INFTDA: A SIMPLE TOPDOWN MECHANISM FOR HIERARCHICAL DIFFERENTIALLY PRIVATE COUNTING QUERIES

Fabrizio Boninsegna University of Padova fabrizio.boninsegna@phd.unipd.it

ABSTRACT

This paper extends InfTDA, a mechanism proposed in (Boninsegna, Silvestri, PETS 2025) for mobility datasets with origin and destination trips, in a general setting. The algorithm presented in this paper works for any dataset of d categorical features and produces a differentially private synthetic dataset that answers all hierarchical queries, a special case of marginals, each with bounded maximum absolute error. The algorithm builds upon the TopDown mechanism developed for the 2020 US Census.

Introduction 1

Consider a dataset $D \in \mathcal{X}^n$, where the data universe \mathcal{X} is defined as the Cartesian product of d categorical features, $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$. For example, this setting can represent a healthcare dataset with attributes such as ("medical condition", "nationality", "age", "gender"), or a geographic dataset with attributes such as ("country", "state", "city"). One of the most relevant classes of queries in these datasets is marginals. Given a subset $S \subseteq [d]$ and specific values $y_i \in \mathcal{X}_i$ for each $i \in S$, a marginal query asks: "how many records in D have, for each attribute $i \in S$, the value equal to y_i ?" The aim is to release a differentially private (DP) synthetic dataset \tilde{D} that can accurately answer all such marginal queries. InfTDA, introduced in [1] for mobility data, constructs a synthetic dataset \tilde{D} that achieves a bounded maximum absolute error of $\tilde{O}(\sqrt{k^3 d})$ for any k-hierarchical query, where a k-hierarchical query is a marginal query with $S = [1, \dots, k]$. The algorithm employs a TopDown approach, initially developed by the US Census Bureau [2], which leverages differentially private and optimized k-hierarchical queries as constraints to refine the estimation of (k+1)-hierarchical queries.

Related Work The most straightforward method is to generate a *contingency table* of D, which is its histogram representation ¹, and add Gaussian noise to each count ². The resulting differentially private contingency table can answer all marginal queries; however, the variance grows linearly with the number of noisy counts required, leading to an absolute error of $\tilde{O}(e^{(d-k)/2})$ for any marginal with |S| = k. By focusing solely on k-hierarchical queries, an alternative approach would be to release a differentially private contingency table D_i for each query. In this case, each query could be answered with a maximum absolute error of $\tilde{O}(\sqrt{kd})^3$. However, this strategy requires handling d separate tables, which may be inconsistent with one another. Hay et al. [3] were the first to introduce the concept of hierarchical queries and to observe that accuracy can be improved by resolving such inconsistencies. The authors proposed a hierarchical tree structure, similar to the one defined for InfTDA, to organize the queries and developed the hierarchical mechanism, which enforces consistency through constrained optimization. A key limitation of this approach is that it produces a full DP contingency table with non-integral counts and without ensuring non-negativity. The authors identified the incorporation of non-negativity constraints as an open direction for future work. The most notable work with real-world application is undoubtedly the Disclosure Avoidance System TopDown algorithm, developed by the US Census Bureau [2]. This method follows a similar principle to the hierarchical mechanism but differs in

¹The histogram representation of D is $h_D \in \mathbb{N}_0^{|\mathcal{X}|}$ defined for each $x \in \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ as $h_D(x) = |\{y \in D : x = y\}|$. ²It can be used also Laplace noise, however here we focus in Gaussian noise as it provides better tail bounds.

 $^{{}^{3}}k$ comes from a union bound on Gaussian tails over $\prod_{i=1}^{k} |\mathcal{X}_{i}| = O(2^{k})$ elements.



Figure 1: An example of the non-negative hierarchical tree defined by hierarchical queries for a data universe $\mathcal{X} = \{y_1^1, y_1^2\} \times \{y_2^1, y_2^2\}$. The hierarchical consistency assures that the tree is hierarchical, so that child nodes' attributes sum to their father's attribute.

that it does not answer all k-hierarchical queries simultaneously. Instead, queries are answered sequentially, starting from the root of the hierarchy and proceeding down the tree. The mechanism employs a more sophisticated convex optimization procedure and produces a complete differentially private contingency table \tilde{D} with non-negative and integer counts. Nevertheless, no formal utility analysis of the algorithm has been provided. An alternative method for generating general purpose synthetic DP datasets is the Multiplicative Weights Exponential Mechanism (MWEM) [4], which is perhaps the most well-known in this category. MWEM achieves an error of $\tilde{O}(\sqrt{nd})$, thus its performance is constrained by sampling error.

