Dual-Tree Fast Gauss Transforms

Dongryeol Lee
Computer Science
Carnegie Mellon Univ.
dongryel@cmu.edu

Alexander Gray
Computer Science
Carnegie Mellon Univ.
agray@cs.cmu.edu

Andrew Moore
Computer Science
Carnegie Mellon Univ.
awm@cs.cmu.edu

Abstract

In previous work we presented an efficient approach to computing kernel summations which arise in many machine learning methods such as kernel density estimation. This approach, dual-tree recursion with finite-difference approximation, generalized existing methods for similar problems arising in computational physics in two ways appropriate for statistical problems: toward distribution sensitivity and general dimension, partly by avoiding series expansions. While this proved to be the fastest practical method for multivariate kernel density estimation at the optimal bandwidth, it is much less efficient at larger-than-optimal bandwidths. In this work, we explore the extent to which the dual-tree approach can be integrated with multipole-like Hermite expansions in order to achieve reasonable efficiency across all bandwidth scales, though only for low dimensionalities. In the process, we derive and demonstrate the first truly hierarchical fast Gauss transforms, effectively combining the best tools from discrete algorithms and continuous approximation theory.

1 Fast Gaussian Summation

Kernel summations are fundamental in both statistics/learning and computational physics. This paper will focus on the common form $G(x_q) = \sum_{r=1}^{N_R} e^{\frac{-||x_q - x_r||^2}{2h^2}}$ *i.e.* where the kernel is the Gaussian kernel with scaling parameter, or *bandwidth* h, there are N_R *reference points* x_r, and we desire the sum for N_Q different *query points* x_q. Such kernel summations appear in a wide array of statistical/learning methods [5], perhaps most obviously in kernel density estimation [11], the most widely used distribution-free method for the fundamental task of density estimation, which will be our main example. Understanding kernel summation algorithms from a recently developed unified perspective [6] begins with the picture of Figure 1, then separately considers the discrete and continuous aspects.

Discrete/geometric aspect. In terms of discrete algorithmic structure, the dual-tree framework of [5], in the context of kernel summation, generalizes all of the well-known algorithms. [1] It was applied to the problem of kernel density estimation in [7] using a simple

[1] These include the Barnes-Hut algorithm [2], the Fast Multipole Method [8], Appel's algorithm [1], and the WSPD [4]: the dual-tree method is a node-node algorithm (considers query regions rather than points), is fully recursive, can use distribution-sensitive data structures such as *kd*-trees, and is bichromatic (can specialize for differing query and reference sets).

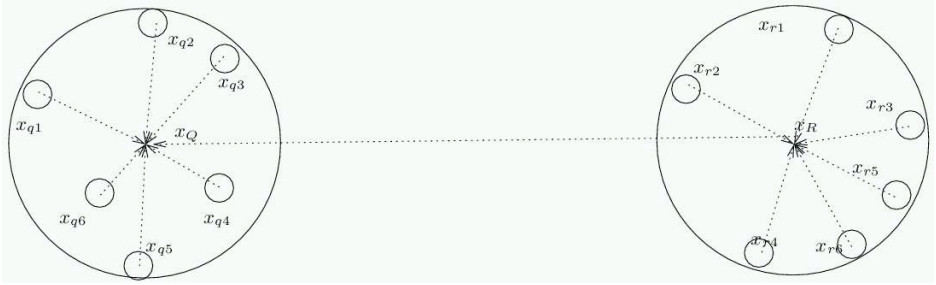

Figure 1: The basic idea is to approximate the kernel sum contribution of some subset of the reference points X_R, lying in some compact region of space R with centroid x_R, to a query point. In more efficient schemes a query region is considered, *i.e.* the approximate contribution is made to an entire subset of the query points X_Q lying in some region of space Q, with centroid x_Q.

finite-difference approximation, which is tantamount to a centroid approximation. Partially by avoiding series expansions, which depend explicitly on the dimension, the result was the fastest such algorithm for general dimension, when operating at the optimal bandwidth. Unfortunately, when performing cross-validation to determine the (initially unknown) optimal bandwidth, both suboptimally small and large bandwidths must be evaluated. The finite-difference-based dual-tree method tends to be efficient at or below the optimal bandwidth, and at very large bandwidths, but for intermediately-large bandwidths it suffers.

