frac{d}{dw_i da_{k-1}} &= \frac{f(a_{k-1})}{w_k}.\end{aligned}$$

For $i \neq k-1$, and

$$\frac{d}{dw_i da_i} = \frac{f(a_i)}{w_i} + \frac{f(a_i)}{1-w_i},$$

when $i = k-1$. □

Combining these two results with the rate theorem, we get this thesis' main result: The limiting distribution of the irregular Kullback-Leibler histogram with variable and constant weights. Looking at the rate theorem, the only conditions left are those concerning a "suitable envelope M_δ", namely the Lindeberg condition $\lim_{\delta \searrow 0} \frac{P^\star M_\delta^2 \{M_\delta > \eta\delta^{-2}\phi^2(\delta)\}}{\phi^2(\delta)} = 0$ together with the condition $\phi^2(\delta) \geq P^\star M_\delta^2$, which we ignore. Both of these can probably be shown by using similar calculations as in Theorem 4.5.2, but this would be laborious and not very enlightening. We assume these two conditions are satisfied in the following.

Theorem 4.5.6. *Assume A1-A5. The rescaled process* $n^{\frac{2}{3}}(P_n m_{(a,w)+n^{-\frac{1}{3}}\widehat{(a,w)}} - P_n m_{(a,w)})$ *converges in distribution to* $\frac{1}{2}h^T V h + G(h)$*, where* $G(h)$ *is the Gaussian process*

$$G(h) = \sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \log \frac{w_{i+1}}{a_{i+1}-a_i} - \log \frac{w_i}{a_i - a_{i-1}} \right| W_i(h_i),$$

with $w_i = P([a_{i-1}, a_i))$ *in the case of variable weights, and the predefined* w_i *otherwise, and* V *is the Hessian of* $Pm_{(a,w)}$ *at* (a,w) *The maximum likelihood estimate converges with rate* $n^{\frac{1}{3}}$ *and its limiting distribution is given by*

$$n^{\frac{1}{3}}(\widehat{(a,w)} - (a,w)) \xrightarrow{d} \arg\max_h \left[\frac{1}{2} h^T V h + G(h) \right],$$

where the elements of V *are given in Proposition 4.5.5.*

Example 4.5.7. We begin by studying the simplest case of $k = 2$ and constant weights; in this case there is only a single true a_0. From Proposition 4.5.5 we

Table 4.2: Convergence of the sample distributions for the split point. The approximate coordinate search (see Section 4.6) has been used, but the exact algorithm gives similar results for the small ns.

	50	100	500	1000	10000	100000
Beta$(1, \frac{1}{2})$	0.255	0.074	0.079	0.068	0.031	0.010
Beta$(2, 7)$	0.124	0.103	0.066	0.057	0.040	0.043
$lN(0, 1)$	0.049	0.036	0.018	0.011	0.006	0.003

find that

$$V = f'(a)(\log(1-a) - \log a) - 2f(a)\left(\frac{1}{1-a} + \frac{1}{a}\right) + \frac{1-F(a)}{(1-a)^2} + \frac{F(a)}{a^2},$$

while $G(h) = f(a)^{\frac{1}{2}} |\log a - \log(1-a)| W(h) = dW(h)$, and the limiting distribution is $\arg\max_h \frac{1}{2}Vh^2 + G(h)$. Using Proposition 2.2.4 from the section on Chernoff's mode estimator, we find the equivalent formulation $|\frac{2d}{V}|^{-\frac{2}{3}}Z$, where $Z = \arg\max_h [W(h) - h^2]$ is Chernoff's distribution.

Next we will use this result in some simulations. Using the results in Groeneboom and Wellner (2001, p.9 - 13, table 1 and 3), we can obtain almost exact quantiles for this distribution, allowing us to construct approximate confidence intervals and check how well the limit distribution approximates the actual distributions for different n. However, this will not work for any choice of distribution F, as the table in Groeneboom and Wellner doesn't contain enough values for the calculation of arbitrary rescalings. Due to this we do one large simulation of the Chernoff distribution with step size $\delta = 0.001$ with 600 000 replications. Our goal is to understand how fast the finite sample distribution of $n^{\frac{1}{3}}(\widehat{q_0} - q_0)$ converges to the stated limiting distribution.

Example 4.5.8. The *Kolmogorov distance* between two distributions F and G is defined by $d_K(F, G) = \sup_x |F(x) - G(x)|$, which corresponds to the greatest vertical distance between the two distributions when they are plotted in the same window. We will use this function to measure the distance between simulated finite sample distributions and the real distribution. Our choices of test distributions are Beta$(1, \frac{1}{2})$ and Beta$(2, 7)$, which have slightly different shapes. For comparison's sake, we perform the same experiment with the mean of $X_1, ..., X_n$ standard log normal variables, which is understood as a case when the CLT is relatively slow working. We perform the experiment for $n = 50, 100, 500, 1000, 10000, 100000$.

Even the fastest converging distributions are very slow compared to the mean of log-normals. We supply some plots of what's going on in Figure 4.5.2.

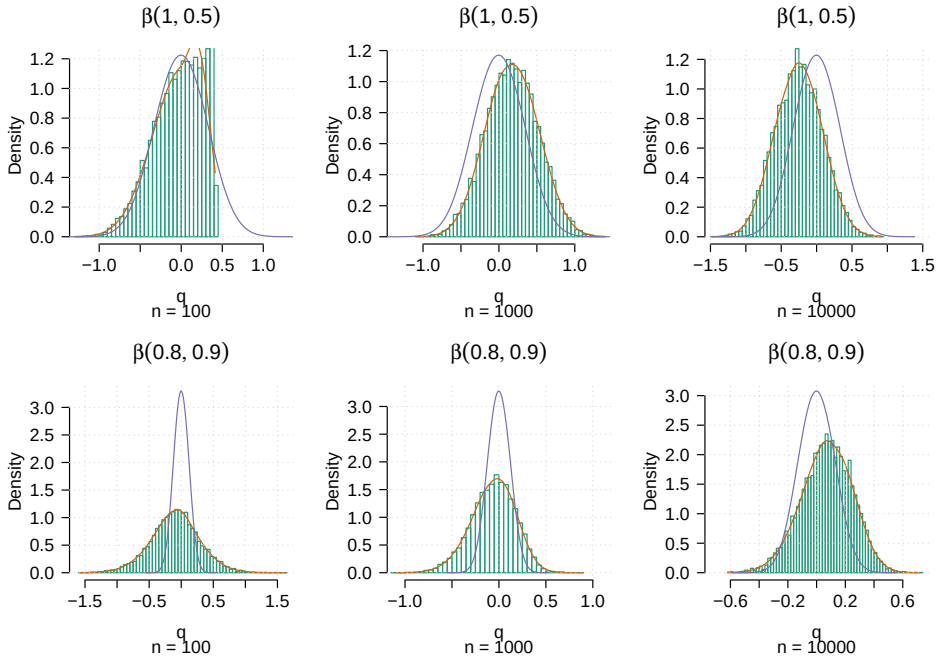

Figure 4.5.2: Plot of simulated split points with KDE from the `locfit` package (Loader, 2013) in orange, the limiting distributions are purple. it is clear that the limit distribution will not work well as an approximation.

Next we study the simplest example of irregular histograms.

Example 4.5.9. We study the special case of $k = 2$ for KL weights. The Hessian V is

$$V = \begin{bmatrix} c_1 & c_2 \\ c_2 & c_3 \end{bmatrix}$$

where

$$\begin{aligned} c_1 &= f'(a)(\log \frac{F(a)}{1-F(a)} - \log \frac{a}{1-a}) - 2f(a)\left(\frac{1}{1-a} + \frac{1}{a}\right) + \frac{1-F(a)}{(1-a)^2} + \frac{F(a)}{a^2}, \\ c_2 &= f(a)(\frac{1}{F(a)} + \frac{1}{1-F(a)}), \\ c_3 &= -(\frac{1}{F(a)} + \frac{1}{1-F(a)}). \end{aligned}$$

The zero-mean Gaussian process is

$$G(h_1) = dW(h_1),$$

where $d = f(a)^{\frac{1}{2}} \left|\log \frac{F(a)}{1-F(a)} - \log \frac{a}{1-a}\right|$. The estimator is $\arg\max_h \frac{1}{2}h^T V h + dW(h_1)$. As in the previous example, this can can be simplified. Use $\frac{1}{2}h^T V h =$

$\frac{1}{2}(c_1h_1^2 + 2c_2h_1h_2 + c_3h_2^2)$ and assume h_1 is known. Then the maximum of h_2 can be found by differentiating the preceding quadratic function, which yields $h_2 = -\frac{eh_1}{2c_3}$. At this point, $\frac{1}{2}h^TVh = \frac{1}{2}(ch_1^2 - \frac{c_2^2h_1^2}{c_3} + \frac{c_2^2h_1^2}{4c_3}) = h_1^2 \cdot \frac{1}{2}(c_1 - \frac{c_2^2}{c_3} + \frac{c_2^2}{4c_3})$. It follows that whenever h_1 maximises $G(h_1) + h_1^2 \cdot \frac{1}{2}(c_1 - \frac{c_2^2}{c_3} + \frac{c_2^2}{4c_3})$, the maximiser of $\frac{1}{2}h^TVh + G(h_1)$ is $(h_1, -\frac{c_2h_1}{2c_3})$. Use the Proposition 2.2.4 to simplify the maximiser of $G(h_1) + h_1^2 \cdot \frac{1}{2}(c - \frac{c_2^2}{c_3} + \frac{c_2^2}{4c_3})$ further, into $(\frac{2d}{(c_1 - \frac{c_2^2}{c_3} + \frac{c_2^2}{4c_3})})^{\frac{2}{3}}Z$, where Z is Chernoff's distribution. We conclude with $n^{\frac{1}{3}}(\widehat{(a,w)} - (a,w)) \xrightarrow{d} (1, -\frac{c_2}{2c_3})\left(\frac{dc_2^2}{c_1 - 3c_2^2}\right)^{\frac{2}{3}}Z$. This representation can be used together with the tables from Groeneboom and Wellner (2001, p.9 - 13, table 1 and 3) or home made simulations to calculate the limit distribution, as in the previous example. Moreover, it allows us to compute quantities like the asymptotic covariance between the weight and the split point.

Remark 4.5.10. This is analogous to 2.4.1 on decision trees, where $n^{\frac{1}{3}}((\widehat{\beta}_l, \widehat{\beta}_u, \widehat{d}) - (\beta_l^0, \beta_u^0, d^0)) \xrightarrow{d} \arg\max_t(c_1, c_2, 1)G(t)$, for some c_1, c_2 and a function $G(t) = a't^2 + b'W(t)$, where $W(t)$ is a standard Brownian motion starting at 0. The main difference between this regression case and our histograms is that our analogue to a_1, namely w, completely determines our analogue to a_2, namely $1 - w$.

4.5.2 A special case with $\sqrt{n}$-consistency

Recall assumption A5': $\frac{w_{i+1}}{a_{i+1} - a_i} = \frac{w_i}{a_i - a_{i-1}}$ for each i. We will translate this condition into something more understandable. First we notice that $\frac{w_j}{a_j - a_{j-1}} = \frac{w_i}{a_i - a_{i-1}}$ for each j by induction. Now we claim that all of these are equal to 1.

Proposition 4.5.11. *If* $\frac{w_{i+1}}{a_{i+1} - a_i} = \frac{w_i}{a_i - a_{i-1}}$ *for every* i*, then* $\frac{w_i}{a_i - a_{i-1}} = 1$ *for every* i *as well.*

Proof. First assume $k = 2$ and recall that $w_2 = 1 - w_1$. Again we use the convention $a_0 = 0$ and $a_k = 1$. Since $\log \frac{w_{i+1}}{a_{i+1} - a_i} = \log \frac{w_i}{a_i - a_{i-1}}$ for each i, we have that $\frac{w_1}{a_1} = \frac{1 - w_1}{1 - a_1}$. Solving this equation yields $w_1 = a_1$, and $\frac{w_i}{a_i - a_{i-1}} = 1$ as claimed. For $k \geq 2$, we use induction. If it is true for k, we consider the distribution $F' = a_k^{-1}F(a_k^{-1}x)$. The rescaled histogram associated with this distribution has weights $\frac{w_i}{a_k}$ and splits $\frac{a_i}{a_k}$, hence every $\frac{w_i/a_k}{(a_i - a_{i-1})/a_k} = \frac{w_i}{(a_i - a_{i-1})} = 1$ for $i = 1, ..., (k-1)$ by the induction hypothesis. We can do this the other way around, starting from a_1, in order to get $\frac{w_k}{a_k - a_{k-1}} = 1$ as well. □

It follows that Theorem 4.5.6 can't be used in the case when each block in the histogram has equal probability as length. This can happen for instance

when the underlying distribution is symmetric and unimodal with $k = 2$ and when the underlying distribution is uniform for any k.

We go on to find the shape of the limiting Gaussian process when $\frac{w_{i+1}}{a_{i+1}-a_i} = \frac{w_i}{a_i-a_{i-1}}$ for every i. In this case, we will have to use the function $\phi(\delta) = \sqrt{\delta}$ in order to find a non-degenerate Gaussian process in the rate theorem (page 22).

Theorem 4.5.12. *Assume conditions (A1-A3,A4,A5'). In that case, we have*

$$\lim_{\delta \searrow 0} \frac{P(m_{\theta_0+\delta g} - m_{\theta_0+\delta h})^2}{\delta^2} = \sum_{i=1}^{k} P([a_{i-1}, a_i))\gamma_i^2,$$

where

$$\gamma_i = \frac{h'_i - g'_i}{w_i} - \frac{(g_i - h_i) - (g_{i-1} - h_{i-1})}{a_i - a_{i-1}}.$$

Thus condition 2.2.5 on page 22 is satisfied for $\phi(\delta) = \delta$. Condition 2.2.6 is satisfied with $G(h) = h^T Z$ for $Z \sim N(0, \Sigma)$, where Σ is a symmetric matrix with upper elements $\sigma_{i,i} = \frac{P(I_i)}{(a_i-a_{i-1})^2} + \frac{P(I_{i+1})}{(a_{i+1}-a_i)^2}$, $\sigma_{i,i+1} = -\frac{P(I_{i+1})}{(a_{i+1}-a_i)^2}$, $\sigma_{i,i'} = \frac{P(I_i)}{w_i(a_i-a_{i-1})}$ $\sigma_{i,(i+1)'} = -\frac{P(I_{i+1})}{w_{i+1}(a_{i+1}-a_i)}$ $\sigma_{i',i'} = \frac{1}{w_i^2}$. Here the primed elements correspond to variable weights; they can be disregarded when using constant weights. Then

$$\sqrt{n}(\widehat{(a, w)} - (a, w)) \xrightarrow{d} N(0, V^{-1}\Sigma V^{-1}),$$

where V is given in Proposition 4.5.5.

Proof. This is a slight modification of the proof in Theorem 4.5.2 on page 99: Instead of finding the terms involving δ, we will search out the terms involving δ^2. In A_1 and A_2, the only terms involving δ^2 also involved $\log \frac{w_j}{a_j-a_{j-1}}$ for some j, but these are equal to 0 by the previous proposition. Similarly, all terms of A_3, A_4 and A_5 containing $\log \frac{w_j}{a_j-a_{j-1}}$ can be disregarded. This leaves the terms containing δ^2 in $\sum_{i=1}^{k} P([a_{i-1}, a_i)) \left[C_i^g(\delta)C_i^h(\delta) + C_i^h(\delta)C_i^h(\delta) - 2C_i^g(\delta)C_i^g(\delta)\right]$:

$$\begin{aligned}
& \sum_{i=1}^{k} P([a_{i-1}, a_i)) \left(\frac{\delta h'_i}{w_i} - \frac{\delta(h_i - h_{i-1})}{a_i - a_{i-1}}\right)^2 \\
+ & \sum_{i=1}^{k} P([a_{i-1}, a_i)) \left(\frac{\delta g'_i}{w_i} - \frac{\delta(g_i - g_{i-1})}{a_i - a_{i-1}}\right)^2 \\
- & 2\sum_{i=1}^{k} P([a_{i-1}, a_i)) \left(\frac{\delta h'_i}{w_i} - \frac{\delta(h_i - h_{i-1})}{a_i - a_{i-1}}\right)\left(\frac{\delta g'_i}{w_i} - \frac{\delta(g_i - g_{i-1})}{a_i - a_{i-1}}\right).
\end{aligned}$$

which clearly simplifies to

$$\delta^2 \sum_{i=1}^{k} P([a_{i-1}, a_i)) \left(\frac{h'_i - g'_i}{w_i} - \frac{(g_i - h_i) - (g_{i-1} - h_{i-1})}{a_i - a_{i-1}} \right)^2 .$$

The covariance kernel of the limiting Gaussian process is

$$E(G(h)G(g)) = \sum_{i=1}^{k} P([a_{i-1}, a_i)) \left(\frac{\delta h'_i}{w_i} - \frac{\delta(h_i - h_{i-1})}{a_i - a_{i-1}} \right) \left(\frac{\delta g'_i}{w_i} - \frac{\delta(g_i - g_{i-1})}{a_i - a_{i-1}} \right).$$

Let $G(h) = h^T Z$, where $Z \sim N(0, \Sigma)$ with Σ as in the statement of the theorem. Then it can be verified that this is the right process by calculating $E(h^T Z(g^T Z)) = h^T \Sigma g$. We write Σg in the form of linear combinations of column vectors:

$$\begin{aligned} h^T \Sigma g &= (h_1, ...h_{k-1}, h'_1, ...h'_{k-1})(\Sigma_1 g_1, ...\Sigma_{k-1} g_{k-1}, \Sigma_k g'_1, ...\Sigma_{2k-2} g'_{k-1}) \\ &= \sum_{i=1}^{k} P(I_i) \left(\frac{h'_i}{w_i} - \frac{(h_i - h_{i-1})}{a_i - a_{i-1}} \right) \left(\frac{g'_i}{w_i} - \frac{(g_i - g_{i-1})}{a_i - a_{i-1}} \right). \end{aligned}$$

To find this Σ we rewrite the sum slightly:

$$\begin{aligned} & \sum_{i=1}^{k-1} g_i (-h_{i-1} \frac{P(I_i)}{(a_i - a_{i-1})^2} + h_i \left(\frac{P(I_i)}{(a_i - a_{i-1})^2} + \frac{P(I_{i+1})}{(a_{i+1} - a_i)^2} \right) \\ & -h_{i+1} \frac{P(I_{i+1})}{(a_{i+1} - a_i)^2} + \frac{P(I_i) h'_i}{w_i (a_i - a_{i-1})} - \frac{P(I_{i+1}) h'_{i+1}}{w_{i+1} (a_{i+1} - a_i)}) \\ + & \sum_{i=1}^{k-1} g'_i \left[\frac{h'_i}{w_i^2} + \frac{P(I_i) h_{i-1}}{w_i (a_{i-1} - a_i)} - \frac{P(I_i) h_i}{w_i (a_i - a_{i-1})} \right], \end{aligned}$$

where $h_0 = g_0 = h_k = g_k = 0$. This gives us the covariance matrix with elements described above. The limiting distribution of $\sqrt{n}(\widehat{(a, w)} - (a, w))$ is the maximiser of $\frac{1}{2} h^T V h + h^T Z$. Differentiation with respect to h gives $Vh + Z = 0$, hence $\widehat{h} = V^{-1} Z \sim N(0, V^{-1} \Sigma V^{-1})$ as claimed. □

Example 4.5.13. Let $F = \mathcal{U}(0, 1)$ and $k \geq 2$. If constant weights are used, each $a_i - a_{i-1} = \frac{1}{k}$ and $P(I_i) = \frac{1}{k}$, hence $\sigma_{i,i-1} = \sigma_{i,i+1} = -k$ and $\sigma_{i,i} = 2k$. As for V, $v_{ii} = -2k$ and $V_{i,i+1} = V_{i,i-1} = k$. Hence the maximiser is $N_k(0, V^{-1} \Sigma V^{-1}) = N_k(0, V^{-1})$-distributed. It can be shown by recursion that $V_{ij}^{-1} = \frac{(k-j)i}{k^2}$ when $j \geq i$. Recall Proposition 4.3.1 on the asymptotic distribution of quantiles, where it is stated that $\sqrt{n}(\widehat{q} - q) \xrightarrow{d} N(0, \Sigma)$ where $\Sigma_{ij} = \frac{(k-j)i}{k^2 f(q_{(i)}) f(q_{(j)})} = \frac{(k-j)i}{k^2}$. It follows that these two approaches are asymptotically equivalent in the context

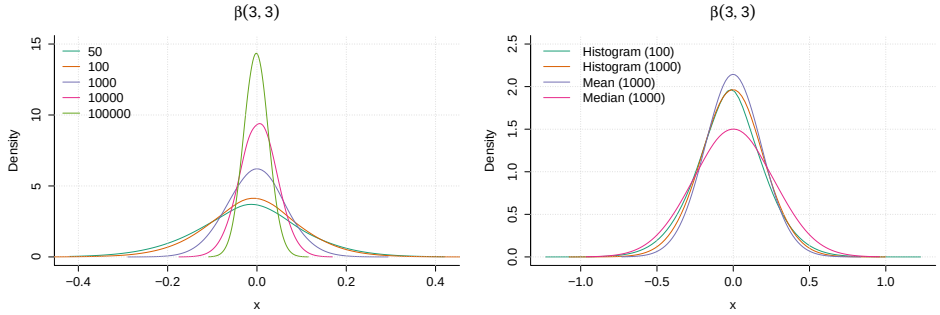

Figure 4.5.3: Example of $n^{\frac{1}{3}}$-convergence to a degenerate distribution, $F = \text{Beta}(3,3)$. However, it is non-degenerate for the $\sqrt{n}$-rate.

of the uniform distribution. Histograms with variable weights will not be unique in this setting.

