Privacy-Preserving Data Analysis via the Johnson-Lindenstrauss Transform

CS798: Course Project — Presentation

Shlok Mishra, Srishti Chandra

IIT Kanpur
Differential Privacy in Machine Learning
Instructor: Dr. Sayak Chowdhury

20.04.2025

Contents

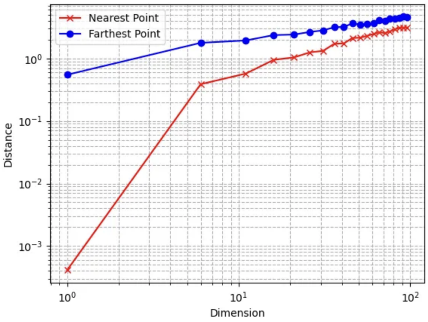
Differential Privacy in Dimensionality Reduction Using the JL Transform

- With the growth of data-intensive applications, privacy-preserving data analysis is becoming crucial in machine learning and statistics.
- Dimensionality reduction techniques, like the Johnson–Lindenstrauss (JL) transform, help to simplify data while preserving key structural properties.
- Two influential studies by Kenthapadi et al. [2013] and Blocki et al. [2012] demonstrate that the JL transform can also enhance differential privacy.
- We explore how the JL transform can be leveraged not only to reduce dimensionality but also to ensure rigorous privacy guarantees.

Motivation: Curse of Dimensionality & Privacy Needs

- High-dimensional data poses challenges:
  - Computational complexity grows significantly with data dimension.
  - Data sparsity makes algorithms inefficient and prone to overfitting.
- Privacy challenges:
  - Sharing detailed, high-dimensional data can inadvertently leak sensitive individual information.
  - Conventional anonymization methods are inadequate against sophisticated privacy attacks.
- Our goal: **Use random projections (JL transform) to simultaneously achieve dimensionality reduction and differential privacy, facilitating safer and more efficient data analysis.**

Illustration of the Curse of Dimensionality



Distances to nearest and farthest points as n increases (Image by the author)

Curse of Dimensionality Illustration

- In high-dimensional spaces, all points tend to become nearly equidistant from each other. This is a hallmark of the curse of dimensionality.
- **Loss of contrast:** It becomes difficult to distinguish between "close" and "far" points, which undermines the effectiveness of algorithms that rely on distance (such as clustering, nearest neighbor search, etc.).
- **Interpretation:** As the dimension increases, the usefulness of distance as a measure of similarity diminishes.

Johnson–Lindenstrauss Lemma: Intuition

- The Johnson–Lindenstrauss lemma enables dimensionality reduction by approximately preserving distances between points.
- Intuition: Randomly projecting data points from a high-dimensional space onto a much lower-dimensional space typically preserves pairwise distances up to a small distortion.
- Useful in scenarios where exact distances are less critical than relative distances.
- Enables efficient storage, computation, and visualization of high-dimensional datasets.

Johnson–Lindenstrauss Lemma: Dimentionality Reduction

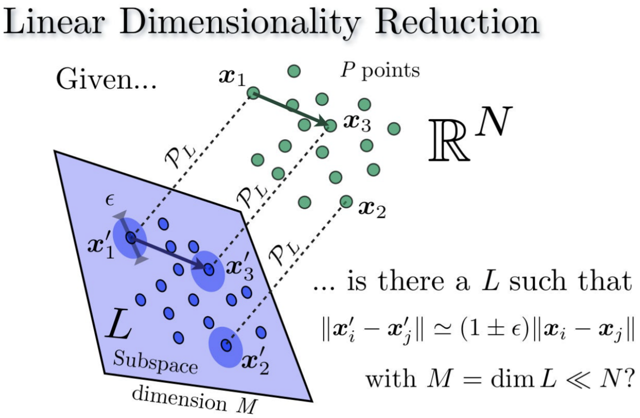


Figure: Johnson-Lindenstrauss Theoram Solves This Question

Johnson–Lindenstrauss Lemma: Formal Statement

- Formally, for any set of points in $\mathbb{R}^d$, there exists a random linear projection to $\mathbb{R}^k$ (with $k = O(\frac{\log n}{\epsilon^2})$) such that:

$$(1-\epsilon)\|u-v\|^2 \leq \|f(u)-f(v)\|^2 \leq (1+\epsilon)\|u-v\|^2$$

- Here, u and v are any two points, ϵ is the allowed distortion, and n is the number of points.
- Projection matrix often chosen randomly from Gaussian or binary distributions for practical ease.