Continuous/approximation aspect. This motivates investigating a multipole-like series approximation which is appropriate for the Gaussian kernel, as introduced by [9], which can be shown the generalize the centroid approximation. We define the Hermite functions $h_n(t)$ by $h_n(t) = e^{-t^2} H_n(t)$, where the Hermite polynomials $H_n(t)$ are defined by the Rodrigues formula: $H_n(t) = (-1)^n e^{t^2} D^n e^{-t^2}, t \in \mathbb{R}^1$. After scaling and shifting the argument t appropriately, then taking the product of univariate functions for each dimension, we obtain the multivariate *Hermite expansion*

$$G(x_q) = \sum_{r=1}^{N_R} e^{\frac{-||x_q - x_r||^2}{2h^2}} = \sum_{r=1}^{N_R} \sum_{\alpha \geq 0} \frac{1}{\alpha!} \left(\frac{x_r - x_R}{\sqrt{2h^2}} \right)^{\alpha} h_\alpha \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right) \tag{1}$$

where we've adopted the usual multi-index notation as in [9]. This can be re-written as

$$G(x_q) = \sum_{r=1}^{N_R} e^{\frac{-||x_q - x_r||^2}{2h^2}} = \sum_{r=1}^{N_R} \sum_{\alpha \geq 0} \frac{1}{\alpha!} h_\alpha \left(\frac{x_r - x_Q}{\sqrt{2h^2}} \right) \left(\frac{x_q - x_Q}{\sqrt{2h^2}} \right)^{\alpha} \tag{2}$$

to express the sum as a *Taylor (local) expansion* about a nearby representative centroid x_Q in the query region. We will be using both types of expansions simultaneously.

Since series approximations only hold locally, Greengard and Rokhlin [8] showed that it is useful to think in terms of a set of three 'translation operators' for converting between expansions centered at different points, in order to create their celebrated hierarchical algorithm. This was done in the context of the Coulombic kernel, but the Gaussian kernel has importantly different mathematical properties. The original Fast Gauss Transform (FGT) [9] was based on a flat grid, and thus provided only one operator ("H2L" of the next section), with an associated error bound (which was unfortunately incorrect). The Improved Fast Gauss Transform (IFGT) [14] was based on a flat set of clusters and thus provided only the H2L operator, though for a rearranged series approximation intended to be more favorable in higher dimensions, also with an incorrect error bound. We will show the derivations of all the translation operators and associated error bounds needed to obtain, for the first time, a *hierarchical* algorithm for the Gaussian kernel.

2 Translation Operators and Error Bounds

The first operator converts a multipole expansion of a reference node to form a local expansion centered at the centroid of the query node, and is our main approximation workhorse.

Lemma 2.1. Hermite-to-local (H2L) translation operator for Gaussian kernel *(as presented in Lemma 2.2 in [9, 10])***:** *Suppose we are given a reference node X_R and a query node X_Q, and given the Hermite expansion centered at a centroid x_R of X_R: $G(x_q) = \sum_{\alpha \geq 0} A_\alpha h_\alpha \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right)$. Then, the Taylor expansion of the Hermite expansion at the centroid x_Q of the query node X_Q is given by $G(x_q) = \sum_{\beta \geq 0} B_\beta \left(\frac{x_q - x_Q}{\sqrt{2h^2}} \right)^\beta$ where $B_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} A_\alpha h_{\alpha+\beta} \left(\frac{x_Q - x_R}{\sqrt{2h^2}} \right)$.*

Proof. (sketch) The proof consists of replacing the Hermite function portion of the expansion with its Taylor series. □

Note that we can rewrite $G(x_q) = \sum_{\alpha \geq 0} \left[\sum_{r=1}^{N_R} \frac{1}{\alpha!} \left(\frac{x_r - x_R}{\sqrt{2h^2}} \right)^\alpha \right] h_\alpha \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right)$ by interchanging the summation order, such that the term in the brackets depends only on the reference points, and can thus be computed indepedent of any query location – we will call such terms Hermite moments. The next operator allows the efficient pre-computation of the Hermite moments in the reference tree in a bottom-up fashion from its children.

Lemma 2.2. Hermite-to-Hermite (H2H) translation operator for Gaussian kernel: *Suppose we are given the Hermite expansion centered at a centroid $x_{R'}$ in a reference node $X_{R'}$: $G(x_q) = \sum_{\alpha \geq 0} A'_\alpha h_\alpha \left(\frac{x_q - x_{R'}}{\sqrt{2h^2}} \right)$. Then, this same Hermite expansion shifted to a new location x_R of the parent node of X_R is given by $G(x_q) = \sum_{\gamma \geq 0} A_\gamma h_\gamma \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right)$ where $A_\gamma = \sum_{0 \leq \alpha \leq \gamma} \frac{1}{(\gamma - \alpha)!} A'_\alpha \left(\frac{x_{R'} - x_R}{\sqrt{2h^2}} \right)^{\gamma - \alpha}$.*

