The idea here is to make a binary tree with weights on each node so that the input data gets binned into each node as it would in a histogram, where the histogram has finer granularity at each level of the binary tree.

Let ϵ be the privacy parameter, $\boldsymbol{x}$ be the input data, m be a lower bound on points in x, and M be an upper bound on points in $\boldsymbol{x}$. Let n be the number of leaves in the tree that will be constructed. Let T be a perfect binary tree with n leaves. For each node v of T, associate a range such that the head node of T has range $[m, M)$, and for all nodes v with range $[m', M')$, v's left child has range $[m', m' + \frac{M'-m'}{2})$ and its right child has range $[m' + \frac{M'-m'}{2}, M')$. Let the weight w of each vertex be the number of points in $\boldsymbol{x}$ that are within v's range. The idea for the differentially private algorithm is to make a histogram at each level of the tree.

Algorithm 1 Differentially private weighted tree using histogram method to add noise

```
Input: $\boldsymbol{x}, \epsilon, T$.
Let $\epsilon' = \epsilon / \lg n$.
Initialize $T'$ as a tree identical to $T$ with the weights of each node $w'$ set to 0.
for every level of $T$ do
  For each node at that level, $w' = w + z$, where $z \sim \text{Lap}(2/\epsilon')$.
end for
return T'
```

Theorem 1. *Algorithm 1 is ϵ-differentially private.*

Proof. By definition of the Laplace mechanism, the histograms on each level of the tree are each ϵ'-differentially private. Then, by composition, if $\boldsymbol{\epsilon}$ is the net privacy budget used across the whole algorithm, if d is the depth of the tree, by composition

$$\begin{aligned}\boldsymbol{\epsilon} &= d\epsilon' \\ &= \lg n(\epsilon/\lg n) \\ &= \epsilon,\end{aligned}$$

since the tree's depth $d = \lg n$. □

Note that since you don't need to distribute each ϵ' across each level of the tree, this gives much better accuracy than doing naive noise addition to each node's weight individually. The resulting tree may then be used to estimate a cumulative distribution function for the data.

As discussed in more detail in [**?**], the fact that the count at a node is equal to the composition of the sums of its child nodes, or of its parent node minus the count of the adjacent node, etc., may be leveraged to calculate a more efficient count at each node of the tree.

You can recursively create an estimate of the count t_i at node i "from below", which we demarcate as t_i^-, by setting $t_i^- = t_i$ and $\sigma(t_i^-) = \sigma_i^-$ at the leaves, then recursively define for all other nodes

$$t_i^- = w_i^- t_i + (1 - w_i^-)(t_{2i}^- + t_{2i+1}^-),$$

where

$$w_i^- = \frac{(\sigma_i)^{-2}}{(\sigma_i)^{-2} + \left[(\sigma_{2i}^-)^2 + (\sigma_{2i+1}^-)^2\right]^{-1}},$$

and

$$\sigma_i^- = \sigma_i \sqrt{w_i^-},$$

which is equation 10 in [**?**].

Note that if each node has noise with variance s added to it, then $\forall i, \sigma_i = s$, $\sigma^-(t_{2i}^-) = \sigma^-(t_{2i+1}^-)$, and w will be the same for every node at depth d. Then, w_i may be recursively defined as

$$\begin{aligned} w_i^- &= \frac{s^{-2}}{s^{-2} + (1/2)(\sigma_{2i}^-)^{-2}} \\ &= \frac{s^{-2}}{s^{-2} + (1/2)\left(s\sqrt{w_{2i}^-}\right)^{-2}} \\ &= \frac{s^{-2}}{s^{-2}(1 + (1/2)(w_{2i}^-)^{-1})} \\ &= \frac{2w_{2i}^-}{2w_{2i}^- + 1}, \end{aligned}$$

and only needs to be calculated once per level of the tree.

For the base case,

$$w_i^- = 1,$$

since $t_i^- = t_i$.

Similarly, you can recursively create an estimate of the count t_i at node i "from above", which we demarcate as t_i^+. Set $t_1^+ = t_1$, and $\sigma_1^+ = \sigma_1$ if N is private and 0 if N is public. Denote the parent node of node i as $i\Phi 1$ and denote the adjacent[1] node to node i as $i\Lambda 1$.

[1] Here, adjacent means "the node which is the other child of node i's parent node"

Then, we can recursively define t_i^+ for all other nodes as

$$t_i^+ = w_i^+ t_i + (1 - w_i^+)(t_{i\Phi 1}^+ - t_{i\Lambda 1}^-),$$

where

$$w_i^+ = \frac{(\sigma_i)^{-2}}{(\sigma_i)^{-2} + \left[(\sigma_{i\Phi 1}^+)^2 + (\sigma_{i\Lambda 1}^-)^2\right]^{-1}},$$

and

$$\sigma_i^+ = \sigma_i \sqrt{w_i^+},$$

which is equation 11 in [**?**].

Note that if each node has noise with variance s, then

$$w_i^+ = \frac{1}{1 + (w_{i\Phi 1}^+ + w_{i\Lambda 1}^-)^{-1}},$$

and

$$w_1^+ = 1,$$

since $t_1^+ = t_i$.

An optimal estimate may then be derived using both the estimates from below and above:

$$t_i^* = w_i t_i^- + (1 - w_i)(t_{i\Phi 1}^+ - t_{i\Lambda 1}^-),$$

where

$$w_i = \frac{(\sigma_i^-)^{-2}}{(\sigma_i^-)^{-2} + \left[(\sigma_{i\Phi 1}^+)^2 + (\sigma_{i\Lambda 1}^-)^2\right]^{-1}},$$

and

$$\sigma_i^* = \sigma_i^- \sqrt{w_i},$$

which is equation 13 in [**?**].

Note that if each node has variance s, then the expression for the weights here is the same as the weight for estimation from above.

References

[1] Honaker, James, "Efficient Use of Differentially Private Binary Trees," http://hona.kr/papers/files/privatetrees.pdf.