Contributions InfTDA is a specific implementation of a hierarchical TopDown algorithm that employs constrained optimisation based on maximum error minimisation (Chebyshev distance), rather than mean squared error minimisation. The algorithm produces a differentially private contingency table \tilde{D} with non-negative integer counts and guarantees a maximum absolute error of $\tilde{O}(\sqrt{k^3 d})$ for each k-hierarchical query. Intuitively, the proof shows that the error introduced by Gaussian noise accumulates linearly across the levels of the hierarchical tree. The method is scalable, as it can be executed independently on each branch of the hierarchical tree, and each privatisation mechanism runs in at most $O(\max_i |\mathcal{X}_i|)$ time. Alongside InfTDA, Boninsegna and Silvestri [2] introduced IntOPT, an optimisation procedure that operates entirely in the integer domain and requires at most $O(\max_i |\mathcal{X}_i|^2)$ steps to solve the constrained problem.

2 Preliminaries

Non-Negative Hierarchical Tree A tree \mathcal{T} of depth d is a tuple $\mathcal{T} = (V, E)$: $V = \bigcup_{k=0}^{d} V_k$ is the set of nodes, with V_k denoting the set of nodes at level $k \in \{0, \ldots, d\}$; E is the set of edges between consecutive levels. A node at level k is indicated with $u_k \in V_k$. The set of children to a node is indicated by a function $\mathcal{C} : V_k \to 2^{V_{k+1}}$ for any level $k \in \{0, \ldots, d\}$. Each node of the tree has an attribute $q(u_k)$. InfTDA works for any tree data structure defined as follows

Definition 1 (Non-Negative Hierarchical Tree [1]). A tree $\mathcal{T} = (V, E)$ is said to be non-negative if it contains nonnegative attributes $q(u) \ge 0$ for any $u \in V$. The tree is hierarchical if the attribute of u is the sum of the attributes of its children $\mathcal{C}(u)$. Specifically, for every $u \in V$ which is not a leaf, we have $q(u) = \sum_{v \in \mathcal{C}(u)} q(v)$.

This paper focuses on integer attributes, so that $q(u) \in \mathbb{N}_0$ for any $u \in V$. Additionally, it will be used $\mathbf{q}_{\mathcal{C}}(u_k)$, which is the vector containing the attributes of the children of u_k .

Hierarchical Queries In this paper, we focus on data points containing d categorical features $\mathcal{X} = \mathcal{X}_1 \times \cdots \times \mathcal{X}_d$ where each \mathcal{X}_i is a set of $|\mathcal{X}_i| \ge 2$ categories. A dataset consists of a multiset of n points for \mathcal{X} , thus $D \in \mathcal{X}^n$.

Definition 2 (Counting Query). A counting query is defined by a function $q_i : \mathcal{X} \to \{0, 1\}$. Given a dataset D, we will abuse notation and write $q_i(D)$ to denote the sum of the values of the function q_i on D as $q_i(D) = \sum_{x \in D} q_i(x)$

A k-way marginal query counts the number of points in D that share the same values for a specified set of k attributes.

Definition 3 (k-way marginal query). A k-way marginal is defined by a subset $S \subseteq [d]$ of |S| = k, together with a particular value of each of the features $y \in \prod_{i \in S} \mathcal{X}_i$ with $y_i \in \mathcal{X}_i$ for any $i \in S$. Given such a pair (S, y), let $\mathcal{X}(S, y) = \{x \in \mathcal{X} : x_i = y_i \ \forall i \in S\}$. The corresponding counting query $q_{S,y}$ is defined as $q_{S,y}(x) = \mathbb{1}(x \in \mathcal{X}(S, y))$

There are 2^d possible k-way marginal queries, corresponding to the cardinality of the power set of [d]. This paper focuses on a specific subset of these marginal queries, referred to as k-hierarchical queries.

Definition 4 (*k*-hierarchical query). A *k*-hierarchical query is a *k*-way marginal query with S = [1, ..., k].

......