Proof. We simply replace the Hermite function part of the expansion by a new Taylor series, as follows:

$$
\begin{aligned}
G(x_q) &= \sum_{\alpha \geq 0} A'_\alpha h_\alpha \left(\frac{x_q - x_{R'}}{\sqrt{2h^2}} \right) \\
&= \sum_{\alpha \geq 0} A'_\alpha \sum_{\beta \geq 0} \frac{1}{\beta!} \left(\frac{x_R - x_{R'}}{\sqrt{2h^2}} \right)^\beta (-1)^{|\beta|} h_{\alpha+\beta} \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right) \\
&= \sum_{\alpha \geq 0} \sum_{\beta \geq 0} A'_\alpha \frac{1}{\beta!} \left(\frac{x_R - x_{R'}}{\sqrt{2h^2}} \right)^\beta (-1)^{|\beta|} h_{\alpha+\beta} \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right) \\
&= \sum_{\alpha \geq 0} \sum_{\beta \geq 0} A'_\alpha \frac{1}{\beta!} \left(\frac{x_{R'} - x_R}{\sqrt{2h^2}} \right)^\beta h_{\alpha+\beta} \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right) \\
&= \sum_{\gamma \geq 0} \left[\sum_{0 \leq \alpha \leq \gamma} \frac{1}{(\gamma - \alpha)!} A'_\alpha \left(\frac{x_{R'} - x_R}{\sqrt{2h^2}} \right)^{\gamma - \alpha} \right] h_\gamma \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right)
\end{aligned}
$$

where $\gamma = \alpha + \beta$. □

The next operator acts as a "clean-up" routine in a hierarchical algorithm. Since we can approximate at different scales in the query tree, we must somehow combine all the approximations at the end of the computation. By performing a breadth-first traversal of the query tree, the L2L operator shifts a node's local expansion to the centroid of each child.

Lemma 2.3. Local-to-local (L2L) translation operator for Gaussian kernel: *Suppose you are given a Taylor expansion centered at a centroid $x_{Q'}$ of a query node $X_{Q'}$: $G(x_q) = \sum_{\beta \geq 0} B_\beta \left(\frac{x_q - x_{Q'}}{\sqrt{2h^2}} \right)^\beta$. Then, the Taylor expansion obtained by shifting this expansion to the new centroid x_Q of the child node X_Q is $G(x_q) = \sum_{\alpha \geq 0} \left[\sum_{\beta \geq \alpha} \frac{\beta!}{\alpha!(\beta-\alpha)!} B_\beta \left(\frac{x_Q - x_{Q'}}{\sqrt{2h^2}} \right)^{\beta-\alpha} \right] \left(\frac{x_q - x_Q}{\sqrt{2h^2}} \right)^\alpha$.*

Proof. We expand the part involving x_q and $x_{Q'}$ in a new Taylor series centered at x_Q:

$$
\begin{aligned}
G(x_q) &= \sum_{\beta \geq 0} B_\beta \left(\frac{x_q - x_{Q'}}{\sqrt{2h^2}} \right)^\beta \\
&= \sum_{\beta \geq 0} B_\beta \sum_{\alpha \geq 0} \frac{1}{\alpha!} \left(D^\alpha \left[\left(\frac{x_q - x_{Q'}}{\sqrt{2h^2}} \right)^\beta \right] (x_Q) \right) (x_q - x_Q)^\alpha \\
&= \sum_{\beta \geq 0} B_\beta \sum_{\alpha \leq \beta} \frac{1}{\alpha!} \left(\frac{1}{\sqrt{2h^2}} \right)^\alpha \left(\prod_{d=1}^{D} \beta_D (\beta_D - 1) \cdots (\beta_D - \alpha_D + 1) \right) \left(\frac{x_Q - x_{Q'}}{\sqrt{2h^2}} \right)^\beta (x_q - x_Q)^\alpha \\
&= \sum_{\beta \geq 0} \sum_{\alpha \leq \beta} B_\beta \frac{\beta!}{\alpha!(\beta-\alpha)!} \left(\frac{x_Q - x_{Q'}}{\sqrt{2h^2}} \right)^{\beta-\alpha} \left(\frac{x_q - x_Q}{\sqrt{2h^2}} \right)^\alpha .
\end{aligned}
$$

whose summation order can be interchanged to achieve the result. □

Because the Hermite and the Taylor expansion are truncated after taking p^D terms, we incur an error in approximation. The original error bounds for the Gaussian kernel in [9, 10] were wrong and corrections were shown in [3]. Here, we will present all necessary three error bounds incurred in performing translation operators. We note that these error bounds place limits on the size of the query node and the reference node. [2]

