Dual Tree Kernel Conditional Density Estimation

NP Slagle and Alexander Gray

February 2012

1 Introduction

1.1 Estimation of Densities

Density estimation, an approach to prediction, is ubiquitous across most scientific and engineering disciplines. Conditional density estimation, in which we estimate $P(Y_1, ..., Y_{D_Y}|X_1, ..., X_{D_X})$ given a random vector $(Y_1, ...Y_{D_Y}, X_1, ..., X_{D_X})$, can capture salient relationships between features not obvious when estimating marginal distributions. Illustrated in figure 1, the marginal distribution of $\mathbf{Y}$ and distributions of $Y_1, ..., Y_{D_Y}|X_1, ..., X_{D_X} = x_1, ..., x_{D_X}$ vary significantly depending on $\mathbf{x}$. Many of the conditionals exhibit bimodality or unimodality, demonstrating that the marginal and any single conditional distribution of $\mathbf{Y}$ often fails to capture the underlying structure. Conditional density estimation when $D_Y = 1$ is also superior to linear regression, since therein we estimate the quantity $E(Y_1|X_1, ..., X_{D_X})$ rather than the entire conditional distribution. Budavari 2009 demonstrates the effectiveness in applying conditional density estimation to red shift in astronomy. Song, Gretton, and Guestrin 20XX demonstrate density estimation effectiveness in graphical models.

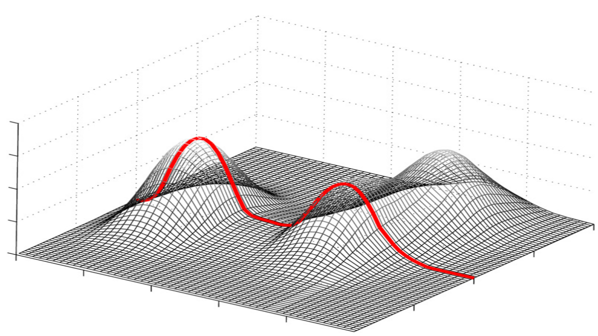

Figure 1: Bivariate distribution

1.2 Kernel Density Estimation

Nonparametric kernel density estimation, introduced in Rosenblatt 1956, assumes only smoothness of the underlying distribution of the data. Given data points $\{\mathbf{X}_i\}_{i=1}^N \subset \mathbb{R}^d$ and a kernel function $K : \mathbb{R}/to/\mathbb{R}^+$, we define the kernel density estimate as the interpolation

$$\hat{f}_h(\mathbf{x}) = \sum_{i=1}^{N} \frac{1}{h^d N} K\left(\frac{\|\mathbf{x} - \mathbf{X}_i\|}{h}\right) \tag{1}$$

Discussed in Silverman 1986, the bandwidth h is critical to the convergence properties of 1, whereas the choice of the kernel K, usually a radial unimodal function integrating to one over $\mathbb{R}^d$, is less crucial[1]. Given a data set $\{\mathbf{X}_i, \mathbf{Y}_i\}_{i=1}^N$ where $\mathbf{X}_i \in \mathbb{R}^{D_X}$ and $\mathbf{Y}_i \in \mathbb{R}^{D_Y}$ for all i, a generalization of the Nadaraya-Watson form (see Gooijer and Zerom 2003) of kernel conditional density estimation is

$$\hat{f}_{a,b,c}(\mathbf{y}|\mathbf{x}) = \frac{\sum_{i=1}^{N} \frac{1}{a^{D_Y} b^{D_X} N} K\left(\frac{\|\mathbf{y}-\mathbf{Y}_i\|}{a}\right) K\left(\frac{\|\mathbf{x}-\mathbf{X}_i\|}{b}\right)}{\sum_{i=1}^{N} \frac{1}{c^{D_X} N} K\left(\frac{\|\mathbf{x}-\mathbf{X}_i\|}{c}\right)} \tag{2}$$

The literature typically abbreviates

$$K_h(t) = \frac{1}{h^d} K(t/h) \tag{3}$$

We express the full form above to indicate clearly the scaling factors, as $\mathbf{X}_i$ and $\mathbf{Y}_i$ are each vectors. Henceforth, we'll adopt the abbreviation.