Clearly, it would be silly to use the KL-histogram with constant weights in order to estimate the quantiles of a uniform distribution. This is a bad idea for several reasons (of descending importance): 1.) If we know that the underlying distribution is in fact uniform, it would be far wiser and more efficient to use $k^{-1}, 2k^{-1}, ..., (k-1)k^{-1}$ as the quantiles. 2.) This only works if the underlying distribution is, in fact, uniform on $[0,1]$. If it isn't, the split points will not converge to the quantiles. 3.) Following the theme of slow convergence, the covariance matrix of the split points takes a very long time to come appreciably close to the covariance matrix of the quantiles. we will investigate this in the next example.

Another example is the case of a symmetric distributions for $k = 2$. We tacitly assume the existence of a unique, well-separated maximum, hence $a = 0.5$ by symmetry. The maximum is typically not unique when we use variable weight histograms, hence we use constant weight histograms in the next example.

Example 4.5.14. Since $k = 2$, the matrix Σ simplifies to $\frac{1/2}{(1/2)^2} + \frac{1/2}{(1/2)^2} = 4$, hence the asymptotic variance is $4V^{-2} = \frac{1}{4}(2f(0.5) - 1)^{-2}$. An illustration of the rate of convergence for $\text{Beta}(3,3)$ can be seen in Figure 4.5.3. As seen in Figure 4.5.4 on the next page, the convergence (to 0.5) is remarkably slow, with the practical consequence that the variance is larger than this for finite sample sizes. The decay of the ratio between the empirical and limiting variance appears to be of the form $n^{-\frac{1}{3}}$.

Example 4.5.15.

Theorem 4.5.12 on page 109 illustrates the power of the rate theorem. It is relatively easy to notice how the special case behaves, as all the information

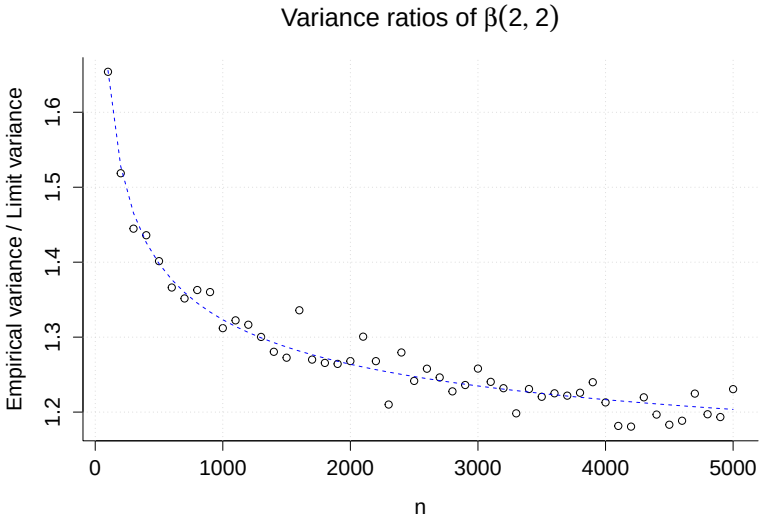

Figure 4.5.4: Plot of the empirical variances divided by the limit variance for the $\sqrt{n}$-rescaled split points for the $k = 2$ quantile histogram with $F = \text{Beta}(2, 2)$, $n = 100, 200, ..., 4900, 5000$. The blue line is fitted by ordinary linear regression, $R = \beta_0 + \beta_1 n^{-\frac{1}{3}}$.

about the convergence rate lies in which function $\phi^2(\delta)$ we can use to divide the function $P(m_{\theta_0+\delta g} - m_{\theta_0+\delta h})^2$ with and still have it "under control". It is also interesting that the symmetry condition on the $\frac{w_i}{a_{i+1}-a_i}$s has such a profound effect on the convergence rate. Still, the convergence to the limiting distribution is very slow.

4.5.3 L_2-histograms

The methods and results for L_2 histograms are very similar to those of KL-histograms, and we are content with simply stating the analogues of the two main theorems (4.5.2, 4.5.6) on Kullback-Leibler histograms.

Theorem 4.5.16. *Let* $m_{(a,w)} = \sum_{i=1}^{k} \left[\frac{2w_i}{a_i - a_{i-1}} 1_{[a_{i-1},a_i)}(x) - \frac{w_i^2}{a_i - a_{i-1}} \right]$ *be the objective associated with the* L_2 *histogram with either variable or constant weights* w_j*. Denote the least false vectors by* a *and* w*, and let* f *be the density of* P*. Provided the conditions (A1-A5) are satisfied and* $f \in L_2$*, the following is true:*

$$\lim_{\delta \searrow 0} \frac{\mathbb{P}\left(m_{(a,w)+\delta(g,g')} - m_{(a,w)+\delta(h,h')}\right)^2}{\delta} = 4 \sum_{i=1}^{k-1} f(a_i) \left(\frac{w_{i+1}}{a_{i+1} - a_i} - \frac{w_i}{a_i - a_{i-1}} \right)^2,$$

thus condition 2.2.5 on page 22 is satisfied for $\phi(\delta) = \sqrt{\delta}$*. Furthermore, condition (2.2.6) is satisfied with*

$$G(h) = 2\sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \frac{w_{i+1}}{a_{i+1} - a_i} - \frac{w_i}{a_i - a_{i-1}} \right| W_i(h_i) \tag{4.5.5}$$

where W_i are independent standard Brownian motions.

The factor 2 in $G(h)$ is a side-effect of the definition of $m^{L_2}_{(a,w)}$; it wouldn't have been there if the we had opted to divide the criterion function by 2. Evidently, this only has an effect on the size of the limit process, not on the actual argmax, but might be considered if one wants the limiting processes of the Kullback-Leibler histogram and the L_2-histogram to look more alike.

Theorem 4.5.17. *Assume conditions A1-A5 are satisfied. The rescaled process $n^{\frac{2}{3}}(P_n m_{(a,w)+n^{-\frac{1}{3}}\widehat{(a,w)}} - P_n m_{(a,w)})$ converges in distribution to $\frac{1}{2}h^T V h + G(h)$, where $G(h)$ is the Gaussian process*

$$G(h) = 2\sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \frac{w_{i+1}}{a_{i+1} - a_i} - \frac{w_i}{a_i - a_{i-1}} \right| W_i(h_i),$$

with $w_i = P([a_{i-1}, a_i))$ in the case of variable weights, and the predefined w_i otherwise, and V is the Hessian of $Pm_{(a,w)}$ at (a, w). The minimum L_2 estimate converges with rate $n^{\frac{1}{3}}$ and its limiting distribution is given by

$$n^{\frac{1}{3}}(\widehat{(a,w)} - (a,w)) \xrightarrow{d} \arg\max_h \frac{1}{2}h^T V h + G(h),$$

where V is the Hessian of $Pm_{(a,w)}$.

Example 4.5.18. This is the L_2-analogue of Example 4.5.7, where we study the constant weight histogram with $k = 2$. The limit objective is$Pm_a = \frac{1}{a}(F(a) - \frac{1}{4}) + \frac{1}{1-a}(\frac{3}{4} - F(a))$, whose first derivative with respect to a is

$$f(a)\left[\frac{1}{a} - \frac{1}{1-a}\right] - F(a)\left[\frac{1}{(1-a)^2} + \frac{1}{a^2}\right] - 4\left[\frac{3}{(1-a)^2} + \frac{1}{a^2}\right],$$

and the second derivative is

$$V = f'(a)\left[\frac{1}{a} - \frac{1}{1-a}\right] - 2f(a)\left[\frac{1}{(1-a)^2} + \frac{1}{a^2}\right] + \frac{2(F(a) - \frac{1}{4})}{a^3} + \frac{2(\frac{3}{4} - F(a)}{(1-a)^3}.$$

Put $G(h) = f(a)^{\frac{1}{2}}\left|\frac{1}{a} - \frac{1}{1-a}\right| W(h) = dW(h)$, and the limiting distribution is $\arg\max_h \frac{1}{2}Vh^2 + dW(h)$. Again using Proposition 2.2.4, we find that $n^{\frac{1}{3}}(\widehat{a} - a) \xrightarrow{d} \frac{2d}{V}^{-\frac{2}{3}} Z$, where $Z = \arg\max_h W(h) - h^2$ is Chernoff's distribution.

4.6 Algorithms

Many, if not most, optimisation problems in statistics can be solved by numerical packages tailored toward solving problems with smooth objective functions. The prototypical example is using Newton-Raphson to solve the maximum likelihood problem for a gamma distributed random variable. Using such procedures works very well, provided the objective function is sufficiently smooth and concave. In the case of irregular histograms, we have neither — our objective function has many local maxima and has jump discontinuities all over the place. Instead, we need combinatorial optimisation.

We describe three algorithms:

1. An exact algorithm based on *dynamic programming* (see e.g. Weiss (1998, chap. 10)). This is essentially the same as the algorithm described in Rozenholc et al. (2010) (first in Kanazawa (1988)). Our new contribution is to solve the problem for all combinations of Kullback-Leibler, L_2, and variable or equal weights, while Rozenholc et al. only considered Kullback-Leibler irregular histograms with variable weights. Unfortunately, this algorithm runs in approximately quadratic time, namely $O(n^2 k)$.

2. A coordinate search algorithm. This usually runs linearly in n, and has worst case complexity bounded by $O(n \log n)$ in our implementation. This is an approximation algorithm, but simulation studies indicate that its performance is satisfactory, which leads us to recommend it above the dynamic programming algorithm. When pre-smoothing, bootstrapping or running simulation studies, the gain in processing time is significant. Unfortunately, it doesn't perform well for the CIC of Section 4.9 on page 128.

3. As was the case for Manski's estimator, we can recast the problem as a mixed integer programming problem. We will not make use of this formulation.

The first two algorithms are implemented in `C++`, with code in Appendix A, and made available in `R` through the `Rcpp`-package (Eddelbuettel and François, 2011). Let $\{x_1, x_2, ..., x_n\}$ be the set of observations. We take note of this important fact,

Proposition 4.6.1. *For any irregular histogram, the solution set is a subset of* $\{x_1, x_2, ..., x_n\}$*, provided the probabilities* $P_n[x_j, x_i)$ *involved in the sum are substituted with* $P_n[x_j.x_i]$*.*

Table 4.3: Definitions of R for different choices of histogram methods. Recall that the objective function for the L_2-histogram is $\sum_{i=1}^{k} w_i \frac{(2P(a_{i-1},a_i)-w_i)}{a_{i-1}-a_i}$. We remove a redundant k^{-1} for the constant weights and contract $2P(a_{i-1},a_i) - w_i = P(a_{i-1},a_i)$ for the variable weights.

	Equal weights	KL/L_2 weights
KL	$\log \frac{k^{-1}}{x_i-x_j} P_n[x_j,x_i)$	$\log \frac{P_n[x_j,x_i)}{x_i-x_j} P_n[x_j,x_i)$
L_2	$\frac{2P_n[x_j,x_i)-k^{-1}}{x_i-x_j}$	$\frac{P_n[x_j,x_i)}{x_i-x_j} P_n[x_j,x_i)$

Proof. Assume $a_1, ..., a_{k-1}$ are the solutions, and put $a_0 = 0, a_k = 1$. Choose a j, and notice that the only part of the objective function that depends on a_j is

$$R(a_{j-1}, a_j) + R(a_j, a_{j+1}),$$

where R is chosen from table 4.3 of histogram weights. Hence it suffices to show that this is maximised for an $x \in \{x_1, x_2, ..., x_n\}$. Let x_l and x_u be the greatest lower bound and least upper bound of a_j in $\{x_1, x_2, ..., x_n\}$, respectively, so we know that $a_j \in [x_l, x_u]$. Assume $R(x_j, x_i) = \log \frac{P_n[x_j,x_i)}{x_i-x_j} P_n[x_j, x_i)]$ for concreteness, and notice that

$$\begin{aligned} & R(a_{j-1}, a_j) + R(a_j, a_{j+1}) \\ = \; & P_n[a_{j-1}, x_l] \log \frac{P_n[a_{j-1}, x_l]}{a_j - a_{j-1}} + P_n[x_u, a_{j+1}] \log \frac{P_n[x_u, a_{j+1}]}{a_{j+1} - a_j}, \end{aligned}$$

hence we reduce the problem to the maximisation of

$$g(a_j) = -p_1 \log(a_j - a_{j-1}) - p_2 \log(a_{j+1} - a_j), \tag{4.6.1}$$

where $p_1 = P_n[a_{j-1}, x_l]$ and $p_2 = P_n[x_u, a_{j+1}]$. For simplicity, extend the candidate solution set to $[x_l, x_u]$. By differentiating g twice we obtain $g''(a_j) = \frac{p_1}{(a_j-a_{j-1})^2} + \frac{p_2}{(a_{j+1}-a_j)^2} > 0$, hence g is strictly convex, and it attains its maximum on either x_l or x_u. For the other choices of R, the argument is similar: Differentiate twice in order to find that the R-analogue of the modified function in (4.6.1) is strictly convex. □

The proposition would not be true if we were to use $P_n[x_j, x_i)$ or $P_n(x_j, x_i]$ in our table, as that would make it possible for the objective function to jump down at the boundary. In this case, the maximum would either fail to exist (get bigger and bigger the closer it gets to x_u or x_l) or be the $x \in \{x_u, x_l\}$ which we thought of as sub maximal in the previous proposition. Either way, it makes

sense to search the observations for a maximiser in these cases as well. Since all these options are asymptotically equivalent and don't make much of a difference in practice, we will restrict our search of optima to the data points.

4.6.1 Dynamic programming

> [There are] two sledgehammers of the algorithms craft, dynamic programming and linear programming, techniques of very broad applicability that can be invoked when more specialized methods fail. Predictably, this generality often comes with a cost in efficiency.
>
> Dasgupta et al. (2006)

We will transform the optimisation problem into the graph theoretic problem of finding the longest path of a weighted acyclic digraph, which is an archetypical problem in dynamic programming (Dasgupta et al., 2006, chap. 6); one which every other problem solvable by dynamic programming can be reduced to. Define, for any set $x_1, ..., x_n$ of points in $[0, 1]$, choice of $k \in \mathbb{N}^+$and histogram type from Table 4.3 on the previous page the associated weighted digraph Γ as follows: Make k sets A_i — which we call levels — of vertices which contains independent copies of the $n - k + 1$ points x_i, x_{i+1}, x_{k-i+1}. Let $x_0 = 0$ and $x_{n+1} = 1$ be source and sink vertices, and put $V = \bigcup_{i=1}^{k} A_i \cup x_0 \cup x_{n+1}$. We define the set E of edges as follows: $(x_0, y) \in E$ for all $y \in A_1$, and $(y, x_{n+1}) \in E$ for all $y \in A_k$. In addition, a tuple (u, v) is in E if the following conditions are satisfied: (1) There is an i such that $u \in A_i$ and $v \in A_{i+1}$, (2) If u is a copy of x_i and v a copy of x_j, then $i > j$. Figure 4.6.1 contains an example.

The weights will depend on which histogram we want to construct, with values in Table 4.3, where the weight assigned to the edge between x_i and x_j is $R(x_i, x_j)$. The following observations is the key to make everything work: A path through the graph, visiting nodes $0, x_{i_1}, x_{i_2}, ..., x_{i_k}, 1$, will have value $R(0, x_{i_1}) + R(x_{i_k}, 1) + \sum_{j=1}^{k} R(x_{i_j}, x_{i_{j+1}})$, which is the objective function corresponding to the choice of type/weights of the histogram. This form of decomposition shows that the problems has an *optimal substructure* , which allows us to do a sequential enumeration of all solutions. In order to compute the objective functions, we use double recursion: For each node we wish to calculate the value $V(x_j^i) = R(0, x_{i_1}) + \sum_{l=1}^{i} R(x_{i_l}, x_{i_{l+1}})$, where i designates the level of x_j. This can be done by applying the recursive formula $V(x_j^i) = \max_{l<j} V(x_l^{i-1}) + R(x_{i_l}, x_{i_j})$, where we store the value $V(x_j^i)$ at the (i, j)th node. When the algorithm terminates at

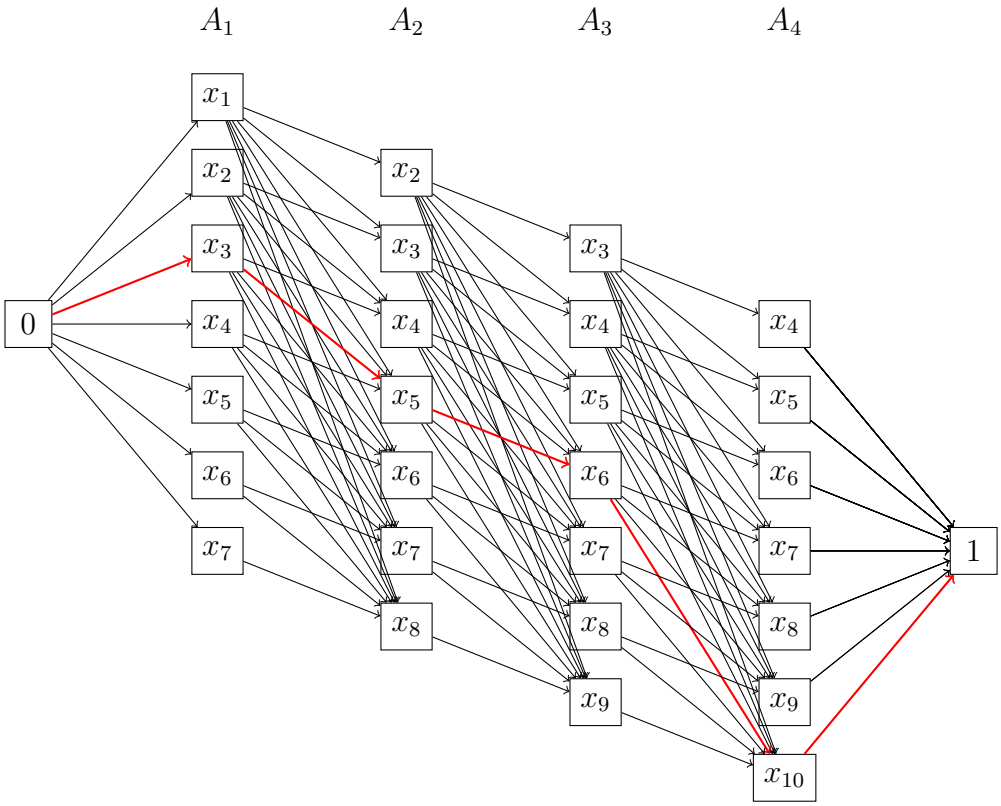

Figure 4.6.1: An example of an associated graph Γ with 10 observations and $k = 5$. The red path is a hypothetical solution path.

$x = 1$, we output the path $0 \to x_{i_1} \to x_{i_2} \to ... \to 1$ which lead to the maximised value.

Proposition 4.6.2. *The DP algorithm runs in $O(kn^2)$ time.*

Proof. The weighted digraph Γ will have $2 + k(n - k + 1) = O(kn)$ vertices and $2(n - k + 1) + (k - 1)\frac{(n-k+1)(n-k+2)}{2} = O(kn^2)$ edges. Since the algorithm just described will visit each edge exactly once, the algorithm will run in $O(kn^2)$ time provided the graph has been constructed. Constructing the graph can be done in $O(kn^2)$ time, as it takes constant time to add a vertex or an edge with all our edge weights W. □

Even though this algorithm is described using graph theoretic language and intuition, the implementation of this algorithm does not rely on graph theoretic data structures of any kind. The graph theoretic description of this algorithm was chosen as it clearly illustrates how the algorithm works and points toward some ideas for making it run faster. It could be possible to remove sections of non-trivial size from the graph before checking them, and there are at least two possible ways to go about this.