Lemma 2.4. Error Bound for Truncating an Hermite Expansion *(as presented in [3])***:** *Suppose we are given an Hermite expansion of a reference node about its centroid x_R: $G(x_q) = \sum_{\alpha \geq 0} A_\alpha h_\alpha \left(\frac{x_q - x_R}{\sqrt{2h^2}} \right)$ where $A_\alpha = \sum_{r=1}^{N_R} \frac{1}{\alpha!} \left(\frac{x_r - x_R}{\sqrt{2h^2}} \right)^\alpha$. For a fixed query point x_q, the error due to truncating the series after the first p^D term is $|\epsilon_M(p)| \leq \frac{N_R}{(1-r)^D} \sum_{k=0}^{D-1} \binom{D}{k} (1 - r^p)^k \left(\frac{r^p}{\sqrt{p!}} \right)^{D-k}$ where each reference data point x_r in the reference node satisfies $||x_r - x_R||_\infty < rh$ for $r < 1$.*

Proof. (sketch) We expand the Hermite expansion as a product of one-dimensional Hermite functions, and utilize a bound on one-dimensional Hermite functions due to [13]: $\frac{1}{n!} |h_n(x)| \leq \frac{2^{\frac{n}{2}}}{\sqrt{n!}} e^{\frac{-x^2}{2}}, n \geq 0, x \in \mathbb{R}^1$. □

[2] Strain [12] proposed the interesting idea of using Stirling's formula (for any non-negative integer n: $\left(\frac{n+1}{e}\right)^n \leq n!$) to lift the node size constraint; one might imagine that this could allow approximation of larger regions in a tree-based algorithm. Unfortunately, the error bounds developed in [12] were also incorrect. We have derived the three necessary corrected error bounds based on the techniques in [3]. However, due to space, and because using these bounds actually degraded performance slightly, we do not include those lemmas here.

Lemma 2.5. Error Bound for Truncating a Taylor Expansion Converted from an Hermite Expansion of Infinite Order: *Suppose we are given the following Taylor expansion about the centroid x_Q of a query node $G(x_q) = \sum_{\beta \geq 0} B_\beta \left(\frac{x_q - x_Q}{\sqrt{2h^2}}\right)^\beta$ where $B_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} A_\alpha h_{\alpha+\beta}\left(\frac{x_Q - x_R}{\sqrt{2h^2}}\right)$ and A_α's are the coefficients of the Hermite expansion centered at the reference node centroid x_R. Then, truncating the series after p^D terms satisfies the error bound $|\epsilon_L(p)| \leq \frac{N_R}{(1-r)^D} \sum_{k=0}^{D-1} \binom{D}{k}(1-r^p)^k \left(\frac{r^p}{\sqrt{p!}}\right)^{D-k}$ where $||x_q - x_Q||_\infty < rh$ for $r < 1$.*

Proof. Taylor expansion of the Hermite function yields

$$\begin{aligned} e^{\frac{-||x_q - x_r||^2}{2h^2}} &= \sum_{\beta \geq 0} \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} \frac{1}{\alpha!} \left(\frac{x_r - x_R}{\sqrt{2h^2}}\right)^\alpha h_{\alpha+\beta}\left(\frac{x_Q - x_R}{\sqrt{2h^2}}\right)\left(\frac{x_q - x_Q}{\sqrt{2h^2}}\right)^\beta \\ &= \sum_{\beta \geq 0} \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha \geq 0} \frac{1}{\alpha!} \left(\frac{x_R - x_r}{\sqrt{2h^2}}\right)^\alpha (-1)^{|\alpha|} h_{\alpha+\beta}\left(\frac{x_Q - x_R}{\sqrt{2h^2}}\right)\left(\frac{x_q - x_Q}{\sqrt{2h^2}}\right)^\beta \\ &= \sum_{\beta \geq 0} \frac{(-1)^{|\beta|}}{\beta!} h_\beta \left(\frac{x_Q - x_r}{\sqrt{2h^2}}\right)\left(\frac{x_q - x_Q}{\sqrt{2h^2}}\right)^\beta \end{aligned}$$