For $D_Y = D_X = 1$, Chen, Linton, and Robinson 2001 summarizes choices for a, b, and c appearing earlier in the literature; herein we assume $b = c$, a simplification resulting in, as discussed in Chen, Linton, and Robinson 2001 for $D_Y = D_X = 1$,

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \hat{f}_{a,b,b}(\mathbf{s}|\mathbf{x}) ds_1 \ldots ds_{D_Y} = 1 \tag{4}$$

Thus, equation 2 with $b = c$ satisfies unit mass over $\mathbb{R}^{D_Y}$.

Many of the aforementioned references with respect to density estimation effectiveness affirm kernel density estimation.

[1] Epanechnikov 1969 offers a proof of the asymptotic minimum variance of the finite-extent Epanechnikov kernel.

1.3 Bandwidth Selection

Conventional bandwidth selection[2] approaches for KCDE include maximization of the log-likelihood function, and likelihood cross validation (LCV), minimization of the least-squares cross validation estimate, or least-squares cross validation (LSCV). Bandwidth selection using LCV, formulated as

$$LCV(a,b) = argmax_{a,b} \sum_{i=1}^{N} log\hat{f}_{a,b,b}(\mathbf{y}_i|\mathbf{x}_i) \quad (5)$$

typically suffers high sensitivity to outliers (Silverman 1986.) The LSCV, discussed in Hansen 2004, attempts to minimize the integrated square error (ISE)

$$ISE(a,b) = \int \ldots \int (f(\mathbf{y}|\mathbf{x}) - \hat{f}_{a,b,b}(\mathbf{y}|\mathbf{x}))^2 f(\mathbf{x}) d\mathbf{y} d\mathbf{x} \quad (6)$$

with a score approximation of

$$LSCV(a,b) = \frac{1}{N} \sum_{i=1}^{N} H_i - 2I_i \quad (7)$$

where

$$H_i = \frac{\sum_{j\neq i} \sum_{k\neq i} K_b(\|\mathbf{x}_i - \mathbf{x}_j\|) K_b(\|\mathbf{x}_i - \mathbf{x}_k\|) J_{j,k}}{\sum_{j\neq i} \sum_{k\neq i} K_b(\|\mathbf{x}_i - \mathbf{x}_j\|) K_b(\|\mathbf{x}_i - \mathbf{x}_k\|)} \quad (8)$$

$$J_{j,k} = \int \ldots \int K_a(\|\mathbf{y} - \mathbf{y}_j\|) K_a(\|\mathbf{y} - \mathbf{y}_k\|) d\mathbf{y} \quad (9)$$

and

$$I_i = \frac{\sum_{j\neq i} K_b(\|\mathbf{x}_i - \mathbf{x}_j\|) K_a(\|\mathbf{y}_i - \mathbf{y}_j\|)}{\sum_{j\neq i} K_b(\|\mathbf{x}_i - \mathbf{x}_j\|)} \quad (10)$$

As discussed in Hansen 2004, if K is the Gaussian kernel, then 9 reduces to

$$J_{j,k} = K_{a\sqrt{2}}(\|\mathbf{y}_j - \mathbf{y}_k\|) \quad (11)$$

Since computing the LSCV requires $O(N^3)$ time, we calculate bandwidths for larger N assuming that there exist c_a, p_a, c_b, p_b such that $a^* = c_a N^{p_a}$ and $b^* = c_b N^{p_b}$. Hansen 2004 and Silverman 1986 offer precedents in the literature, and our empirical studies indicate that log-linear regression offers a reasonable approximation.

[2]Chen, Linton, and Robinson 2001 offer a survey of bandwidth selection choices for various cases of parameters a, b, and c where $D_X = D_Y = 1$.