1. Use upper and lower bounds on the objective in certain sections of the graph. One could implement this by assigning to each vertex x_j^i a lower $l(x_j^i)$ and upper bound $u(x_j^i)$, and disregard any edge emanating from that vertex whenever its upper bound is lower than the current global lower bound l. A naïve choice of lower and upper bounds would be to make $l(x_j^i)$ a random path in the diagram containing x_j^i and $u(x_j^i) = V(x_j^i) + h(x_j^i)$, where $h(x_j^i) = \sum_{i \in C} R(x_i, x_{i+1})$ with $C = \{j, j+1, ..., n\}$. This upper bound will typically be far too large, however. An issue with the upper bound is that there is no theoretical upper bound on $h'(x_j^i) = \sum_{i=j+1}^{k} R(y_{i-1}, y_i)$ which holds for any underlying distribution F: By creating an arbitrarily sharp spike in the density, this value can become arbitrarily large. A workaround is to impose upper bounds that aren't theoretically sound, but might still works well in practice: For instance, we could use bounds obtained from distributions which we have reason to think that behaves worse than the true F.

2. Prune the graph by throwing away x_j^is we consider unlikely to be in the solution path. For instance, it seems very unlikely that x_{n-k}^1 is the first element of the solution path. One could implement this idea by removing quantiles from each level of the graph, e.g. the lower quartile for the first level and upper quartile on the last level.

These lines of thinking will not be pursued further, but might be worth it if we're serious about using the CIC in Section 4.9 on page 128. Here massive resampling has to be carried out, and the coordinate search soon to be described will not do the job properly.

The complexity $O(n^2k)$ means that the algorithm runs in exponential time, at least on the face of it. This is because k is not the *input length* of k. In fact, the input length of k is $\log_2 k$. This is not the case with the data points $x_1, ..., x_n$, which have input size cn for some c. Obviously, whenever k is considered as a fixed parameter, the algorithm runs in quadratic time and should be considered fast. However, if k is not considered fixed, it runs in *pseudo-polynomial time,* the class of algorithms which runs in polynomial time as function of input value, not input length (Papadimitriou, 2003, p. 203). In our case, since $2 \leq k < n$, with k typically much less than n, the algorithm has complexity bounded by $O(n^3)$, cubic time. This phenomenon is not dissimilar to the dynamic programming algorithm for the *knapsack problem*, a problem that is known to be NP-complete in its decision problem formulation (Papadimitriou, 2003, p. 202). Whenever a

certain k is polynomially bounded in the input n, it is solvable in polynomial time; if not, it runs in exponential time.

While this algorithm yields exact estimates in a reasonable time, the resulting estimates will not be *robust* in the sense that they can be very sensitive to slight changes in input values. We will discuss this issue further in Section 4.7.

Finally, in our `C++` program (of Appendix (A)) the difference in run times between the KL and L_2 histograms is very large. The choice of constant or variable weights doesn't seem to matter. This came as a big surprise, and the culprit has to be the `log`s involved in the KL-maximisation. Now we use the `R` package `microbenchmark` (Mersmann, 2014) to time how much faster L_2 is. We replicate $N = 500$ histograms of both types for $k = 2, 3, ..., 80$ and $n = 500$. In addition, we use a modification of the algorithm into a "skeleton", such that only the constructions that both histograms have in common are computed. Denote the L_2 times by $T(L_2)$, KL runtimes by $T(KL)$ and the skeleton runtimes by $T(S)$. Now we look at $\frac{T(KL)-T(S)}{T(L_2)-T(S)}$, which is seen to be about 16 for any k. This gives a reasonable estimate on how much faster L_2 is. This difference in execution time is a large point in favour of L_2, especially if one wishes to do the subsampling approach to the CIC in Section 4.9 on page 128 on information criteria.

4.6.2 Coordinate search

We describe a *coordinate search* approximation algorithm, a type of local search algorithm (Hromkovič, 2013, section 3.6). Let k be fixed and define an initial vector $a^0 = (a_1^0, \ldots, a_k^0)$. Put $a_0 = 0$ and $a_{k+1} = 1$, and define recursively

$$a_i^j = \underset{a_{i-1}^j < p < a_{i+1}^{j-1}}{\arg\max} R(a_{i-1}^j, p) + R(p, a_{i+1}^{j-1}).$$

On ties, choose the smallest possible p. We say that the algorithm terminates in iteration j whenever $a_i^j = a_i^{j+1}$ for all i.

Proposition 4.6.3. *The coordinate search algorithm terminates.*

Proof. Notice that the objective function can only increase in value when a_i^j is swapped for a_i^{j+1}. Now suppose the algorithm doesn't terminate. Since there are finitely many states, it has to by cyclic. But this is impossible, as it would either lead to decreasing values of the objective function for at least one a_i^j to a_i^{j+1}, or make the objective function constant. However, an a_i^j can't change more

than once when the objective function doesn't increase, due to the tie-breaking rule. □

This algorithm yields consistent estimates. Heuristically, this can be seen as follows: For each i it finds the maximum in the i-th dimension. It converges to a "local" maximum in the sense that it can't move along any i-direction without diminishing in value. Now, whenever the underlying function is concave, this can only happen when it is at an actual maximum. Since the likelihood converges in distribution towards a concave function, it is consistent. Hence we expect the results of this algorithm to have the same asymptotic properties as the exact algorithm, as the rate theorem (2.2.4) only requires $P_n m_{\hat{\theta}_n} \geq \arg\max_\theta P_n m_\theta - o_p(1)$, a condition which is probably satisfied.

In our implementation, we will add an upper bound to the number of full iterations. We will let this be a used defined c. With this modification, at worst this algorithm will terminate after c full iterations. Since each full iteration will check at most n points, it is complexity is bounded, very roughly, by $O(n \log n)$. This is an order of magnitude faster than the exact algorithm, and doesn't depend on k. (It is $O(n \log n)$ instead of $O(n)$ since we need to sort the list of observations first.)

This algorithm requires an initial input to run on. We have not studied the effects of initial input thoroughly, but it certainly matters. In all our uses of the coordinate search algorithm we use an "evenly spaced" initial input, namely the vector of $\frac{i}{k}$th quantiles, with i ranging from 1 to $k-1$. This choice is clearly justifiable in the context of irregular quantile histograms.

4.6.3 Integer programming

As was the case for Manski's estimator (Section 3.4), we can formulate the histogram problem as a mixed integer program. We wish to maximise the linear objective

$$\sum_{i,j} z_{ij} w(i,j), \tag{4.6.2}$$

where $w(i,j) = W(x_i, x_j)$ and W is chosen from table 4.3. This objective should satisfy the constraints

$$\begin{aligned}
\sum_{j=1}^{n-k+1} z_{1i} &= 1, \\
\sum_{j=n-k+1}^{n} z_{i(n+1)} &= 1, \\
\sum_{j=1}^{n} z_{ij} &\leq 1, \quad \forall i = 1, ..., n, \\
\sum z_{ij} &= k+2, \\
\sum_{i<j<k} z_{ij} z_{jk} &= k+2, \\
z_{ij} &\in \{0, 1\}.
\end{aligned}$$

While the last constraint is not linear, it can be reformulated as a linear constraint, basically because $z_{ij}z_{jk}$ can be understood as $z_{ij} \wedge z_{jk}$. The method is as follows: Define a set of new variables w_{ijk}, add the constraint $\sum_{i<j<k} w_{ijk} = k+2$ and add, for every suitable i, j, k,

$$\begin{aligned}
w_{ijk} &\leq z_{ij}, \\
w_{ijk} &\leq z_{jk}, \\
w_{ijk} &\geq z_{ij} + z_{jk} - 1.
\end{aligned}$$

Why these constraints? The first forces there to be exactly one edge connecting the source to the body of the graph, while the second assures us that there is exactly one edge connecting the sink to the body of the graph. The following constraint force every i to appear at most once, while the constraint $\sum z_{ij} = k+2$ says how many edges we want to include. In order to ascertain that the z_{ij} describe a connected path of k vertices in the graph, we require the multiplicative constraint.

Finally we mention the approximation algorithm of choice for Mildenberger et al. (2009), used in the `R` package `histogram`. This is a greedy algorithm. Let $k > 2$ be a given bin count. The algorithm starts by performing an inexpensive $k = 2$ histogram calculation. When $k = i$ bins are calculated, it finds the $(i+1)$th split point by minimizing the divergence in each bin separately, choosing the split point which maximises the global reduction of divergence.

4.7 Pre-smoothing and instability

When estimating our histograms, we minimise the $\theta \mapsto P_n m_\theta$, where P_n is the empirical measure. However, since we assume some smoothness in the true distribution P, it should also work to use a smoothed variant $\widetilde{P}_n$ of P, such as the Gaussian copula estimator. Here we demonstrate that this is the case, and also discuss some advantages of this approach. First we discuss a very important anomaly of the irregular histograms.

4.7.1 Instability

Recall the general histogram formula (4.2.1),

$$h(x) = \sum_{i=1}^{k} \frac{w_i}{a_i - a_{i-1}} 1_{[a_{i-1}, a_i]}(x),$$

and the minimiser of the empirical L_2-distance with variable weights, given by

$$\arg\max_a \sum_{i=1}^{k} \log \frac{P_n(a_{i-1}, a_i)}{a_i - a_{i-1}} P_n[a_{i-1}, a_i],$$

where the a_is are constrained to be inside the set of observations $x_1, ..., x_n$.

Let $x_{(i)}$ denote the ith order statistic. Then $P_n[x_{(i)}, x_{(i+j)})^2 = \frac{4+(j-1)^2}{n^2}$. When j is small, this quantity is also small, $P_n[x_{(i)}, x_{(i+j)}]^2 \approx n^{-2}$. But it might also happen that $x_{(i+j)} - x_{(i)}$ is very small: In this case $x_{(i+1)} - x_{(i)}, x_{(i+2)} - x_{(i+1)}$ etc. will be even smaller. For $x_{(i+1)} - x_{(i)} \lll \frac{1}{n^2}$, the term $\frac{P_n[x_{(i)}, x_{(i+1)})}{x_{(i+1)} - x_{(i)}} P_n[x_j, x_i) = \frac{4}{n^2(x_{(i+1)} - x_{(i)})}$ will become very large, forcing the histogram to include the pair $x_{(i)}, x_{(i+1)}$ as split points even tough their closeness is just noise. This *instability*[1] is especially prevalent in real data sets, where numbers are rounded to fewer digits than in simulations. It manifests itself in the histograms as tall spikes with very high density, as in Figure 4.7.1. The phenomenon affects both the exact algorithm and the coordinate search algorithm, but the exact algorithm to a much higher degree, essentially because it actively ferrets these anomalies out. The coordinate search will have to be unlucky in order to stumble across such spikes. We can remedy this fault by imposing a minimum distance between pairs (x_i, x_j) considered in the dynamic programming algorithm and the coordinate search, for instance by enforcing $|x_i - x_j| \geq \delta$ for some $\delta > 0$ as in the proof

[1] Typically one would say that the solution non-robust or sensitive to changes in the data. We will not use these terms here, as the word "robustness" is reserved by the usual statistical concept of an estimator being robust to changes in the underlying model. This is not the issue here, the instability phenomenon occurs when the model assumptions hold perfectly.

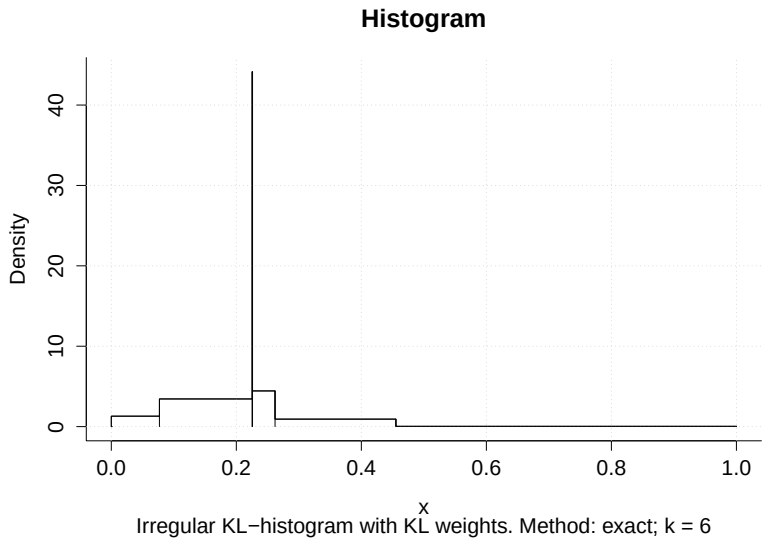

Figure 4.7.1: Example of a spike when $X_1, ...X_n \sim \text{Beta}(2, 7)$. The culprits are the points $(0.2252, 0.2258)$. More spikes can be found in the illustrations section.

of consistency (Theorem 4.4.6). A choice of such a $\delta > 0$ is unavoidable, at least for the variable weight histograms. In most cases, this choice is easy to make: What's important is that δ is so large that the really small differences $x_{(i+1)} - x_{(i)}$ are avoided,

4.7.2 Pre-smoothing

The pre-smoothing procedure is simple to describe and set up. Given data $x_1, ..., x_n$, we find the Gaussian copula KDE estimate $\widehat{f}$ of the true density f, as described in Section 4.1. If we wish to calculate the pre-smoothed KL-histogram for a given k, we need to find $\arg\max_a \sum_{i=1}^{k} \widehat{P}[a_{i-1}, a_i) \log \frac{\widehat{P}[a_{i-1},a_i)}{a_i - a_{i-1}}$, where $\widehat{P}$ is the measure associated with $\widehat{f}$. As calculating this requires intensive numerical integration, we approximate it by resampling a large amount of observations from $\widehat{f}$, which will be $50kn$ in our studies. Recall from Section 4.1 that the Gaussian copula KDE is easy to sample from, and this is the main reason why we use it. The `R` code for this procedure is included in Appendix A.

In addition to fixing the instability issue, the pre-smoother has superior performance in terms of Hellinger distance from the true distribution and mean integrated squared error (MISE), as we will demonstrate in a special case shortly.

Table 4.4: Mean integrated squared error for $F = \text{Beta}(2, 7)$ with different choices of n, k.

		KL/var			L_2/var		
n	k	Coord	Exact	Smooth	Coord	Exact	Smooth
100	2	0.362	0.363	0.397	0.7	0.712	1.065
	8	0.181	16.072	0.087	0.179	12.17	0.084
	50	2.182	19.386	0.053	2.218	29.71	0.055
500	10	0.084	10.835	0.036	0.092	38.19	0.037
	60	0.298	74.025	0.016	0.315	18.037	0.016
		KL/const			L_2/const		
100	2	0.634	0.629	0.62	0.641	0.64	0.616
	8	1.885	1.54	0.142	1.459	0.684	0.136
	50	2.136	62.748	0.068	1.941	3.519	0.071
500	10	2.474	0.164	0.088	2.218	0.189	0.08
	60	0.337	441.356	0.024	0.364	1.156	0.023

4.7.3 Simulations

We compare the MISE and Hellinger distances for the exact solution, coordinate search solution and the smoothed solution for some choices of n,k with underlying $F = \text{Beta}(2, 7)$. The results are in Table 4.5 (Hellinger) and Table 4.4 (MISE). Several tentative conclusions can be made from these simulations.

1. The exact solutions perform very poorly, with enormous MISE in all cases except those where $k \lll n$, a miserable result which probably can be attributed to the *instability* discussed in the preceding section.

2. The smoothed histogram outperforms the others on both measures for almost every choice of k and n, the exception is when $k = 2$, where smoothing seems to worsen things.

3. The quality of the smoothed histogram increases with k, contrary to the coordinate search histogram. This is not surprising, as it approximates the underlying Gaussian copula KDE better and better.

4. L_2 and KL histograms have very similar performance, with a small victory to KL. The difference is large for $k = 2$ and $n = 100$, however.

5. Constant weights don't fare well at all compared to variable weights. This serves to further discourage the use of this kind of histogram.

Also of interest is the performance of estimating the split points. The results of a small simulation study is found in Figure 4.7.2. It seems likely that the

Table 4.5: Hellinger distance between the histogram and the true distribution for $F = \text{Beta}(2, 7)$ with different choices of n, k.

		KL/var			L_2/var		
n	k	Coord	Exact	Smooth	Coord	Exact	Smooth
100	2	0.216	0.218	0.221	0.278	0.281	0.391
	8	0.137	0.195	0.103	0.147	0.215	0.123
	50	0.257	0.331	0.079	0.252	0.318	0.081
500	10	0.098	0.116	0.072	0.114	0.185	0.095
	60	0.127	0.218	0.048	0.127	0.201	0.053
		KL/const			L_2/const		
100	2	0.386	0.385	0.385	0.385	0.385	0.384
	8	0.306	0.251	0.211	0.306	0.262	0.208
	50	0.263	0.358	0.11	0.26	0.29	0.115
500	10	0.276	0.195	0.186	0.274	0.198	0.183
	60	0.156	0.234	0.087	0.156	0.216	0.088

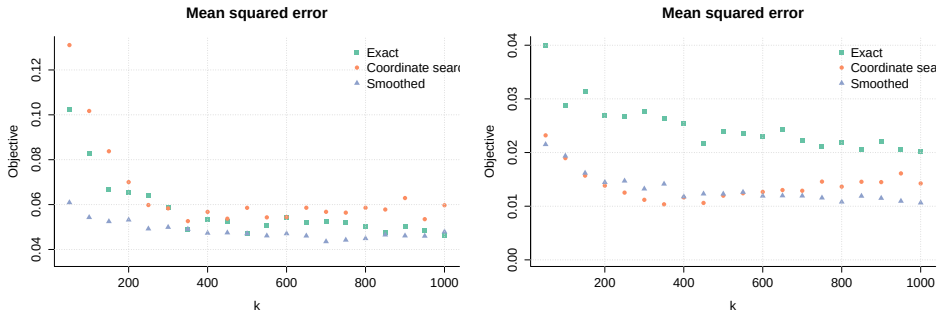

Figure 4.7.2: Plots of $n^{\frac{2}{3}}$-rescaled MSE for two different split points for Beta$(1, 3)$ with $k = 11$ and $n = 50, 100..., 1000$ (left) MSE for the split point $i = 2$, (right) for $i = 5$. Curiously, the exact algorithm performs much worse than the coordiante search for $i = 5$, but better for $i = 2$.

smoothed estimator will perform at least as well as the exact estimator and the coordinate search.

4.8 Illustrations

4.8.1 Police percentage data

We apply the pre-smoothed and exact Kullback-Leibler histograms ($k = 6$) on the racial police data set from New York Times Ashkenas and Park (2014). Each observations x_i is the percentage of police officers in a big city in the United States that is white ($n = 543$). The resulting histograms are in Figure 4.8.1. Notice the dotted spike in the in the exact histogram at the right side:

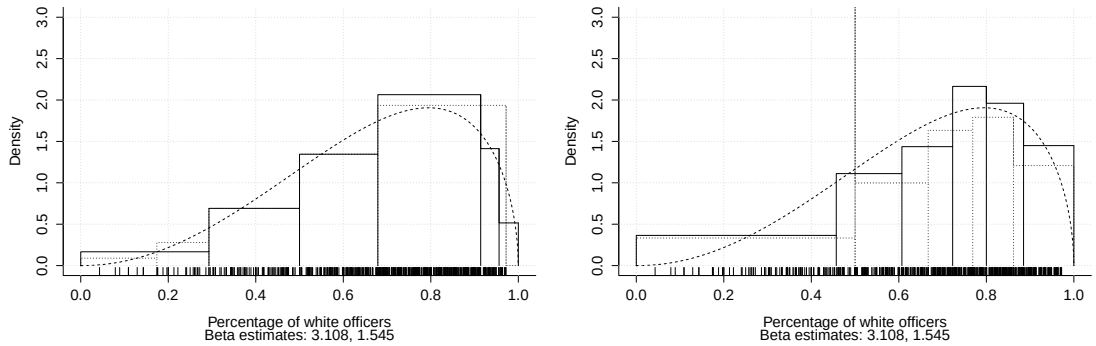

Figure 4.8.1: (left) Kullback-Leibler irregular histogram with variable weights, dotted is the exact histogram, while solid is pre-smoothed one. (right) Same histogram with constant weights.

This sort of thing happens even when n is large compared to k. Also notice that the exact histogram on the left hand side gives almost zero weight to the rightmost part of the picture, an "error" rectified by smoothing.