Use $e^{\frac{-||x_q - x_r||^2}{2h^2}} = \prod_{i=1}^{D} \left(u_p(x_{q_i}, x_{r_i}, x_{Q_i}) + v_p(x_{q_i}, x_{r_i}, x_{Q_i})\right)$ for $1 \leq i \leq D$, where

$$\begin{aligned} u_p(x_{q_i}, x_{r_i}, x_{Q_i}) &= \sum_{n_i=0}^{p-1} \frac{(-1)^{n_i}}{n_i!} h_{n_i}\left(\frac{x_{Q_i} - x_{r_i}}{\sqrt{2h^2}}\right)\left(\frac{x_{q_i} - x_{Q_i}}{\sqrt{2h^2}}\right)^{n_i} \\ v_p(x_{q_i}, x_{r_i}, x_{Q_i}) &= \sum_{n_i=p}^{\infty} \frac{(-1)^{n_i}}{n_i!} h_{n_i}\left(\frac{x_{Q_i} - x_{r_i}}{\sqrt{2h^2}}\right)\left(\frac{x_{q_i} - x_{Q_i}}{\sqrt{2h^2}}\right)^{n_i}. \end{aligned}$$

These univariate functions respectively satisfy $u_p(x_{q_i}, x_{r_i}, x_{Q_i}) \leq \frac{1-r^p}{1-r}$ and $v_p(x_{q_i}, x_{r_i}, x_{Q_i}) \leq \frac{1}{\sqrt{p!}} \frac{r^p}{1-r}$, for $1 \leq i \leq D$, achieving the multivariate bound. □

Lemma 2.6. Error Bound for Truncating a Taylor Expansion Converted from an Already Truncated Hermite Expansion: *A truncated Hermite expansion centered about the centroid x_R of a reference node $G(x_q) = \sum_{\alpha < p} A_\alpha h_\alpha \left(\frac{x_q - x_R}{\sqrt{2h^2}}\right)$ has the following Taylor expansion about the centroid x_Q of a query node: $G(x_q) = \sum_{\beta \geq 0} C_\beta \left(\frac{x_q - x_Q}{\sqrt{2h^2}}\right)^\beta$ where the coefficients C_β are given by $C_\beta = \frac{(-1)^{|\beta|}}{\beta!} \sum_{\alpha < p} A_\alpha h_{\alpha+\beta}\left(\frac{x_Q - x_R}{\sqrt{2h^2}}\right)$. Truncating the series after p^D terms satisfies the error bound $|\epsilon_L(p)| \leq \frac{N_R}{(1-2r)^{2D}} \sum_{k=0}^{D-1} \binom{D}{k}((1 - (2r)^p)^2)^k \left(\frac{((2r)^p)(2-(2r)^p)}{\sqrt{p!}}\right)^{D-k}$ for a query node X_Q for which $||x_q - x_Q||_\infty < rh$, and a reference node X_R for which $\forall x_r \in x_R, ||x_r - x_R||_\infty < rh$ for $r < \frac{1}{2}$.*

Proof. We define the following for $1 \leq i \leq D$:

$$u_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) = \sum_{n_i=0}^{p-1} \frac{(-1)^{n_i}}{n_i!} \sum_{n_j=0}^{p-1} \frac{1}{n_j!} \left(\frac{x_{R_i} - x_{r_i}}{\sqrt{2h^2}}\right)^{n_j} (-1)^{n_j} h_{n_i+n_j} \left(\frac{x_{Q_i} - x_{R_i}}{\sqrt{2h^2}}\right) \left(\frac{x_{q_i} - x_{Q_i}}{\sqrt{2h^2}}\right)^{n_i}$$

$$v_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) = \sum_{n_i=0}^{p-1} \frac{(-1)^{n_i}}{n_i!} \sum_{n_j=p}^{\infty} \frac{1}{n_j!} \left(\frac{x_{R_i} - x_{r_i}}{\sqrt{2h^2}}\right)^{n_j} (-1)^{n_j} h_{n_i+n_j} \left(\frac{x_{Q_i} - x_{R_i}}{\sqrt{2h^2}}\right) \left(\frac{x_{q_i} - x_{Q_i}}{\sqrt{2h^2}}\right)^{n_i}$$

$$w_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) = \sum_{n_i=p}^{\infty} \frac{(-1)^{n_i}}{n_i!} \sum_{n_j=0}^{\infty} \frac{1}{n_j!} \left(\frac{x_{R_i} - x_{r_i}}{\sqrt{2h^2}}\right)^{n_j} (-1)^{n_j} h_{n_i+n_j} \left(\frac{x_{Q_i} - x_{R_i}}{\sqrt{2h^2}}\right) \left(\frac{x_{q_i} - x_{Q_i}}{\sqrt{2h^2}}\right)^{n_i}$$