For finite extent kernels (such as spherical and Epanechnikov), the LSCV can be problematic, as the denominator and numerator of some terms in the summation can be zero. To mitigate this, we observe that given a weighted sum over nonzero weights $S(\{\alpha_i, \omega_i(t)\}_{i=1}^n) = \frac{\sum_{i=1}^n \alpha_i \omega_i(t)}{\sum_{i=1}^n \omega_i(t)}$ such that $\lim_{t\to 0} \omega_i(t) = 0$ for all i and such that for all $i,j \in \{1,...,n\}$, $\lim_{t\to 0} \frac{\omega_i(t)}{\omega_j(t)} = \infty$, 1, or 0, then given $\{\omega_{i_1}(t), ...\omega_{i_k}(t)\}$ that produce ∞ limits in numerators the most frequently, we have $\lim_{t\to 0} S(\{\alpha_i, \omega_i(t)\}_{i=1}^n) = \frac{1}{k}\sum_{j=1}^k \alpha_{i_j}$. Less formally, the sum approaches the arithmetic mean of the components whose weights approach zero the most slowly. Since our kernel functions over various point pairs serve as weight functions akin to the ω functions, we can apply this technique when calculating LSCV over the finite-extent kernels. Also, we can apply this approach to infinite-extent kernels since they exhibit numerical imprecision on smaller bandwidths.

1.4 Tree Methods for Kernel Density Estimation

Gray 2000 introduces an efficient algorithm for kernel density estimation that organizes both the reference and query sets into space-partitioning trees (ball trees or *kd*-trees) such that coordinates over each node are maximally tight. Figure 1.4 shows leaf nodes of a *kd* tree applied to a bivariate distribution. Gray's dual tree algorithm recurses on the query and reference trees to approximate upper and lower bounds on each p_q, the mass of query point $x_q \in Q$, pruning subtrees with a rule guaranteeing that the relative error between the bounds is less than or equal to a user-specified ϵ. For each $x_q \in Q$, the estimate of p_q is $\hat{p}_q = \frac{\hat{p}_q^{max} + \hat{p}_q^{min}}{2}$ with $\left|\frac{\hat{p}_q - p_q}{p_q}\right| \leq \epsilon$. The algorithm appears in figure 3.

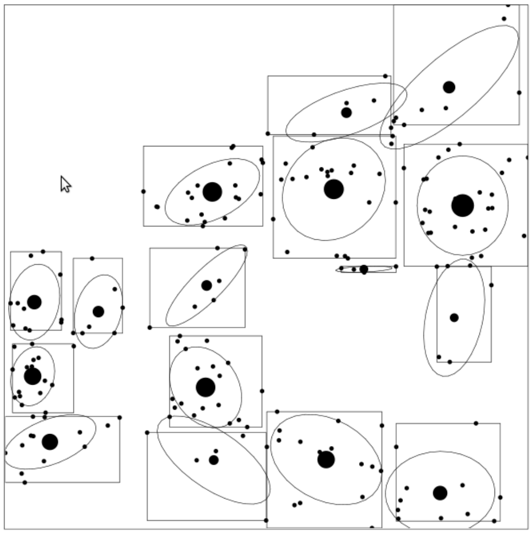

Figure 2: Bivariate distribution partition: The ellipses represent covariance matrices; the large dots indicate centroid locations and cluster masses.

DualTree(Q, T)
$dl = N_T K_h(\delta_{QT}^{max})$
$du = N_T(K_h(\delta_{QT}^{min}) - 1)$
if $K_h(\delta_{QT}^{min}) - K_h(\delta_{QT}^{max}) \leq \frac{2\epsilon}{N\hat{p}_Q^{min}}$ **then**

  for all $x_q \in Q$ **do**
    $\hat{p}_q^{max} += dl$, $\hat{p}_q^{min} += du$
  end for
  $\hat{p}_Q^{min} += dl$, $\hat{p}_Q^{max} += du$
  return
else
  if leaf(Q) and leaf(T) **then**
    DualTreeBase(Q,T)
    return
  else
    DualTree(Q.left,closer-of(Q.left,{T.left,T.right}))
    DualTree(Q.left,farther-of(Q.left,{T.left,T.right}))
    DualTree(Q.right,closer-of(Q.right,{T.left,T.right}))
    DualTree(Q.right,farther-of(Q.right,{T.left,T.right}))
  end if
end if