Algorithm I InfTDA		
Require: Tree $\mathcal{T} = (V, E)$ of depth d, privacy budget $\rho > 0$.	1: p	rocedure IntOpt (\mathbf{x}, c) .
1: $\tilde{V}_0 \leftarrow \{(u_0, n)\}$	2:	$\mathbf{z} \leftarrow \max\left(\left\lceil \frac{c-\sum_{i} x_{i}}{\dim(\mathbf{x})} \right\rceil, -\mathbf{x}\right)$
2: for $k \in (1,, d)$ do	3:	$t \leftarrow \ \mathbf{z}\ _{\infty}$
3: $V_k \leftarrow \{\}$	4:	$I \leftarrow$ sorted indices of x in ascending order
4: for $(u,c) \in V_{k-1}$ do	5:	$j \leftarrow 0$
5: $\tilde{\mathbf{q}} \leftarrow \mathbf{q}_{c}(u) + \mathcal{N}_{\mathbb{Z}}(0 \ d/o)^{ \mathcal{X}_{k} }$	6:	while $\sum_i z_i > c - \sum_i x_i$ do
$= - \mathbf{r} \cdot \mathbf{q} \cdot (\mathbf{a}) + \mathbf{r} \cdot \mathbf{z} \cdot (\mathbf{a}, \mathbf{a}, \mathbf{p})$	7:	$\Delta \leftarrow \sum_i z_i - c + \sum_i x_i$
6: $\mathbf{q} \leftarrow \texttt{IntUpt}(\mathbf{q}, c)$	8:	$z_{I[j]} \leftarrow \max(z_{I[j]} - \Delta, -x_{I[j]}, -t)$
7: $C \leftarrow C(u)$	9:	$i \leftarrow (i+1) \mod I $
8: $X \leftarrow \{(C_j, \bar{q}_j) : \bar{q}_j > 0\}$	10:	j = 0 then
9: $V_k \leftarrow V_k \cup X$	11:	$t \leftarrow t + 1$
10: return $D = V_d$	12:	return $\mathbf{x} + \mathbf{z}$

The hierarchy follows from a simple observation, a single k-hierarchical query can be obtained by summing all the k + 1 hierarchical queries with the same features.

Observation 1 (Hierarchical Consistency). For any $k \in [d-1]$ and for any $y \in \prod_{i=1}^{k} X_i$ we have that

$$q_{S_k,y}(D) = \sum_{y_{k+1} \in \mathcal{X}_{k+1}} q_{S_{k+1},(y,y_{k+1})}(D).$$

For convenience, we define $q_{S_0}(D) = n$ so that $q_{S_0}(D) = \sum_{y_1 \in \mathcal{X}_1} q_{S_1,y_1}(D)$.

Thus, all the k-hierarchical queries can be arranged into a non-negative hierarchical tree, as shown in Figure 1. The tree contains at each level k all the k-hierarchical queries, thus for any $y \in \prod_{i=1}^{k} \mathcal{X}_i$ there is a unique $u_k \in V_k$ s.t. $q(u_k) = q_{S_k,y}(D)$. The error considered in this work is the maximum absolute error for each k-hierarchical query, hence $\max_{y \in \prod_{i=1}^{k} \mathcal{X}_i} |q_{S_{k,y}}(D) - q_{S_{k,y}}(\tilde{D})|$ for any $k \in \{0, \ldots, d\}$.

Differential Privacy The neighbouring relation considered in this work is by *substitution* of a single data point, which is more precisely called *bounded* differential privacy [5]. With this definition, the size of the dataset does not change among neighbouring datasets, making the query $q_{S_0}(D) = n$ non-sensitive. The mechanism considered in this work is the *multivariate discrete Gaussian mechanism* [6], which adds integer noise sampled from a discrete Gaussian distribution satisfying zero-concentrated differential privacy (zCDP) [7].

3 TopDown Mechanism with Chebyshev Distance

This section introduces InfTDA. The algorithm constructs a privatized non-negative hierarchical tree with integer attributes, where the leaves correspond to the counts of a private full contingency table \tilde{D} . The mechanism employs a TopDown approach: at each level k, discrete Gaussian noise is added to the attributes, and they are optimized to satisfy the consistency relationship with level k - 1, and non-negativity, by minimizing the Chebyshev distance (i.e., the maximum error) with the private estimates.

