L_∞ Error and Bandwidth Selection for Kernel Density Estimates of Large Data

Yan Zheng
yanzheng@cs.utah.edu
University of Utah

Jeff M. Phillips
jeffp@cs.utah.edu
University of Utah

ABSTRACT

Kernel density estimates are a robust way to reconstruct a continuous distribution from a discrete point set. Typically their effectiveness is measured either in L_1 or L_2 error. In this paper we investigate the challenges in using L_∞ (or worst case) error, a stronger measure than L_1 or L_2. We present efficient solutions to two linked challenges: how to evaluate the L_∞ error between two kernel density estimates and how to choose the bandwidth parameter for a kernel density estimate built on a subsample of a large data set. [1]

1. INTRODUCTION

Kernel density estimates (KDEs) are essential tools [33, 31, 11, 12] for understanding a continuous distribution represented by a finite set of points. For instance, KDEs are used in data mining amid uncertainty to provide an effective intermediate representation, which captures information about the noise in the underlying data [2]. They are also used in classification problems by constructing the class of conditional probability density functions that are used in a Bayesian classifier [25]. They have many applications in other areas, such as network outlier detection [8], human motion tracking [6], financial data modeling [3] and geometric inference [27].

Given a point set $P \subset \mathbb{R}^d$ and a kernel $K_\sigma : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ with *bandwidth* parameter σ, for any point $x \in \mathbb{R}^d$, a kernel density estimate is defined as $\textsc{kde}_P(x) = \frac{1}{|P|}\sum_{p \in P} K_\sigma(p, x)$. We focus on *symmetric, shift-invariant* kernels which depend only on $z = \|p - x\|$ and σ, then a kernel can be written as function $K_\sigma(p, x) = k_\sigma(\|p - x\|) = k_\sigma(z)$. Intuitively, $\textsc{kde}_P(x)$ smoothes the effect of each $p \in P$ for the evaluation point x. For $d = 1$ this object can be used in place of an equi-width histogram; it removes the choice of how to shift the boundary of bins and thus KDEs are more robust. Moreover, they generalize naturally to higher dimensions.

The brute force solution of evaluating a kernel density estimate requires $O(|P|)$ time, and is thus untenable as a data structure for large data sets. And a lot research has gone towards speeding up these queries [7, 37, 9, 40]. One of the techniques [40] is to produce a *coreset* representation Q of the data which can be used as proxy for the true data P while guaranteeing approximation error. The size of Q depends only on the required error, not on any properties of P; these go beyond just randomly sampling Q from P. Written concretely, given P, and some error parameter $\varepsilon > 0$, the goal is to construct a point set Q to ensure

$$L_\infty(P, Q) = \mathsf{err}(P, Q) = \max_{x \in \mathbb{R}^d} |\textsc{kde}_P(x) - \textsc{kde}_Q(x)| \leq \varepsilon,$$

or written $\mathsf{err}(P, \sigma, Q, \omega)$ if the bandwidths σ and ω for $\textsc{kde}_{P,\sigma}$ and $\textsc{kde}_{Q,\omega}$ are under consideration. This line of work shows that an L_∞ error measure, compared with L_1 or L_2 error, is a more natural way to assess various properties about kernel density estimates. This work (like other work [7, 37, 9]) assumes σ is given, and then implicitly also assumes $\omega = \sigma$. In this paper, we will investigate choosing a bandwidth ω for $\textsc{kde}_Q$ under L_∞ error given P, σ, Q.

Thus, we empirically study two concrete problems:

1. Given two point sets $P, Q \subset \mathbb{R}^d$ and a kernel K, estimate $\mathsf{err}(P, Q)$.

2. Given two point sets $P, Q \subset \mathbb{R}^d$, a kernel K, and a bandwidth σ, estimate $\omega = \arg\min_\omega \mathsf{err}(P, \sigma, Q, \omega)$.

It should be apparent that the first problem is a key subproblem for the second, but it is also quite interesting in its own right. We will observe that L_∞ is a strictly stronger measure than L_1 or L_2, yet can still be assessed. To the best of our knowledge, we provide the first rigorous empirical study of how to measure this L_∞ error in practice in an efficient way, following theoretical investigations demonstrating it should be possible.

Bandwidth parameter is hugely important in the resulting KDE, and hence, there have been a plethora of proposed approaches [33, 31, 11, 12, 24, 34, 17, 28, 4, 32, 18, 29, 14, 30, 21, 36, 16, 23, 39, 10, 20, 19] to somehow automatically choose the "correct" value. These typically attempt to minimize the L_2 [33, 31] or L_1 error [11, 12] (or less commonly other error measures [24]) between $\textsc{kde}_P$ and some unknown distribution μ that it is assumed P is randomly drawn from. Perhaps unsurprisingly, for such an abstract problem different methods produce wildly different results. In practice, many practitioners choose a bandwidth value in a ad-hoc manner through visual inspection and domain knowledge.

[1] Thanks to NSF Awards 1350888, 1251019, and 1443046.

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.
© 2015 ACM. ISBN 978-1-4503-3664-2/15/08 ...$15.00.
DOI: http://dx.doi.org/10.1145/2783258.2783357.

In this paper we argue that the choice of bandwidth should not be completely uniquely selected. Rather this value provides a choice of scale at which the data is inspected, and for some data sets there can be more than one correct choice depending on the goal. We demonstrate this on real and synthetic data in 1 and 2 dimensions. As an intuitive 1-dimensional example, given temperature data collected from a weather station, there are very obvious modal trends at the scale of 1 day and at the scale of 1 year, and depending at which phenomenon one wishes to study, the bandwidth parameter should be chosen along the corresponding scale, so it is totally reasonable if we assume σ for $\textsc{kde}_P$ is given.

Via examinations of problem (2), we observe that in some cases (but not all), given P, Q, and σ, we can choose a new bandwidth ω (with $\omega > \sigma$) so that $\mathsf{err}(P, \sigma, Q, \omega)$ is significantly smaller than the default $\mathsf{err}(P, \sigma, Q, \sigma)$. This corresponds with fine-grained phenomenon disappearing with less data (as $|Q| < |P|$), and has been prognosticated by theory about L_2 [33] or L_1 [11] error where the optimal bandwidth for $\textsc{kde}_Q$ is a strictly shrinking function of $|Q|$. Yet, we urge more caution than this existing bandwidth theory indicates since we only observe this phenomenon in specific data sets with features present at different scales (like the daily/yearly temperature data example in Section 2.3).

Organization. Section 2 formalizes and further motivates the problem. Section 3 addresses problem (1), and Section 4 problem (2). Then Section 5 describes detailed experimental validations of our proposed approaches. Finally, Section 6 provides some concluding thoughts.

2. BACKGROUND AND MOTIVATION

In addition to the symmetric, shift-invariant properties of the kernels, it is convenient to enforce one of two other properties. A *normalized* kernel satisfies

$$\int_{x \in \mathbb{R}^d} K_\sigma(p, x) dx = 1,$$

so that the kernel and the kernel density estimate are probability distributions. A *unit* kernel satisfies

$$K_\sigma(x, x) = 1 \text{ so that } 0 \leq K_\sigma(x, p) \leq 1,$$

which ensures that $\textsc{kde}_P(x) \leq 1$. Unlike with the normalized kernel, the changing of bandwidth does not affect the coefficient of kernel function, so $K_\sigma(p, x) = k(\|p - x\|/\sigma)$.

Although this paper focuses on the Gaussian kernel $K_\sigma(p, x) = \frac{1}{\sigma^d (2\pi)^{d/2}} \exp(-\|p - x\|^2/2\sigma^2)$, probably the most commonly used kernel, there are many other symmetric, shift invariant kernels such as

- Laplace Kernel:
  $K_\sigma(p, x) = \frac{1}{\sigma^d c_d d!} \exp(-\|x - p\|/\sigma)$,
- Triangular Kernel:
  $K_\sigma(p, x) = \frac{d}{\sigma^d c_{d-1}} \max\{0, 1 - \|x - p\|/\sigma\}$,
- Epanechnikov Kernel:
  $K_\sigma(p, x) = \frac{d+2}{2\sigma^d c_d} \max\{0, 1 - \|x - p\|^2/\sigma^2\}$, or
- Ball Kernel:
  $K_\sigma(p, x) = \{\frac{1}{\sigma^d c_{d-1}} \text{ if } \|p - x\| \leq \sigma; \text{ o.w. } 0\}$,

where $c_d = \frac{r^d \pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}$ is the volume of the unit d-dimensional sphere. These are shown as normalized kernels, to make them unit kernels, the coefficient is simply set to 1.

