KOÇ UNIVERSITY

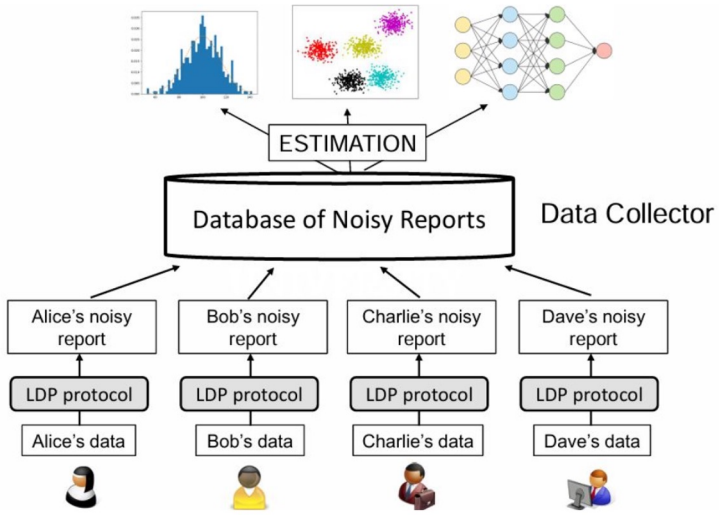
Fast Simulation Algorithms for OLH using Binomial Modeling

Berkay Kemal Balioglu, Alireza Khodaie, M. Emre Gürsoy

Department of Computer Engineering
Koç University, Istanbul, Türkiye

Local Differential Privacy

Modern analytics relies heavily on data collection, much of which involves **privacy-sensitive** information such as health, financial, and location data.



In **Local Differential Privacy (LDP)**:

- Each user perturbs their value locally
- The server applies estimation to recover aggregate statistics such as frequencies
- Lower ε yields **stronger privacy**, higher ε yields **higher utility**

Definition 1 (ε-**LDP**). *A randomized mechanism ψ satisfies ε-local differential privacy (ε-LDP) if and only if, for any two input values $v_1, v_2 \in \mathcal{D}$:*

$$\forall\, y \in \mathrm{Range}(\psi): \quad \frac{\Pr[\psi(v_1) = y]}{\Pr[\psi(v_2) = y]} \leq e^{\varepsilon}, \tag{1}$$

Optimized Local Hashing

Each user randomly selects a hash function from a hash family and uses it to encode their true value to an integer in [0, g-1], where $g = e^{\varepsilon} + 1$

User-Side Perturbation:

$$\Pr[x'_u = i] = \begin{cases} p = \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}, & \text{if } i = x_u \\ q = \frac{1}{e^{\varepsilon}+g-1}, & \text{otherwise} \end{cases}$$

Server-side support and frequency estimation:

$$\mathrm{Sup}(v) = \left|\{u \in \mathcal{P} : x'_u = H_u(v)\}\right| \qquad \hat{f}_v = \frac{(e^{\varepsilon}+g-1)\left(g \cdot \mathrm{Sup}(v) - n\right)}{(e^{\varepsilon}-1)(g-1)\,n}$$

Encode
map v_u to $x_u = H_u(v_u)$ in [0,...,g-1]

→

Perturb
Keep x_u with probability p; otherwise, sample uniformly from [0,...,g-1]\{x_u}

→

Report
Send $<H_u, x'_u>$ to the server

Simulating OLH

- In research, we run simulations of OLH on **one machine**: every user, then the server
- User phase: $O(n)$
- Server phase: recomputes $H_u(v)$ for every report-value pair
- Bottleneck: Line 17 executes $O(nd)$ hash calls, which dominate runtime

Algorithm 1 Existing OLH Simulation

Input: Domain $\mathcal{D}$, budget ε, values of users in $\mathcal{P}$ $(v_{u_1}, v_{u_2}, \ldots, v_{u_n})$
Output: Estimated frequencies $\hat{f}_v$ for all $v \in \mathcal{D}$