4.8.2 Church services

On the website of Statistics Norway (SSB) one can find the data of church services from the Church of Norway per 1000 residents for each Norwegian commune SSB (2014). Theoretically speaking, this data may not be contained in a bounded interval — it is conceivable for a commune of 1000 residents to have e.g. $10,000,000$ church services per year (if all residents are priests and do 24 services each day, we're getting close). Nevertheless, the histogram works nicely when we assume, innocently enough, that the distribution is supported on $[0, 1000]$. This data can be found for $1999 - 2014$, and our focus is on 2014. Out of the 428 communes, 5 where removed due to NAs in the data set. We used the smoothed and exact L_2-histogram. The result is displayed in Figure 4.8.2. This and other examples illustrates the huge impact of pre-smoothing. Also, in this case, the approximate coordinate algorithm gives much better results than the exact algorithm. The observations on $64, 66, 70$ and 82 are Loppa (residents: 1 027), Modalen (372) Vevelstad (495) and Solund (815) respectively. Oslo, with a population of 634 463, have 7.3 services per resident, placing it into the second to last bin.

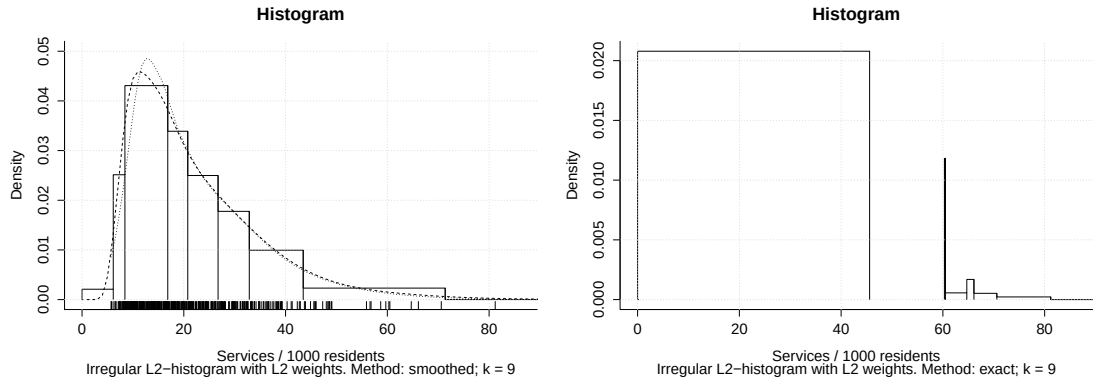

Figure 4.8.2: L_2-histogram of church services per 1000 residents for Norwegian communes. The dashed line is the Gaussian copula kernel density estimator, while the dotted line is fitted through the `locfit` package, (Loader, 2013). As for the exact histogram, we get a similarly horrible result for the KL-histograms.

Table 4.6: 95% confidence intervals, with coverage and length for different methods. The numbers $(2, 7, 10, 20)$ are the block sizes in the subsample. The αs are the nominal levels for the smoothed bootstrap. We supplied results for $\alpha = 0.90$ since $\alpha = 0.95$ only yielded coverages of 1.

	$n = 50$		$n = 100$		$n = 500$		$n = 1000$	
	Cov	Len	Cov	Len	Cov	Len	Cov	len
5	0.958	0.1456	0.959	0.1195	0.962	0.0715	0.974	0.0569
7	0.958	0.1436	0.97	0.119	0.96	0.0717	0.971	0.0571
10	0.946	0.1422	0.981	0.1194	0.979	0.0727	0.976	0.058
20	0.901	0.1246	0.968	0.1127	0.98	0.0724	0.984	0.0579
$\alpha' = 0.90$	0.98	0.1359	0.98	0.1053	0.99	0.0594	1	0.0461
$\alpha' = 0.95$	1	0.1571	1	0.1223	1	0.0686	1	0.0529

4.8.3 Confidence intervals

We perform a small Monte Carlo study of confidence intervals (CIs) for the split points when $k = 2$ for L_2-histograms with equal weights. We will compare subsample CIs and smoothed bootstrap CIs. Both the ordinary non-parametric bootstrap and m-out-of-n bootstrap are inappropriate for irregular histograms, as they resample with replacement, something the histogram can't handle well. This is not a big problem, as the nonparametric bootstrap is inconsistent for cube root estimators, and the m-out-of-n bootstrap and the subsample tend to have similar performance.

This combination of L_2 and equal weights is chosen because L_2 is faster to calculate than KL and equal weights allows us to estimate only one parameter.

The results are displayed in Table 4.6. The smoothed bootstrap appears inconsistent. A likely explanation for this is that the bandwidth choice, obtained

by the normal reference rule, isn't tailored to this situation. Still, the smoothed bootstrap intervals tend to have both smaller length and higher coverage than the subsample intervals. A similar kind of problem appears for the limiting distribution, as we require a consistent estimate of

$$V = f'(a)\left[\frac{1}{a} - \frac{1}{1-a}\right] - 2f(a)\left[\frac{1}{(1-a)^2} + \frac{1}{a^2}\right] + \frac{2(F(a) - \frac{1}{4})}{a^3} + \frac{2(\frac{3}{4} - F(a))}{(1-a)^3},$$

(see Example 4.5.18), but we have no guarantee that $f'(a)$ is well estimated by the Gaussian copula KDE.

4.9 Information criteria

4.9.1 Akaike's information criterion

The most famous model selection criterion is the AIC, Akaike's information criterion. Let $X_1, X_2, ..., X_n \overset{i.i.d.}{\sim} f$, for some density f, and $\mathcal{M} = (M_1, M_2, ..., M_k)$ be a list of parametric models with likelihoods $f^1, f^2, ..., f^k$. Let d_j be the size of the free parameter vector θ^j in f^j, and $\widehat{\theta^j}$ be its maximum likelihood estimate. Then the AIC is defined as[2],

$$AIC_j = 2\sum_{i=1}^{n} \log f^j(X_i; \widehat{\theta}) - 2d_j,$$

and the index of the chosen model will be $\arg\max_{j=\{1,2,...,k\}} AIC_j$. A better justified cousin of the AIC is the TIC, Takeuchi's information criterion, defined as $TIC_j = 2\sum_{i=1}^{n} \log f^j(X_i; \widehat{\theta}) - 2\text{Tr}JK^{-1}$. Here θ_0 is the least false value of θ, defined as the argmin of the Kullback-Leibler divergence between $f(\cdot, \theta)$ and f. Here $J = E\left[\frac{\partial^2}{\partial\theta^2} \log f(X, \theta) \mid_{\theta=\theta_0}\right]$, and $K = \text{Var}\left[\frac{\partial}{\partial\theta} f(X, \theta) \mid_{\theta=\theta_0}\right]$. When the underlying density f has the likelihood $f(\cdot, \theta_0)$ for some θ_0, $\text{Tr}JK^{-1} = d_j$, and the TIC and AIC are equal. These model selection procedures can and will lead to different choices of the best model. While the TIC is better theoretically justified, the AIC performs well in practice and avoids the noise induced through the estimation of the J and K matrices.

We will describe the theoretical justification for the TIC. The quantity we wish to minimise is the *expected discrepancy* of a statistical divergence. As in

[2]There are in fact two different definitions of the AIC. In addition to the one mention, the variant $AIC' = -AIC$ is also commonly used. In this case, the "best model" will minimise the AIC instead of maximising it.

the foregoing sections, the divergences of interest to us are the L_2-divergence and the Kullback-Leibler divergence. To these divergences, we can associate a *discrepancy* which does not involve the unknown density. The discrepancy has the same shape as the divergence written as a function of θ. For Kullback-Leibler,

$$\begin{aligned} d_{KL}(p, f_\theta) &= \int \log \frac{p(x)}{f_\theta(x)} p(x)dx \\ &= \int p(x) \log p(x)dx - \int p(x) \log f_\theta(x)dx. \end{aligned}$$

Here $\int p(x) \log p(x)dx$ is a constant, and we will only need to maximise

$$\int p(x) \log f_\theta(x)dx = P \log f_\theta.$$

Substituting P_n for P, we obtain the discrepancy $P_n \log f_\theta$. Similarly for the L_2-divergence, we obtain the discrepancy $-2P_n f_\theta + F_\theta f_\theta$.

The rationale behind the AIC/TIC is to chose the model that has the smallest expected Kullback-Leibler discrepancy, $P\left[\max_\theta P_n \log f_\theta\right]$. Define $\widehat{\theta} = \arg\max_\theta P_n f_\theta$, and observe that $P\left[\max_\theta P_n \log f_\theta\right] = P\left[\log f_{\widehat{\theta}}\right]$. Since P is unavailable, we use the plug-in P_n instead. But this move introduces bias. In order to make fair comparisons between models, we estimate this bias and subtract it from the estimate. This is a crude method of estimating the discrepancy, and there are other alternatives, like the bootstrap (yielding the *EIC*) (Shibata, 1997, Konishi and Kitagawa, 2008, chapter 8) and leave-one-out cross validation of the likelihood (Stone, 1977, Claeskens and Hjort, 2008, section 2.9). In the following proposition, $l(\theta) = \log f_\theta$ denotes the log-likelihood of f.

Proposition 4.9.1. *The bias of $P_n\left[\max_\theta P_n f_\theta\right]$ as an estimator of $P\left[\max_\theta P_n f_\theta\right]$ equals $\frac{1}{n} Tr J^{-1} K$. Under model conditions, this bias simplifies to $\frac{1}{n} d$, where d is the number of free parameters.*

Proof. We wish to estimate the true value $\mathbb{P}l(\widehat{\theta})$, and the purpose of this exercise is to estimate the bias of the naïve estimator $\mathbb{P}_n l(\widehat{\theta})$. In Theorem 2.2.1, we derived the limit process of

$$n(P_n - P)(l(\theta_0 + tn^{-\frac{1}{2}}) - l(\theta_0),$$

namely $t^T Z - \frac{1}{2} t^T J t$, with $Z \sim N(0, K)$ and $K = \mathrm{Var} U(X, \theta_0)$. Also, its maximiser is $s = J^{-1} Z$. we will deduce this bias by using this rescaled process. For

this purpose, put $s_n = \sqrt{n}(\widehat{\theta} - \theta) \to s$. We calculate

$$\begin{aligned} n(\mathbb{P}_n l(\widehat{\theta}) - \mathbb{P} l(\widehat{\theta})) &= n(\mathbb{P}_n l(\theta_0 + n^{-\frac{1}{2}} t_n) - \mathbb{P}_n l(\theta_0)) \\ &\quad -n(\mathbb{P} l(\theta_0 + n^{-\frac{1}{2}} t_n) - \mathbb{P} l(\theta_0)) \\ &\quad +n(\mathbb{P}_n l(\theta_0) - \mathbb{P} l(\theta_0)). \end{aligned}$$

The third summand has expectation 0, the second summand converges to $\frac{1}{2} s^T J s$, while first term converges to $s^T Z - \frac{1}{2} s^T J s$, with $Z = N(0, K)$ as above. Combining these, we get the random variable $Z^T J^{-1} Z$. Observe that $Z^T J^{-1} Z = \sum Z_i Z_j j^{ik}$, where j^{ik} is the $(i,k)th$ element of J^{-1} and Z_i is the i-th element of Z. Since $EZ_i Z_j = K_{ij}$, we obtain $EU^T J^{-1} U = \sum k_{ij} j^{jk} = \mathrm{Tr} K J^{-1} = \mathrm{Tr} J^{-1} K$. When $J = K$, this trace is equal to d. □

The AIC formula can be used for bin selection in regular histograms, but is not recommended (Birgé and Rozenholc, 2006). The log likelihood at the argmax is

$$l(x) = \sum_{i=1}^{k} \log \left(\frac{P_n[a_i, a_{i-1})}{a_i - a_{i-1}} \right) 1_{[a_{i-1}, a_i)}(x),$$

hence the empirical log likelihood at the data $x_1, x_2, ..., x_n$ is

$$n \sum_{i=1}^{k} \log \left(\frac{P_n[a_i, a_{i-1})}{a_i - a_{i-1}} \right) P_n[a_i, a_{i-1}). \tag{4.9.1}$$

Assume the split points are equally spaced, *i.e.* $a_i - a_{i-1} = k^{-1}$ for every i, and define $n_i = n P_n[a_i, a_{i-1})$ as the number of observations in the i-th bin. Then (4.9.1) becomes

$$\sum n_i \log \left(\frac{n_i}{n} \right) - n \log(h) = \sum n_i \log n_i + n \log(\frac{k}{n}),$$

and a variant of the AIC for regular histograms on the unit interval is

$$2 \left[\sum n_i \log n_i + n \log(\frac{k}{n}) - (k-1) \right].$$

It can be argued that this isn't the "correct variant" of the AIC for histograms, as the condition for the approximation $k - 1$ isn't satisfied. Indeed, in 1990, Atilgan demonstrates that the proper variant of the AIC for regular histograms for positive densities on the unit interval is

$$\left[\sum n_i \log n_i + n \log(\frac{k}{n}) - k\right].$$

4.9.2 The cube root information criterion

We will now copy the reasoning behind the AIC into the setting of irregular Kullback-Leibler histograms, in the hope of arriving at a reasonable bin selection procedure.

Proposition 4.9.2. *The bias of* $\mathbb{P}_n m(\widehat{q})$ *equals*

$$d^*_{KL} = n^{-\frac{2}{3}} \sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \log\left(\frac{w_{i+1}}{a_{i+1}-a_i}\right) - \log\left(\frac{w_i}{a_i - a_{i-1}}\right)\right| EW_i(s_i).$$

Proof. Define $s_n = n^{\frac{1}{3}}(\widehat{q} - q^0) \to s = \arg\max G(t)$, and observe

$$\begin{aligned}(\mathbb{P}_n m(\widehat{q}) - \mathbb{P}m(\widehat{q})) &= (\mathbb{P}_n m(a + n^{-\frac{1}{3}}s_n) - \mathbb{P}_n m(a)) \\ &\quad -(\mathbb{P}m(a + n^{-\frac{1}{3}}s_n) - \mathbb{P}m(a)) \\ &\quad +(\mathbb{P}_n m(a) - \mathbb{P}m(a)).\end{aligned}$$

The third line has expected value 0, while the first converges to $n^{-\frac{2}{3}}G(s)$ and the second line has limit $-n^{-\frac{2}{3}}\frac{1}{2}s^T V s$. Since

$$G(s) = \frac{1}{2}s^T V s + \sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \log\left(\frac{w_{i+1}}{a_{i+1}-a_i}\right) - \log\left(\frac{w_i}{a_i - a_{i-1}}\right)\right| W_i(s_i),$$

the bias is $n^{-\frac{2}{3}} \sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \log\left(\frac{w_{i+1}}{a_{i+1}-a_i}\right) - \log\left(\frac{w_i}{a_i-a_{i-1}}\right)\right| EW_i(s_i)$. □

Similarly, the bias corresponding to the the L_2-histogram is

$$d^*_{L_2} = n^{-\frac{2}{3}} 2 \sum_{i=1}^{k-1} f(a_i)^{\frac{1}{2}} \left| \frac{w_{i+1}}{a_{i+1}-a_i} - \frac{w_i}{a_i - a_{i-1}}\right| EW_i(s_i).$$

Now we can define four new bin selection criteria for irregular histograms, one for each combination of constant / variable weights and KL / L_2:

$$
\begin{aligned}
CIC_{KLv} &= n\sum_{i=1}^{k} P_n(\widehat{a_{i-1}}, \widehat{a_i}) \log \frac{P_n(\widehat{a_{i-1}}, \widehat{a_i})}{\widehat{a_i} - \widehat{a_{i-1}}} - n\widehat{d^*_{KL}}, \\
CIC_{KLc} &= n\sum_{i=1}^{k} P_n(\widehat{a_{i-1}}, \widehat{a_i}) \log \frac{w_i}{\widehat{a_i} - \widehat{a_{i-1}}} - n\widehat{d^*_{KL}}, \\
CIC_{L_2v} &= n\sum_{i=1}^{k} \frac{P(\widehat{a_{i-1}}, \widehat{a_i})^2}{\widehat{a_{i-1}} - \widehat{a_i}} - n\widehat{d^*_{L_2}}, \\
CIC_{L_2c} &= n\sum_{i=1}^{k} w_i \frac{(2P(\widehat{a_{i-1}}, \widehat{a_i}) - w_i)}{\widehat{a_{i-1}} - \widehat{a_i}} - n\widehat{d^*_{L_2}}.
\end{aligned}
$$

Here CIC is an abbreviation for "cube root information criterion", a term first used by Hjort in a workshop report (2007, p. 33). His CIC is equivalent to our CIC_{KLc}, the cube root information criterion of KL-histograms with constant weights.

As noted in the introduction, the value $\widehat{d^*}$ is hard to estimate, as the limiting distribution is difficult to work with analytically and intractable numerically. This last claim demands justification. we will work with KL-histograms with variable weights for simplicity, in which case a natural estimator for d^*_{KLv} is

$$
\widehat{d^*_{KLv}} = n^{-\frac{2}{3}} \sum_{i=1}^{k-1} \widehat{f}(\widehat{a_i})^{\frac{1}{2}} \left| \log\left(\frac{P_n(\widehat{a_{i+1}}, \widehat{a_i})}{\widehat{a_{i+1}} - \widehat{a_i}} \right) - \log\left(\frac{P_n(\widehat{a_i}, \widehat{a_{i-1}})}{\widehat{a_i} - \widehat{a_i}} \right) \right| EW_i(s_i),
$$

where $\widehat{f}$ is the Gaussian copula KDE estimate of f. The difficulty lies in finding $EW_i(s_i)$. This would be done by simulation from the limiting distribution, which also requires consistent estimation of $f'(a_i)$ as part of the Hessian V (see Proposition 4.5.5). This is not the biggest problem though, since simulating the $\arg\max$ with $k > 2$ will require the computation of a $(k-1)$-dimensional grid of values in which we simulate $k-1$ Brownian motions, which has cardinality N^{k-1} for some N. To compute the argmax, we would have to find the value over each combinations of $(t_1, t_2, ..., t_{k-1})$, and in order to calculate the expectation, this has to be done several times, say 100. This is clearly infeasible for relatively large k, even for $k = 3$ when implemented in `R`.

Fortuitously, it turns out that a subsampling approach works reasonably well. In order to calculate this term we will use a subsampling procedure on the scaled asymptotic bias $n^{2/3} E\left(\mathbb{P}_n m_{\widehat{s}} - \mathbb{P} m_{\widehat{s}}\right)$. Let G_b be the subsample distribution with block size b. A reasonable estimator for the bias is

$$E_{G_b}\left[\mathbb{P}_b m_{q^*} - \mathbb{P}_n m_{q^*}\right],$$

obtained by substituting $\widehat{q}$ with q^*, $\mathbb{P}_n$ with $\mathbb{P}_b$ and $\mathbb{P}$ with $\mathbb{P}_n$. We give an indication why this should work when $b = \frac{1}{2}n$. The block size $b = \frac{1}{2}n$ can be justified, very handwavingly, by considering the alternatives

1. $b < \frac{1}{2}n$. The estimate m_{q^*} is skewed away from the $m_{\widehat{q}}$,

2. $b > \frac{1}{2}n$. The estimate $m_{q^*}$is skewed towards $m_{\widehat{q}}$.

This procedure estimates the expected bias reasonably well up to $k \approx 2n^{\frac{1}{3}}$, according to our experiments.

Unfortunately, we will have to use the exact algorithm when performing this subsampling, together with a smart choice $\delta > 0$ to bound the bin widths, $a_i - a_{i-1} \leq \delta$. We will not use the coordinate search algorithm as it doesn't approximate the objective function well enough, and can't drop the δ, as delta-free maximisation has a very erratic behaviour.

In the following subsections, we will demonstrate that the subsampling approach works reasonably well. We could, as much simpler approach, create an empirical reference rule for the bias based on e.g. the Beta$(2, 7)$-distribution. This could be obtained through extensive simulations and stored as an `R`-function. we will demonstrate empirically that the asymptotic bias term $\widehat{d^*_{KL}}$ doesn't approximate the finite sample bias well even for $n \approx 1000$, an observation which fits nicely into the picture given in example (4.5.8). Finally, we will apply the information criterion on two real data sets and do a small Monte Carlo study on their performance as compared to the approach of Rozenholc et al. (2010) and the AIC.

4.9.3 Bias and subsampling

First we demonstrate that subsampling the bias works, then we look at the theoretical behaviour of the procedure, where we use the simulated true biases instead of the subsample estimates. The point of the CIC is to attempt to chose the histogram with the minimal discrepancy. In order for this to work well, the expected bias must be reasonably close to the actual bias. Recall the discussion at the end of Section 4.7, where it was noted that the L_2-histogram is much faster to compute that the KL-histogram, by factor around 16. Since all these simulations are time consuming, we restrict our attention to L_2-histograms.

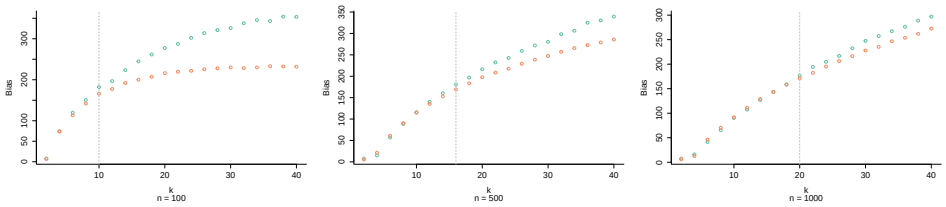

Figure 4.9.1: The green points are the true bias, while the red points are subsampled, underlying distribution is Beta(2, 7).