Differential Privacy: Concept and Definition

- Differential privacy provides strong guarantees against privacy breaches by limiting the influence of individual data points.
- Definition: A randomized algorithm A is (ϵ, δ)-differentially private if for any two datasets X and X' differing by one individual's data, and for any event $S \subseteq \text{Range}(A)$:

$$\Pr[A(X) \in S] \leq e^{\epsilon} \cdot \Pr[A(X') \in S] + \delta$$

- ϵ: privacy budget (smaller ϵ means stronger privacy).
- δ: probability of privacy guarantee failure (usually very small).

Mechanisms for Differential Privacy (Noise & Sensitivity)

- Key mechanisms to achieve differential privacy include adding carefully calibrated random noise to query outputs.
- Laplace mechanism: Adds Laplace-distributed noise scaled to the query's ℓ_1-sensitivity, suitable for $(\epsilon, 0)$-privacy.
- Gaussian mechanism: Adds Gaussian noise scaled to ℓ_2-sensitivity, suitable for (ϵ, δ)-privacy and commonly used for its tighter concentration around zero.
- Sensitivity: Measures maximum possible change in the query result from altering one individual's data. Lower sensitivity allows adding smaller noise, thereby preserving more utility.

Key Differences

Aspect	**Kenthapadi et al. [2013]**	**Blocki et al. [2012]**
Noise Addition	Explicit Gaussian noise added after projection.	No explicit noise; relies solely on randomness from projection.
Data Requirements	No stringent requirements on data structure.	Requires specific data conditions (large singular values, rank-1 bounded differences).
Applications	General-purpose distance-based analysis (e.g., clustering, nearest neighbor).	Specific linear-algebraic and graph-based queries (cuts, covariance matrices).
Utility Advantage	Good accuracy and minimal distortion in distances due to controlled noise.	Superior scalability (dimension-independent noise magnitude).

Table: Key Differences Between Kenthapadi et al. [2013] and Blocki et al. [2012]

Privacy via JL Transform – Algorithm

- Kenthapadi et al. [2013] propose using JL transform combined with Gaussian noise addition to achieve differential privacy.
- Procedure:
  1. Generate a random JL projection matrix P.
  2. Project original data points using $y = Px$.
  3. Add Gaussian noise $\mathcal{N}(0, \sigma^2 I_k)$ to each projected point to preserve privacy.
- Resulting noisy projection $\tilde{y}$ provides rigorous privacy guarantees while maintaining geometric structures.

JL Projection Workflow with Privacy

High-Dimensional Data
$x \in \mathbb{R}^d$

↓

Generate Random Matrix
$P \in \mathbb{R}^{k\times d}, \ P_{ij} \sim \mathcal{N}\left(0, \frac{1}{k}\right)$

↓

Project: $y = Px, \quad y \in \mathbb{R}^k$

↓

Add Gaussian Noise:
$\tilde{y} = y + \mathcal{N}(0, \sigma^2 I_k)$

↓

Private Sketch $\tilde{y}$
(Low-dimensional & DP-protected)

↓

Use for Tasks
(e.g., clustering, classification)

Utility – Preserving Distances with Noise

- Despite adding Gaussian noise, pairwise distances are preserved in expectation.
- Adjusted distance calculation:

$$\|\tilde{y}_i - \tilde{y}_j\|^2 - 2k\sigma^2$$

  removes the expected noise contribution, yielding an unbiased estimate of the original squared distance.
- High probability guarantees ensure distances remain close to true values, enabling tasks like clustering and nearest neighbor search effectively.

Comparisons – Direct Noise Addition and Randomized Response

- Direct noise addition: Adds noise directly to each entry in pairwise distance matrix.
  - Requires significantly higher noise levels to achieve comparable privacy.
  - High sensitivity due to influence of individual changes on many distances.
- Randomized response: Flips individual bits randomly to obscure true data.
  - Performs poorly for moderately different data points.
  - Good for very large or very small differences but ineffective in mid-range scenarios.
- JL transform-based method balances privacy and accuracy effectively, outperforming direct noise addition and randomized response in typical scenarios.

JL-Based Differential Privacy for Graphs

- Blocki et al. [2012] proposed that a single Johnson–Lindenstrauss (JL) random projection can provide both dimensionality reduction and differential privacy.
- **Goal:** Answer many private cut queries efficiently.
- **Steps:**
  - Smooth edge weights to reduce sensitivity.
  - Apply JL projection to edge-incidence matrix.
  - Publish a noisy Laplacian sketch.
- **Output:**
  - Use the sketch to estimate any cut size.
  - No extra noise or privacy cost per query.
- **Result:** One-time release $\rightarrow$ unlimited, private, accurate cut queries.