1: Compute $g \leftarrow \lfloor e^{\varepsilon} + 1 \rfloor$, $p \leftarrow \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}$
2: Initialize empty list of reports L
3: ▷ **User-side encoding and perturbation**
4: **for** each user $u_i \in \mathcal{P}$ **do**
5: Sample a random seed s_{u_i} and initialize H_{u_i} accordingly
6: $x_{u_i} \leftarrow H_{u_i}(v_{u_i}) \bmod g$
7: $r \leftarrow$ sample a random float in $[0, 1]$
8: **if** $r \leq p$ **then**
9: $x'_{u_i} \leftarrow x_{u_i}$
10: **else**
11: $x'_{u_i} \leftarrow$ sample uniformly from $\{0, \ldots, g-1\} \setminus \{x_{u_i}\}$
12: Store $\langle x'_{u_i}, H_{u_i} \rangle$ in L
13: ▷ **Server-side estimation**
14: **for** each value $v \in \mathcal{D}$ **do**
15: Initialize $\mathrm{Sup}(v) \leftarrow 0$
16: **for** each $\langle x'_{u_i}, H_{u_i} \rangle$ in L **do**
17: Increment $\mathrm{Sup}(v)$ by $+1$ iff $x'_{u_i} = H_{u_i}(v) \bmod g$
18: Compute $\hat{f}_v \leftarrow \frac{(e^{\varepsilon}+g-1)(g \cdot \mathrm{Sup}(v)-n)}{(e^{\varepsilon}-1)(g-1)\,n}$
19: **return** $\hat{f}_v$ for all $v \in \mathcal{D}$

$O(nd)$ complexity: with n in the millions and d in the hundreds, a single simulation takes minutes.

Contributions

- We propose two simulation algorithms (**2-Binom** and **3-Binom**) for OLH to capture the collective behavior of n users using Binomial sampling, avoiding per-user simulation

- We formally prove that 2-Binom and 3-Binom yield **unbiased** frequency **estimates** with variances equal to original OLH

- We empirically validate 2-Binom and 3-Binom on multiple datasets
  - Orders-of-magnitude reductions in runtime and memory
  - Estimates are statistically indistinguishable from OLH

The Binomial Insight

Each user contributes 0 or 1 to Sup(v).
Partition the population by their true value:

Group 1: User's value $v_u = v$ (n_v such users) $X \sim \text{Binomial}(n_v, p)$

Contributes iff the perturbation keeps the hashed value, w.p. p.

Group 2: User's value $v_u \neq v$ ($n - n_v$ users)

A. hash collision, value kept: p/g

B. no collision, flips onto $H_u(v)$: $(g\text{-}1)q/g$

$$Y \sim \text{Binomial}\left(n - n_v, \frac{1}{g}\right)$$

Total Sup(v) is: $\text{Sup}(v) = X + Y$

2-Binom Algorithm

- No reports are stored and no hash is ever computed
- Total complexity $O(n+d)$, down from $O(nd)$
- When the same dataset is reused (repetitions, or a sweep over ε) the count phase runs once, so each further run is $O(d)$

Algorithm 2 Proposed 2-Binom Simulation Algorithm

Input: Domain $\mathcal{D}$, budget ε, values of users in $\mathcal{P}$ $(v_{u_1}, v_{u_2}, \ldots, v_{u_n})$
Output: Estimated frequencies $\hat{f}_v$ for all $v \in \mathcal{D}$

Compute $g \leftarrow \lfloor e^{\varepsilon} + 1 \rfloor, \quad p \leftarrow \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}$
▷ **Find n_v and n from user values**
for each user $u_i \in \mathcal{P}$ **do**
    $n_{v_{u_i}} \leftarrow n_{v_{u_i}} + 1, \quad n \leftarrow n + 1$
▷ **Binomial approach**
for each value $v \in \mathcal{D}$ **do**
    $x \leftarrow$ draw a random sample from Binomial(n_v, p)
    $y \leftarrow$ draw a random sample from Binomial$\left(n - n_v, \frac{1}{g}\right)$
    $\mathrm{Sup}(v) \leftarrow x + y$
    $\hat{f}_v \leftarrow \frac{\mathrm{Sup}(v) - (n/g)}{n \cdot (p - (1/g))}$
return $\hat{f}_v$ for all $v \in \mathcal{D}$