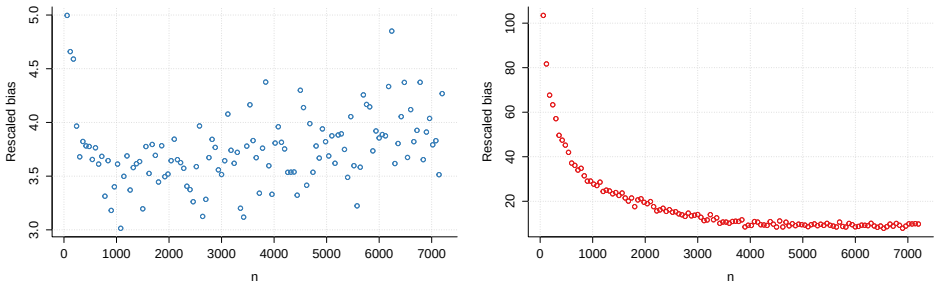

Figure 4.9.2: Large study of biases for (blue) KL/c and (red) L_2/c.

Subsampling the bias

We illustrate the performance of subsampling $n^{\frac{2}{3}} \times$ bias, for both Beta(2, 7) with $n = 100, 500, 1000$ in Figure 4.9.1. The green points are the true bias, while the red points are subsampled. The dashed lines are $2n^{\frac{1}{3}}$, which appears to be a reasonable break point for where the subsampled bias approximate the true bias well.

Behaviour of the bias

We investigate how fast the finite sample converges in Figure 4.9.2. This is done for KL/c (Kullback-Leibler histogram with constant weights) and L_2/c, (L_2 with constant weights) both with $\delta = 0.0001$ and underlying $F = \text{Beta}(2, 7)$. The ns under consideration are $60, 120, 180, ..., , 7140, 7200$. For each n, we simulated n observations from F and calculated the bias $n^{\frac{2}{3}}(P_n m_{\widehat{a}} - P m_{\widehat{a}})$, replicated $k = 100$ times. The resulting values are the means of these samples. Strikingly, the L_2-bias stabilises only around $n = 4000$ or so, while the KL biases stabilise far earlier. From this small study, it seems likely that the asymptotic approximation to the real bias works earlier for KL than L_2.

Since subsampling is computationally expensive, it would be nice if we could get a reference rule up and running. Some experiments suggests this might be the case. In the following plot we have simulated the biases using $n = 1000$

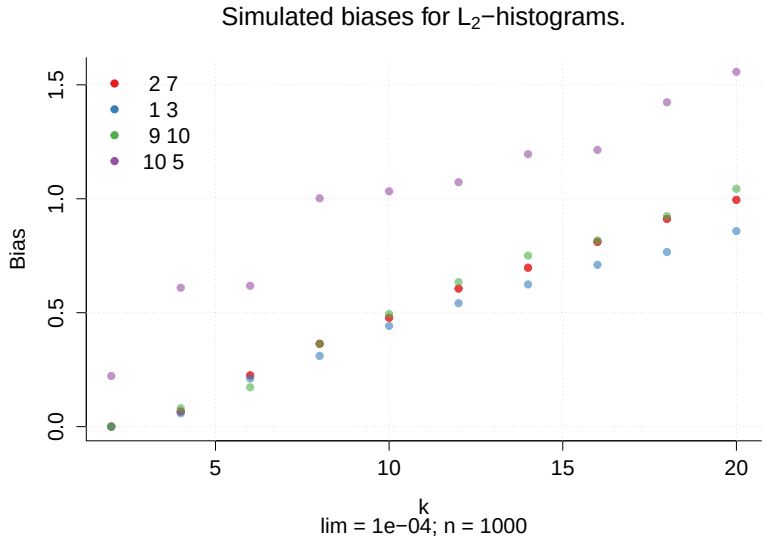

Figure 4.9.3: Comparison of simulated biases for a selection of Beta(a, b)-distributions, $n = 1000$.

and $N = 100$ replications for several choices of Beta-distributions. The shape parameters were $(2, 7)$,$(1, 3)$,$(9, 10)$ and $(10, 5)$. The results are in Figure 4.9.3. Notice that all reasonably regular distributions have pretty similar biases. This suggests that it might be workable to do a single, big simulation in order to get a reference rule for the bias.

4.9.4 A small Monte Carlo Study

Rozenholc et al. (2010) propose several information criteria for the irregular histogram obtained by penalising the likelihood

$$n \sum_{i=1}^{k} P_n(\widehat{a_{i-1}}, \widehat{a_i}) \log \frac{P_n(\widehat{a_{i-1}}, \widehat{a_i})}{\widehat{a_i} - \widehat{a_{i-1}}}.$$

Their penalties are

$$\begin{aligned} \text{pen}_n^B &= \log \binom{n-1}{k-1} + (k-1) + \log^{\frac{5}{2}} k, \\ \text{pen}_n^R &= \log \binom{n-1}{k-1} + \frac{1}{2} \sum_{j=1}^{D} \frac{P_n(\widehat{a_{i-1}}, \widehat{a_i})}{\widehat{a_i} - \widehat{a_{i-1}}} + \log^{\frac{5}{2}} k, \end{aligned}$$

For each choice of penalty, the histogram with the largest penalised likelihood is chosen. Since it requires a slightly different computation procedure, we ignore pen_n^R. In addition to pen_n^B, $CIC_{L_2 v}$ and CIC_{KLv}, we will use the "classical"

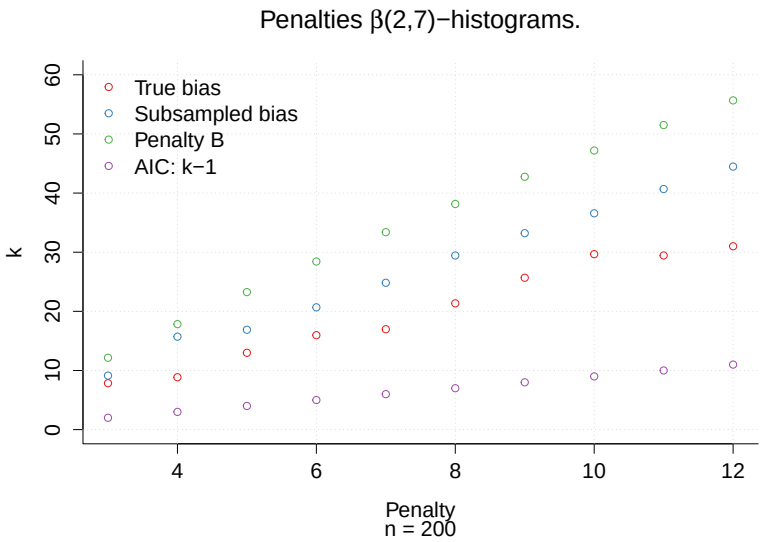

Figure 4.9.4: A comparison of the different penalties when the underlying distribution is Beta$(2, 7)$.

penalty

$$\text{pen}_n^{AIC} = d - 1.$$

In Figure (4.9.4) we supply a plot of the penalties for comparisons sake. It appears that the AIC penalty is far too small, butpen$_n^B$ lies interestingly close to the subsampled bias. This is likely due to the regular shape of Beta$(2, 7)$.

For each n, we calculate the penalised likelihood for $k = 3, 4, ...2\left\lceil n^{\frac{1}{3}}\right\rceil$, which should be more than enough. The case when $k = 2$ is dropped, as it is deemed unlikely that it would ever be used in practice. For the CICs, we subsample $K = 50$ times in order to keep the execution time down. For $n = 50, 100, 200, 500$ we obtain the Hellinger risks and mean k chosen for all the different approaches, each replicated $N = 100$ times. The distributions under consideration are Beta$(2, 7)$, Beta$(\frac{5}{6}, \frac{7}{8})$ (in Table (4.7)), andBeta$(3, 3)$ and a bimodal $(0.5, 0.5)$-mixture of Beta$(10, 40)$ and Beta$(40, 10)$ in Table (4.8).

This study suggests a couple of tentative conclusions:

1. The Kullback-Leibler CICs are superior to the L_2 CICs; the L_2 CICs select too few bins.

2. Constant weights increase the Hellinger risk, unless the underlying distribution is "difficult", like Beta$(\frac{5}{6}, \frac{7}{8})$.

3. pen$_n^B$ performs very well, but CIC_{KLv} appears to be best.

Table 4.7: Hellinger distances for Beta$(2,7)$ and Beta$(\frac{5}{6},\frac{7}{8})$. The best results are in bold.

		Beta$(2,7)$				Beta$(\frac{5}{6},\frac{7}{8})$			
		50	100	200	500	50	100	200	500
k	$CIC_{L_2 v}$	5.04	3.69	3.5	3.23	3.79	3.14	3.09	3.03
	CIC_{KLv}	5.16	5.56	6.82	7.86	4.15	4.02	3.47	3.32
	$CIC_{L_2 e}$	4.96	3.9	3.3	3.28	3.89	3.15	3.07	3.03
	CIC_{KLe}	5.23	5.77	6.36	7.77	4.28	4.25	3.48	3.38
	pen_n^B	3.22	3.51	4.28	5.35	3.01	3.01	3.01	3
	pen_n^{AIC}	3.11	3.02	3	3	3	3	3	3
Hell	$CIC_{L_2 v}$	0.2199	0.1826	0.1682	0.1675	**0.1251**	0.085	**0.0642**	**0.0475**
	CIC_{KLv}	0.2103	**0.1718**	0.1424	**0.1061**	0.1648	0.1327	0.0927	0.0639
	$CIC_{L_2 e}$	0.3132	0.3141	0.3203	0.3138	0.1322	**0.0844**	0.0671	0.0532
	CIC_{KLe}	0.3004	0.2756	0.2513	0.2205	0.148	0.107	0.0732	0.0532
	pen_n^B	0.2005	0.1751	**0.1405**	0.1102	0.1436	0.1166	0.0874	0.0599
	pen_n^{AIC}	**0.1979**	0.1795	0.1661	0.1595	0.1431	0.1163	0.0871	0.0599

Table 4.8: Hellinger distances for Beta$(3,3)$ and the bimodal distribution. The best results are in bold.

		Beta$(3,3)$				Bimodal			
		50	100	200	500	50	100	200	500
k	$CIC_{L_2 v}$	4.47	3.28	3.06	3.41	4.96	3.52	4.8	5.74
	CIC_{KLv}	5.01	5.3	6.11	6.51	6.94	7.96	9.81	12.6
	$CIC_{L_2 e}$	4.48	3.19	3.06	3.32	4.8	3.64	4.72	5.52
	CIC_{KLe}	5.14	5.23	6.23	6.24	6.88	8.22	9.57	12.41
	pen_n^B	3.04	3.12	3.69	4.97	5.11	5.53	6.72	9.65
	pen_n^{AIC}	3.01	3	3	3	4.96	4.94	4.99	5
Hell	$CIC_{L_2 v}$	0.2066	0.1742	0.1711	0.1689	0.3874	0.4089	0.3328	0.2466
	CIC_{KLv}	0.2	0.1671	**0.1373**	**0.1045**	**0.2672**	**0.2199**	**0.183**	**0.1344**
	$CIC_{L_2 e}$	0.2809	0.2646	0.2606	0.2489	0.4583	0.4891	0.4484	0.3928
	CIC_{KLe}	0.2603	0.2376	0.2163	0.1965	0.386	0.3464	0.3173	0.2727
	pen_n^B	**0.179**	0.1648	0.1437	0.1077	0.271	0.2315	0.1994	0.146
	pen_n^{AIC}	0.1794	**0.1639**	0.1542	0.1484	0.2726	0.2437	0.2264	0.2158

In applications, pen_n^B would be preferred since it is much faster to compute, while AIC is too greedy with the bins when n is large.

All the distributions we have used are smooth, but there is no a priori reason why the CIC wouldn't work in more general settings when we use the subsample to estimate the bias. It would be of interest to run a larger scale comparison study with other bin selection procedures, using e.g. the distributions described in figure 1 of Rozenholc et al. (2010).

Chapter 5

Summing it up

From this it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

- Alfred North Whitehead and Bertrand Russell in p. 379 of the *Principia Mathematica*

5.1 On the `R` programs

5.1.1 Manski's maximum score estimator

In Appendix B on page 167 we supply the code for a small program which can be used for Manski's estimator in one and two dimensions. This uses the formula convention from `R`, and can be called by using $\texttt{mms}(\texttt{resp} \sim \texttt{cov1} + \texttt{cov2})$, where $\texttt{resp}, \texttt{cov1}, \texttt{cov2}$ are the responses and covariate vectors. In addition to this functionality, we have supplied plotting generics. The program is ready to use.

5.1.2 Histograms

A program for computing irregular L_2/KL-histograms is supplied in Appendix A on page 151. We support the calculation of exact histograms through the dynamic programming algorithm (with the important modulating $\delta > 0$ discussed in Section 4.7), approximate histograms through the coordinate search algorithm, and pre-smoothed histograms by means of the coordinate search algorithm combined with the Gaussian copula kernel density estimator if Section 4.1. Histograms are only supported for data on $[0, 1]$. Additionally, we

provide functions to compute the subsampling based CIC. Furthermore, we have provided plotting and printing generics for the new `histogram` class of objects. A function for the computation of the Gaussian copula kernel density estimator with the normal reference is supplied.

5.2 Things one might do

In the authors opinion, two unfinished problems stand out as worth completing in this thesis: Finding the breakdown point of Manski's estimator for higher dimensions, and the L_1-consistency of the irregular histograms. We describe some other things it could be worth checking out.

5.2.1 Manski's maximum score estimator

As already said, it remains to find the breakdown point in dimensions higher than one. An important part of this work is to analyse the multidimensional colour depth properly. Other robustness properties should also be investigated, for instance the maximal asymptotic bias (Maronna et al., 2006, chapter 3.3). One can attempt to mimic the algorithm in van Kreveld et al. (1999), which solves the deepest regression problem in covariate dimension $d = 1$ in $O(n \log^2 n)$ time, in order to compute Manski's estimator when $d = 2$ faster than we have with our algorithm. Can one constructively prove constructively that the bootstrap fails, by doing something similar as in the of the proof of the bootstrap failure for the uniform distribution? It might be possible to do this by getting bounds on how far away the bootstrapped solution sets can be from the real solution sets as $n \to \infty$, maybe by using some clever combinatorial argument.

Problems with $d > 2$ hasn't been touched in this thesis, and should be studied. For instance, Florios and Skouras (2008) (F&L) show that the estimates of Horowitz (1993) for his work-trip data ($d = 4$) are probably wrong. However, F&Ls estimates conflict sharply with the "safe" estimates of Horowitz' smoothed maximum score estimator (1992), and they fail to mention the geometry of Manski's estimator which makes it possible for the solution sets to be very large. This could have practical consequences: The interpretation of F&L differ dramatically from Horowitz'. A proper empirical (or theoretical) study of Manski's estimator in $d > 2$ would give us a greater understanding of how the solution sets behaves, and whether it is important to calculate the whole solution set or not. Also, an implemented algorithm would answer the question of whether both Horowitz and F&L are "right" at the same time. (As F&L

aptly demonstrated, Horowitz' estimates, originated from an approximation algorithm, were far off the mark. This is consistent with our warning against approximating algorithms for this problem, recall Theorem 3.4.4 on page 53.)

An `R`-package for Manski's estimator could be developed. Such a package should include a function for the calculation of entire solution sets for $d > 2$, and some kind of error reporting. Currently, only subsampling and the smoothed bootstrap can do the job. Finally, we need a model selection procedure. This could work by counting the number of correct classifications compared to the expected number of correct classification given covariates with no information.

5.2.2 Histograms

One could mimic the approach in Banerjee and McKeague (2007) in order to obtain analogues of their RSS confidence intervals. It still remains to be proved rigorously that the bootstrap is inconsistent, and it should be shown that the smoothed bootstrap is consistent for some rate and additional constraints on the fs. Also, the properties of the pre-smoothed histograms should be investigated, and it could be worthwhile to develop bin selection procedures for such kinds of histograms, especially with specific goals (like compressing the data for a specific application) in mind. It might also be worth it to investigate the BHHJ histograms, especially since a master theorem for the BHHJ histograms would include everything about both KL and L_2 histograms, including a generalised CIC for the entire family of divergences. Also the Kolmogorov distance can be investigated, despite its apparent drawbacks: For instance, it might converge at a different rate than the BHHJ histograms, which almost certainly would converge with the cube root rate for any choice of α. More importantly, the L_1-histograms should be investigated.

There is much that could be done on the algorithmic side. Like investigating the idea mentioned in Section 4.6 on "ensmarting" the dynamic programming algorithm. Also completely different algorithms might be viable. If we could get the run time down to for instance $O(nk \log n)$, subsampling would be far more viable. It would also be nice to find efficient approximation algorithms with proven worst case performance both with regards to the objective function and the estimates. These algorithms should be incorporated with an associated δ bounding the minimum distance between split points.

There is a package for irregular histograms on `CRAN`, namely `histogram` of Mildenberger et al. (2009), a package which can do all the different forms of bin selection procedures considered in Rozenholc et al. (2010). In this thesis, we

have developed the CIC, quantile histograms and L_2-histograms, which is not included in this package. Also, the package doesn't handle the instability issue in clear-cut way, and hasn't implemented the dynamic programming algorithm in C++, making it too slow for practical use. Finally, as pre-smoothing dramatically reduces the Hellinger risk and the MISE for k large enough, it should be made publicly available.

Bibliography

Abrevaya, J. and Huang, J. (2005), 'On the bootstrap of the maximum score estimator', *Econometrica* pp. 1175–1204.

Amaldi, E. and Kann, V. (1994), On the approximability of finding maximum feasible subsystems of linear systems, *in* 'STACS 94', Springer, pp. 521–532.

Amaldi, E. and Kann, V. (1995), 'The complexity and approximability of finding maximum feasible subsystems of linear relations', *Theoretical computer science* **147**(1), 181–210.

Andrews, D. F. and Hampel, F. R. (2015), *Robust estimates of location: survey and advances*, Princeton University Press.

Ashkenas, J. and Park, H. (2014), 'The race gap in america's police departments', `http://www.nytimes.com/interactive/2014/09/03/us/the-race-gap-in-americas-police-departments.html`. Web. 24 Sept. 2015.

Athreya, K. (1987), 'Bootstrap of the mean in the infinite variance case', *The Annals of Statistics* pp. 724–731.

Atilgan, T. (1990), 'On derivaton and application of aic as a data-based criterion for histograms', *Communications in Statistics-Theory and Methods* **19**(3), 885–903.

Babu, G. J. and Rao, C. R. (1988), 'Joint asymptotic distribution of marginal quantiles and quantile functions in samples from a multivariate population', *Journal of Multivariate Analysis* **27**(1), 15–23.

Banerjee, M. and McKeague, I. W. (2007), 'Confidence sets for split points in decision trees', *The Annals of Statistics* **35**(2), 543–574.

Basu, A., Harris, I. R., Hjort, N. L. and Jones, M. C. (1998), 'Robust and efficient estimation by minimising a density power divergence', *Biometrika* **85**(3), 549–559.

Beran, R. (1982), 'Estimated sampling distributions: the bootstrap and competitors', *The Annals of Statistics* pp. 212–225.

Beran, R. (1983), *Bootstrap methods in statistics*, Univ., Sonderforschungsbereich 123.

Bickel, P. J. and Freedman, D. A. (1981), 'Some asymptotic theory for the bootstrap', *The Annals of Statistics* pp. 1196–1217.

Billingsley, P. (2008), *Probability and measure*, John Wiley & Sons.

Billingsley, P. (2013), *Convergence of probability measures*, John Wiley & Sons.

Birgé, L. and Rozenholc, Y. (2006), 'How many bins should be put in a regular histogram', *ESAIM: Probability and Statistics* **10**, 24–45.

Büchlmann, P. and Yu, B. (2002), 'Analyzing bagging', *Annals of Statistics* pp. 927–961.

Chamberlain, G. (1986), 'Asymptotic efficiency in semi-parametric models with censoring', *journal of Econometrics* **32**(2), 189–218.

Chen, S. X. (1999), 'Beta kernel estimators for density functions', *Computational Statistics & Data Analysis* **31**(2), 131–145.

Chernoff, H. (1964), 'Estimation of the mode', *Annals of the Institute of Statistical Mathematics* **16**(1), 31–41.

Chinneck, J. W. (2007), *Feasibility and Infeasibility in Optimization:: Algorithms and Computational Methods*, Vol. 118, Springer Science & Business Media.

Claeskens, G. and Hjort, N. L. (2008), *Model selection and model averaging*, Vol. 330, Cambridge University Press Cambridge.