2.1 Unit Kernels or Normalized Kernels?

Unit kernels are more natural to estimate the L_∞ errors of kernel density estimates [26, 38] since the range of values are in $[0, 1]$. For normalized kernels as σ varies, the only bound in the range is $[0, \infty)$.

Moreover, unit kernels, under a special case, correspond to the total variation distance of probability measures. In probability theory, the total variation distance for two probability measures P and Q on a sigma-algebra $\mathcal{F}$ of subsets of sample space Ω is defined as:

$$\delta(P, Q) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)|.$$

Terms $P(A)$, resp. $Q(A)$, refer to the probability restricted to subset A. If we use $\mathcal{F}$ as the set of all balls of radius σ, so A is one such ball, then $P(A)$ is the fraction of points of P falling in A. Hence $P(A)$ can be viewed as the $\textsc{kde}_{P,\sigma}(x)$ under the ball kernel, where x is the center of ball A. When Q is the coreset of P, then $Q(A)$ is the fraction of points of Q falling in A, so it can be viewed as the $\textsc{kde}_{Q,\sigma}(x)$ under the ball kernel. In this sense, the total variance distance is the L_∞ error, specifically $\mathsf{err}(P, Q)$ where K is the ball kernel. The total variation distance also maps to other unit kernels if $\mathcal{F}$ can admit weighted subsets, not just subsets.

However, normalized kernels are more useful in bandwidth selection. In this case, there is a finite value for $\sigma \in (0, \infty)$ which minimizes the L_1 or L_2 error between $\textsc{kde}_{P,\sigma}$ and $\textsc{kde}_{Q,\sigma}$, whereas for unit kernels this is minimized for $\sigma \to 0$.

But recall that unit and normalized kernels are only different in the scaling coefficient, so given one setting it is simple to convert to the other without changing the bandwidth. Hence we use *both* types of kernels in different scenarios: unit kernels for choosing the coresets, and normalized kernel for problem (1) and problem (2).

2.2 Why Coresets?

In the big data era, we are creating and accessing vastly more data than ever before. For example, mobile phones are consistently (implicitly) generating positional data along with various aspects of meta data including call duration and quality. To analyze or monitor the quality of signals or demand for this connections, we rarely need the entire data set, just an approximate version of it. A coreset can provide such a summary with accuracy guarantees, and by virtue of smaller size much more efficient and affordable access to it.

More formally, a *coreset* of a point set P is a subset Q such that (1) one can perform a family of queries on Q instead of P and the returned results are guaranteed to have bounded errors, and (2) the size of Q is much smaller than P, often independent of the size of P and only depends on the guaranteed error on the queries. For this paper, we consider coresets which preserve properties about the kernel density estimate, namely that for any query point x that $|\textsc{kde}_Q(x) - \textsc{kde}_P(x)| \leq \varepsilon$ for some error parameter $\varepsilon > 0$. The study of the worst case error was initiated by Phillips [26], and similar results under the L_2 error have been studied by Chen *et al.* [9] using an approach called *kernel herding*. Zheng *et.al.* [40] empirically improved these approaches to finding such a coreset in one and two dimensions, using methods based on random sampling, iteratively matching and halving of the data set, and Z-order curves. For instance, the experiments in [40] show that in two dimension, a coreset of 10,000 points can be constructed in less than 5 seconds from a 160 million record data set with approximation $\varepsilon = 0.01$.

2.3 Why σ is given?

Recall that problem (2) takes as given two point sets P and Q as well as a bandwidth σ associated with P, and then tries to find the best bandwidth ω for Q so that $\textsc{kde}_{P,\sigma}$ is close to $\textsc{kde}_{Q,\omega}$. This is different from how the "bandwidth selection problem" is typically posed [11, 33]: a single point set Q is given with no bandwidth, and it is assumed that Q is drawn randomly from an unknown distribution.

We break from this formulation for two reasons. First, we often consider the point set Q chosen as a coreset from P, and this may not be randomly from P, as more intricate techniques [40] can obtain the same error with much smaller size sets Q. These non-random samples break most modeling assumptions essential to the existing techniques.

Second, the choice of bandwidth may vary largely within the same data set, and these varied choices may each highlight a different aspect of the data. As an extended example consider temperature data (here we treat a reading of 50 degrees as 50 data points at that time) from a MesoWest weather station KSLC read every hour in all of 2012. This results in 8760 total readings, illustrated in Figure 1. For three bandwidth values of 3, 72, and 1440, $\textsc{kde}$s are shown to represent daily, weekly, and yearly trends. All are useful representations of the data; there is no "one right bandwidth." Section 5 shows a 2-dimensional example of population densities where similarly there are several distinct reasonable choices of bandwidths.

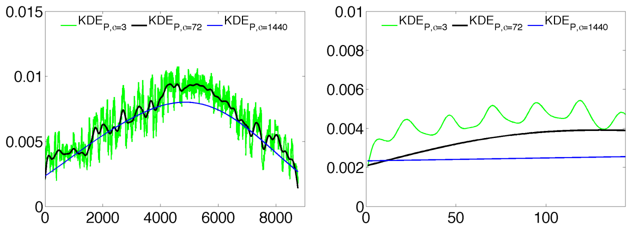


Figure 1: KDEs with different bandwidths showing daily, weekly and yearly temperature trends. Left shows the full year data, and right shows the one week data.

2.4 Why L_∞ Error?

As mentioned the most common measures for comparing $\textsc{kde}$s are the L_1 or L_2 error, defined for $p = \{1, 2\}$ as

$$L_p(P,Q) = \|\textsc{kde}_P - \textsc{kde}_Q\|_p$$
$$= \left(\int_{x\in\mathbb{R}^d} |\textsc{kde}_P(x) - \textsc{kde}_Q(x)|^p\right)^{1/p}.$$

Although this integral can be reasoned about, it is difficult to estimate precisely. Rather many techniques only evaluate at the points P and simply calculate

$$\left(\frac{1}{|P|}\sum_{q\in P} |\textsc{kde}_P(p) - \textsc{kde}_Q(p)|^p\right)^{1/p}.$$

These average over the domain or P; hence if $|\textsc{kde}_P(x) - \textsc{kde}_Q(x)| \leq \varepsilon$ *for all* x, then $L_p(P,Q)$ is also at most ε. That "for all" bound is precisely what is guaranteed by $L_\infty(P,Q)$, hence it is a stronger bound.

Another reason to study L_∞ error is that it preserves the worst case error. This is particularly important when $\textsc{kde}_P(x)$ values above a threshold trigger an alarm. For instance in tracking densities of tweets, too much activity in one location may indicate some event worth investigating. L_1 or L_2 error from a baseline may be small, but still have high error in one location either triggering a false alarm, or missing a real event.

2.5 Related Work on Bandwidth Selection

There is a vast literature on bandwidth selection under the L_1 [11, 12] or L_2 [33, 31] metric. In these settings Q is drawn, often at random, from an unknown continuous distribution μ (but μ can be evaluated at any single point x). Then the goal is to choose ω to minimize $\|\mu - \textsc{kde}_{Q,\omega}\|_{\{1,2\}}$. This can be conceptualized in two steps as $\|\mu - \textsc{kde}_{\mu,\omega}\|$ and $\|\textsc{kde}_{\mu,\omega} - \textsc{kde}_{Q,\omega}\|$. The first step is minimized as $\omega \to 0$ and the second step as $\omega \to \infty$. Together, there is a value $\omega_{\{1,2\}} \in (0, \infty)$ that minimizes the overall objective.

The most common error measure for ω under L_2 are Integrated Squared Errors(ISE) $ISE(\omega) = \int_{x\in\mathbb{R}^d}(\textsc{kde}_{Q,\omega} - \mu)^2 dx$ and its expected value, the Mean Integrated Squared Error (MISE) $MISE(\omega) = E_{Q\sim\mu}[\int_{x\in\mathbb{R}^d}(\textsc{kde}_{Q,\omega} - \mu)^2 dx]$. As MISE is not mathematically tractable, often approximations such as the Asymptotic Mean Integrated Squared Error (AMISE) or others [33, 34] are used. Cross-validation techniques [17, 28, 4, 32, 29, 14] are used to evaluate various parameters in these approximations. Alternatively, plug-in methods [30, 21, 36] recursively build approximations to μ using $\textsc{kde}_{Q,\omega_i}$, and then refine the estimate of ω_{i+1} using $\textsc{kde}_{Q,\omega_i}$. Bayesian approaches [5, 16, 23, 39, 10, 20] build on these models and select ω using MCMC approaches.