Flowchart: JL-Based DP for Graphs

Input: Graph with Edge Set

Smooth Edge Weights
(Add small weight to all edges)

Apply JL Projection
to Edge-Incidence Matrix

Compute and Publish
Noisy Laplacian Sketch

Estimate Any Cut Size
Using Published Sketch

Accurate, Private
Answers for All Cuts

Comparison of Differential Privacy Mechanisms

Feature	**Input Pert.**	**Output Pert.**	**MW Mech.**	**JL-Based**
High-Dim Utility	✗	✓ (limited)	✓	✓
Many Queries	✓	✗	✓	✓
One-Time Publish	✓	✓	✗	✓
Simple to Use	✓	✓	✗	✓
Non-Interactive	✓	✓	✗	✓

✓ = *Advantage*, ✗ = *Limitation*

Theoretical Guarantees – Privacy & Utility Summary

- Both Kenthapadi et al. [2013] and Blocki et al. [2012] methods provide formal differential privacy guarantees.
- Utility guarantees:
  - Distances preserved within small multiplicative and additive errors.
  - Error bounds independent of original data dimensionality.
- Projection-based methods significantly enhance the trade-off between privacy protection and data utility.

Discussion & Open Questions

- How can privacy guarantees be extended when data distributions are not well-conditioned?
- Improving accuracy for very similar data points remains a challenge.
- Optimal parameter selection (projection dimension, noise levels) for practical applications.
- Exploring JL transform applicability beyond Euclidean distance scenarios.
- Future research directions to expand the practicality and effectiveness of JL-based privacy preservation.

Thank You!

Questions?

Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pages 410–419, 2012. doi: 10.1109/FOCS.2012.67.

Krishnaram Kenthapadi, Aleksandra Korolova, Ilya Mironov, and Nina Mishra. Privacy via the johnson-lindenstrauss transform. *Journal of Privacy and Confidentiality*, 5(1), Aug. 2013. doi: 10.29012/jpc.v5i1.625. URL `https://journalprivacyconfidentiality.org/index.php/jpc/article/view/625`.

Appendix A: Privacy Proof Sketch

- **DP Guarantee:** Add Gaussian noise $\Delta \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma \geq w_2(P) \frac{\sqrt{2(\ln \frac{1}{2\delta} + \varepsilon)}}{\varepsilon}$, where $w_2(P) = \max_i \|e_i P\|_2$.
  (Noise scale matches maximum row-sensitivity of P.)
- **Core Lemma:** If $\|Y' - Y\|_2 \leq w$, then for any measurable $S \subset \mathbb{R}^k$:

$$\Pr[Y' + \Delta \in S] \leq e^{\varepsilon} \Pr[Y + \Delta \in S] + \delta.$$

  (Extends 1D Gaussian mechanism to k dimensions via spherical symmetry.)
- **Proof Sketch:**
  1. A bit-flip in X changes a single row of XP, so $\|Y' - Y\|_2 \leq w_2(P)$.
  2. Rotate coordinates to align change along one axis (simplifies densities).
  3. Partition output space into:
     - *Inner region*: bound density ratios by e^{ε}.
     - *Outer region*: use Gaussian tail bound to cap mass by δ.

See Section 3.2.1 of the paper for full derivation.

Appendix B: Utility Proof Sketch

- **Unbiasedness:**
  $E[\|xP + \Delta - (yP + \Delta')\|_2^2 - 2k\sigma^2] = \|x - y\|_2^2.$
  (Noise has zero mean; projection preserves expectation.)
- **Variance Decomposition:**
  Write error $= Z_1 + Z_2 + Z_3$ with
  $Z_1 = \|(x-y)P\|_2^2,\ Z_2 = 2\langle (x-y)P, \Delta - \Delta' \rangle,\ Z_3 = \|\Delta - \Delta'\|_2^2 - 2k\sigma^2.$
  (Analyzes distortion from projection and from noise.)
- **Variance Formula:**
  [error] $= 2\|x-y\|_2^4/k + 8\sigma^2\|x-y\|_2^2 + 8\sigma^4 k.$
  (Term1: JL randomness; Term23: cross- and noise variance.)
- **Deviation Bound:**
  With prob. $1 - (\delta_{JL} + \delta_{\chi^2} + \delta_N)$,

$$|\text{error} - \|x-y\|_2^2| \leq \lambda_{JL}\|x-y\|_2^2 + 4\sigma^2(\sqrt{k}\lambda_{\chi^2} + \lambda_{\chi^2}^2) + 4\sigma(1+\lambda_{JL})\lambda_N\|x-y\|_2.$$

(Combine JL lemma for Z_1, chi-square tails for Z_3, and Gaussian tail for Z_2.)