DualTreeBase(Q, T)
for all $x_q \in Q$ **do**
  for all $x_t \in T$ **do**
    $\hat{p}_q^{min} += K_h(\|x_q - x_t\|)$, $\hat{p}_q^{max} += K_h(\|x_q - x_t\|)$
  end for
  $\hat{p}_q^{max} -= N_T$
end for
$\hat{p}_Q^{min} = \min_{q \in Q} \hat{p}_q^{min}$, $\hat{p}_Q^{max} = \max_{q \in Q} \hat{p}_q^{max}$

Figure 3: Gray and Moore's dual tree algorithm

1.5 Related Work

Early efforts to reduce time complexity in kernel density estimation include Scott 1992 and Fan 1994, applying univariate methods to the multivariate case. Gray and Moore 2000, stated earlier, introduces efficient tree methods applicable to various machine learning techniques, including kernel density estimation. Gray 2003 and Gray and Moore 2003 build on tree methods for N-body problems and kernel density estimation, respectively. Holmes, Gray, and Isbell 2010 applies the dual tree approach to log-likelihood kernel conditional density estimation for bandwidth selection, assuming $D_Y = 1$.

1.6 New Approach

In this paper, we introduce a fast algorithm for kernel conditional density estimation based on Gray's dual tree approach. Heretofore, we believe this is the fastest kernel conditional density estimation algorithm for prediction. The generalized algorithm presented herein allows for arbitrary D_Y and D_X, extending the univariate case in both the label and conditioning variable.

2 Our Approach

2.1 Dual Tree for KCDE

Based on equation 2, we can apply the dual tree algorithm to both the numerator and denominator summations, then simply perform a pointwise division over the query set. Naively, we can build four trees, one query/reference pair for the set of attributes $Y_1, ..., Y_{D_Y}, X_1, ..., X_{D_X}$ and one query/reference pair for the conditional attributes $X_1, ..., X_{D_X}$. Performing a modification (the approximation of the numerator requires calculating the product of kernel functions, evident in equation 2; the minimum and maximum bounds between nodes requires filtering on both the set of conditional attributes and its complement) of the dual tree algorithm on each of the two pair of trees gives estimates for the numerators and denominators of query point masses. However, applying ϵ in both the numerator and denominator dual tree approximations fails to give an ϵ error rate in the quotients. Fortunately, we can leverage algebra to obtain component-wise error bounds.

Lemma 1. *Let $\epsilon > 0$, $n > 0$, and $d > 0$. If $\left|\frac{\hat{n}-n}{n}\right| \leq \alpha = \frac{\epsilon}{2+\epsilon}$ and $\left|\frac{\hat{d}-d}{d}\right| \leq \beta = \frac{\epsilon}{3+\epsilon}$, then $\left|\frac{\frac{\hat{n}}{\hat{d}} - \frac{n}{d}}{\frac{n}{d}}\right| \leq \epsilon$.*

Proof. We can transform the hypothesized inequalities to

$$\left(1 - \frac{\epsilon}{2+\epsilon}\right) n \leq \hat{n} \leq \left(1 + \frac{\epsilon}{2+\epsilon}\right) n \tag{12}$$

and

$$\left(1 - \frac{\epsilon}{3+\epsilon}\right) d \leq \hat{d} \leq \left(1 + \frac{\epsilon}{3+\epsilon}\right) d \tag{13}$$

The desired inequality is

$$(1-\epsilon)\frac{n}{d} \leq \frac{\hat{n}}{\hat{d}} \leq (1+\epsilon)\frac{n}{d} \tag{14}$$

Inverting equation 13, then combining equations 12 and 13, we have

$$\left(\frac{1-\frac{\epsilon}{2+\epsilon}}{1+\frac{\epsilon}{3+\epsilon}}\right)\frac{n}{d} \leq \frac{\hat{n}}{\hat{d}} \leq \left(\frac{1+\frac{\epsilon}{2+\epsilon}}{1-\frac{\epsilon}{3+\epsilon}}\right)\frac{n}{d} \tag{15}$$

Clearing and simplifying the bounds in equation 15, we have

$$\left(\frac{6+2\epsilon}{6+7\epsilon+2\epsilon^2}\right)\frac{n}{d} \leq \frac{\hat{n}}{\hat{d}} \leq \left(\frac{6+8\epsilon+2\epsilon^2}{6+3\epsilon}\right)\frac{n}{d} \tag{16}$$