An alternative to these L_2 approaches is using an L_1 measure, like integrated absolute error (IAE) of $\textsc{kde}_{Q,\omega}$ is $IAE(\omega) = \int_{x\in\mathbb{R}^d} |\textsc{kde}_{Q,\omega} - \mu| dx$, which has simple interpretation of being the area between the two functions. Devroye and Györfi [11] describe several robustness advantages (better tail behavior, transformation invariance) to these approaches. Several of the approximation approaches from L_2 can be extended to L_1 [19].

Perplexingly, however, the bandwidths generated by these methods can vary quite drastically! In this paper, we assume that some bandwidth is given to indicate the intended scale, and then we choose a bandwidth for a sparser point set. Hence the methods surveyed above are not directly comparable to our proposed approaches. We include the experiment results from some of the above methods to show that different approaches give quite different "optimal" bandwidth, which in another way shows us there are more than one correct bandwidth for some data sets.

3. COMPUTING $\mathsf{err}(P,Q)$

The goal of this section is to calculate

$$\mathsf{err}(P,Q) = \max_{x\in\mathbb{R}^d} |\textsc{kde}_P(x) - \textsc{kde}_Q(x)|.$$

For notational convenience let $G(x) = |\textsc{kde}_P(x) - \textsc{kde}_Q(x)|$. We focus on the case where the kernel K is a unit Gaussian. Since even calculating $\max_{x\in\mathbb{R}^d} \textsc{kde}_P(x)$ (which is a special case of $\mathsf{err}(P,Q)$ where Q is empty) appears hard, and only constant factor approximations are known [1, 27], we will not calculate $\mathsf{err}(P,Q)$ exactly. Unfortunately these approximation techniques [1, 27] for $\max_{x\in\mathbb{R}^d} \textsc{kde}_P(x)$ do not easily extend to estimating $\mathsf{err}(P,Q)$. They can focus on dense areas of P, since the maximum must occur there, but in

$\mathsf{err}(P,Q)$, these dense areas may perfectly cancel out. These approaches are also much more involved than the strategies we will explore.

3.1 Approximation Strategy

Towards estimating $\mathsf{err}(P,Q)$, which is optimized over all of $\mathbb{R}^d$, our strategy is to generate a finite set $X \subset \mathbb{R}^d$, and then return $\mathsf{err}_X(P,Q) = \max_{x \in X} G(x)$. Our goal in the generation of X is so that in practice our returned estimate $\mathsf{err}_X(P,Q)$ is close to $\mathsf{err}(P,Q)$, but also so that under this process as $|X| \to \infty$ then formally $\mathsf{err}_X(P,Q) \to \mathsf{err}(P,Q)$. We say such a process *converges*.

We formalize this in two steps. First we show $G(x)$ is Lipschitz-continuous, hence a point $\hat{x} \in \mathbb{R}^d$ close to the point $x^* = \arg\max_{x \in \mathbb{R}^d} G(x)$ will also have error close to x^*. Then given this fact, we show that our strategy will, for any radius r, as $|X| \to \infty$ generate a point $\hat{x} \in X$ so that $\|x^* - \hat{x}\| \le r$. This will be aided by the following structural theorem on the location of x^*, with proofs in 1 and 2 dimensions deferred to Appendix A. (M is illustrated in Figure 2.)

THEOREM 1. *For K_σ a unit Gaussian kernel, and two point sets $P, Q \in \mathbb{R}^d$, then $x^* = \arg\max_{x \in \mathbb{R}^d} G(x)$ must be in M, the Minkowski sum of a ball of radius σ and the convex hull of $P \cup Q$.*

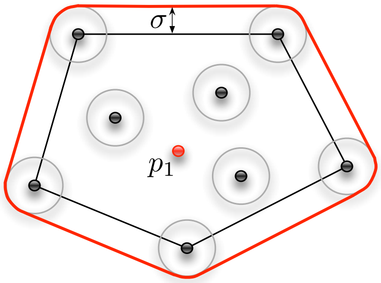


Figure 2: Illustration of the Minkowski sum of a ball of radius σ and convex hull of $P \cup Q$.

We will not focus on proving theoretical bounds on the rate of convergence of these processes since they are quite data dependent, but will thoroughly empirically explore this rate in Section 5. As $|X|$ grows, the max error value in X will consistently approach some error value (the same value for several provably converging approaches), and we can then have some confidence that as these processes plateau, they have successfully estimated $\mathsf{err}(P,Q)$. Our best process WCen6 converges quickly (e.g. $|X| = 100$); it is likely that the maximum error is approximately achieved in many locations.

Now as a basis for formalizing these results we first show $G(x)$ is Lipschitz continuous. Recall a function $f : \mathbb{R}^d \to \mathbb{R}$ is Lipschitz continuous if there exists some constant β such that for *any* two points $x, y \in \mathbb{R}^d$ that $|f(x) - f(y)|/\|x - y\| \le \beta$. This result follows from the Gaussian kernel (as well as all other kernels mentioned in Section 1 except the Ball kernel) also being Lipschitz continuous. Then since the function $f(x) = \text{KDE}_P(x) - \text{KDE}_Q(x)$ is a finite weighted sum of Gaussian kernels, each of which is Lipschitz continuous, so is $f(x)$. Since taking absolute value does not affect Lipschitz continuity, the claim holds.

3.2 Generation of Evaluation Points

We now consider strategies to generate a set of points X so that $\mathsf{err}_X(P,Q)$ is close to $\mathsf{err}(P,Q)$. Recall that M, the Minkowski sum of a ball of radius σ with the convex hull of $P \cup Q$ must contain the point x^* which results in $\mathsf{err}(P,Q)$. In practice, it is typically easier to use $\mathcal{B}$, the smallest axis-aligned bounding box that contains M. For discussion we assume $Q \subset P$ so $P = P \cup Q$.

Rand: *Choose each point uniformly at random from $\mathcal{B}$.*

Since $x^* \in M \subset \mathcal{B}$, eventually some point $x \in X$ will be close enough to x^*, and this process converges.

Orgp: *Choose points uniformly at random from P.*

This process does not converge since the maximum error point may not be in P. Yet Section 5 shows that this process converges to its limit very quickly. So many of the following proposed approaches will attempt to adapt this approach while still converging.

Orgp+N: *Choose points randomly from the original point set P then add Gaussian noise with bandwidth σ, where σ is the bandwidth of K.*

Since the Gaussian has infinite support, points in X can be anywhere in $\mathbb{R}^d$, and will eventually become close enough to x^*. So this process converges.

Grid: *Place a uniform grid on $\mathcal{B}$ (we assume each grid cell is a square) and choose one point in each grid.* For example in 2 dimension, if four evaluation points are needed, the grid would be 2×2 and if nine points are needed, it would be 3×3. So with this method, the number of evaluation points is a non-prime integer.

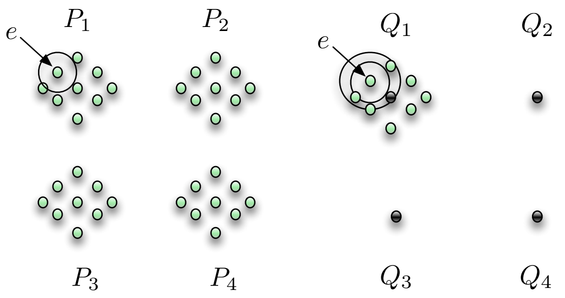
Since $x^* \in \mathcal{B}$, and eventually the grid cell radius is arbitrarily small, then some point $x \in X$ is close enough to x^*. Thus this process converges.

Cen{E[m]}: *Randomly select one point p_1 from the original point set P and randomly choose m neighbor points of p_1 within the distance of 3σ. m is chosen through a Exponential process with rate $1/E[m]$. Then we use the centroid of the selected neighbor points as the evaluation point.* This method is inspired by [15], which demonstrates interesting maximums of KDEs at the centroids of the data points.

Since P is fixed, the centroid of any combination of points in P is also finite, and the set of these centroids may not include x^*. So this process does not converge. We next modify it in a way so it does converge.

WCen{E[m]}: *Randomly select one point p_1 from the original point set P and select the neighbor point $p_n \in P$ as candidate neighbor proportional to $\exp(-\frac{||p_n - p_1||^2}{2\sigma^2})$, where σ is the bandwidth for K. The smaller the distance between p_n and p_1, the higher probability it will be the chosen. Repeat to choose m total points including p_1, where again m is from an Exponential process with rate $1/E[m]$. Now refine the m neighbor points so with probability 0.9, it remains the original point $p_n \in P$, with the remaining probability it is chosen randomly from a ball of radius σ centered at p_n. Next, we assign each point a random weight in $[0,1]$ so that all weights add to 1. Then finally the evaluation point is the weighted centroid of these points.*

This method retains much of the effectiveness of Cen, but does converge. Without the 0.1 probability rule of being in a ball of radius σ around each point, this method can generate any points within the convex hull of P. That 0.1 probability allows it to expand to M, the Minkowski sum of the convex hull of P with a ball of radius σ. Since $x^* \in M$, by Theorem 1, this process converges.