Conforti, M., Cornuéjols, G. and Zambelli, G. (2014), *Integer programming*, Vol. 271, Springer Berlin.

Dasgupta, S., Papadimitriou, C. H. and Vazirani, U. (2006), *Algorithms*, McGraw-Hill, Inc.

Delgado, M. A., Rodriguez-Poo, J. M. and Wolf, M. (2001), 'Subsampling inference in cube root asymptotics with an application to manski's maximum score estimator', *Economics Letters* **73**(2), 241–250.

Devroye, L. and Györfi, L. (1985), *Nonparametric density estimation: the L1 view*, Vol. 119, John Wiley & Sons Inc.

Devroye, L., Györfi, L. and Lugosi, G. (2013), *A probabilistic theory of pattern recognition*, Vol. 31, Springer Science & Business Media.

Devroye, L. and Lugosi, G. (2012), *Combinatorial methods in density estimation*, Springer Science & Business Media.

Donoho, D. L. and Gasko, M. (1992), 'Breakdown properties of location estimates based on halfspace depth and projected outlyingness', *The Annals of Statistics* pp. 1803–1827.

Donoho, D. L. and Huber, P. J. (1983), 'The notion of breakdown point', *A Festschrift for Erich L. Lehmann* **157184**.

Eddelbuettel, D. and François, R. (2011), 'Rcpp: Seamless R and C++ integration', *Journal of Statistical Software* **40**(8), 1–18.
URL: *http://www.jstatsoft.org/v40/i08/*

Edelsbrunner, H. and Souvaine, D. L. (1990), 'Computing least median of squares regression lines and guided topological sweep', *Journal of the American Statistical Association* **85**(409), 115–119.

Efron, B. (1979), 'Bootstrap methods: another look at the jackknife', *The annals of Statistics* pp. 1–26.

Florios, K. and Skouras, S. (2008), 'Exact computation of max weighted score estimators', *Journal of Econometrics* **146**(1), 86–91.

Folland, G. B. (1984), *Real analysis: modern techniques and their applications*, John Wiley & Sons.

Garey, M. R., Johnson, D. S. and Stockmeyer, L. (1976), 'Some simplified np-complete graph problems', *Theoretical computer science* **1**(3), 237–267.

Geenens, G. (2014), 'Probit transformation for kernel density estimation on the unit interval', *Journal of the American Statistical Association* **109**(505), 346–358.

Groeneboom, P., Jongbloed, G. and Wellner, J. A. (2014), *Nonparametric estimation under shape constraints*, Vol. 38, Cambridge University Press.

Groeneboom, P. and Wellner, J. A. (2001), 'Computing chernoff's distribution', *Journal of Computational and Graphical Statistics* **10**(2).

Hao, L. and Naiman, D. Q. (2007), *Quantile regression*, number 149, Sage.

Hastie, T., Tibshirani, R., Friedman, J. and Franklin, J. (2005), 'The elements of statistical learning: data mining, inference and prediction', *The Mathematical Intelligencer* **27**(2), 83–85.

Hettmansperger, T. P. and Sheather, S. J. (1992), 'A cautionary note on the method of least median squares', *The American Statistician* **46**(2), 79–83.

Hjort, N. L. (2007), Model selection for cube root asymptotics, *in* U. Gather, P. Hall and H.-R. Künsch, eds, 'Reassessing the Paradigms of Statistical Model-Building'. Workshop report, accessed 7/10 2015.

Hjort, N. L. and Jones, M. C. (1996), 'Locally parametric nonparametric density estimation', *The Annals of Statistics* pp. 1619–1647.

Horowitz, J. L. (1992), 'A smoothed maximum score estimator for the binary response model', *Econometrica: journal of the Econometric Society* pp. 505–531.

Horowitz, J. L. (1993), 'Semiparametric estimation of a work-trip mode choice model', *Journal of Econometrics* **58**(1), 49–70.

Hössjer, O., Rousseeuw, P. J. and Croux, C. (1994), 'Asymptotics of the repeated median slope estimator', *The Annals of Statistics* pp. 1478–1501.

Hromkovič, J. (2013), *Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics*, Springer Science & Business Media.

Huber, P. J. (1981), *Robust statistics*, Wiley.

Johnson, D. S. and Preparata, F. P. (1977), The densest hemisphere problem., Technical report, DTIC Document.

Jones, M. C. and Henderson, D. A. (2007*a*), 'Kernel-type density estimation on the unit interval', *Biometrika* **94**(4), 977–984.

Jones, M. C. and Henderson, D. A. (2007*b*), 'Kernel-type density estimation on the unit interval', http://oro.open.ac.uk/22541/1/kernel-type.pdf. Accessed: 2015-07-15.

Kanazawa, Y. (1988), 'An optimal variable cell histogram', *Communications in Statistics-Theory and Methods* **17**(5), 1401–1422.

Kechris, A. (2012), *Classical descriptive set theory*, Vol. 156, Springer Science & Business Media.

Kim, J. and Pollard, D. (1990), 'Cube root asymptotics', *The Annals of Statistics* pp. 191–219.

Knight, K. (1989), 'On the bootstrap of the sample mean in the infinite variance case', *The Annals of Statistics* pp. 1168–1175.

Konishi, S. and Kitagawa, G. (2008), *Information criteria and statistical modeling*, Springer Science & Business Media.

Kosorok, M. R. (2007), *Introduction to empirical processes and semiparametric inference*, Springer Science & Business Media.

Kosorok, M. R. (2008), Bootstrapping the grenander estimator, *in* 'Beyond parametrics in interdisciplinary research: Festschrift in honor of Professor Pranab K. Sen', Institute of Mathematical Statistics, pp. 282–292.

Lee, S. M. S. and Pun, M. (2006), 'On m out of n bootstrapping for nonstandard m-estimation with nuisance parameters', *Journal of the American Statistical Association* **101**(475), 1185–1197.

Léger, C. and MacGibbon, B. (2006), 'On the bootstrap in cube root asymptotics', *Canadian Journal of Statistics* **34**(1), 29–44.

Loader, C. (2013), *locfit: Local Regression, Likelihood and Density Estimation.* R package version 1.5-9.1.
URL: *http://CRAN.R-project.org/package=locfit*

Lugosi, G. and Nobel, A. (1996), 'Consistency of data-driven histogram methods for density estimation and classification', *The Annals of Statistics* **24**(2), 687–706.

Manski, C. F. (1975), 'Maximum score estimation of the stochastic utility model of choice', *Journal of econometrics* **3**(3), 205–228.

Manski, C. F. (1985), 'Semiparametric analysis of discrete response: Asymptotic properties of the maximum score estimator', *Journal of econometrics* **27**(3), 313–333.

Manski, C. F. and Thompson, T. S. (1986), 'Operational characteristics of maximum score estimation', *Journal of Econometrics* **32**(1), 85–108.

Maronna, R. A., Martin, D. and Yohai, V. (2006), *Robust statistics*, John Wiley & Sons, Chichester. ISBN.

Matoušek, J. (2002), *Lectures on discrete geometry*, Vol. 212, Springer New York.

Maxima (2014), 'Maxima, a computer algebra system. version 5.34.1'.
URL: *http://maxima.sourceforge.net/*

Megginson, R. E. (2012), *An introduction to Banach space theory*, Vol. 183, Springer Science & Business Media.

Mendelson, E. (2009), *Introduction to mathematical logic*, CRC press.

Mersmann, O. (2014), *microbenchmark: Accurate Timing Functions*. R package version 1.4-2.
URL: *http://CRAN.R-project.org/package=microbenchmark*

Mildenberger, T., Rozenholc, Y. and Zasada., D. (2009), *histogram: Construction of regular and irregular histograms with different options for automatic choice of bins*. R package version 0.0-23.
URL: *http://CRAN.R-project.org/package=histogram*

Nobel, A. et al. (1996), 'Histogram regression estimation using data-dependent partitions', *The Annals of Statistics* **24**(3), 1084–1105.

Papadimitriou, C. H. (2003), *Computational complexity*, John Wiley and Sons Ltd.

Pinkse, C. (1993), 'On the computation of semiparametric estimates in limited dependent variable models', *Journal of Econometrics* **58**(1), 185–205.

Politis, D. N. and Romano, J. P. (1994), 'Large sample confidence regions based on subsamples under minimal assumptions', *The Annals of Statistics* pp. 2031–2050.

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

Pollard, D. (1991), 'Asymptotics for least absolute deviation regression estimators', *Econometric Theory* **7**(02), 186–199.

R Core Team (2014), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*

Rosenthal, J. S. (2006), *A first look at rigorous probability theory*, World Scientific.

Ross, S. M. (2014), *Introduction to probability models*, Academic press.

Rousseeuw, P. J. (1984), 'Least median of squares regression', *Journal of the American statistical association* **79**(388), 871–880.

Rousseeuw, P. J. and Hubert, M. (1999*a*), 'Depth in an arrangement of hyperplanes', *Discrete & Computational Geometry* **22**(2), 167–176.

Rousseeuw, P. J. and Hubert, M. (1999*b*), 'Regression depth', *Journal of the American Statistical Association* **94**(446), 388–402.

Rousseeuw, P. J. and Leroy, A. M. (2005), *Robust regression and outlier detection*, Vol. 589, John Wiley & Sons.

Rozenholc, Y., Mildenberger, T. and Gather, U. (2010), 'Combining regular and irregular histograms by penalized likelihood', *Computational Statistics & Data Analysis* **54**(12), 3313–3323.

Rudin, W. (1987), *Real and complex analysis*, Tata McGraw-Hill Education.

Scheffé, H. (1947), 'A useful convergence theorem for probability distributions', *The Annals of Mathematical Statistics* pp. 434–438.

Seijo, E. and Sen, B. (2011), 'Bootstrapping manski's maximum score estimator', *arXiv preprint arXiv:1105.1976* .

Sen, B., Banerjee, M. and Woodroofe, M. (2010), 'Inconsistency of bootstrap: The grenander estimator', *The Annals of Statistics* **38**(4), 1953–1977.

Shao, J. (2007), *Mathematical statistics, 2nd edition*, Springer.

Shao, J. and Tu, D. (2012), *The jackknife and bootstrap*, Springer Science & Business Media.

Shibata, R. (1997), 'Bootstrap estimate of kullback-leibler information for model selection', *Statistica Sinica* **7**(2), 375–394.

Siegel, A. F. (1982), 'Robust regression using repeated medians', *Biometrika* **69**(1), 242–244.

Silverman, B. W. (1986), *Density estimation for statistics and data analysis*, Vol. 26, CRC press.

SSB (2014), 'Church data on statistics norway', `https://www.ssb.no/kultur-og-fritid/statistikker/kirke_kostra/aar/2014-05-06`. Web. 24 Sept. 2015.

Steele, J. and Steiger, W. (1986), 'Algorithms and complexity for least median of squares regression', *Discrete Applied Mathematics* **14**(1), 93–100.

Stone, M. (1977), 'An asymptotic equivalence of choice of model by cross-validation and akaike's criterion', *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 44–47.

Stroustrup, B. (1986), *The C++ programming language*, Pearson Education India.

Swaab, R. I., Schaerer, M., Anicich, E. M., Ronay, R. and Galinsky, A. D. (2014), 'The too-much-talent effect team interdependence determines when more talent is too much or not enough', *Psychological science* **25**(8), 1581–1591.

Theussl, S., Hornik, K., Buchta, C., Makhorin, A., Davis, T. A., Sorensson, N., Adler, M., Gailly, J.-l. and Theussl, M. S. (2015), 'Package 'rglpk".

Tukey, J. W. (1975), Mathematics and the picturing of data, *in* 'Proceedings of the international congress of mathematicians', Vol. 2, pp. 523–531.

van Aelst, S. and Rousseeuw, P. J. (2000), 'Robustness of deepest regression', *Journal of Multivariate Analysis* **73**(1), 82–106.

van der Vaart, A. W. (2000), *Asymptotic statistics*, Vol. 3, Cambridge university press.

van der Vaart, A. W. and Wellner, J. A. (1996), *Weak Convergence*, Springer.

van Kreveld, M., Mitchell, J. S., Rousseeuw, P., Sharir, M., Snoeyink, J. and Speckmann, B. (1999), Efficient algorithms for maximum regression depth, *in* 'Proceedings of the fifteenth annual symposium on Computational geometry', ACM, pp. 31–40.

Van Ryzin, J. (1973), 'A histogram method of density estimation', *Communications in Statistics-Theory and Methods* **2**(6), 493–506.

Vapnik, V. N. and Chervonenkis, A. Y. (1971), 'On the uniform convergence of relative frequencies of events to their probabilities', *Theory of Probability & Its Applications* **16**(2), 264–280.

Wand, M. P. and Jones, M. C. (1994), *Kernel smoothing*, Crc Press.

Wegener, I. (2005), *Complexity theory: exploring the limits of efficient algorithms*, Springer Science & Business Media.

Weiss, M. A. (1998), *Data structures and algorithm analysis in Java*, Addison-Wesley Longman Publishing Co., Inc.

Yeh, A. B. and Singh, K. (1997), 'Balanced confidence regions based on tukey's depth and the bootstrap', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **59**(3), 639–652.

Zhao, L. C., Krishnaiah, P. R. and Chen, X. R. (1991), 'Almost sure l_r-norm convergence for data-based histogram density estimates', *Theory of Probability & Its Applications* **35**(2), 396–403.

Appendix A

Histogram code

We present all the code involved in the calculation of histograms.

1. The C++ code for the computation of exact histograms via the DP algorithm,
2. the C++ code for the coordinate search,
3. the R code for the Gaussian copula KDE,
4. the R wrapper with generics,

A.1 C++ code for the DP algorithm

```
1  #include <cmath>
2  #include <vector>
3  #include <Rcpp.h>
4  #include <iostream>
5
6  using namespace std;
7
8  /* Functions for combos of real and l2*/
9
10 double (*residual)(int,int,double*,int,double,double);
11 double (*residual_final)(int,double*,int,double,double);
12
13 /* These correspond to Kullback-Leibler and equal weights. */
14 double rkf(int i,int j,double* data,int len,double k,double lim){
15   double a = -(j-i);
16   double b = data[j]-data[i];
17   if (b < lim) return -0.1/0.0;
18   else return(a*log(b));
19 }
20
21 double rkf_final(int i,double* data,int len,double k,double lim){
22   double a = -(len-i);
```

```
23    double b = 1-data[i];
24    if (b < lim) return -0.1/0.0;
25    else return(a*log(b));
26  }
27
28  /* These correspond to L2 and equal weights. */
29  double rlf(int i,int j,double* data,int len,double k,double lim){
30    double a = 2*(j-i)/((double) len)-1/k;
31    double b = data[j]-data[i];
32    if (b < lim) return -0.1/0.0;
33    else return(a/b);
34  }
35
36  double rlf_final(int i,double* data,int len,double k,double lim){
37    double a = 2*(len-i)/((double) len)-1/k;
38    double b = 1-data[i];
39    if (b < lim) return -0.1/0.0;
40    else return(a/b);
41  }
42
43  /* ... and these correspond to KL weights and splits! */
44  double rkt(int i,int j,double* data,int len,double k,double lim){
45    double b = data[j]-data[i];
46
47    if (b < lim) return -0.1/0.0;
48
49    /* A special case occurs whener i = 0. Then log(j-i) should equal
50       log(j-1) instead */
51    if (i == 0) i = 1;
52
53    /* This is required in order to avoid log(0)-c, which is indeterminate. */
54    if (i == 1 && j == 1) return -0.1/0.0;
55    return((j-i)*(log(j-i)-log(len)-log(b)));
56  }
57
58  double rkt_final(int i,double* data,int len,double k,double lim){
59    double b = 1-data[i];
60
61    if (b < lim) return -0.1/0.0;
62
63    if (i == 0) i = 1;
64    return((len-i)*(log(len-i)-log(len)-log(b)));
65  }
66
67  /* Finally, L2 weights and splits. */
68  double rlt(int i,int j,double* data,int len,double k,double lim){
69    double b = data[j]-data[i];
70
71    if (b < lim) return -0.1/0.0;
72
73    if (i == 0) i = 1;
74    if (i == 1 && j == 1) return -0.1/0.0;
75    double a = pow(j-i,2);
76    return(a/b);
77  }
78
79  double rlt_final(int i,double* data,int len,double k,double lim){
```

```
80    double b = 1-data[i];
81
82    if (b < lim) return -0.1/0.0;
83
84    if (i == 0) i = 1;
85    double a = pow(1-i,2);
86    return(a/b);
87  }
88
89  // [[Rcpp::export]]
90  Rcpp::NumericMatrix cpp_exact_alg(bool real_hist,bool l2,Rcpp::NumericVector
        data,int len,double k,double lim){
91
92      /* residual and residual_final point to the functions needed in
            maximisation.
93       * The definition of these functions vary as L2 and real varies, every
             other aspect
94       * of the algorithm stays constant.
95       */
96
97      if (real_hist) {
98          if (l2) {
99              residual = &rlt;
100             residual_final = &rlt_final;
101         }
102         else {
103             residual = &rkt;
104             residual_final = &rkt_final;
105         }
106     }
107
108     else {
109         if (l2) {
110             residual = &rlf;
111             residual_final = &rlf_final;
112
113         }
114         else {
115             residual = &rkf;
116             residual_final = &rkf_final;
117         }
118     }
119
120     /* We define the pretty "matrix" of estimates indices, with the correct
            dimensions.
121      * This matrix contains the ML / L2 estimate indices, as j-ary vectors. The
             (i,j)-th
122      * element corresponds to ML-estimate with k=j and data[0:j]. */
123
124     vector < vector < vector < int > > > estimates;
125     estimates.resize(len+2);
126     for (int i = 0; i <= len+1; i++){
127         estimates[i].resize(k);
128         for (int j = 0;j<(k-1);j++){
129             estimates[i][j].resize(j+1);
130         }
131     }
```

```
132
133     /* This matrix contains the optimal objective values at (i,j) instead of
            the estimates.
134      * It doesn't need crazy dimensions. */
135
136     vector < vector < double > > objective;
137     objective.resize(len+2);
138     for (int i = 0; i <= len+1; i++){
139         objective[i].resize(k-1);
140     }
141
142     /* We begin on the actual algorithm. */
143
144
145     /* The special case when j = 0. Needed in order to get initial values. Also
            , the case when i = len+1 is
146      * extra special. However, it is only needed when k = 2. */
147
148     vector <int> est;
149     est.resize(1);
150
151     for (int i=2; i<=len;i++){
152         double maxer = residual(0,1,data.begin(),len,k,lim)+residual(1,i,data.
                begin(),len,k,lim);
153         double temp_max;
154         int ind = 1;
155         for (int p=2; p<i; p++){
156             temp_max = residual(0,p,data.begin(),len,k,lim)+residual(p,i,data.
                    begin(),len,k,lim);
157             if (temp_max > maxer){
158                 maxer = temp_max;
159                 ind = p;
160             }
161         }
162
163         est[0] = ind;
164         estimates[i][0] = est;
165         objective[i][0] = maxer;
166     }
167
168     /* Now we can handle the case when j=0 and i = len + 1! */
169
170     int i = len + 1;
171     double maxer = residual(0,1,data.begin(),len,k,lim)+residual_final(1,data.
            begin(),len,k,lim);
172     double temp_max;
173     int ind = 1;
174     for (int l=2; l<i; l++){
175         temp_max = residual(0,l,data.begin(),len,k,lim)+residual_final(l,data.
                begin(),len,k,lim);
176         if (temp_max > maxer){
177             maxer = temp_max;
178             ind = l;
179         }
180     }
181
182     est[0] = ind;
```

```
183     estimates[i][0] = est;
184     objective[i][0] = maxer;
185
186
187
188     /* The main program follows, the generation of the two matrices objective
            and estimates.
189      * We begin with iteration through j, as the calculation of estimates[i,j]
            depends on knowing (almost) every
190      * value estimates[i',j], with i' < i. */
191
192     for (int j=1; j<(k-1);j++){
193         // Initialisation of variables used in loop.
194         vector <int> est;
195         est.resize(j+1);
196         double maxer, temp_max;
197         int ind;
198
199
200         /* Calculates the matrix for every term except i=len+1, which is a
                special case. */
201         for (int i=j+2; i<len+1;i++){
202
203             /* Given an i, we wish to find the best estimates for data[0,i]
204              * given that k=j. We use i = j+2 in order to have enough points to
                     fit the data:
205              * The "best" case is that (i-1) is the optimal index, and this one
                     needs j points
206              * of data below it. We start with i-1, and continue through the
                     loop. */
207
208             maxer = objective[i-1][j-1]+residual(i-1,i,data.begin(),len,k,lim);
209             ind = i-1;
210
211             /* We have the condition p>=j+1 for the same reason as above. If p
                    = j or less,
212              * there will not be the needed j points below it. */
213
214             for (int p = (i-2);p>=(j+1);p--){
215                 temp_max = objective[p][j-1]+residual(p,i,data.begin(),len,k,
                        lim);
216                 if (temp_max > maxer){
217                     maxer = temp_max;
218                     ind = p;
219                 }
220             }
221
222             /* Our resulting objective is maxer, while our indices are, the
                    winning estimates'
223              * indices concatenated with with the index which makes them win.
                     */
224
225             objective[i][j] = maxer;
226             est = estimates[ind][j-1];
227             est.push_back(ind);
228             estimates[i][j] = est;
229         }
```

```
230
231         /* We proceed with the special case i = len+1. The reason why this is a
                special case is
232          * that Pn(1) = Pn(x_n), the final observation in the data set. This
                makes the residual function
233          * return incorrect values.
234         */
235
236         int i = len+1;
237         maxer = objective[i-1][j-1];
238         ind = i-1;
239
240         for (int p = (i-2);p>=(j+1);p--){
241             temp_max = objective[p][j-1]+residual_final(p,data.begin(),len,k,
                    lim);
242             if (temp_max > maxer){
243                 maxer = temp_max;
244                 ind = p;
245             }
246         }
247
248         objective[i][j] = maxer;
249         est = estimates[ind][j-1];
250         est.push_back(ind);
251         estimates[i][j] = est;
252
253         /* We print out the content of the vector. */
254
255     }
256
257
258     Rcpp::NumericMatrix xx = Rcpp::NumericMatrix(Rcpp::Dimension(k-1, k-1));
259
260     for (int i=0;i<k-1;i++){
261       for (int j=0;j<=i;j++) {
262         xx(i,j) = estimates[len+1][i][j];
263       }
264     }
265
266     return(xx);
267
268 }
```