InfTDA The detailed pseudocode of InfTDA is provided in Algorithm 1. The process begins by constructing the root \tilde{V}_0 of the differentially private tree in line 1. Here, u_0 denotes the root node of the input tree, while *n* represents its attribute. The algorithm then starts the TopDown loop in line 2. Each iteration aims to construct the set of nodes at level k of the differentially private tree, instantiated in line 3. Each node of the previous level k - 1 is sampled in line 4 and used as a constraint. In line 5 the discrete Gaussian mechanism with ρ/d privacy budget (for zCDP) is applied to the attributes $q_{\mathcal{C}}(u)$ of the child nodes of the constraint. Then, in line 6 the private attributes \tilde{q} are post-processed to satisfy the constraints. The algorithm IntOpt solves the following integer optimization problem by minimizing the Chebyshev distance

$$\mathcal{P}(\mathbf{x},c) := \arg\min_{\mathbf{y}} ||\mathbf{x} - \mathbf{y}||_{\infty} \quad \text{s.t.} \quad \sum_{i=1}^{\dim(\mathbf{x})} y_i = c. \quad \text{and} \quad y_i \in \mathbb{N}_0 \quad \forall i \in [1,\ldots,\dim(\mathbf{x})]$$
(1)

In line 7, the set of child nodes of the constraint is constructed, and it is augmented with the corresponding postprocessed DP attributes in line 8, dropping nodes with zero attributes. This final step effectively reduces the size of

the DP tree and the running time of the algorithm, particularly for sparse datasets. In fact, if there is a node u_k with optimized attribute $\bar{q}(u_k) = 0$, then, by consistency, the entire branch of the tree starting at u_k will also have nodes with zero attributes. Finally, in line 9 the set of DP nodes is updated, and this set will serve as constraints in the next iteration. The algorithm outputs the leaves of the DP tree, which constitute a full contingency table \tilde{D} .

Theorem 1 (Utility and Privacy of InfTDA). Let $D = (\mathcal{X}_1 \times \mathcal{X}_d)^n$ be a dataset containing n points of d categorical values. Let \tilde{D} be the dataset returned by InfTDA. Then, $|\tilde{D}| = n$ and for any $k \in [1, ..., d]$ and $\beta \in (0, 1)$ we have that

$$\Pr\left[\max_{y\in\prod_{i=1}^{k}\mathcal{X}_{i}}\left|q_{S_{k},y}(D)-q_{S_{k},y}(\tilde{D})\right|\leq\sum_{\ell=1}^{k}\sqrt{\frac{8d}{\rho}\log\left(\frac{k\cdot\prod_{i=1}^{\ell}|\mathcal{X}_{i}|}{\beta}\right)}\right]\geq1-\beta.$$
(2)

Additionally, InfTDA satisfies ρ -zCDP.

Proof. We refer the reader to Appendix A for the utility proof. Privacy is ensured through composition, parallel composition, and post-processing. A privacy budget of ρ/d is allocated to each level of the tree, as the ℓ_2 -global sensitivity of each k-hierarchical query is $\sqrt{2}$, and each level represents a partition of the data. By applying the composition rule of zCDP across d levels, the privacy guarantee is established.

IntOpt The main difference between InfTDA and the TopDown algorithm developed by the US Census [2] lies in the optimization procedure. In InfTDA, Chebyshev minimization was chosen for two key reasons: to enable a utility analysis of the mechanism and to design a straightforward optimizer that operates directly in the integer domain. In contrast, in [2] the mechanism solves a Euclidean minimization in two steps. First, the problem is solved in the real domain using convex optimization techniques; then it searches for the best rounding using an integer linear program. IntOPT solves Chebyshev minimization in one shot entirely in the integer domain. Algorithm 1 shows a simple version that runs in O(c) steps. A version that runs in $O(\min(|\dim(\mathbf{x})|^2, c))$ is presented in Appendix C in [1].