(a) Original dataset. (b) Coreset.

Figure 3: The example of necessary of larger bandwidth for coreset Q. The radius of the circle represents the chosen bandwidth.

Comb: Rand + Orgp: *The combination of method Rand and Orgp, of which* 20% *points generated from* $\mathcal{B}$ *and* 80% *points generated from original points.*

The 20% of points from Rand guarantees convergence, but retain most empirical properties of Orgp. This was used before with little discussion [40].

Section 5 describes extensive experiments on both synthetic and real data to evaluate these methods. The weighted centroid method WCen{E[m]} with large parameter (e.g. $E[m] = 6$) works very well for 1 and 2 dimensions, and also converges, so in general this technique is recommended. Although in some situations, it does not perform significantly better than other approaches like Rand+Orgp, which are simpler and also converge, so those may be a good option.

4. BANDWIDTH SELECTION

In this section we consider being given two point sets $P, Q \subset \mathbb{R}^2$, a kernel K, and a bandwidth σ associated with P. We consider K as a normalized Gaussian kernel, and where Q is a coreset of P. The goal is to find another bandwidth ω to associate with Q so that $\mathsf{err}(P, \sigma, Q, \omega)$ is small.

4.1 Refining the Bandwidth for Coresets

In [40], coresets are constructed assuming that KDE_Q uses the same bandwidth σ as KDE_P. Can we improve this relationship by using a different bandwidth ω for Q? The theory for L_1 or L_2 error (assuming Q is a random sample from P) dictates that as $|Q|$ decreases, the bandwidth ω should increase. This intuition holds under any error measure since with fewer data points, the KDE should have less resolution. It also matches the L_∞ theoretical error bounds described previously [26].

We first reinforce this with a simple 2-dimensional example. Consider point set $P = \cup\{P_1, P_2, P_3, P_4\}$ in Figure 3(a), the radius of the circle represents the bandwidth σ for P. Figure 3(b) gives the coreset Q of P: $Q = \cup\{Q_1, Q_2, Q_3, Q_4\}$, each Q_i contains only one black point. Now suppose our evaluation point is point e. If we use the original bandwidth σ, $\text{KDE}_{Q,\sigma}(e) = 0$ with ball kernel, but if we use ω, which is the radius of larger circle centered at e, then $\text{KDE}_{Q,\omega}(e) > 0$, so the error is decreased. But, we don't want ω too large, as it would reach the points in other Q_i, which is not the case for σ in P, so the error would be increased again. Thus there seems to be a good choice for $\omega > \sigma$.

But the situation of finding the ω_{opt} that minimizes $h(\omega) = \mathsf{err}(P, \sigma, Q, \omega)$ is more complicated. For each ω, $\mathsf{err}(P, \sigma, Q, \omega)$ is a maximization over $x \in \mathbb{R}^d$. There may in fact be more than one local minimum for ω in $h(\omega)$.

However, equipped with the WCen6 procedure to evaluate $\mathsf{err}(P, Q)$, we propose a relatively simple optimization algorithm. We can perform a golden section search over ω, using WCen6 to obtain a set X and evaluate $\mathsf{err}_X(P, \sigma, Q, \omega)$. Such a search procedure requires a convex function for any sort of guarantees, and this property may not hold. However, we show next that $h(\omega)$ has some restricted Lipschitz property, so that with random restarts it should be able to find reasonable local minimum. This is illustrated in Figure 4, where the curve that is Lipschitz either has a large, relatively convex region around the global minimum, or has shallow local minimums. The other curve without a Lipschitz property has a very small convex region around the global minimum, and any search procedure will have a hard time finding it.

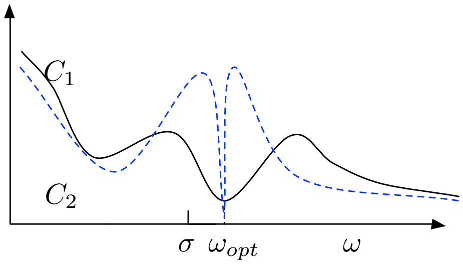


Figure 4: Two curves, dark one is Lipschitz, dashed curve is not.

4.2 Lipschitz Properties of h

In general, however, $h(\omega)$ is not Lipschitz in ω. But, we can show it is Lipschitz over a restricted domain, specifically when $\omega \geq \sigma \geq 1/A$ for some absolute constant A. Define $y(\omega, a) = \frac{1}{2\pi\omega^2}\exp(-a^2/(2\omega^2))$.

LEMMA 1. *For any $\omega \geq \sigma \geq 1/A$, $y(\omega, a)$ is β-Lipschitz with respect to ω, with $\beta = |a^2 - 1/\pi|A^3$.*

PROOF. By taking the first derivative of $y(\omega)$, we have

$$\frac{\mathrm{d}y(\omega, a)}{\mathrm{d}\omega} = (a^2 - \frac{1}{\pi})\omega^{-3}\exp(-a^2/(2\omega^2)).$$

And thus

$$\left|\frac{\mathrm{d}y(\omega, a)}{\mathrm{d}\omega}\right| = |a^2 - 1/\pi|\omega^{-3}\exp(-a^2/(2\omega^2))$$
$$\leq |a^2 - 1/\pi|\sigma^{-3} \leq |a^2 - 1/\pi|A^3.$$

So the absolute value of largest slope of function $y(\omega, a)$ is $\beta = |a^2 - 1/\pi|A^3$, thus $y(\omega, a)$ is β-Lipschitz continuous on ω. □

THEOREM 2. *For any $\omega \geq \sigma \geq 1/A$, $h(\omega)$ is β-Lipschitz with respect to ω, for $\beta = \frac{1}{|Q|}\sum_{q \in Q}|(x^* - q)^2 - 1/\pi|A^3$ where $x^* = \arg\max_{x \in \mathbb{R}^2}|\text{KDE}_{P,\sigma}(x) - \text{KDE}_{Q,\omega}(x)|$.*

PROOF. If $\text{KDE}_{P,\sigma}(x^*) \geq \text{KDE}_{Q,\omega}(x^*)$ then

$$h(\omega) = |\text{KDE}_{P,\sigma}(x^*) - \text{KDE}_{Q,\omega}(x^*)|$$
$$= \text{KDE}_{P,\sigma}(x^*) - \frac{1}{|Q|}\sum_{q \in Q}\frac{1}{2\pi\omega^2}\exp\left(\frac{-(x^* - q)^2}{2\omega^2}\right)$$
$$= \text{KDE}_{P,\sigma}(x^*) - \frac{1}{|Q|}\sum_{q \in Q} y(\omega, (x^* - q))).$$

Since $h(\omega)$ is linear combination of $|Q|$ functions of $y(\omega, a)$ plus a constant and $y(\omega, a)$ is Lipschitz continuous, based on the Lemma 1, $h(\omega)$ is Lipschitz continuous on ω. We can get the same result if $\text{KDE}_{P,\sigma}(x^*) \leq \text{KDE}_{Q,\omega}(x^*)$. In both directions, the first derivative of the function is bounded, so $h(\omega)$ is bounded. □

4.3 Random Golden Section Search

From the above properties, we design a search procedure that will be effective in finding the bandwidth ω minimizing $\mathsf{err}(P, \sigma, Q, \omega)$. The *random golden section search* is based on the golden section search [22], a technique to find extremum in a strictly unimodal function. To find a minimum, it successively narrows a range $[\ell, r]$ with known function values $h(\ell)$, $h(m_1)$, $h(m_2)$, and $h(r)$ with $\ell < m_1 < m_2 < r$ and with both $h(m_1), h(m_2)$ less than $h(\ell)$ and $h(r)$. If $h(m_1) < h(m_2)$ the new search range is $[\ell, m_2]$ and otherwise it is $[m_1, r]$. In either case a new fourth point is chosen according to the golden ratio in such a way that the interval shrinks by a constant factor on each step.

However, $h(\omega)$ in our case can be a multi-modal function, thus golden section search is not guaranteed to work. We apply random restarts as follows. Starting with a range $[\ell = \sigma, r = 10\sigma]$ we choose one middle point at $m = \lambda\sigma$ for $\lambda \sim \mathsf{Unif}(1, 10)$. If $h(m) > h(r)$ we increase r by a factor 10 until it is (e.g. $r = 100\sigma$). Then the second middle point is chosen using the golden ratio, and the search is run deterministically. We repeat with several random values λ.