Theorem 1 (Unbiasedness). *The frequency estimations $\hat{f}_v$ produced by Algorithm 2 are unbiased, i.e., $\mathbb{E}[\hat{f}_v] = f_v$ for all $v \in \mathcal{D}$.*

Theorem 2 (Variance). *The variance of the frequency estimations $\hat{f}_v$ produced by Algorithm 2 is:*

$$\mathrm{Var}[\hat{f}_v] = \frac{\frac{1}{g}\left(1 - \frac{1}{g}\right)}{n\left(p - \frac{1}{g}\right)^2} + \frac{f_v\left(1 - p - \frac{1}{g}\right)}{n\left(p - \frac{1}{g}\right)} \tag{9}$$

3-Binom Algorithm

- Instead of merging the two sub-cases of Group 2, draw the number of colliding users n_c first and condition on it.
- Three Binomial draws:
  - X: true (v), survives perturbation
  - Y_1: colliding users, kept by perturbation
  - Y_2: non-colliding users, flip to $H_u(v)$ by chance
- Result: three draws per value; complexity remains $O(n+d)$

Algorithm 3 Proposed 3-Binom Simulation Algorithm

Input: Domain $\mathcal{D}$, budget ε, values of users in $\mathcal{P}$ $(v_{u_1}, v_{u_2}, \ldots, v_{u_n})$
Output: Estimated frequencies $\hat{f}_v$ for all $v \in \mathcal{D}$

Compute $g \leftarrow \lfloor e^{\varepsilon} + 1 \rfloor$, $p \leftarrow \frac{e^{\varepsilon}}{e^{\varepsilon}+g-1}$, $q \leftarrow \frac{1}{e^{\varepsilon}+g-1}$
▷ **Find n_v and n from user values**
for each user $u_i \in \mathcal{P}$ **do**
    $n_{v_{u_i}} \leftarrow n_{v_{u_i}} + 1$, $n \leftarrow n + 1$
▷ **3-Binom approach**
for each value $v \in \mathcal{D}$ **do**
    $x \leftarrow$ draw a random sample from Binomial(n_v, p)
    $n_c \leftarrow$ draw a random sample from Binomial$\left(n - n_v, \frac{1}{g}\right)$
    $y_1 \leftarrow$ draw a random sample from Binomial(n_c, p)
    $y_2 \leftarrow$ draw a random sample from Binomial$(n - n_v - n_c, q)$
    $\mathrm{Sup}(v) \leftarrow x + y_1 + y_2$
    $\hat{f}_v \leftarrow \frac{\mathrm{Sup}(v) - (n/g)}{n \cdot (p - (1/g))}$
return $\hat{f}_v$ for all $v \in \mathcal{D}$

Theorems 3-4 in our paper show 3-Binom produces unbiased estimates with variance **identical to 2-Binom**: The two decompositions are statistically equivalent.

Experiment Setup

Datasets:

- **Adult:** 45K records, 14 attributes; Age as private value (d=74)
- **MSNBC:** User browsing activity logs (d=17)
- **Kosarak:** Click-stream data; top 128 URLs (d=128)
- **BMS-POS:** 516K transactions, 1.7K items; top 256 items (d=256)

Experimental Settings:

- $\varepsilon \in \{0.5, 1.0, \ldots, 4.0\}$
- Runtime: 10 repetitions
- Error & variance: 5,000 repetitions

Metrics:

- Execution time & peak memory
- ℓ_1 and ℓ_2 estimation error
- Per-value empirical variance

Execution Time Results

Table 1. Total execution times (in seconds) for 10 repetitions across different ε values.