The main approach to solve $\mathcal{P}(\mathbf{x}, c)$ is to first compute a lower bound for the Chebyshev distance. Based on this bound, a non-negative solution is then constructed with ℓ_{∞} norm no greater than the lower bound and with elements summing to a value exceeding c. Subsequently, IntOPT reduces the elements to ensure that the summation constraint is satisfied, while increasing the Chebyshev distance as little as possible. More precisely, $\mathcal{P}(\mathbf{x}, c)$ is equivalent to the following problem in the real domain for $\mathbf{z} = \mathbf{y} - \mathbf{x}$

$$\min \alpha \ge 0 \quad \text{s.t. } z_i \in [-\alpha, \alpha] \quad \text{and} \quad z_i \ge -x_i \quad \text{and} \quad \sum_j z_j = c - \sum_j x_j \quad \forall i \in [1, \dots, \dim(\mathbf{x})].$$

This problem has a lower bound $\alpha \ge \max\left(\left|\frac{c-\sum_{i=1} x_i}{\dim(\mathbf{x})}\right|; -\min_i x_i\right)$ in the real domain, which translates to a lower bound in the integer domain by taking the ceiling function. In line 2 of IntOPT, \mathbf{z} is initialized such that $\|\mathbf{z}\|_{\infty} \le \max\left(\left[\left|\frac{c-\sum_{i=1} x_i}{\dim(\mathbf{x})}\right|\right]; -\min_i x_i\right)$. Furthermore, since $\sum_i z_i \ge c - \sum_i x_i^4$, it is sufficient to reduce the elements of \mathbf{z} until this constraint is satisfied. The reduction is applied uniformly across the elements to minimise the increase in Chebyshev distance. This procedure can be implemented through different strategies, resulting in multiple possible solutions. Indeed, Chebyshev optimisation does not admit a unique solution. The method adopted in IntOPT is based on two main choices: (a) the reduction is applied iteratively following the indices of \mathbf{x} that sort it in ascending order (line 4), and (b) during the first iteration, each element is reduced by the maximum possible amount that does not alter $\|\mathbf{z}\|_{\infty}$ (line 3). The underlying idea is to prioritise setting to zero the private counts that are negative or small, as these are more likely to correspond to sensitive counts that were originally zero. Such choices have been demonstrated to effectively reduce the presence of false positives in the private contingency table [1], when compared to alternative optimisation strategies.

4 Conclusions and Open Questions

This paper has presented a generalisation of InfTDA for a generic collection of data points characterised by d categorical features. The proposed mechanism offers theoretical guarantees for any k-hierarchical query, ensuring a maximum absolute error of $O(\sqrt{k^3d})$. Furthermore, the mechanism has been shown to perform effectively in practice on both real-world and synthetic datasets [1]. However, it remains an open question whether it is possible to achieve an error of $\tilde{O}(\sqrt{kd})$, as attained by the hierarchical mechanism, while still producing a contingency table \tilde{D} with non-negative integer counts.

$${}^{4}\sum_{i=1} \max\left(\left\lceil \frac{c-\sum_{i=1} x_{i}}{\dim(\mathbf{x})} \right\rceil, -x_{i} \right) \ge \sum_{i=1} \left\lceil \frac{c-\sum_{i=1} x_{i}}{\dim(\mathbf{x})} \right\rceil \ge c - \sum_{i=1} x_{i}$$

References

- [1] Fabrizio Boninsegna and Francesco Silvestri. Differentially private release of hierarchical origin/destination data with a topdown approach, 2025. Accepted for publication in PETS 2025.
- [2] John M. Abowd, Robert Ashmead, Ryan Cumings-Menon, Simson Garfinkel, Micah Heineck, Christine Heiss, Robert Johns, Daniel Kifer, Philip Leclerc, Ashwin Machanavajjhala, Brett Moran, William Sexton, Matthew Spence, and Pavel Zhuravlev. The 2020 Census Disclosure Avoidance System TopDown Algorithm. *Harvard Data Science Review*, (Special Issue 2), jun 24 2022. https://hdsr.mitpress.mit.edu/pub/7evz361i.
- [3] Michael Hay, Vibhor Rastogi, Gerome Miklau, and Dan Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3(1–2):1021–1032, September 2010.
- [4] Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [5] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 193–204, 2011.
- [6] Clement Canonne, Gautam Kamath, and Thomas Steinke. The discrete gaussian for differential privacy. *Journal of Privacy and Confidentiality*, 12(1), Jul. 2022.
- [7] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of cryptography conference*, pages 635–658. Springer, 2016.

A Utility of InfTDA

We start with the utility of the discrete Gaussian mechanism.

Corollary 1 (Corollary 9 [6]). Let
$$Z \sim \mathcal{N}_{\mathbb{Z}}(0, \sigma^2)$$
. Then $\operatorname{Var}[Z] \leq \sigma^2$ and $\Pr[Z \geq t] \leq e^{-\frac{t^2}{2\sigma^2}}$ for any $t \geq 0$.