5. EXPERIMENTS

Here we run an extensive set of experiments to validate our techniques. We compare KDE_P where P is in 1 and 2 dimensions with kernel density estimate under smaller coreset KDE_Q for both synthetic and real data. To show our methods work well in large data sets, we use the large synthetic data set(0.5 million) and real data set(1 million) in 2 dimension.

5.1 Data Sets

We consider data sets that have different features at various scales, so that as more data is present using a smaller bandwidth more fine-grain features are brought out, and a larger bandwidth only shows the coarse features. Our real data set in 1 dimension is the temperature data in Figure 1, with default $\sigma = 72$ (3 days). We use parameter $\varepsilon = 0.02$ to generate a coreset Q with the Sort-selection technique [40].

We can also simulate data with multi-scale features. On a domain $[0, 1]$ we generate P recursively, starting with $p_1 = 0$ and $p_2 = 1$. Next we consider the interval between $[p_1, p_2]$ and insert two points at $p_3 = 2/5$ and $p_4 = 3/5$. There are now 3 intervals $[p_1, p_3]$, $[p_3, p_4]$, and $[p_4, p_2]$. For each interval $[p_i, p_j]$ we recursively insert 2 new points at $p_i + (2/5) \cdot (p_j - p_i)$ and at $p_i + (3/5) \cdot (p_j - p_i)$, until $|P| = 19684$. The KDE of this data set with $\sigma = 0.01$ is shown in Figure 5(d), along with that of a coreset Q of size $|Q| = 100$.

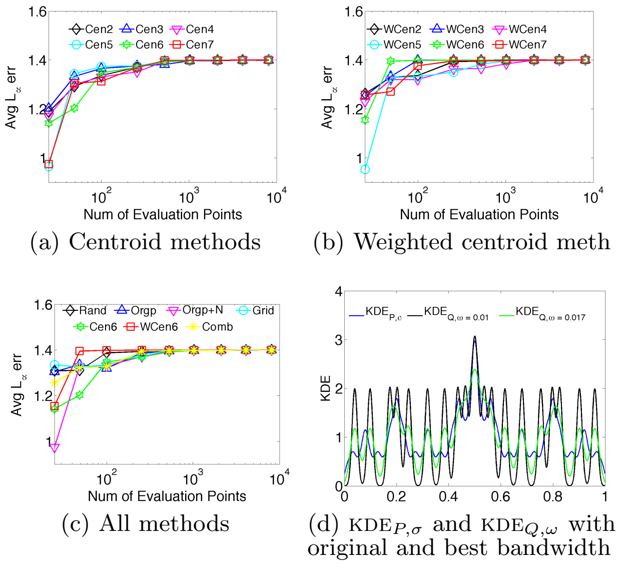
We construct the 2-dimensional synthetic data set in a similar way. The data is in $[0, 1]^2$ starting with four points $p_1 = (0, 0), p_2 = (0, 1), p_3 = (1, 0), p_4 = (1, 1)$. We recurse on this rectangle by adding 4 new point in the middle m: the x-coordinates are either at the 0.5-quantile or 0.8-quantile of the x-coordinates, and same for new y-coordinates. These 4 new points creates 9 smaller empty rectangles. We further recurse on each of these rectangles until $|P| = 532900$. The KDE_P with $\sigma = 0.01$ is shown in Figure 10(a). We use Grid matching [40] to generate a coreset Q with $\varepsilon = 0.1$ and size $|Q| = 1040$. Under the original bandwidth σ, the KDE_Q is shown in Figure 10(b); due to a small bandwidth this KDE has many more modes than the original, which motivates the larger bandwidth KDE shown in Figure 10(c).

For real data with multiple scales in 2 dimension we consider OpenStreetMap data from the state of Iowa. Specifically, we use the longitude and latitude of all highway data points, then rescale so it lies in $[0, 1] \times [0, 1]$. It was recognized in the early 1900s [35] that agricultural populations, such as Iowa, exhibited population densities at several scales. In experiment, we use the original data of size $|P| = 1155102$ with $\sigma = 0.01$, and Q as a smaller coreset with $\varepsilon = 0.1$ and $|Q| = 1128$. These are illustrated in Figure 11.

5.2 Evaluating Point Generation for $\mathsf{err}_X(P, Q)$

To find the best evaluation point generation techniques, we compare the various ways to generate a set X to evaluate $\mathsf{err}_X(P, Q)$. The larger numbers are better, so we want to find point sets X so that $\mathsf{err}_X(P, Q)$ is maximized with $|X|$ small. As most of our methods are random, five evaluation point sets are generated for each method and the average $\mathsf{err}_X(P, Q)$ is considered.

We start in 1 dimension, and investigate which parameter of the Cen and WCen methods work best. We will then compare the best in class against the remaining approaches. Recall the parameter $E[m]$ determines the expected number of points (under a Exponential process) chosen to take the centroid or weighted centroid of, respectively. We only show the test result with $E[m]$ from 2 to 7, since the results are similar when $E[m]$ is larger than 7, and the larger the parameter the slower (and less desirable) the process. The results are plotted in Figure 5 on the 1-dimensional synthetic data. Specifically, Figure 5(a) shows the Cen method and Figure 5(b) the WCen method. Both methods plateau, for some parameter setting, after around $|X| = 100$, with WCen more robust to parameter choice. In particular WCen converges



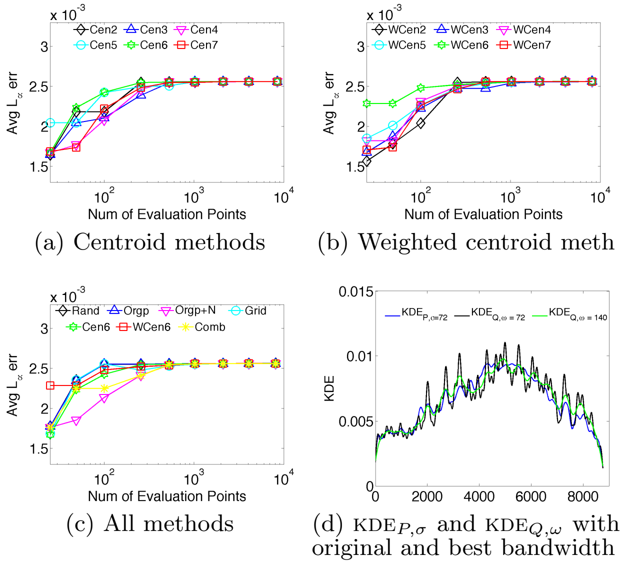
(a) Centroid methods

(b) Weighted centroid meth

(c) All methods

(d) $\text{KDE}_{P,\sigma}$ and $\text{KDE}_{Q,\omega}$ with original and best bandwidth

Figure 5: Choosing best evaluating point generation techniques for 1-dimensional synthetic data.



(a) Centroid methods (b) Weighted centroid meth

(c) All methods (d) $\text{KDE}_{P,\sigma}$ and $\text{KDE}_{Q,\omega}$ with original and best bandwidth

Figure 6: Choosing best evaluating point generation techniques for 1-dimensional real data.

slightly faster but with not much pattern across the choice of parameter. We use Cen6 and WCen6 as representatives. We next compare these approaches directly against each other as well as Rand, Orgp, Orgp+N, Grid, and Comb in Figure 5(c). WCen6 appears the best in this experiment, but it has been selected as best WCen technique from random trials. The Rand and Grid techniques which also converge perform well, and are simpler to implement.

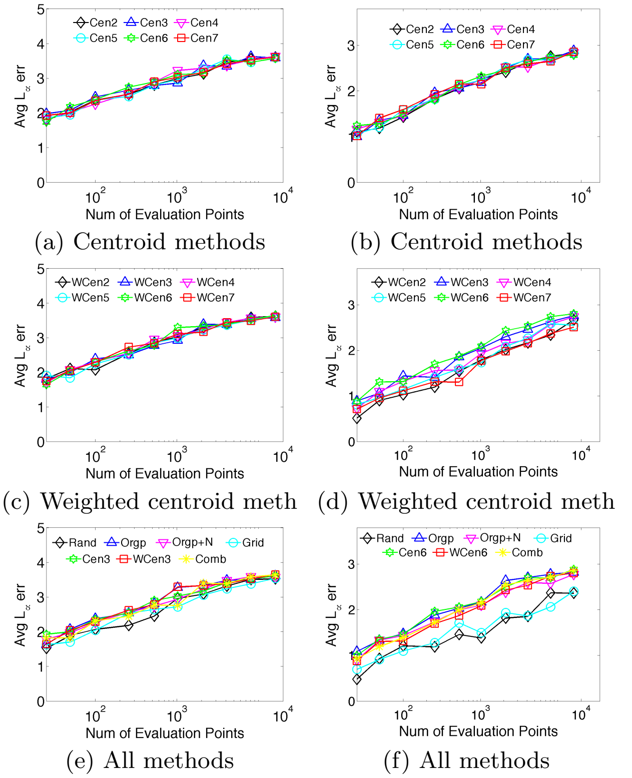
Similar results are seen on the real 1-dimensional data in Figure 6. We can take best in class from Cen and WCen parameter choices, shown as Cen6 and WCen6 in Figure 6(a) and Figure 6(b). These perform well and similar to the simpler Rand, Grid, and Orgp in Figure 6(c). Since Rand and Grid also converge, in 1 dimension we would recommend one of these simple methods.