Dataset	Method	ε=0.5	ε=1.0	ε=1.5	ε=2.0	ε=2.5	ε=3.0	ε=3.5	ε=4.0
Adult	OLH	8.6496	8.3656	8.1026	7.8320	7.7127	7.6063	7.4373	7.3598
	2-Binom	0.0008	0.0009	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
	3-Binom	0.0012	0.0014	0.0013	0.0013	0.0013	0.0013	0.0012	0.0013
MSNBC	OLH	63.037	60.954	59.021	57.635	56.345	55.832	54.491	54.867
	2-Binom	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
	3-Binom	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
Kosarak	OLH	159.75	151.31	147.21	141.59	138.33	135.33	133.87	133.36
	2-Binom	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
	3-Binom	0.0018	0.0019	0.0019	0.0019	0.0019	0.0019	0.0018	0.0019
BMS-POS	OLH	304.37	289.84	284.95	274.70	265.88	262.16	257.32	252.03
	2-Binom	0.0025	0.0025	0.0029	0.0025	0.0025	0.0027	0.0025	0.0025
	3-Binom	0.0036	0.0036	0.0041	0.0036	0.0036	0.0036	0.0036	0.0036

- Binomial methods: Have run time <5 ms across all configurations
- OLH: Scales with both n and d, reaching several minutes
- 3-Binom: Slight overhead from the additional draw per v, but negligible in practice

Effect of Multi-Threading

Parallel OLH with η threads against single-threaded Binomial methods, $\varepsilon = 1$

	OLH (parallel, η threads)					Binomial	
Dataset	$\eta = 1$	$\eta = 2$	$\eta = 4$	$\eta = 8$	$\eta = 16$	**2-Binom**	**3-Binom**
Adult	8.5531	4.3803	2.6380	1.7852	1.4543	0.0005	0.0010
MSNBC	62.739	34.907	20.468	14.625	12.581	0.0002	0.0003
Kosarak	155.14	78.902	48.702	32.588	25.403	0.0009	0.0017
BMS-POS	304.15	154.50	91.823	65.495	52.596	0.0018	0.0034

- Multithreading accelerates OLH, but speedup is sublinear as threads increase
- Even with 16 threads, OLH remains orders of magnitude slower than single-threaded Binomial
- The bottleneck remains: parallelization reduces runtime but does not remove the $O(nd)$ computation

Memory Consumption

Table 3. Memory consumptions (in MB) of simulations across different ε values.

Dataset	Method	ε=0.5	ε=1.0	ε=1.5	ε=2.0	ε=2.5	ε=3.0	ε=3.5	ε=4.0
Adult	OLH	3.8444	3.8434	3.8434	3.8433	3.8446	3.8435	3.8433	3.8435
	2-Binom	0.0157	0.0151	0.0152	0.0158	0.0152	0.0158	0.0150	0.0151
	3-Binom	0.0152	0.0157	0.0158	0.0157	0.0157	0.0151	0.0153	0.0157
MSNBC	OLH	83.591	83.591	83.593	83.591	83.591	83.593	83.590	83.591
	2-Binom	0.0150	0.0150	0.0150	0.0152	0.0151	0.0151	0.0153	0.0153
	3-Binom	0.0151	0.0150	0.0150	0.0152	0.0152	0.0151	0.0151	0.0153
Kosarak	OLH	42.138	42.140	42.138	42.142	42.138	42.140	42.140	42.138
	2-Binom	0.0161	0.0161	0.0157	0.0157	0.0156	0.0161	0.0161	0.0154
	3-Binom	0.0157	0.0160	0.0160	0.0161	0.0161	0.0162	0.0161	0.0161
BMS-POS	OLH	42.143	42.139	42.143	42.139	42.139	42.139	42.143	42.139
	2-Binom	0.0165	0.0170	0.0171	0.0166	0.0164	0.0170	0.0170	0.0172
	3-Binom	0.0166	0.0172	0.0169	0.0171	0.0170	0.0171	0.0165	0.0166

- 2-Binom and 3-Binom cut memory usage by orders of magnitude (from megabytes down to tens of kilobytes)
- Population-independent: Binomial methods scale as $O(d)$ vs OLH's $O(n+d)$
- 2-Binom ≈ 3-Binom: Both have a negligible memory footprint