Now we are ready to prove the full version of the utility Theorem for InfTDA.

Proof of utility in Theorem 1. Let us focus on one k-hierarchical query, $q_{S_k,y}(D)$ for some $y \in \prod_{i=1}^k \mathcal{X}_i$ and $k \in [1, \ldots, d]$. The non-negative hierarchical tree represents this query with an attribute of a node u_k at level k. Let this attribute be $q(u_k)$, while $\tilde{q}(u_k)$ be the attribute returned by the Gaussian mechanism before the optimization is applied, and $\bar{q}(u_k)$ after the optimization. Since $\bar{q}(u_k)$ is the differentially private query returned by InfTDA, we have by triangle inequality that

$$\operatorname{err}(u_k)| = |q(u_k) - \bar{q}(u_k)| \le |q(u_k) - \tilde{q}(u_k)| + |\tilde{q}(u_k) - \bar{q}(u_k)|$$

The first term is just the absolute value of a Gaussian random variable, so we focus on the second term. Let u_{k-1} be the father node of u_k , then $\bar{q}(u_k)$ is an element of the vector $\mathbf{\bar{q}}_{\mathcal{C}}(u_{k-1}) \in \mathbb{N}_0^{|\mathcal{X}_k|}$ solution to the optimization problem $\mathcal{P}(\mathbf{\tilde{q}}_{\mathcal{C}}(u_{k-1}), c)$, where $c = \bar{q}(u_{k-1})$, thus

$$|\tilde{q}(u_k) - \bar{q}(u_k)| \le \|\mathbf{\tilde{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{\bar{q}}_{\mathcal{C}}(u_{k-1})\|_{\infty}.$$

Let us consider another vector $\boldsymbol{\xi} \in \mathbb{Z}^{|\mathcal{X}_k|}$, called the *offset*, such that $\mathbf{q}_{\mathcal{C}}(u_{k-1}) + \boldsymbol{\xi}$ lies within the feasible region of the constrained optimization problem \mathcal{P} , then

$$q_{\mathcal{C},j}(u_{k-1}) + \xi_j \ge 0 \tag{3}$$

$$\sum_{j=1}^{|\mathcal{X}_k|} q_{\mathcal{C},j}(u_{k-1}) + \xi_j = \bar{q}(u_{k-1}).$$
(4)

As the vector $\bar{\mathbf{q}}_{\mathcal{C}}(u_{k-1})$ is a solution to the optimization problem, it minimizes the Chebyshev distance with $\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1})$ under the non-negativity and summation constraints, then by triangle inequality

$$\begin{aligned} \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1}) - \bar{\mathbf{q}}_{\mathcal{C}}(u_{k-1})\|_{\infty} &\leq \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{q}_{\mathcal{C}}(u_{k-1}) - \boldsymbol{\xi}\|_{\infty} &\{\bar{\mathbf{q}}_{\mathcal{C}}(u_{k-1})\} \text{ is a optimal solution}\}\\ &\leq \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{q}_{\mathcal{C}}(u_{k-1})\|_{\infty} + \|\boldsymbol{\xi}\|_{\infty} &\{\text{due to triangle inequality}\}. \end{aligned}$$

As $|q(u_k) - \tilde{q}(u_k)| \le \|\mathbf{\tilde{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{q}_{\mathcal{C}}(u_{k-1})\|_{\infty}$, the upper bound becomes

1.22

$$|\operatorname{err}(u_k)| \le 2 \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{q}_{\mathcal{C}}(u_{k-1})\|_{\infty} + \|\boldsymbol{\xi}\|_{\infty}.$$
(5)

The problem is now to find an upper bound for $\|\boldsymbol{\xi}\|_{\infty}$.