For 2-dimensional data, the techniques perform a bit differently. We again start with Cen and WCen methods as shown in Figure 7 on real and synthetic data. The convergence results are not as good as in 1 dimension, as expected, and it takes roughly $|X| = 10000$ points to converge. All methods perform roughly the same for various parameter settings, so we use Cen6 and WCen6 as representatives. Comparing against all techniques in Figure 7(e), most techniques perform roughly the same relative to each other, and again WCen6 appears to be a good choice to use. The notable exceptions is that Grid and Rand perform worse in 2-d than in 1-d; likely indicating that the data dependent approaches are more important in this setting.

5.3 Choosing New Bandwidth Evaluation

We now apply a random golden section search to find new bandwidth values for coresets on 1-dimensional and 2-dimensional synthetic and real data. In all 10 random trials we always find the same local minimum, and report this value. We will see that a value $\omega > \sigma$ can often give better error results, both visually and empirically, by smoothing out the noise from the smaller coresets.

Figure 8 shows evaluation of $\text{err}_X(P, \sigma, Q, \omega)$ for various ω values chosen while running the random golden section search on synthetic and real 1-dimensional data. In both

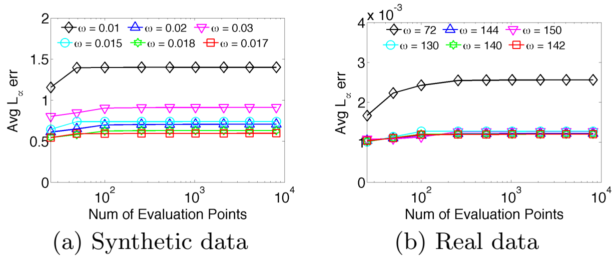


(a) Centroid methods (b) Centroid methods

(c) Weighted centroid meth (d) Weighted centroid meth

(e) All methods (f) All methods

Figure 7: Choosing the best evaluation set X for 2-dimensional synthetic (left) and real (right) data.



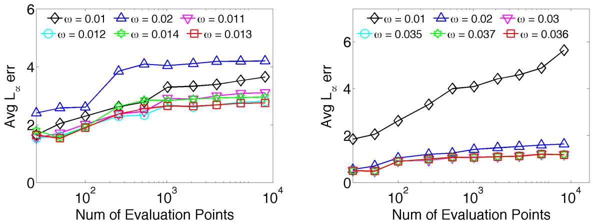
(a) Synthetic data (b) Real data

Figure 8: $\omega^* = \arg\min_\omega \text{err}_X(P, \sigma, Q, \omega)$ in $\mathbb{R}^1$.

cases, setting $\omega = \sigma$ (as $\omega = 0.01$ and $\omega = 72$, respectively) gives roughly twice as much error as using an omega roughly twice as large ($\omega = 0.017$ and $\omega = 142$, respectively).

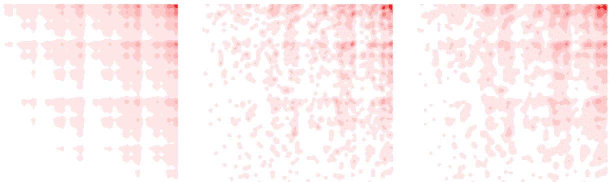
We can see the same results in 2-dimensional data sets in Figure 9. We observe in Figure 9(a) on synthetic data that with the original $\omega = \sigma = 0.01$ the error is roughly 3.6, but by choosing $\omega = 0.013$ that we can reduce the error to roughly 2.7. This is also shown visually in Figure 10, where a small coreset Q is chosen to do $\text{KDE}_{Q,\sigma}$ (Figure 10(b)) and the large-scale pattern in $\text{KDE}_{P,\sigma}$ is replaced by many isolated points; $\text{KDE}_{Q,\omega=0.013}$ (Figure 10(c)) increases the bandwidth and the desired visual pattern re-emerges. On real data, a similar pattern is seen in Figure 9(b). The original $\omega = \sigma = 0.01$ has error roughly 3.0, and an $\omega = 0.024$ (more than 2 times larger) gives error about 1.1. This extra smoothing is illustrated in Figure 11.

Thus we see that it is indeed useful to increase the bandwidth of kernel density estimates on a coreset, even though theoretical bounds already hold for using the same bandwidth. We show that doing so can decrease the error by



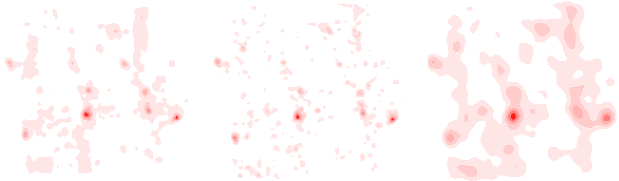
(a) Synthetic data (b) Real data

Figure 9: $\omega^* = \arg\min_\omega \exp_X(P, \sigma, Q, \omega)$ **in** $\mathbb{R}^2$**.**



(a) $\text{KDE}_{P,\sigma=0.01}$. (b) $\text{KDE}_{Q,\omega=0.01}$. (c) $\text{KDE}_{Q,\omega=0.013}$.

Figure 10: Visualization of KDE$_P$ and KDE$_Q$ for 2-dimensional synthetic data using different bandwidth.



(a) $\text{KDE}_{P,\sigma=0.01}$. (b) $\text{KDE}_{Q,\omega=0.01}$. (c) $\text{KDE}_{Q,\omega=0.024}$.

Figure 11: Visualization of KDE$_P$ and KDE$_Q$ for 2-dimensional real data using different bandwidth.

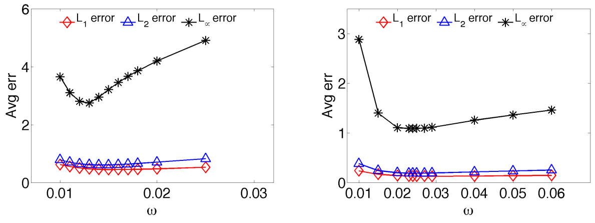
a factor of 2 or more. Since we consider $\omega = \sigma$, and only decrease the error in the process, we can claim the same theoretical bounds for the new ω value. It is an open question of whether one can prove tighter coreset bounds by adapting the bandwidth value.

5.4 New Bandwidth for L_1 and L_2 Error

The above bandwidth selection method can be extended to minimizing the L_1 and L_2 errors. Differing from L_∞ error, the L_1 and L_2 errors do not require finding a witness point of large error, but rather are the averaged over a region or, more commonly, the input points P. Figure 12 shows the L_1, L_2, and L_∞ errors for 2-dimensional synthetic and real data; other settings gave similar results. The results show that minimizing L_∞ does not give significantly worse errors than minimizing L_1 or L_2 in our setting. For example, in Figure 12(a), we see that the choice of $\omega = 0.013$ minimizes L_∞ errors, $\omega = 0.014$ gave a minimum L_2 error of 0.608 and $\omega = 0.016$ minimizes L_1 error of 0.450. Comparing instead to $\omega = 0.013$ which provided the minimum L_∞ error, then we get L_2 error of 0.618 and L_1 error of 0.476; both are within 1% of the minimum solutions.

5.5 Comparing Bandwidth Selection Methods

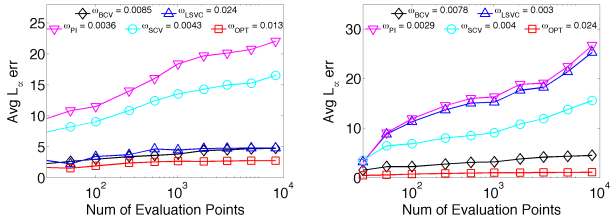
We compare against some traditional bandwidth selection methods for the 2-dimensional synthetic and real data using.