Note that $e^{\frac{-||x_q - x_r||^2}{2h^2}} = \prod_{i=1}^{D} \left(u_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) + v_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) + w_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i})\right)$
for $1 \leq i \leq D$. Using the bound for Hermite functions and the property of geometric series, we obtain the following upper bounds:

$$u_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) \leq \sum_{n_i=0}^{p-1} \sum_{n_j=0}^{p-1} (2r)^{n_i} (2r)^{n_j} = \left(\frac{1 - (2r)^p)}{1 - 2r}\right)^2$$

$$v_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) \leq \frac{1}{\sqrt{p!}} \sum_{n_i=0}^{p-1} \sum_{n_j=p}^{\infty} (2r)^{n_i} (2r)^{n_j} = \frac{1}{\sqrt{p!}} \left(\frac{1 - (2r)^p}{1 - 2r}\right) \left(\frac{(2r)^p}{1 - 2r}\right)$$

$$w_p(x_{q_i}, x_{r_i}, x_{Q_i}, x_{R_i}) \leq \frac{1}{\sqrt{p!}} \sum_{n_i=p}^{\infty} \sum_{n_j=0}^{\infty} (2r)^{n_i} (2r)^{n_j} = \frac{1}{\sqrt{p!}} \left(\frac{1}{1 - 2r}\right) \left(\frac{(2r)^p}{1 - 2r}\right)$$

Therefore,

$$\left| e^{\frac{-||x_q - x_r||^2}{2h^2}} - \prod_{i=1}^{D} u_p(x_{q_i}, x_{r_i}, x_{R_i}) \right| \leq (1 - 2r)^{-2D} \sum_{k=0}^{D-1} \binom{D}{k} ((1 - (2r)^p)^2)^k \left(\frac{((2r)^p)(2 - (2r)^p)}{\sqrt{p!}}\right)^{D-k}$$

$$\left| \sum_{r=1}^{N_R} e^{\frac{-||x_q - x_r||^2}{2h^2}} - \sum_{\beta < p} C_\beta \left(\frac{x_q - x_Q}{\sqrt{2h^2}}\right)^\beta \right| \leq \frac{N_R}{(1 - 2r)^{2D}} \sum_{k=0}^{D-1} \binom{D}{k} ((1 - (2r)^p)^2)^k \left(\frac{((2r)^p)(2 - (2r)^p)}{\sqrt{p!}}\right)^{D-k}$$

which completes the proof. □

3 Algorithm and Results

Algorithm. The algorithm mainly consists of making the function call **DFGT**(Q.root,R.root), *i.e.* calling the recursive function **DFGT**() with the root nodes of the query tree and reference tree. After the **DFGT**() routine is completed, the pre-order traversal of the query tree implied by the L2L operator is performed. Before the **DFGT**() routine is called, the reference tree could be initialized with Hermite coefficients stored in each node using the H2H translation operator, but instead we will compute them as needed on the fly. It adaptively chooses among three possible methods for approximating the summation contribution of the points in node R to the queries in node Q, which are self-explanatory, based on crude operation count estimates. G_Q^{min}, a running lower bound on the kernel sum $G(x_q)$ for any $x_q \in X_Q$, is used to ensure locally that the global relative error is ϵ or less. This automatic mechanism allows the user to specify only an error tolerance ϵ rather than other tweak parameters. Upon approximation, the upper and lower bounds on G for Q and all its children are updated; the latter can be done in an $O(1)$ delayed fashion as in [7]. The remainder of the routine implements the characteristic four-way dual-tree recursion. We also tested a hybrid method (DFGTH) which approximates if either of the DFD or DFGT approximation criteria are met.

DFGT(Q, R)

`if` R.maxside $< 2h$, $p_{DH} =$ the smallest $p \geq 1$ such that
$\frac{N_R}{(1-r)^D} \sum_{k=0}^{D-1} \binom{D}{k}(1-r^p)^k \left(\frac{r^p}{\sqrt{p!}}\right)^{D-k} < \epsilon G_Q^{min}$, `else` $p_{DH} = \infty$.

`if` Q.maxside $< 2h$, $p_{DL} =$ the smallest $p \geq 1$ such that
$\frac{N_R}{(1-r)^D} \sum_{k=0}^{D-1} \binom{D}{k}(1-r^p)^k \left(\frac{r^p}{\sqrt{p!}}\right)^{D-k} < \epsilon G_Q^{min}$, `else` $p_{DL} = \infty$.

`if` $\max(Q$.maxside,R.maxside$) < 2h$, $p_{H2L} =$ the smallest $p \geq 1$ such that
$\frac{N_R}{(1-2r)^{2D}} \sum_{k=0}^{D-1} \binom{D}{k}((1-(2r)^p)^2)^k \left(\frac{((2r)^p)(2-(2r)^p)}{\sqrt{p!}}\right)^{D-k} < \epsilon G_Q^{min}$, `else` $p_{H2L} = \infty$.