ℓ_1 and ℓ_2 Errors

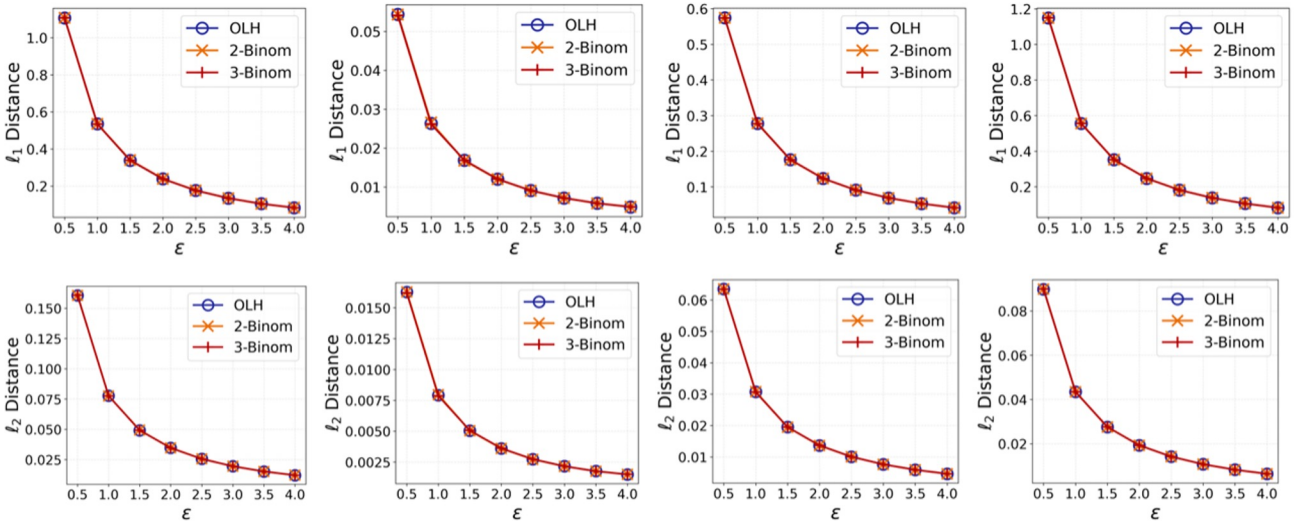


Fig. 1. Average ℓ_1 and ℓ_2 estimation errors of OLH, 2-Binom, and 3-Binom across varying ε values. Datasets from left to right: Adult, MSNBC, Kosarak, BMS-POS.

- Error curves of OLH, 2-Binom and 3-Binom closely overlap across all $\boldsymbol{\varepsilon}$ values
- All three decrease monotonically as $\boldsymbol{\varepsilon}$ grows
- Residual differences are within the sampling noise of 5,000 repetitions

Error Distributions (KDE)