For the upper bound, note that since $0 < \epsilon + \epsilon^2$,

$$6+8\epsilon+2\epsilon^2 < \epsilon+\epsilon^2+6+8\epsilon+2\epsilon^2 = (1+\epsilon)(6+3\epsilon) \tag{17}$$

Thus, we have

$$\frac{6+8\epsilon+2\epsilon^2}{6+3\epsilon} < 1+\epsilon \tag{18}$$

The lower bound follows similarly. □

Theorem 1. *Applying error bounds from 1, applying the dual tree algorithm to both the numerators and denominators of the query set evaluations specified in equation 2 gives relative error ϵ for each query point $x_q \in Q$.*

We can eliminate much of the memory footprint of our approach by generating a single tree for each of the query and reference sets, calculating upper and lower bounds on both the numerators and denominators simultaneously. If we reach our stopping criterion for either the numerator or denominator, but not both, we can simply filter our remaining recursion on the set failing to meet its respective criterion. We can share efforts further in that the maximum and minimum pairwise node distances along the conditioning attributes appear in both the numerator and denominator calculations. Key to the algorithm is the observation that the maximum and minimum distances between nodes are greedily selected in kd-trees, preserving optima not just in a single monotonic kernel expression but in the product of kernels. Unfortunately, ball trees fail to share this property; however, a slight adjustment in selecting the maximum and minimum distances over conditioning attributes solves this minor issue. We present the shared algorithm in figure 4.

3 Empirical Study[3]

3.1 Data Sets

We apply the KCDE algorithm to selections from the Sloan Digital Sky Survey (SDSS) DR6[4]. Unless stated

DualTreeKCDE$(Q, T, yContinue, xContinue)$
$dl_Y = N_T K_a(\delta^{max}_{y,QT}) K_b(\delta^{max}_{x,QT})$
$du_Y = N_T(K_a(\delta^{min}_{y,QT}) K_b(\delta^{min}_{x,QT}) - 1)$
$dl_X = N_T K_b(\delta^{max}_{x,QT})$, $du_X = N_T(K_b(\delta^{min}_{x,QT}) - 1)$
if $yContinue$ and $K_a(\delta^{min}_{y,QT}) - K_a(\delta^{max}_{y,QT}) \leq \frac{2\alpha}{N\hat{p}^{min}_{y,Q}}$ **then**
  for all $y_q \in Q$ **do**
    $\hat{p}^{max}_{y,q} += dl_Y$, $\hat{p}^{min}_{y,q} += du_Y$
  end for
  $\hat{p}^{min}_{y,Q} += dl_Y$, $\hat{p}^{max}_{y,Q} += du_Y$, $yContinue = false$
end if
if $xContinue$ and $K_b(\delta^{min}_{x,QT}) - K_b(\delta^{max}_{x,QT}) \leq \frac{2\beta}{N\hat{p}^{min}_{x,Q}}$ **then**
  for all $x_q \in Q$ **do**
    $\hat{p}^{max}_{x,q} += dl_X$, $\hat{p}^{min}_{x,q} += du_X$
  end for
  $\hat{p}^{min}_{x,Q} += dl_X$, $\hat{p}^{max}_{x,Q} += du_X$, $xContinue = false$
end if
if not $yContinue$ and not $xContinue$ **then**
  return
else
  if leaf(Q) and leaf(T) **then**
    DualTreeBaseKCDE(Q,T,$xContinue$,$yContinue$)
    return
  else
    DualTreeKCDE(Q.left,closer-of(Q.left,{T.left,T.right}), $xContinue$, $yContinue$)
    DualTreeKCDE(Q.left,farther-of(Q.left,{T.left,T.right}), $xContinue$, $yContinue$)
    DualTreeKCDE(Q.right,closer-of(Q.right,{T.left,T.right}), $xContinue$, $yContinue$)
    DualTreeKCDE(Q.right,farther-of(Q.right,{T.left,T.right}), $xContinue$, $yContinue$)
  end if
end if