$c_{DH} = p_{DH}^D N_Q$. $c_{DL} = p_{DL}^D N_R$. $c_{H2L} = D p_{H2L}^{D+1}$. $c_{Direct} = D N_Q N_R$.

`if` no Hermite coefficient of order p_{DH} exists for X_R,
Compute it. $c_{DH} = c_{DH} + p_{DH}^D N_R$.

`if` no Hermite coefficient of order p_{H2L} exists for X_R,
Compute it. $c_{H2L} = c_{H2L} + p_{H2L}^D N_R$.

$c = \min(c_{DH}, c_{DL}, c_{H2L}, c_{Direct})$.

`if` $c = c_{DH}$,
Evaluate each x_q at the Hermite series of order p_{DH} centered about x_R of X_R using Equation 1. (Direct Hermite)

`if` $c = c_{DL}$,
Accumulate each $x_r \in X_R$ as the Taylor series of order p_{DL} about the center x_Q of X_Q using Equation 2. (Direct Local)

`if` $c = c_{H2L}$,
Convert the Hermite series of order p_{H2L} centered about x_R of X_R to the Taylor series of the same order centered about x_Q of X_Q using Lemma 2.1. (Hermite-to-Local)

`if` $c \neq c_{Direct}$,
Update G^{min} and G^{max} in Q and all its children. `return` .

`if` leaf(Q) and leaf(R),
Perform the naive algorithm on every pair of points in Q and R.
`else`
DFGT(Q.left, R.left). **DFGT**(Q.left, R.right).
DFGT(Q.right, R.left). **DFGT**(Q.right, R.right).

Experimental results. We empirically studied the runtime [3] performance of five algorithms on five real-world datasets for kernel density estimation at every query point with a range of bandwidths, from 3 orders of magnitude smaller than optimal to three orders larger than optimal, according to the standard least-squares cross-validation score [11]. The naive algorithm computes the sum explicitly and thus exactly. We have limited all datasets to 50K points so that true relative error, *i.e.* $\left(|\widehat{G}(x_q) - G_{true}(x_q)|\right) / G_{true}(x_q)$, can be evaluated, and set the tolerance at 1% relative error for all query points. When any method fails to achieve the error tolerance in less time than twice that of the naive method, we give up. Codes for the FGT [9] and for the IFGT [14] were obtained from the authors' websites. Note that both of these methods require the user to tweak parameters, while the others are automatic. [4] DFD refers to the depth-first dual-tree finite-difference method [7].

[3] All times include all preprocessing costs including any data structure construction. Times are measured in CPU seconds on a dual-processor AMD Opteron 242 machine with 8 Gb of main memory and 1 Mb of CPU cache. All the codes that we have written and obtained are written in C and C++, and was compiled under `-O6 -funroll-loops` flags on Linux kernel 2.4.26.

[4] For the FGT, the number of grid points in each dimension was set to 100 after some trial and error based on the true values (which are unknown in practice). For the IFGT, which has multiple