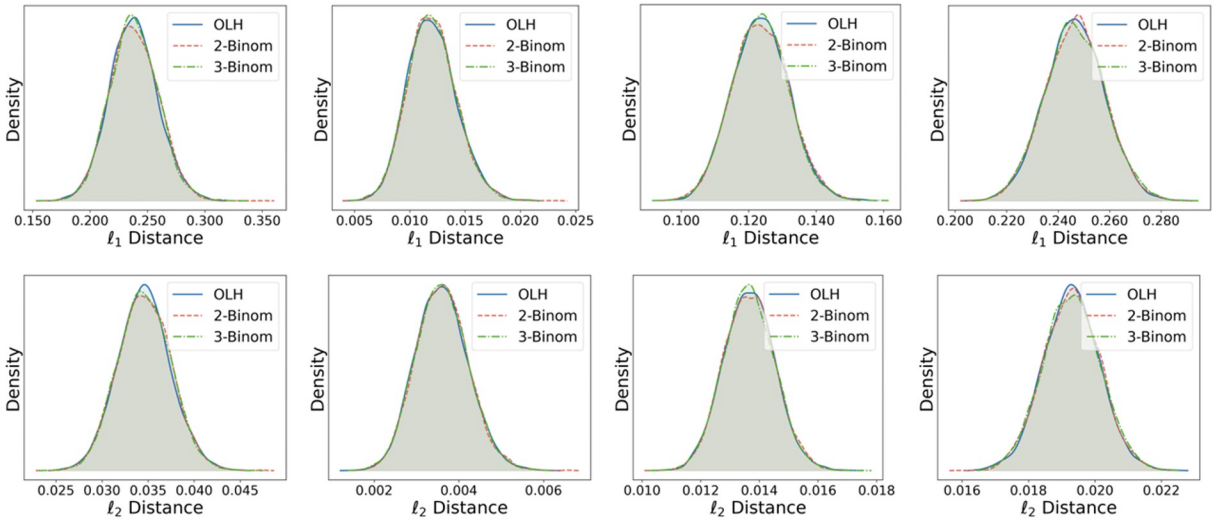


Fig. 2. KDE of ℓ_1 and ℓ_2 distances across 5,000 repetitions at $\varepsilon = 2.0$. Datasets from left to right: Adult, MSNBC, Kosarak, BMS-POS.

- The KDE curves overlap closely and are bell-shaped, sharing the same center and the same spread
- Agreement is tightest on the smallest domain (MSNBC, d = 17)
- Binomial methods match OLH not only in mean error, but in the full error distribution

Per-Value Variance

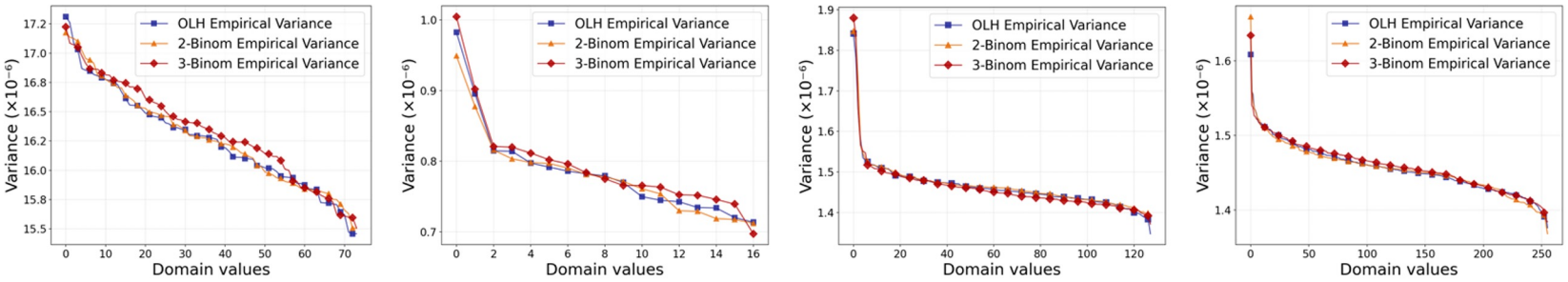


Fig. 3. Empirical variances of OLH, 2-Binom, and 3-Binom at $\varepsilon = 2.0$. Datasets from left to right: Adult, MSNBC, Kosarak, BMS-POS.

- Per-value variances of the three methods track one another across the whole domain
- Variance decreases with f_v, the behaviour predicted analytically by **Theorems 2 and 4**
- Together with the error results, this establishes the Binomial methods as **empirically equivalent** to OLH

Conclusion

- **2-Binom & 3-Binom:** Replace per-user OLH simulation with Binomial sampling to improve simulation efficiency
- Complexity drops from $O(nd)$ to $O(n+d)$, or $O(d)$ with cached counts, while unbiasedness and variance are **provably preserved**
- Execution times fall from minutes to milliseconds and memory from megabytes to kilobytes, with no change in utility
- **Future work:**
  - Integrate the proposed methods into LDP benchmarking platforms
  - Accelerate downstream OLH-based applications

Thank You!

Berkay Kemal Balioglu
bbalioglu23@ku.edu.tr

Alireza Khodaie
akhodaie22@ku.edu.tr

M. Emre Gürsoy
emregursoy@ku.edu.tr