(a) Synthetic data (b) Real data

Figure 12: Relations of $L1$, $L2$ and L_∞ error and ω

We consider the following exemplars, among those surveyed in Section 2.5: biased cross-validation (BCV), least-squares cross-validation (LSCV), plug-in (PI), and smoothed cross-validation (SCV). We use the kernel smoothing R package (`ks`), which was originally introduced by Duong in 2007 [13] and improved in 2014. In the experiment, our data set is normalized and we assume data in each dimension is independent and share the same bandwidth; so we use the largest value from the main diagonal of bandwidth matrix computed from the R package. For the 2-dimensional synthetic data set, we use the same coreset with $|Q| = 1040$. The four exemplar methods, respectively, resulted in the following bandwidths $\omega_{BCV} = 0.0085$, $\omega_{LSCV} = 0.024$, $\omega_{PI} = 0.0036$, and $\omega_{SCV} = 0.0043$. For the 2-dimensional real data set, with the coreset $|Q| = 1128$, the bandwidth chosen by the four exemplar methods, respectively, are $\omega_{BCV} = 0.0078$, $\omega_{LSCV} = 0.0003$, $\omega_{PI} = 0.0029$, $\omega_{SCV} = 0.004$. The corresponding error trends compared to our method for these two data sets are shown in Figure 13, where ω_{OPT} denotes the optimal bandwidth from our method. Both of these figures show our method achieves the smallest error compared, and sometimes it is much (a factor of 20) smaller.



(a) Synthetic data (b) Real data

Figure 13: $\omega^* = \arg\min_\omega \exp_X(P, \sigma, Q, \omega)$ **in** $\mathbb{R}^2$**.**

6. CONCLUSION

This paper considers evaluating kernel density estimates under L_∞ error, and how to use these criteria to select the bandwidth of a coreset. The L_∞ error is stronger than the more traditional L_1 or L_2 error, it provides approximation guarantees for *all* points in the domain, and it aligns with recent theoretical results [26] of kernel range space. Thus it is worth rigorously investigating, and this paper presents the first such study.

We propose several methods to efficiently evaluate the L_∞ error between two kernel density estimates and provide a convergence guarantee. The method Grid works well, and is very simple to implement in $\mathbb{R}^1$. In $\mathbb{R}^2$, methods that adapt more to the data perform much better, and our technique

WCen is shown accurate and efficient on real and synthetic data. We then use these technique to select a new bandwidth value for coresets which can improve the error by a factor of 2 to 3. We demonstrate this both visually and empirically on real and synthetic large data sets.

7. REFERENCES

[1] P. K. Agarwal, S. Har-Peled, H. Kaplan, and M. Sharir. Union of random minkowski sums and network vulnerability analysis. In *SOCG*, 2013.

[2] C. C. Aggarwal. On density based transforms for uncertain data mining. In *ICDE*, 2007.

[3] T. Bouezmarni and J. V. Rombouts. Nonparametric density estimation for multivariate bounded data. *J. Statistical Planning and Inference*, 140:139–152, 2010.

[4] A. W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

[5] M. J. Brewer. A bayesian model for local smoothing in kernel density estimation. *Statistics and Computing*, 10(4):299–309, 2000.

[6] T. Brox, B. Rosenhahn, D. Cremers, and H.-P. Seidel. Nonparametric density estimation with adaptive, anisotropic kernels for human motion tracking. In *Human Motion–Understanding, Modeling, Capture and Animation*, pages 152–165. Springer, 2007.

[7] P. B. Callahan and S. R. Kosaraju. Algorithms for dynamic closest-pair and n-body potential fields. In *SODA*, 1995.

[8] Y. Cao, H. He, H. Man, and X. Shen. Integration of self-organizing map (SOM) and kernel density estimation (KDE) for network intrusion detection. In *SPIE Europe Security+ Defence*, 2009.

[9] Y. Chen, M. Welling, and A. Smola. Super-samples from kernel hearding. In *UAI*, 2010.

[10] M. S. de Lima and G. S. Atuncar. A bayesian method to estimate the optimal bandwidth for multivariate kernel estimator. *Journal of Nonparametric Statistics*, 23(1):137–148, 2011.

[11] L. Devroye and L. Györfi. *Nonparametric Density Estimation: The L_1 View.* Wiley, 1984.

[12] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation.* Springer-Verlag, 2001.

[13] T. Duong et al. ks: Kernel density estimation and kernel discriminant analysis for multivariate data in r. *Journal of Statistical Software*, 21(7):1–16, 2007.

[14] T. Duong and M. L. Hazelton. Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scandinavian J. of Stat.*, 32:485–506, 2005.

[15] H. Edelsbrunner, B. T. Fasy, and G. Rote. Add isotropic Gaussian kernels at own risk: More and more resiliant modes in higher dimensions. *SOCG*, 2012.

[16] A. Gangopadhyay and K. Cheung. Bayesian approach to choice of smoothing parameter in kernel density estimation. *J. of Nonparam. Stat.*, 14:655–664, 2002.

[17] J. Habbema, J. Hermans, and K. van den Broek. A stepwise discrimination analysis program using density estimation. *Proc. in Computational Statistics*, 1974.

[18] P. Hall, J. Marron, and B. U. Park. Smoothed cross-validation. *Prob. The. and Rel. Fields*, 92:1–20, 1992.

[19] P. Hall and M. P. Wand. Minimizing L_1 distance in nonparametric density estimation. *Journal of Multivariate Analysis*, 26(1):59–88, 1988.

[20] S. Hu, D. S. Poskitt, and X. Zhang. Bayesian adaptive bandwidth kernel density estimation of irregular multivariate distributions. *CS&DA*, 56:732–740, 2012.

[21] M. Jones and S. Sheather. Using non-stochastic terms to advantage in kernel-based estimation of integrated squared density derivatives. *Statistics & Probability Letters*, 11:511–514, 1991.

[22] J. Kiefer. Sequential minimax search for a maximum. *Proc. Am. Mathematical Society*, 4:502–506, 1953.

[23] K. Kulasekera and W. Padgett. Bayes bandwidth selection in kernel density estimation with censored data. *Nonparametric statistics*, 18(2):129–143, 2006.

[24] J. Marron and A. Tsybakov. Visual error criteria for qualitative smoothing. *Journal of the American Statistical Association*, 90(430):499–507, 1995.

[25] A. Pérez, P. Larrañaga, and I. Inza. Bayesian classifiers based on kernel density estimation: Flexible classifiers. *Int. J. Approximate Reasoning*, 50:341–362, 2009.

[26] J. M. Phillips. eps-samples for kernels. *SODA*, 2013.

[27] J. M. Phillips, B. Wang, and Y. Zheng. Geometric inference on kernel density estimates. In *SoCG*, 2015.

[28] M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandin. J. of Stat.*, 9:65–78, 1982.

[29] S. R. Sain, K. A. Baggerly, and D. W. Scott. Cross-validation of multivariate densities. *J. American Statistical Association*, 89:807–817, 1994.

[30] D. Scott, R. Tapia, and J. Thompson. Kernel density estimation revisited,. *Nonlinear Analysis, Theory, Methods and Appplication*, 1:339–372, 1977.

[31] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization.* Wiley, 1992.

[32] D. W. Scott and G. R. Terrell. Biased and unbiased cross-validation in density estimation. *J. ASA*, 82:1131–1146, 1987.

[33] B. W. Silverman. *Density Estimation for Statistics and Data Analysis.* Chapman & Hall/CRC, 1986.

[34] G. R. Terrell. Maximal smoothing principle in density estimation. *J. ASA*, 85:470–477, 1990.

[35] E. Ullman. A theory of location for cities. *American Journal of Sociology*, pages 853–864, 1941.

[36] M. Wand and M. Jones. Multivariate plug-in bandwidth selection. *J. Comp. Stat*, 9:97–116, 1994.

[37] C. Yang, R. Duraiswami, and L. S. Davis. Efficient kernel machines using the improved fast gauss transform. In *NIPS*, 2004.

[38] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast Gauss transform and efficient kernel density estimation. In *ICCV*, 2003.

[39] X. Zhang, M. L. King, and R. J. Hyndman. A bayesian approach to bandwidth selection for multivariate kernel density estimation. *CS&DA*, 50:3009–3031, 2006.

[40] Y. Zheng, J. Jestes, J. M. Phillips, and F. Li. Quality and efficiency in kernel density estimates for large data. In *SIGMOD*, 2013.

APPENDIX

A. STRUCTURAL PROOFS

To prove the weighted centroid method converges, we want to prove Theorem 1 in 1 and 2 dimension. For simplicity, we assume $Q \subset P$ so $P = P \cup Q$.