Algorithm \ scale	*0.001*	*0.01*	*0.1*	*1*	*10*	*100*	*1000*
sj2-50000-2 (astronomy: positions), $D = 2$, $N = 50000$, $h^* = 0.00139506$							
Naive	301.696	301.696	301.696	301.696	301.696	301.696	301.696
FGT	out of RAM	out of RAM	out of RAM	3.892312	2.01846	0.319538	0.183616
IFGT	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	7.576783
DFD	0.837724	1.087066	1.658592	6.018158	62.077669	151.590062	1.551019
DFGT	0.849935	1.11567	4.599235	72.435177	18.450387	2.777454	2.532401
DFGTH	**0.846294**	**1.10654**	**1.683913**	**6.265131**	**5.063365**	**1.036626**	**0.68471**
colors50k (astronomy: colors), $D = 2$, $N = 50000$, $h^* = 0.0016911$							
Naive	301.696	301.696	301.696	301.696	301.696	301.696	301.696
FGT	out of RAM	out of RAM	out of RAM	$> 2\times$Naive	$> 2\times$Naive	0.475281	0.114430
IFGT	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	7.55986
DFD	1.095838	1.469454	2.802112	30.294007	280.633106	81.373053	3.604753
DFGT	1.099828	1.983888	29.231309	285.719266	12.886239	5.336602	3.5638
DFGTH	**1.081216**	**1.47692**	**2.855083**	**24.598749**	**7.142465**	**1.78648**	**0.627554**
edsgc-radec-rnd (astronomy: angles), $D = 2$, $N = 50000$, $h^* = 0.00466204$							
Naive	301.696	301.696	301.696	301.696	301.696	301.696	301.696
FGT	out of RAM	out of RAM	out of RAM	2.859245	1.768738	0.210799	0.059664
IFGT	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	7.585585
DFD	0.812462	1.083528	1.682261	5.860172	63.849361	357.099354	0.743045
DFGT	0.84023	1.120015	4.346061	73.036687	21.652047	3.424304	1.977302
DFGTH	**0.821672**	**1.104545**	**1.737799**	**6.037217**	**5.7398**	**1.883216**	**0.436596**
mockgalaxy-D-1M-rnd (cosmology: positions), $D = 3$, $N = 50000$, $h^* = 0.000768201$							
Naive	354.868751	354.868751	354.868751	354.868751	354.868751	354.868751	354.868751
FGT	out of RAM	out of RAM	out of RAM	out of RAM	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive
IFGT	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive
DFD	0.70054	0.701547	0.761524	0.843451	1.086608	42.022605	383.12048
DFGT	0.73007	0.733638	0.799711	0.999316	50.619588	125.059911	109.353701
DFGTH	**0.724004**	**0.719951**	**0.789002**	**0.877564**	**1.265064**	**22.6106**	**87.488392**
bio5-rnd (biology: drug activity), $D = 5$, $N = 50000$, $h^* = 0.000567161$							
Naive	364.439228	364.439228	364.439228	364.439228	364.439228	364.439228	364.439228
FGT	out of RAM	out of RAM	out of RAM	out of RAM	out of RAM	out of RAM	out of RAM
IFGT	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive
DFD	2.249868	2.4958865	4.70948	12.065697	94.345003	412.39142	107.675935
DFGT	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive
DFGTH	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive	$> 2\times$Naive

Discussion. The experiments indicate that the DFGTH method is able to achieve reasonable performance across all bandwidth scales. Unfortunately none of the series approximation-based methods do well on the 5-dimensional data, as expected, highlighting the main weakness of the approach presented. Pursuing corrections to the error bounds necessary to use the intriguing series form of [14] may allow an increase in dimensionality.

References

[1] A. W. Appel. An Efficient Program for Many-Body Simulations. *SIAM Journal on Scientifi c and Statistical Computing*, 6(1):85–103, 1985.

[2] J. Barnes and P. Hut. A Hierarchical $O(NlogN)$ Force-Calculation Algorithm. *Nature*, 324, 1986.

[3] B. Baxter and G. Roussos. A new error estimate of the fast gauss transform. *SIAM Journal on Scientifi c Computing*, 24(1):257–259, 2002.

[4] P. Callahan and S. Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, 62(1):67–90, January 1995.

[5] A. Gray and A. W. Moore. N-Body Problems in Statistical Learning. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13 (December 2000)*. MIT Press, 2001.

[6] A. G. Gray. *Bringing Tractability to Generalized N-Body Problems in Statistical and Scientifi c Computation*. PhD thesis, Carnegie Mellon University, 2003.

[7] A. G. Gray and A. W. Moore. Rapid Evaluation of Multiple Density Models. In *Artifi cial Intelligence and Statistics 2003*, 2003.

[8] L. Greengard and V. Rokhlin. A Fast Algorithm for Particle Simulations. *Journal of Computational Physics*, 73, 1987.

[9] L. Greengard and J. Strain. The fast gauss transform. *SIAM Journal on Scientifi c and Statistical Computing*, 12(1):79–94, 1991.

[10] L. Greengard and X. Sun. A new version of the fast gauss transform. *Documenta Mathematica*, Extra Volume ICM(III):575–584, 1998.

[11] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[12] J. Strain. The fast gauss transform with variable scales. *SIAM Journal on Scientifi c and Statistical Computing*, 12:1131–1139, 1991.

[13] O. Szász. On the relative extrema of the hermite orthogonal functions. *J. Indian Math. Soc.*, 15:129–134, 1951.

[14] C. Yang, R. Duraiswami, N. A. Gumerov, and L. Davis. Improved fast gauss transform and efficient kernel density estimation. *International Conference on Computer Vision*, 2003.

parameters which must be tweaked simultaneously, an automatic scheme was created, based on the recommendations given in the paper and software documentation: For $D = 2$, use $p = 8$; for $D = 3$, use $p = 6$; set ρ_x = 2.5; start with $K = \sqrt{N}$ and double K until the error tolerance is met. When this failed to meet the tolerance, we resorted to additional trial and error by hand, again using knowledge which is unavailable in practice. The costs of parameter selection for these methods in both computer and human time is not included in the table.