DualTreeBaseKCDE$(Q, T, xContinue, yContinue)$
if $yContinue$ **then**
  for all $y_q \in Q$ **do**
    for all $y_t \in T$ **do**
      $\hat{p}^{min}_{y,q} += K_a(\|y_q - y_t\|)K_b(\|x_q - x_t\|)$, $\hat{p}^{max}_{y,q} += K_a(\|y_q - y_t\|)K_b(\|x_q - x_t\|)$
    end for
    $\hat{p}^{max}_{y,q} -= N_T$
  end for
  $\hat{p}^{min}_{y,Q} = \min_{q\in Q}\hat{p}^{min}_{y,q}$, $\hat{p}^{max}_{y,Q} = \max_{q\in Q}\hat{p}^{max}_{y,q}$
end if
if $xContinue$ **then**
  for all $x_q \in Q$ **do**
    for all $x_t \in T$ **do**
      $\hat{p}^{min}_{x,q} += K_b(\|x_q - x_t\|)$, $\hat{p}^{max}_{x,q} += K_b(\|x_q - x_t\|)$
    end for
    $\hat{p}^{max}_{x,q} -= N_T$
  end for
  $\hat{p}^{min}_{x,Q} = \min_{q\in Q}\hat{p}^{min}_{x,q}$, $\hat{p}^{max}_{x,Q} = \max_{q\in Q}\hat{p}^{max}_{x,q}$
end if

Figure 4: Shared Dual Tree for KCDE

[3] To determine optimal bandwidths, we apply the LSCV on data sets of smaller sizes of N (under 1K), performing a uniform search over $[0.0001, 10] \times [0.0001, 10]$; we obtain optimal bandwidth pairs, then calculate c_a, c_b, p_a, p_b where $(a^*, b^*) = (c_a N^{p_a}, c_b N^{p_b})$. With these formulas, we estimate optimal bandwidths for larger values of N.

[4] The first two features are x and y location coordinates of celestial objects; subsequent features are visual attributes.

otherwise, we apply the Epanechnikov kernel and sphere the data (subtract empirical feature means and scale by empirical standard deviations). We also apply shared dual tree to the MiniBooNE particle data set (see Frank and Asuncion 2010).

3.2 Scaling with Data Set Size

Table 1 exhibits run times using optimal bandwidths on various sizes of data taken from SDSS DR6 with $D_X = D_Y = 1$. Empirically, the shared dual tree algorithm requires a decaying fraction of the time required to execute the naive summation.

Shared Dual Tree on SDSS, $D_X = D_Y = 1$				
N	Shared Dual Tree	Naive	a^*	b^*
1K	0.089355	1.037543	0.00814	0.0004765
10K	2.299951	102.464691	0.00663	0.000227
100K	72.103015	10246.469*	0.0054	0.000108
1M	1885.179322	1024647*	0.00439	5.156e-5
10M	99699.903932	102464691*	0.003578	2.4567e-5

Table 1: Shared Dual Tree on SDSS, $D_X = D_Y = 1$

3.3 Scaling by Bandwidths

Over various data sets, values of N, and kernels, execution times with shared dual tree exhibit a similar pattern over the bandwidth pair a, b. Figure 5 exhibits this pattern. Notice that optimal runtimes occur when either both bandwidths are large (greater than 10) or either bandwidth is quite small (less than 0.1.) Bandwidths exhibiting suboptimal runtimes lie along the two crested regions along each bandwidth axis. The optimal bandwidth rests between the crested intersection and the origin, somewhat on the downward slope. The crested region seems to coincide with naive time complexity, and the pattern persists with higher N.