To upper bound $\|\boldsymbol{\xi}\|_{\infty}$, we now construct an example of $\boldsymbol{\xi}$ satisfying the constraints and having a bounded ℓ_{∞} norm. By construction, from Equation 4 and hierarchical consistency, we have that

$$\sum_{j=1}^{|\mathcal{X}_k|} \xi_j = \bar{q}(u_{k-1}) - \sum_{j=1}^{|\mathcal{X}_k|} q_{\mathcal{C},j}(u_{k-1}) = \bar{q}(u_{k-1}) - q(u_{k-1}) = \operatorname{err}(u_{k-1})$$

If $\operatorname{err}(u_{k-1}) \ge 0$ we can take $\xi_j = \frac{\operatorname{err}(u_{k-1})}{|\mathcal{X}_k|}$ for any $j \in [|\mathcal{X}_k|]$ as a solution satisfying the summation constraint and the inequality constraint in Equation 3. However, this is not sufficient. If $\operatorname{err}(u_{k-1}) < 0$ the inequality constraint might not be satisfied. In this scenario we might consider a solution where $\xi_i = 0$ for any $i \in [|\mathcal{X}_k|] \setminus \{i^*\}$ where $\xi_{i^*} = -|\operatorname{err}(u_{k-1})|$. Any zero element satisfies the constraint in Equation 3 as $q_{\mathcal{C},j}(u_{k-1}) \ge 0$. If $\xi_{i^*} \ge -q_{\mathcal{C},i^*}(u_{k-1})$ then we finish and obtain an upper bound $\|\boldsymbol{\xi}\|_{\infty} \le |\operatorname{err}(u_{k-1})|$. In the other case where $\xi_{i^*} < -q_{\mathcal{C},i^*}(u_{k-1})$ we need to augment ξ_{i^*} up to meet $-q_{\mathcal{C},i^*}(u_{k-1})$. By doing so we increase $\sum_i \xi_i$ making necessary to decrease some elements of the offset. As we are reducing elements that initially are zero, the new offset still contains only negative elements, and as $\sum_i \xi_i = -|\operatorname{err}(u_{k-1})|$ any element cannot be less than $-|\operatorname{err}(u_{k-1})|$. Thus, we conclude that there always exists an offset such that $\|\boldsymbol{\xi}\|_{\infty} \le |\operatorname{err}(u_{k-1})|$.

Now we continue from the upper bound in Equation 5

$$|\operatorname{err}(u_k)| \le 2 \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{q}_{\mathcal{C}}(u_{k-1})\|_{\infty} + |\operatorname{err}(u_{k-1})|.$$
(6)

By taking the maximum we obtain

$$\max_{u_k \in V_k} |\operatorname{err}(u_k)| \le 2 \cdot \max_{u_{k-1} \in V_{k-1}} \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-1}) - \mathbf{q}_{\mathcal{C}}(u_{k-1})\|_{\infty} + \max_{u_{k-1} \in V_{k-1}} |\operatorname{err}(u_{k-1})|.$$

Completing the recurrence relation by ending at $err(u_0) = 0$ we get

$$\max_{u_k \in V_k} |\operatorname{err}(u_k)| \le 2 \cdot \sum_{\ell=1}^k \max_{u_{k-\ell} \in V_{k-\ell}} \| \tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-\ell}) - \mathbf{q}_{\mathcal{C}}(u_{k-\ell}) \|_{\infty}.$$
(7)

Note that in the previous equation u_{k-1} is the father node of u_k , as previously defined, whereas u_{k-2} is the father node of u_{k-1} and so on. We may upper bound each term in Equation 7 by using Corollary 1 with a union bound over the dimension of each vector, which is $|\mathcal{X}_{k-\ell+1}|$, times the number of nodes at level $V_{k-\ell}$, which are $\prod_{i=1}^{k-\ell} |\mathcal{X}_i|$. Thus, we obtain

$$\Pr\left[\max_{u_{k-\ell}\in V_{k-\ell}} \|\tilde{\mathbf{q}}_{\mathcal{C}}(u_{k-\ell}) - \mathbf{q}_{\mathcal{C}}(u_{k-\ell})\|_{\infty} \le \sqrt{\frac{2d}{\rho}}\log\left(\frac{k \cdot \prod_{i=1}^{k-\ell+1} |\mathcal{X}_i|}{\beta}\right)\right] \ge 1 - \frac{\beta}{k}.$$

By re-indexing Equation 7, summing the previous upper bounds, and applying an union bound, we finally obtain our claim

$$\Pr\left[\max_{u_k \in V_k} |\operatorname{err}(u_k)| \le \sum_{\ell=1}^k \sqrt{\frac{8d}{\rho} \log\left(\frac{k \cdot \prod_{i=1}^\ell |\mathcal{X}_i|}{\beta}\right)}\right] \ge 1 - \beta.$$