First we work on the weighted 1-dimensional data, and extend to 2 dimension using that the cross section of a 2-dimensional Gaussian is still a 1-dimensional Gaussian. We focus on when P and Q use the same bandwidth σ, and a unit kernel K_σ. We start to examine two points in 1 dimension, and without loss of generality, we assume $p_1 = d$ and $p_2 = -d$ for $d \geq 0$, and that the coreset of P is $Q = \{p_2\}$. We assign the weight for p_1 as w_1 and the weight for p_2 as w_2. Plug in P, Q and the weight for each point, $G(x) = |\text{KDE}_P(x) - \text{KDE}_Q(x)|$ is expanded as following:

$$G(x) = \left|\frac{1}{2}w_1 \exp\big(-\frac{(x-d)^2}{2\sigma^2}\big) - \frac{1}{2}w_2 \exp\big(-\frac{(x+d)^2}{2\sigma^2}\big)\right|.$$

We assume $w_1 \geq w_2$, the largest error point must be closer to p_1. So we only need to discuss when $x \geq 0$, then $\frac{1}{2}w_1 \exp\big(-\frac{(x-d)^2}{2\sigma^2}\big) \geq \frac{1}{2}w_2 \exp\big(-\frac{(x+d)^2}{2\sigma^2}\big)$, so

$$G(x) = \frac{1}{2}w_1 \exp\big(-\frac{(x-d)^2}{2\sigma^2}\big) - \frac{1}{2}w_2 \exp\big(-\frac{(x+d)^2}{2\sigma^2}\big).$$

LEMMA 2. *For K_σ a unit Gaussian kernel, $P = \{p_1, p_2\}$ and $Q = \{p_2\}$ where $p_1 = d$ and $p_2 = -d$, when $x \geq 0$, function $G(x)$ has only one local maximum, which is between d and $d + \sigma$ and $G(x)$ is decreasing when $x > d$.*

PROOF. By taking the derivative of $G(x)$, we can get

$$\begin{aligned}\frac{\mathrm{d}G(x)}{\mathrm{d}x} &= \frac{1}{2}w_2 \exp\big(-\frac{(x+d)^2}{2\sigma^2}\big)\frac{x+d}{\sigma^2} \\ &\quad - \frac{1}{2}w_1 \exp\big(-\frac{(x-d)^2}{2\sigma^2}\big)\frac{x-d}{\sigma^2}.\end{aligned}$$

When $0 \leq x < d$, both $\frac{1}{2}w_2 \exp\big(-\frac{(x+d)^2}{2\sigma^2}\big)\frac{x+d}{\sigma^2}$ and $-\frac{1}{2}w_1 \exp\big(-\frac{(x-d)^2}{2\sigma^2}\big)\frac{x-d}{\sigma^2} > 0$, thus $\frac{\mathrm{d}G(x)}{\mathrm{d}x} > 0$, so $G(x)$ is always increasing.

When $x = d$,

$$\frac{\mathrm{d}G(x)}{\mathrm{d}x} = \frac{1}{2}w_2 \exp\big(-\frac{2d^2}{\sigma^2}\big)\frac{2d}{\sigma^2} \geq 0.$$

To understand $x > d$ we examine the ratio function

$$r(x) = \frac{\frac{1}{2}w_2 \exp\big(-\frac{(x+d)^2}{2\sigma^2}\big)\frac{x+d}{\sigma^2}}{\frac{1}{2}w_1 \exp\big(-\frac{(x-d)^2}{2\sigma^2}\big)\frac{x-d}{\sigma^2}} = \frac{w_2}{w_1}\exp\big(-\frac{2xd}{\sigma^2}\big)\frac{x+d}{x-d}.$$

Since both $\exp\big(-\frac{2xd}{\sigma^2}\big)$ and $\frac{x+d}{x-d}$ are decreasing and positive, $r(x)$ and thus $\frac{\mathrm{d}G(x)}{\mathrm{d}x}$ is decreasing when $x > d$.

When $x = d + \sigma$, the ratio function is

$$r(d+\sigma) = \frac{w_2}{w_1}\exp(-\frac{2\sigma d + 2d^2}{\sigma^2})\frac{\sigma + 2d}{\sigma}.$$

We can view the above equation as a function of variable d.

$$r(d) = \frac{w_2}{w_1}\exp(-\frac{2\sigma d + 2d^2}{\sigma^2})\frac{\sigma + 2d}{\sigma},$$

and take the derivative of $r(d)$:

$$\frac{\mathrm{d}r(d)}{\mathrm{d}d} = -\frac{4d(d+\sigma)}{\sigma^3}\frac{w_2}{w_1}\exp(-\frac{2\sigma d + 2d^2}{\sigma^2}) \leq 0.$$

With $d \geq 0$ then $\frac{\mathrm{d}r(d)}{\mathrm{d}d} \leq 0$ and thus $r(d)$ is a decreasing function which attains maximum $\frac{w_2}{w_1} \leq 1$ when $d = 0$; thus $r(d) \leq 1$. So when $x = d + \sigma$, $\frac{\mathrm{d}G(x)}{\mathrm{d}x} \leq 0$. With the above fact that $\frac{\mathrm{d}G(x)}{\mathrm{d}x} \geq 0$ when $x = d$ and $\frac{\mathrm{d}G(x)}{\mathrm{d}x}$ is decreasing when $x > d$, there is only one point between d and $d + \sigma$ making $\frac{\mathrm{d}G(x)}{\mathrm{d}x} = 0$. Since when $0 \leq x < d$, $\frac{\mathrm{d}G(x)}{\mathrm{d}x} > 0$. There is only one maximum point of $G(x)$ between d and $d + \sigma$ when $x \geq 0$. □

From Lemma 2, we show that the evaluation point having largest error is between d and $d + \sigma$. Due to the symmetry of p_1 and p_2, when $w_1 \leq w_2$,$G(x)$ gets its largest error between $-d$ and $-d - \sigma$.

With the results on both sides, we now show the maximum value point of $G(x)$ can't be outside σ distance of $Conv(P)$.

Now we discuss the case for n points in 1 dimension.

LEMMA 3. *For K_σ a unit Gaussian kernel, P has n points and $|Q| = |P|/2$, $\arg\max_{x \in \mathbb{R}^1} G(x)$ for 1-dimensional data is within σ distance of $Conv(P)$.*

PROOF. Suppose $n = 2k$, $P = \{p_1, p_2, p_3, p_4, ..., p_{2k-1}, p_{2k}\}$, choose any k points in Q. Then pair any point in Q with any point in P not in Q, so each point in P is in exactly one pair. For simplicity we set $Q = \{p_1, p_3, ..., p_{2k-1}\}$ and the pairs are $\{p_1, p_2\}, \{p_3, p_4\}, ..., \{p_{2k-1}, p_{2k}\}$.

Suppose $e_1 = \arg\max_{x \in \mathbb{R}^1} G(x)$ is not within σ distance of $Conv(P)$ and p_1 is the point closest to e_1. Based on Lemma 2, for P has only two points, function $G(x)$ is decreasing as a point outside σ moves away from p_1. So if we choose another point e_2 infinitesimally closer to p_1, and we set $P_1 = \{p_1, p_2\}$, $Q_1 = \{p_1\}$, $G_{P_1,Q_1}(e_2)$ has larger value than $G_{P_1,Q_1}(e_1)$. Since p_1 is the closest point in P, for any other set $P_2 = \{p_3, p_4\}$, $Q_2 = \{p_3\}$, e_2 is closer to P_2 than e_1 is to P_2, hence $G_{P_2,Q_2}(e_2)$ is also larger than $G_{P_2,Q_2}(e_1)$. The same result holds for all pairs $\{p_{2i-1}, p_{2i}\}$, where i is from 1 to k. So $G(e_2) > G(e_1)$, which contradicts the assumption that $e_1 = \arg\max_{x \in \mathbb{R}^1} G(x)$. So the largest error evaluation point should be within σ distance of $Conv(P)$. □

In 2 dimensions we show a similar result. We illustrate the Minkowski sum M of a set of points P with a ball of radius σ in Figure 2.

THEOREM 3. *For K_σ a unit Gaussian kernel, and two point sets $P, Q \in \mathbb{R}^2$, $|Q| = |P|/2$, $\arg\max_{x \in \mathbb{R}^2} G(x)$ should be within the Minkowski sum M of a ball of radius σ and $Conv(P)$.*

PROOF. Now we have n points in $P \in \mathbb{R}^2$. Suppose the largest error position $e_1 = \arg\max_{x \in \mathbb{R}^2} G(x) \notin M$, then for some direction v no point in the convex hull of P is closer than σ to e_1 after both are projected onto v. Then since any cross section of a Gaussian is a 1-dimensional Gaussian (with reduced weight), we can now invoke the 1-dimensional result in Lemma 3 to show that e_1 is not the largest error position along the direction v, thus $e_1 \neq \arg\max_{x \in \mathbb{R}^2} G(x)$. So $\arg\max_{x \in \mathbb{R}^2} G(x)$ should be within the Minkowski sum M of a ball of radius σ and $Conv(P)$. □