See Section 3.2.2 of the paper for full details.

Appendix C: Supporting Lemmas

- **Gaussian DP Mechanism:** Noise σ gives (ε, δ)-DP if $\sigma \geq S\sqrt{2(\ln\frac{1}{2\delta} + \varepsilon)}/\varepsilon$ where S=sensitivity.
- **Johnson–Lindenstrauss Lemma:** $M \in^{r\times m} \sim N(0,1)$ for any x, $(1-\lambda)\|x\|^2 \leq \frac{1}{r}\|Mx\|^2 \leq (1+\lambda)\|x\|^2$ w.p. $1 - 2e^{-r\lambda^2/8}$.
- **Tail Bounds:**
  - Gaussian: $\Pr[|N(0,1)| > t] \leq e^{-t^2/2}$.
  - Chi-square: $\Pr[\chi_k^2 > k + 2\sqrt{kx} + 2x] \leq e^{-x}$.

  (Used to bound Z_3 and DJL deviations.)

Appendix D: Graph DP Proof Sketch

- **Row-wise DP:** Each sample $y^T E_G$ is a Gaussian with cov L_G, so by 1D analysis, it satisfies $(\varepsilon_0, \delta_0)$-DP for $\varepsilon_0 = \varepsilon/\sqrt{4r\ln(2/\delta)}$, $\delta_0 = \delta/(2r)$.
- **Matrix-level bounds:**
  - $L_G \preceq L_{G'}$ after smoothing covariances monotonic.
  - Eigenvalues $\sigma_i(L_G) \geq w$ well-conditioned.
  - Determinant ratio $\sqrt{\det L_{G'} / \det L_G} \leq e^{\varepsilon_0/2}$.
- **PDF Ratio:**

$$\frac{G'(x)}{G(x)} = \frac{e^{-\frac{1}{2}x^T L'^{\dagger}_G x}}{e^{-\frac{1}{2}x^T L^{\dagger}_G x}} \cdot \sqrt{\frac{\det L_G}{\det L_{G'}}} \leq e^{\varepsilon_0} + \delta_0.$$

- Compose r independent rows overall (ε, δ)-DP.

See Section 3.1 of Blocki et al. for derivation.

Appendix E: Graph Utility Proof Sketch

- **JL Preservation:** $\frac{1}{r}\|ME_G 1_S\|^2 = (1 \pm \eta) 1_S^T L_G 1_S$ w.p. $1 - \nu$ by JL lemma.
- **Smooth Laplacian:** Published $L_H = \frac{w}{n}K + (1 - \frac{w}{n})L_G$ shifts all eigenvaluesw/n.
- **Query Recovery:** $R(S) = \frac{1}{1-w/n}(y - \frac{w}{n}s(n-s))$ exacts the cut size.
- **Error:** Multiplicative $1 \pm \eta$ plus additive $O\big(s \cdot \eta\big) = O\big(s \frac{\sqrt{\ln(1/\delta)\ln(1/\nu)}}{\varepsilon}\big)$.
  (Combine JL distortion with arithmetic on L_H.)

See Theorem 3.2 of Blocki et al. for full proof.

Appendix F: Covariance Algorithm Proof Sketch

- **Data Translation:** Add deterministic offset so singular values $\sigma_i(A) \geq w$.
  (Ensures 'small' directions get lifted above sensitivity floor.)
- **Row-wise DP:** Each projected row of MA is Gaussian with cov AA^T, so DP as in D.
- **JL-Utility:** For any unit vector x,

$$\|(MA/\sqrt{r})^T x\|^2 = (1 \pm \eta)\|Ax\|^2 + w^2 \eta.$$

  (JL lemma on rows of A in translated basis.)
- **Resulting Error:** Subtract w^2 to center variance multiplicative $(1 \pm \eta)$, additive $O(\eta w^2) = O\big(\frac{\ln(1/\delta)\ln(1/\nu)}{\varepsilon^2 \eta}\big)$.

See Section 4 of Blocki et al. for details.