A.2 C++ code for the coordinate search

```
1  #include <cmath>
2  #include <vector>
3  #include <Rcpp.h>
4
5  using namespace std;
6
7  /* Functions for combos of real and l2*. These pointers are needed in order to
8   * avoid many repeated if calls and make the code more readable. */
9
10 double (*residual)(int,int,double*,int,double,double);
11 double (*residual_final)(int,double*,int,double,double);
```

```
12
13 /* These correspond to Kullback-Leibler and equal weights. */
14 double rkf(int i,int j,double* data,int len,double k,double lim){
15   double a = -(j-i);
16   double b = data[j]-data[i];
17   if (b < lim) return -0.1/0.0;
18   else return(a*log(b));
19 }
20
21 double rkf_final(int i,double* data,int len,double k,double lim){
22   double a = -(len-i);
23   double b = 1-data[i];
24   if (b < lim) return -0.1/0.0;
25   else return(a*log(b));
26 }
27
28 /* These correspond to L2 and equal weights. */
29 double rlf(int i,int j,double* data,int len,double k,double lim){
30   double a = 2*(j-i)/((double) len)-1/k;
31   double b = data[j]-data[i];
32   if (b < lim) return -0.1/0.0;
33   else return(a/b);
34 }
35
36 double rlf_final(int i,double* data,int len,double k,double lim){
37   double a = 2*(len-i)/((double) len)-1/k;
38   double b = 1-data[i];
39   if (b < lim) return -0.1/0.0;
40   else return(a/b);
41 }
42
43 /* ... and these correspond to KL weights and splits! */
44 double rkt(int i,int j,double* data,int len,double k,double lim){
45   double b = data[j]-data[i];
46   if (b < lim) return -0.1/0.0;
47   else return((j-i)*(log(j-i)-log(len)-log(b)));
48 }
49
50 double rkt_final(int i,double* data,int len,double k,double lim){
51   double b = 1-data[i];
52   if (b < lim) return -0.1/0.0;
53   else return((len-i)*(log(len-i)-log(len)-log(b)));
54 }
55
56 /* Finally, L2 weights and splits. */
57 double rlt(int i,int j,double* data,int len,double k,double lim){
58   double a = pow(j-i,2);
59   double b = data[j]-data[i];
60   if (b < lim) return -0.1/0.0;
61   else return(a/b);
62 }
63
64 double rlt_final(int i,double* data,int len,double k,double lim){
65   double a = pow(1-i,2);
66   double b = 1-data[i];
67   if (b < lim) return -0.1/0.0;
68   else return(a/b);
```

```
69 }
70
71
72 // [[Rcpp::export]]
73 Rcpp::NumericVector cpp_greedy_alg(bool real_hist,bool l2,Rcpp::NumericVector
       data,
74                                    int len,double k,int modulator,Rcpp::
                                           NumericVector init,
75                                    double lim){
76
77     /* residual and residual_final point to the functions needed in
           maximisation.
78      * The definition of these functions vary as L2 and real varies, every
            other aspect
79      * of the algorithm stays constant.
80      */
81
82     if (real_hist) {
83         if (l2) {
84             residual = &rlt;
85             residual_final = &rlt_final;
86         }
87         else {
88             residual = &rkt;
89             residual_final = &rkt_final;
90         }
91     }
92
93     else {
94         if (l2) {
95             residual = &rlf;
96             residual_final = &rlf_final;
97
98         }
99         else {
100            residual = &rkf;
101            residual_final = &rkf_final;
102        }
103    }
104
105    /* We define the pretty "matrix" estimatse indices, with the correct
           dimensions.
106     * This matrix contains the ML / L2 estimate indices, as j-ary vectors. The
            (i,j)-th
107     * element corresponds to ML-estimate with k=j and data[0:j]. */
108
109    vector <int> estimates;
110    vector <int> test_estimates;
111
112    estimates.resize((int) k+1);
113    test_estimates.resize((int) k+1);
114    estimates[0] = 0;
115    estimates[k] = 1;
116
117    for(int i = 1;i<k;i++){
118      estimates[i] = init[i-1];
119    }
```

```
120
121     int ended = 0;
122     int over;
123     int under;
124     double max;
125     double temp;
126     for (int j = 0;j<modulator*k;j++){
127
128       test_estimates = estimates;
129
130       /* The loop takes care of all the values except the final. */
131
132       for (int i=1;i<(k-1);i++){
133         over = estimates[i+1];
134         under = estimates[i-1];
135         max = -0.1/0.0;
136         for (int p = under; p<over;p++){
137           if (data[p]-data[under]>lim && data[over]-data[p]>lim){
138             temp = residual(under,p,data.begin(),len,k,lim)+residual(p,over,
                    data.begin(),len,k,lim);
139           }
140           else temp = -0.1/0.0;
141           if (max < temp) {
142             max = temp;
143             estimates[i] = p;
144           }
145         }
146
147       }
148
149       /* And now is the time for the last value. */
150       int i = (k-1);
151       under = estimates[i-1];
152       max = -0.1/0.0;
153       for (int p = under; p<(len+1);p++){
154         if (data[p]-data[under]>lim && 1-data[p]>lim){
155             temp = residual(under,p,data.begin(),len,k,lim)+residual_final(p,
                    data.begin(),len,k,lim);
156           }
157         else temp = -0.1/0.0;
158         if (max < temp) {
159             max = temp;
160             estimates[i] = p;
161           }
162       }
163
164       /* We test the break condition. */
165
166       if (test_estimates == estimates){
167         ended = j;
168         break;
169       }
170
171     }
172
173     Rcpp::NumericVector xx((int) k);
174
```

```
175     for (int i=0;i<k-1;i++){
176         xx[i] = estimates[i+1];
177       }
178
179     xx[k-1] = ended;
180
181     return(xx);
182
183 }
```

A.3 R code for the Gaussian copula KDE

```
1
2  # Functions for the Gaussian copula ------------------------------------------
3  rgc = function(n,X,rho){
4    N = length(X)
5    X = qnorm(X)
6    xs = rnorm(n,rho*X,rep(sqrt(1-rho^2),times=N),1882)
7    pnorm(xs)
8  }
9
10 dgc = Vectorize(function(x,X,rho){
11   inside = rho^2*(qnorm(x)^2+qnorm(X)^2)-2*rho*qnorm(x)*qnorm(X)
12   1/sqrt(1-rho^2)*exp(-inside/(2*(1-rho^2)))
13 })
14
15 dgcd = Vectorize(function(x,X,rho){
16   dgc(x,X,rho)*rho/(1-rho^2)*(-rho*qnorm(x)+qnorm(X))/dnorm(qnorm(x))
17 })
18
19
20 # Gaussian copula KDE functions ----------------------------------------------
21 hgcde = function(data,lims=NULL){
22   if (!is.null(lims)) data = (data-lims[1])/lims[2]
23   trans = qnorm(data)
24   trans = trans[trans != Inf & trans != -Inf]
25   s = sd(trans)
26   m = mean(trans)
27   n = length(data)
28   min(s*(2*m^2*s^2+3*(1-s^2)^2)^(-1/5)*n^(-1/5),0.5)
29 }
30
31 gcde = function(data,h=NULL,lims=NULL){
32   if (is.null(h)) h = hgcde(data)
33   if (!is.null(lims)) data = (data-lims[1])/lims[2]
34   else lims = c(0,1)
35   trans = qnorm(data)
36   data = data[trans != Inf & trans != -Inf]
37   rho = 1-h^2
38   function(x){
39     mean(dgc((x-lims[1])/lims[2],data,rho))/lims[2]
40   }
41 }
42
43 gcded = function(data,h=NULL,lims=NULL){
44   if (is.null(h)) h = hgcde(data)
```

```
45    if (!is.null(lims)) data = (data-lims[1])/lims[2]
46    else lims = c(0,1)
47    trans = qnorm(data)
48    data = data[trans != Inf & trans != -Inf]
49    rho = 1-h^2
50    function(x){
51      mean(dgcd((x-lims[1])/lims[2],data,rho))/lims[2]
52    }
53  }
54
55  hgcde = function(data,lims=NULL){
56    if (!is.null(lims)) data = (data-lims[1])/lims[2]
57    trans = qnorm(data)
58    trans = trans[trans != Inf & trans != -Inf]
59    s = sd(trans)
60    m = mean(trans)
61    n = length(data)
62    min(s*(2*m^2*s^2+3*(1-s^2)^2)^(-1/5)*n^(-1/5),0.5)
63  }
64
65  rgcde = function(n,data,h=NULL,lims=NULL){
66    if (!is.null(lims)) data = (data-lims[1])/lims[2]
67    else lims = c(0,1)
68    if (is.null(h)) h = hgcde(data)
69    samples = sample(data,n,replace=TRUE)
70    rho = 1-h^2
71    (rgc(n,samples,rho))*lims[2]+lims[1]
72  }
```

A.4 R wrapper and generics

```
1
2  library("Rcpp")
3  source("copula_base.R")
4  sourceCpp("coordinate.cpp")
5  sourceCpp("exact.cpp")
6
7  # Support functions and variables ------------------------------------------
8
9  `%||%` <- function(a, b) if (! is.null(a)) a else b
10
11 # Algorithms ---------------------------------------------------------------
12 coordHistogram = function(data,k,type="KL",weights="equal",lim=0.0001,
13                           modulator=10,init=NULL){
14    ret_object = list()
15    ret_object$specification = c(type=type,weights=weights)
16    ret_object$k = k
17    n = length(data)
18    type = (type == "L2")
19    weights = (weights!="equal")
20    init = (init %||% quantile(1:n,(1:(k-1))/k))
21    vals = cpp_greedy_alg(weights,type,c(0,data,1),n,k,
22              modulator=modulator,init=init,lim=lim)
23    as = data[vals[-k]]
24    ret_object$splits = as
25    if (!weights) {
```

```
26      ret_object$weights = rep(1/k,k)
27    } else {
28      ret_object$weights = c(sapply(1:k,
29                                    function(i) c(0,vals[-k]/n,1)[i+1]-c(0,vals[-
                                        k]/n,1)[i]))
30      ret_object$weights[1] = ret_object$weights[1] - 1/n
31      ret_object$weights[k] = ret_object$weights[k] + 1/n
32    }
33    ret_object$iterations = vals[k]
34    ret_object$method = "greedy"
35    ret_object$lim = lim
36    class(ret_object) = c("hist")
37    ret_object
38  }
39
40  dpHistogram = function(data,k,type="KL",weights="equal",lim=0.0001){
41    ret_object = list()
42    ret_object$specification = c(type=type,weights=weights)
43    ret_object$k = k
44    n = length(data)
45    type = (type=="L2")
46    weights = (weights!="equal")
47    vals = cpp_exact_alg(weights,type,c(0,data,1),n,k,lim=lim)
48    vals = vals[k-1,]
49    as = data[vals]
50    ret_object$splits = as
51    if (!weights) {
52      ret_object$weights = rep(1/k,k)
53    } else {
54      ret_object$weights = c(sapply(1:k,
55              function(i) c(0,vals[-k]/n,1)[i+1]-c(0,vals[-k]/n,1)[i]))
56      ret_object$weights[1] = ret_object$weights[1] - 1/n
57      ret_object$weights[k] = ret_object$weights[k] + 1/n
58    }
59    ret_object$method = "exact"
60    ret_object$lim = lim
61    class(ret_object) = c("hist")
62    ret_object
63  }
64
65  # "log likelihood"-function
      ------------------------------------------------
66
67  Pnm = function(obj,useData=data){
68    q      = obj$splits
69    w      = obj$weights
70    k      = obj$k
71    n      = length(useData)
72    Pn     = c(0,sapply(q,function(j) sum(useData<=j)/n),1)
73    qAug   = c(0,q,1)
74    if (obj$specification[1] == "KL") {
75      dis    = sapply(1:k,function(i) - log(qAug[i+1]-qAug[i]) + log(w[i]))
76      probs = sapply(1:k,function(i) Pn[i+1]-Pn[i])
77      probs[1] = probs[1] - 1/n
78      probs[k] = probs[k] + 1/n
79      return(sum(probs*dis))
80    } else {
```

```
81        dis    = sapply(1:k,function(i) 1/(qAug[i+1]-qAug[i]))
82        probs = sapply(1:k,function(i) Pn[i+1]-Pn[i])
83        probs[1] = probs[1] - 1/n
84        probs[k] = probs[k] + 1/n
85        return(sum(w*(2*probs-w)*dis))
86      }
87    }
88
89    # Histogram class constructor ------------------------------------------------
90
91    histogram = function(data,k,method="exact",seed=1882,lim=0.0001, ...){
92      data = sort(data)
93
94      if (method == "exact") {
95        obj          = dpHistogram(data,k,lim=lim,...)
96        obj$loglik = Pnm(obj,data)
97      } else if (method == "smoothed") {
98        n = length(data)
99        new_data = sort(rgcde(50*k*n,data))
100       obj = coordHistogram(new_data,k,lim=lim,...)
101       obj$method = "smoothed"
102       obj$loglik = Pnm(obj,new_data)
103       return(obj)
104     } else {
105       obj = coordHistogram(data,k,lim=lim,...)
106       obj$method = "coordinateSearch"
107       obj$loglik = Pnm(obj,data)
108     }
109
110     return(obj)
111
112   }
113
114   # Shortcuts ------------------------------------------------------------------
115
116   L2histogram = function(data,k,lim=0.0001,method="exact") {
117     histogram(data,k,method=method,lim=lim,type="L2",weights="L2")
118   }
119
120   KLhistogram = function(data,k,lim=0.0001,method="exact") {
121     histogram(data,k,method=method,lim=lim,type="KL",weights="KL")
122   }
123
124   # Bin selection procedures ---------------------------------------------------
125
126   CIC = function(data,k,b=.5,K=50,type="L2",weights="L2",lim=0.001){
127     n = length(data)
128     obj = histogram(data,k=k,type=type,weights=type,lim=lim,method="exact")
129     disc = obj$loglik
130     bias = biasSubs(data, b=b, k=k, K=K, type=type, weights=type, lim=lim)
131     n*disc - n*bias
132   }
133
134   CICselect = function(data,b=.5,K=50,type="L2",weights="L2",lim=0.0001,ks = seq
        (3,2*ceiling(nobs^(1/3)))) {
135     nobs = length(data)
136     CICs = sapply(ks,function(k) CIC(data,k,b,K,type,weights,lim=lim))
```

```
137   which.max(CICs) + 2
138 }
139
140 penB = function(data,k){
141   n = length(data)
142   obj = histogram(data,k=k,type=type,weights=type,lim=lim,method="exact")
143   lik = n*obj$loglik
144   pen = log(choose(n-1,k-1)) + (k-1) + (log(k))^(2.5)
145   lik - pen
146 }
147
148 penBselect = function(data,ks = seq(3,2*ceiling(nobs^(1/3)))){
149   nobs = length(data)
150   penBs = sapply(ks,function(k) penB(data,k))
151   which.max(penBs) + 2
152 }
153
154 penR = function(data,k,lim=.0001){
155   n = length(data)
156   obj = histogram(data,k=k,type=type,weights=type,lim=lim,method="exact")
157   lik = n*obj$loglik
158   pen = log(choose(n-1,k-1)) + 0.5*sum(obj$weights/diff(c(0,obj$splits,1))) + (
        log(k))^(2.5)
159   lik - pen
160 }
161
162 penRselect = function(data,ks = seq(3,2*ceiling(nobs^(1/3)))){
163   nobs = length(data)
164   penRs = sapply(ks,function(k) penR(data,k))
165   which.max(penRs) + 2
166 }
167
168 AICselect = function(data,ks = seq(3,2*ceiling(nobs^(1/3)))){
169   nobs = length(data)
170   AICs = sapply(ks,function(k) -AIC(histogram(data,k=k,type=type,weights=type,
        lim=lim,method="exact")))
171   which.max(AICs) + 2
172 }
173
174 # Generics ----------------------------------------------------------------
175
176 plot.hist = function(hist_obj,main=NULL,sub=NULL,xlab=NULL,ylab=NULL,rescale =
      1,grid=TRUE,...){
177   splits <- c(0,hist_obj$splits,1)
178   xlab = ifelse(is.null(xlab),"x",xlab)
179   ylab = ifelse(is.null(ylab),"Density",ylab)
180   ys <- c(0,hist_obj$weights*sapply(1:hist_obj$k,function(i) 1/(splits[i+1]-
        splits[i])))
181   if (is.null(main)) main = "Histogram"
182   if (is.null(sub)) sub = paste0("Irregular ",(hist_obj$specification)[1],"-
        histogram with ",(hist_obj$specification)[2]," weights. Method: ",hist_
        obj$method,"; k = ",hist_obj$k)
183   plot(rescale*splits,ys/rescale,type="S",bty="l",
184        xlab=xlab,ylab=ylab,main=main,sub=sub,...)
185   if(grid) {
186     grid()
187     lines(rescale*splits,ys/rescale,type="S",bty="l",
```

```
188             xlab=xlab,ylab=ylab,main=main,sub=sub,...)
189   }
190   lines(rescale*splits,ys/rescale,type="h",...)
191 }
192
193 lines.hist = function(hist_obj,...){
194   splits <- c(0,hist_obj$splits,1)
195   ys <- c(0,hist_obj$weights*sapply(1:hist_obj$k,function(i) 1/(splits[i+1]-
          splits[i])))
196   lines(splits,ys,type="S",...)
197   lines(splits,ys,type="h",...)
198 }
199
200 print.hist = function(hist_obj){
201   cat("***----------***\n")
202   cat("*** Irregular histogram object. ***\n")
203   cat("***----------***")
204   cat("\n Type:",(hist_obj$specification)[1])
205   cat("\n Weights:",(hist_obj$specification)[2])
206   cat("\n Method:",hist_obj$method)
207   cat("\n Splits:",hist_obj$splits)
208   cat("\n Weights:",hist_obj$weights)
209 }
210
211 logLik.hist = function(hist_obj) obj$loglik
212
213 AIC.hist = function(hist_obj) {
214   -2*n*hist_obj$loglik + 2*(hist_obj$k-1)
215 }
216
217 # Error measurements ----------
218
219 imse = function(obj,dist){
220   k = obj$k
221   q = obj$splits
222   w = obj$weights
223   qAug = c(0,q,1)
224   error = 0
225   for (i in 2:(k+1)){
226     error = error + integrate(function(x) (dist(x) - w[i-1]/(qAug[i]-qAug[i-1]))
          ^2,
227                               lower = qAug[i-1],upper=qAug[i])$value
228   }
229
230   error
231 }
232
233 hellinger = function(obj,dist){
234   k = obj$k
235   q = obj$splits
236   w = obj$weights
237   qAug = c(0,q,1)
238   error = 0
239   for (i in 2:(k+1)){
240     error = error + integrate(function(x) (sqrt(dist(x)) - sqrt(w[i-1]/(qAug[i
          ]-qAug[i-1])))^2,
241                               lower = qAug[i-1],upper=qAug[i])$value
```

```
242     }
243
244     sqrt(0.5*error)
245   }
```

Appendix B

Manski's estimator code

We provide C++code for the algorithm in one and two dimensions, together with an R wrapper.