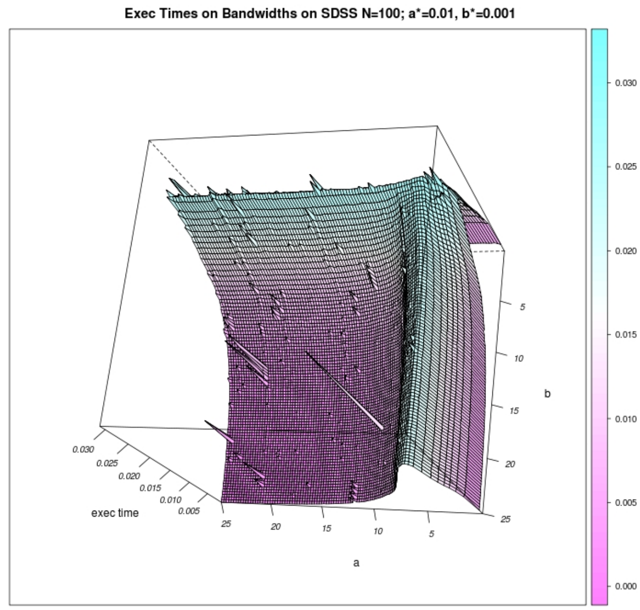

Figure 5: Execution Times Per Bandwidth, $N = 100$, $D_X = D_Y = 1$

3.4 Scaling by Dimension

Table 2 exhibits run times over various values of D_X.

Shared Dual Tree on MiniBooNE, $D_Y = 1, N = 100,000$				
D_X	Shared Dual Tree	Naive	a^*	b^*
4	4650	20500*	4.3e-7	0.17
8	???	41000*	3e-8	0.25
16	3500	82000*	2.2e-9	0.32

Table 2: Shared Dual Tree on MiniBooNe, $D_Y = 1, N = 100,000$

3.5 Kernel Choice

Comparisons between the spherical, Epanechnikov, and Gaussian kernels across selections from SDSS DR6 appear in table 3. Notice, as expected, that the Gaussian kernel requires the most time, roughly 50% more than that of the Epanechnikov. The spherical kernel, as expected, offers the fastest computation time.

Kernels Using Shared Dual Tree on SDSS, $D_X = D_Y = 1$			
N	Spherical	Epanechnikov	Gaussian
1K	0.070923	0.089355	0.128079
10K	2.240860	2.299951	3.313672
100K	71.934361	72.103015	104.501412

Table 3: Shared Dual Tree with Various Kernels on SDSS, $D_X = D_Y = 1$

4 Conclusions

The shared dual tree algorithm offers significant speed-up over the naive computation when bandwidths a and b are sufficiently small or sufficiently large. Building on a series of dual tree approaches to density estimation (Gray and Moore 2003, Holmes, Gray, Isbell 2010), shared dual tree extends the framework to compute kernel conditional density estimates with an approximation guarantee and an approach in cases in which the weights are zero.

5 References

1. T. Budavari. A Unified Framework for Photometric Redshifts. *The Astrophysical Journal*, Volume 695, Issue 1, 747-754, 2009.
2. X. Chen, O. Linton, P.M. Robinson. The Estimation of Conditional Densities, unpublished discussion paper, No.EM/01/415, 2001.
3. V.A. Epanechnikov. Non-parametric estimation of a multivariate probability density. *Theory of Probability and its Applications 14*, 153158, 1969.
4. J. Fan and J. Marron. Fast Implementations of Nonparametric Curve Estimators. *Journal of Computational and Graphical Statistics*, 3:35-56, 1994.
5. A. Frank and A. Asuncion. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science, 2010.
6. J.G.D. Gooijer and D. Zerom, On conditional density estimation. *Statistica Neerlandica 57 (2)*, 159176, 2003.
7. A.G. Gray and A.W. Moore. N-Body Problems in Statistical Learning. T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Information Processing Systems 13 (December 2000)*. MIT Press, 2001.
8. A.G. Gray and A. W. Moore. Rapid Evaluation of Multiple Density Models. In *Artificial Intelligence and Statistics 2003*, 2003.
9. B.E. Hansen. Nonparametric conditional density estimation, unpublished manuscript. URL: http://www.economics.ucr.edu/seminars/fall04/ Bruce-Hansen.pdf, 2004.
10. M.P. Holmes, A.G. Gray, and C.L. Isbell Jr. Fast Kernel Conditional Density Estimation: A Dual Tree Monte Carlo Approach. *Computational Statistics and Data Analysis*, 1707-1718, 2010.
11. M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Annals of Mathematical Statistics* 27: 832837, 1956.
12. D.W. Scott. *Multivariate Density Estimation*. Wiley, 1992.
13. B.W. Silverman. *Density Estimation for Statistics and Data Analysis*, Chapman and Hall/CRC, 1986.
14. L. Song, A. Gretton, C. Guestrin. Nonparametric Tree Graphical Models via Kernel Embeddings. 765-772, 20XX.