B.1 One dimension

```
1  #include <cmath>
2  #include <vector>
3  #include <Rcpp.h>
4
5  class Point {
6  public:
7    /* These values should be there from the beginning, when the struct is
8    * instantiated. */
9    double value;
10   double weight;
11   bool isRed;
12
13   double redLeft;
14   double blueRight;
15
16   double posWeight;
17
18   Point(double _value, double _weight, bool _isRed);
19   Point();
20
21 };
22
23 typedef std::vector <Point> Points;
24
25 inline bool operator < (const Point pointOne, const Point pointTwo) {
26   return pointOne.value < pointTwo.value;
27 }
28
29 inline bool operator > (const Point pointOne, const Point pointTwo) {
30   return pointOne.value > pointTwo.value;
31 }
32
```

```
33 Point::Point(){
34 }
35
36 Point::Point(double _value, double _weight, bool _isRed) {
37   /* The basic values of the object is defined. */
38   value      = _value;
39   weight     = _weight;
40   isRed      = _isRed;
41   posWeight     = 0;
42
43   /* The quickredblue procedure will never update a point
44   * with its own colors. This is done here. */
45     if (isRed){
46       redLeft   = weight;
47       blueRight = 0;
48     } else {
49       redLeft   = 0;
50       blueRight = weight;
51     }
52 }
53
54 void updateColors(Points &_points){
55   long n = _points.size() - 1;
56
57   /* We iterate through all points. At each point, we update its redLeft
58   * etc values by adding what was known at the previous step of the iteration
59   */
60
61     for (long i = (n-1); i >= 0; i--){
62       _points[n].redLeft += _points[i].redLeft;
63     }
64
65   for (long i = (n-1); i >= 0; i--){
66     _points[i].blueRight += _points[i+1].blueRight;
67     if (_points[i+1].isRed){
68       _points[i].redLeft   = _points[i+1].redLeft - _points[i+1].weight;
69     } else {
70       _points[i].redLeft   = _points[i+1].redLeft;
71     }
72   }
73 }
74
75 void updateWeights(Points &_points){
76   long n = _points.size();
77   for (long i = 0; i < n; i++){
78     _points[i].posWeight = _points[i].blueRight + _points[i].redLeft;
79   }
80 }
81
82
83 // [[Rcpp::export]]
84 Rcpp::NumericMatrix redBlue(Rcpp::NumericVector values,Rcpp::NumericVector
       weights,
85                             Rcpp::IntegerVector isReds,bool isSorted){
86
87   long n = values.size();
88   Points points(n+2);
```

```
89    for (int i = 1; i < (n+1); i++){
90      points[i] = Point(values[i-1],weights[i-1],isReds[i-1]);
91    }
92
93    points[n+1]   = Point( 1.0/0.0, 0, false);
94    points[0]     = Point(-1.0/0.0, 0, true);
95
96    if (!isSorted) std::sort(points.begin(),points.end());
97
98    updateColors(points);
99    updateWeights(points);
100
101   Rcpp::NumericMatrix valuesAndWeights(3,n+2);
102   for (int i = 0; i < n+2; i++){
103     valuesAndWeights(0,i) = points[i].value;
104     valuesAndWeights(1,i) = points[i].isRed;
105     valuesAndWeights(2,i) = points[i].posWeight;
106   }
107
108   return (valuesAndWeights);
109 }
110 /*
111 Points updatePoints(Points &points,const Point newPoint){
112   long n = points.size();
113   int index;
114
115   if (newPoint.isRed){
116   index =
117   } else {
118
119   }
120 }
121 */
```

B.2 Two dimensions

```
1  #include <cmath>
2  #include <vector>
3  #include <algorithm>
4  #include <list>
5  #include <Rcpp.h>
6
7
8  class Point {
9  public:
10   double value;
11   long weight;
12   long posWeight;
13   double redLeft;
14   double blueRight;
15   bool isRed;
16   Point(double _value, long _weight, bool _isRed);
17   Point();
18 };
19
20 typedef std::vector <Point> Points;
```

```
21
22 inline bool operator < (const Point pointOne, const Point pointTwo) {
23   return pointOne.value < pointTwo.value;
24 }
25
26 inline bool operator > (const Point pointOne, const Point pointTwo) {
27   return pointOne.value > pointTwo.value;
28 }
29
30 Point::Point(){
31 }
32
33 Point::Point(double _value, long _weight, bool _isRed) {
34   /* The basic values of the object is defined. */
35   value      = _value;
36   weight     = _weight;
37   isRed      = _isRed;
38   posWeight     = 0;
39
40   /* The quickredblue procedure will never update a point
41   * with its own colors. This is done here. */
42     if (isRed){
43       redLeft   = weight;
44       blueRight = 0;
45     } else {
46       redLeft   = 0;
47       blueRight = weight;
48     }
49 }
50
51 void updateColors(Points &_points){
52   long n = _points.size() - 1;
53
54   /* We iterate through all points. At each point, we update its redLeft
55   * etc values by adding what was known at the previous step of the iteration
56   */
57
58     for (long i = (n-1); i >= 0; i--){
59       _points[n].redLeft += _points[i].redLeft;
60     }
61
62   for (long i = (n-1); i >= 0; i--){
63     _points[i].blueRight += _points[i+1].blueRight;
64     if (_points[i+1].isRed){
65       _points[i].redLeft   = _points[i+1].redLeft - _points[i+1].weight;
66     } else {
67       _points[i].redLeft   = _points[i+1].redLeft;
68     }
69   }
70 }
71
72 template <typename T> int signum(T val) {
73     return (T(0) < val) - (val < T(0));
74 }
75
76 void updateWeights(Points &_points){
77   long n = _points.size();
```

```
78    for (long i = 0; i < n; i++){
79      _points[i].posWeight = _points[i].blueRight + _points[i].redLeft;
80    }
81  }
82
83  long getWeightMax(Points points){
84      long tempweight = 0;
85      long n = points.size();
86      for(int i = 0; i < n; i++){
87          tempweight = std::max(tempweight,points[i].posWeight);
88      }
89      return(tempweight);
90  }
91
92  inline bool getColor(double const slopeCurrent,double const slope,
93                       bool const isRed){
94    bool whichColor;
95
96    if (slopeCurrent > slope) whichColor = isRed;
97    else whichColor = !isRed;
98
99    return (whichColor);
100 }
101
102
103
104 Points redBlue(std::vector < double > values,
105                std::vector < long > weights,
106                std::vector < bool   >   isReds){
107
108   long n = values.size();
109   Points points(n+2);
110
111   for (int i = 1; i < (n+1); i++){
112     points[i] = Point(values[i-1],weights[i-1],isReds[i-1]);
113   }
114
115   points[n+1]   = Point( 1.0/0.0, 0, false);
116   points[0]     = Point(-1.0/0.0, 0, true);
117
118   std::sort(points.begin(),points.end());
119   updateColors(points);
120   updateWeights(points);
121
122   return (points);
123 }
124
125
126 // [[Rcpp::export]]
127 Rcpp::List redBlue2(std::vector < double > xx1,
128                                std::vector < double > xx2,
129                                std::vector < bool > isReds,
130                                std::vector < long > weights) {
131
132   long nobs  = xx1.size();
133   long currentWeight;
134   long maxWeight = -10000;
```

```
135
136   std::vector < double > points(nobs-1);
137   std::vector < bool >   colors(nobs-1);
138   std::vector < Points > storage(nobs-1);
139   std::list < double > xCoord;
140   std::list < double > yCoord;
141   Points redBlues;
142
143   /* The main part of the algorithm consists of computing redBlue for each line
          in
144    * in the arrangement. In order to this, we will have to calculate all points
          of
145    * intersection and all the corresponding colors. The following loop will run
          through
146    * this algorithm for us. */
147
148   for (int lineCount = 0; lineCount < nobs; lineCount++){
149     double slopeCurrent = -xx2[lineCount]/xx1[lineCount];
150     double x11 = xx1[lineCount];
151     double x12 = xx2[lineCount];
152
153     for (int intersectionCount = 0; intersectionCount < lineCount;
          intersectionCount++){
154       bool isRed = isReds[intersectionCount];
155       double slope = -xx2[intersectionCount]/xx1[intersectionCount];
156       double x21 = xx1[intersectionCount];
157       double x22 = xx2[intersectionCount];
158
159       colors[intersectionCount] = getColor(slopeCurrent,slope,isRed);
160       points[intersectionCount] = (x21-x11)/(x22*x11-x21*x12);
161
162     }
163
164     for (int intersectionCount = lineCount +1; intersectionCount < nobs;
          intersectionCount++){
165       bool isRed = isReds[intersectionCount];
166       double slope = -xx2[intersectionCount]/xx1[intersectionCount];
167       double x21 = xx1[intersectionCount];
168       double x22 = xx2[intersectionCount];
169
170       colors[intersectionCount-1] = getColor(slopeCurrent,slope,isRed);
171       points[intersectionCount-1] = (x21-x11)/(x22*x11-x21*x12);
172     }
173
174     /* In order to call redBlue we will erase the values. When we're done it is
           safe
175      * to add them once again. */
176     currentWeight   = weights[lineCount];
177     weights.erase(weights.begin()+lineCount);
178     redBlues = redBlue(points,weights,colors);
179
180
181     /* We begin the updating operation. */
182     long tempWeight = maxWeight;
183     maxWeight = std::max(getWeightMax(redBlues),maxWeight);
184     if (tempWeight < maxWeight){
185
```

```
186         /* In this case, we will remove all entries from our
187          * xCoord and yCoord lists, and replace them with the new
188          * and superior entries from redBlues! */
189
190         xCoord.clear();
191         yCoord.clear();
192
193         double slope = -xx2[lineCount]/xx1[lineCount];
194         double intercept = -1/xx1[lineCount];
195
196         for (int i = 0; i < (nobs + 1); i++){
197             if (redBlues[i].posWeight >= maxWeight) {
198                 if(redBlues[i].value >= pow(10,10)) {
199                   xCoord.push_front(pow(10,10));
200                   yCoord.push_front(slope*pow(10,10)+intercept);
201                 } else if (redBlues[i].value <= - pow(10,10)) {
202                   xCoord.push_front(-pow(10,10));
203                   yCoord.push_front(slope*(-pow(10,10)+intercept));
204                 } else {
205                   xCoord.push_front(redBlues[i].value);
206                   yCoord.push_front(slope*redBlues[i].value+intercept);
207                 }
208             }
209         }
210     }
211
212     else if (tempWeight == maxWeight){
213
214         double slope = -xx2[lineCount]/xx1[lineCount];
215         double intercept = -1/xx1[lineCount];
216
217         for (int i = 0; i < (nobs + 1); i++){
218             if (redBlues[i].posWeight >= maxWeight) {
219                 if(redBlues[i].value >= pow(10,10)) {
220                   xCoord.push_front(pow(10,10));
221                   yCoord.push_front(slope*pow(10,10)+intercept);
222                 } else if (redBlues[i].value <= - pow(10,10)) {
223                   xCoord.push_front(-pow(10,10));
224                   yCoord.push_front(slope*(-pow(10,10)+intercept));
225                 } else {
226                   xCoord.push_front(redBlues[i].value);
227                   yCoord.push_front(slope*redBlues[i].value+intercept);
228                 }
229             }
230         }
231
232     }
233
234     weights.insert(weights.begin()+lineCount,currentWeight);
235   }
236
237
238  return Rcpp::List::create(Rcpp::Named("xCoord")      = xCoord,
239                            Rcpp::Named("yCoord")      = yCoord,
240                            Rcpp::Named("maxWeight") = maxWeight + 1);
241 }
```

B.3 R wrapper

Here we provide generics and wrappers for the C++-code. The mms-function can be used as the lm-function, by using formulas. An examples is mms(yy ∼ xx1 + xx2).

```
1  library("Rcpp")
2  sourceCpp("manskiAlgorithm[1d].cpp")
3  sourceCpp("manskiAlgorithm[2d].cpp")
4
5  #
6  # Begin the one dimensional algorithm ------------------------------------
7  #
8
9  # The objective function. ------------------------------------------------
10
11 manskiObjective = function(beta,yy,xx,weights=rep(1,length(yy)),positive = TRUE
       ){
12   if (positive) {
13     sum(weights*yy*(1+beta*xx>=0)) + sum(weights*(1-yy)*(1+beta*xx<0))
14   }
15   else {
16     sum(weights*yy*(-1+beta*xx>=0)) + sum(weights*(1-yy)*(-1+beta*xx<0))
17   }
18 }
19
20 # Wrapper. --------------------------------------------------------------
21
22 manski1d = function(yy, xx, xxConst = 1, weights=rep(1,length(yy)),
23                     isSorted = FALSE) {
24
25   # xxConst is used to modulate the values of the constant beta. By
26   # putting it unequal to 1, one can let y = x1 + bx2, for instance.
27   #
28   # isSorted = TRUE if the values yy,xx are already sorted.
29
30
31   # First we find the colors and transform the input.
32   # isRed = c((yy & xx >=0 ) | (!yy & xx < 0))
33
34   values = -xxConst/xx
35   isRed = c((yy & xx >=0 ) | (!yy & xx < 0))
36
37   redBlues = redBlue(values,weights,isRed,isSorted=isSorted)
38
39   maxWeight = max(redBlues[3,])
40   points    = matrix(redBlues[1,][redBlues[3,] == maxWeight],
41                      ncol=2,byrow=TRUE)
42   colnames(points) = c("beta1","beta2")
43   sol1d = list()
44   class(sol1d)     = c("manski1d","manski")
45   sol1d$maxWeight = maxWeight
46   sol1d$total     = sum(weights)
47   sol1d$points    = points
48   sol1d$yy        = yy
49   sol1d$xx        = xx
50   sol1d$weights   = weights
```

```
51
52    return(sol1d)
53  }
54
55  # Generics. ------------------------------------------------------------------
56
57  plot.manski1d = function(manski_obj,main=NULL,sub=NULL,lim=NULL,...){
58
59    redBlues = manski_obj$details
60    nobs = length(redBlues[1,]) - 2
61    maxWeight = manski_obj$maxWeight
62    total = manski_obj$total
63    weights = manski_obj$weights
64    positives = manski_obj$points
65    xx = manski_obj$xx
66    yy = manski_obj$yy
67
68    if(max(positives)==Inf) {
69      positives[positives==Inf] = max(positives[positives!=Inf])+4
70      upper = max(positives[positives!=Inf])
71    } else {
72      upper = max(positives,na.rm=TRUE) + abs(min(positives,na.rm=TRUE))
73    }
74
75    xs = seq(min(positives,na.rm=TRUE)-abs(min(positives,na.rm=TRUE)),
76             upper,by=0.001)
77
78    plot(xs,sapply(xs,function(beta) manskiObjective(beta,yy,xx,weights)/total),
         type="s",
79         col=adjustcolor("purple",alpha.f=0.6),xlab=expression(beta[1]),ylab="
             Percent hits",
80         main = main, bty="l",
81         xlim=c(min(positives)-0.01,max(positives)+0.01))
82    for (i in 1:dim(positives)[1]){
83      lines(positives[i,],c(maxWeight,maxWeight)/total)
84    }
85
86  }
87
88  breakdown = function(obj){
89    nobs = length(obj$yy)
90    isReds = obj$details[2,][-c(1,nobs+2)]
91    vals   = obj$details[1,][-c(1,nobs+2)]
92    left   = (vals <= obj$positives[1,1])
93    left   = sum(ifelse(isReds,1,-1)*left)
94    right  = (vals >= obj$positives[1,2])
95    right  = sum(ifelse(isReds,-1,1)*right)
96    ceiling(min(left,right))/2
97  }
98
99  coef.manski1d = function(obj,rand=FALSE){
100   xs = obj$positives
101   xsFiltered = xs[xs<=10^9 & xs >= -10^9]
102   if (rand) {
103     return (sample(xsFiltered,1))
104   } else {
105     xsAbs = abs(xsFiltered)
```

```
106       index = which.min(xsAbs)
107       return(xsFiltered[index])
108     }
109   }
110
111   #
112   # Begin the code for 2D algorithms ------------------------------------------
113   #
114
115   # Wrapper for the algorithm. ------------------------------------------------
116
117   manski2d = function(yy,xx1,xx2,weights=rep(1,length(yy))) {
118     isReds = (xx1 >= 0 & yy) | ( xx1 < 0 & !yy)
119     res = redBlue2(xx1,xx2,isReds,weights)
120     points = cbind(beta1 = res$yCoord,beta2 = res$xCoord)
121     sol2d = list()
122     class(sol2d)     = c("manski2d","manski")
123     sol2d$maxWeight = res$maxWeight
124     sol2d$total      = sum(weights)
125     sol2d$points     = points
126     sol2d$yy         = yy
127     sol2d$xx1        = xx1
128     sol2d$xx2        = xx2
129     sol2d$weights    = weights
130     return(sol2d)
131   }
132
133   # Generics. -----------------------------------------------------------------
134
135   plot.manski2d = function(manski_obj,main=NULL,sub=NULL,
136                            full=FALSE,lim=NULL,...){
137
138     len = length(manski_obj$points[,1])/2
139     xCoord = matrix(manski_obj$points[,1],len,2,byrow=TRUE)
140     yCoord = matrix(manski_obj$points[,2],len,2,byrow=TRUE)
141     xx1 = manski_obj$xx1
142     xx2 = manski_obj$xx2
143     yy  = manski_obj$yy
144     par(pty="m")
145
146     plot(NULL,xlim=c(min(xCoord),max(xCoord)),
147          col=adjustcolor("black",alpha.f=0.6),
148          ylim=c(min(yCoord),max(yCoord)),type="l",
149          xlab=expression(beta[2]),ylab=expression(beta[1]),
150          bty="l")
151     grid()
152
153     if (full){
154       nobs = length(xx1)
155       xs = seq(-1000,1000,by=2)
156       isReds = (xx1 >= 0 & yy) | ( xx1 < 0 & !yy)
157       colIndex = ifelse(isReds,2,4)
158       for (i in 1:nobs){
159         lines(xs,-1/xx2[i]-xs/xx2[i]*xx1[i],col=adjustcolor(colIndex[i],alpha.f
              =0.7),lty=3)
160       }
161
```

```
162     }
163
164     if (is.null(lim)){
165
166       for (i in 1:len) {
167         lines(xCoord[i,],yCoord[i,],lwd=1,col=adjustcolor("black",alpha.f=0.6))
168       }
169     } else {
170       plot(xCoord[1,],yCoord[1,],xlim=c(min(xCoord),lim),col=adjustcolor("black",
            alpha.f=0.6),
171            ylim=c(min(yCoord),lim),type="l",xlab=expression(beta[2]),ylab=
                 expression(beta[1]))
172
173       for (i in 2:len) {
174         lines(xCoord[i,],yCoord[i,],lwd=1,col=adjustcolor("black",alpha.f=0.6))
175       }
176     }
177
178   }
179
180   coef.manski2d = function(obj,type="min"){
181     # The point is to automatically select coefficients in a smart way.
182     # Most importantly, we _don't_ want beta1 or beta2 to be Inf (that
183     # is 10^10 in this case.)
184
185     if (type == "min") {
186       points = obj$points
187       index = which.min(apply(points,1,function(z) (z[1]^2+z[2]^2)))
188       return(c(points[index,1],points[index,2]))
189     } else {
190       xs = obj$points[,1]
191       ys = obj$points[,2]
192       for (i in 1:length(xs)){
193         if (xs[i] <= 10^9 & xs[i] >= -10^9 & ys[i] <= 10^9 & ys[i] >= -10^9){
194           return(c(beta1=xs[i],beta2=ys[i]))
195         }
196       }
197     }
198   }
199
200
201   #
202   # Define the mms function ------------------------------------------------
203   #
204
205   mms = function(formula,weights=NULL) {
206     vars = all.vars(formula)
207     resp = eval(parse(text=vars[1]))
208
209     if ( is.null(weights) ) weights = rep(1,length(resp))
210
211     if ( length(vars) == 2 ) {
212       xx = eval(parse(text=vars[2]))
213       return (manski1d(resp,xx,weights=weights))
214     } else {
215       xx1 = eval(parse(text=vars[2]))
216       xx2 = eval(parse(text=vars[3]))
```

```
217       return(manski2d(resp,xx1,xx2,weights=weights))
218     }
219   }
220
221   print.manski = function(obj) {
222     cat("***----------***␣\n")
223     cat("***␣Manski's␣maximum␣score␣␣␣␣␣***␣\n")
224     cat("***----------***")
225     if("manski1d" %in% class(obj)) {
226       cat("\n␣␣␣Covariates:␣1")
227     } else {
228       cat("\n␣␣␣Covariates:␣2")
229     }
230     cat("\n␣␣␣Objective:",obj$maxWeight)
231     cat("\n␣␣␣Sum␣of␣weights:",obj$total)
232     cat("\n␣␣␣Solution␣edges:␣\n")
233     print(as.data.frame(obj$points))
234   }
```