UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CLUSTERING DIFERENCIALMENTE PRIVADO MEDIANTE LA GENERACIÓN DE SINOPSIS PRIVADAS DE DATOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

BRUNO GERMÁN RODRÍGUEZ SEPÚLVEDA

PROFESOR GUÍA:
FEDERICO OLMEDO BERÓN

PROFESOR CO-GUÍA:
MATÍAS TORO IPINZA

MIEMBROS DE LA COMISIÓN:
ÉRIC TANTER
EDUARDO RIVEROS ROCA

SANTIAGO DE CHILE
2024

Resumen

Actualmente estamos en una era donde la generación de datos masivos es constante y crucial para diferentes sectores como empresas, gobiernos y centros de investigación. Esta realidad ha impulsado el desarrollo del campo de la minería de datos, enfocado en extraer información valiosa de grandes volúmenes de datos. En particular, se destaca la importancia del agrupamiento (*clustering*) de datos y el algoritmo *k-means*, un método para la creación de grupos de datos, aunque enfrenta desafíos como la dificultad en encontrar el agrupamiento óptimo, un problema muy difícil (NP-Hard).

Existe una creciente preocupación por el manejo de datos sensibles, especialmente en lo que respecta a la privacidad en el uso de algoritmos de *clustering* como *k-means*. Para abordar esto, se han incorporado técnicas de privacidad diferencial. En pocas palabras, estas técnicas modifican el algoritmo para **enmascarar** la contribución de cada individuo.

Nuestro enfoque se basa en crear una aproximación de los datos que tenga garantías de privacidad. La denominaremos *sinopsis privada*. Esta sinopsis privada se emplea como entrada para el algoritmo *k-means*, el cual preservara las garantías de privacidad. La elaboración de la sinopsis privada se llevará a cabo mediante técnicas de particionamiento espacial y conteo privado. En otras palabras, contamos de manera privada cuantos puntos hay en regiones del espacio (definidas por una partición) y este resultado lo representamos como un conjunto de puntos ponderados, lo que seria la sinopsis privada.

Para evaluar la eficacia de nuestros algoritmos de generación de sinopsis privadas, realizamos una comparativa entre los resultados obtenidos mediante *k-means* aplicados a sinopsis privadas y aquellos obtenidos mediante algoritmos de *clustering*, tanto privados como no privados. Investigamos la eficiencia de distintos tipos de particionadores en combinación con un análisis exhaustivo de hiperparámetros, incluyendo aspectos como el presupuesto de privacidad, las estrategias para su distribución, los umbrales de la condición de parada y la profundidad máxima en los pasos recursivos.

Este estudio propuso y evaluó distintas estrategias de particionamiento del espacio para crear sinopsis privadas. Se descubrió que, aunque no siempre superaron los *benchmarks*, los métodos de particionamiento multi-cuantil y uniforme resultaron ser prometedores, sobre todo al ajustar adecuadamente los hiperparámetros. Los resultados obtenidos con estos particionadores son alentadores, mostrándose competentes en términos de privacidad y precisión.

Esto se lo dedico a mi familia, por su apoyo incondicional en todo lo que he querido hacer.

A mis amigos, por estar siempre ahí y disfrutar conmigo esta etapa de nuestras vidas.

A Inés, por impulsarme a ser una mejor persona y compartir nuestras vidas.

Agradecimientos

Quiero agradecer a mis padres Rocío y Oscar, por todo lo que me han enseñado y lo que me siguen enseñando. A mis hermanas Catalina y Gabriela, y mi hermano José Pablo. A mi abuela María Angélica.

A mis amigos de la universidad: Nicolás Fierro, Victoria, Nicolás Cornejo, Max, Cristian.

A mis amigos de computación, con quienes tuve la suerte de coincidir y compartir el último tramo de esta carrera: Albani, Asunción, Camila, Daniel, José Tomas, Alonso, María José, Diego, Julia, Konrad, Louise, Paula, Roberto, Sebastián, Pablo, Tomás, Valentina.

A mis profesores guías: Matías Toro y Federico Olmedo, por su paciencia e increíble disposición a ayudarme.

Tabla de Contenido

1. Introducción **1**

1.1. Contexto 1

1.2. Problema 2

1.3. Objetivos 2

1.4. Solución 3

2. Preliminares **4**

2.1. Privacidad diferencial 4

2.2. *Clustering* 7

2.3. *k-means* 9

2.4. Cálculo distribuido 11

3. Estado del arte **13**

3.1. *k-means* diferencialmente privado 13

3.2. Enfoque interactivo 13

3.3. Enfoque no interactivo 14

4. Solución **16**

4.1. Estrategia de particionamiento recursivo 19

4.2. Particionamiento por nivel 21

4.2.1. Partición binaria 21

4.2.2. Partición mediana 24

4.2.3. Partición uniforme 26
4.2.4. Partición multi-cuantil 28
4.3. Asignación de presupuesto 31
4.3.1. Problema de optimización 32
4.4. Implementación y tecnologías 34

5. Evaluación **37**
5.1. Conjuntos de datos 37
5.2. Métricas 38
5.3. Marco experimental 40
5.4. *Baseline* y *benchmarks* 42
5.5. Resultados 43
5.6. Interpretación de resultados 53

6. Conclusión **55**
6.1. Trabajo futuro 56

Bibliografía **59**

Anexo A. Gráficos **60**

Anexo B. Tablas **70**

Índice de Tablas

4.1. Resumen de notaciones. 19

4.2. Tabla de comparación de particionadores 31

5.1. Descripción de los conjuntos de datos 38

5.2. Espacio de configuraciones 41

A.1. Espacio de configuraciones de resultados 60

B.1. S1 - NICV for Binary partitioner with Threshold 10 70

B.2. S1 - NICV for Median partitioner with Threshold 10 70

B.3. S1 - NICV for Uniform partitioner with Threshold 10 71

B.4. S1 - NICV for Multi-quantile partitioner with Threshold 10 71

B.5. S2 - NICV for Binary partitioner with Threshold 10 72

B.6. S2 - NICV for Median partitioner with Threshold 10 72

B.7. S2 - NICV for Uniform partitioner with Threshold 10 73

B.8. S2 - NICV for Multi-quantile partitioner with Threshold 10 73

B.9. S3 - NICV for Binary partitioner with Threshold 10 74

B.10.S3 - NICV for Median partitioner with Threshold 10 74

B.11.S3 - NICV for Uniform partitioner with Threshold 10 75

B.12.S3 - NICV for Multi-quantile partitioner with Threshold 10 75

B.13.S4 - NICV for Binary partitioner with Threshold 10 76

B.14.S4 - NICV for Median partitioner with Threshold 10 76

B.15.S4 - NICV for Uniform partitioner with Threshold 10 77

B.16.S4 - NICV for Multi-quantile partitioner with Threshold 10 77
B.17.Shuttle - NICV for Binary partitioner with Threshold 80 78
B.18.Shuttle - NICV for Median partitioner with Threshold 80 78
B.19.Shuttle - NICV for Uniform partitioner with Threshold 80 79
B.20.Shuttle - NICV for Multi-quantile partitioner with Threshold 80 79
B.21.Skin Segmentation - NICV for Binary partitioner with Threshold 80 80
B.22.Skin Segmentation - NICV for Median partitioner with Threshold 80 80
B.23.Skin Segmentation - NICV for Uniform partitioner with Threshold 80 81
B.24.Skin Segmentation - NICV for Multi-quantile partitioner with Threshold 80 . 81
B.25.Tarvel Review Ratings - NICV for Binary partitioner with Threshold 80 . . . 82
B.26.Tarvel Review Ratings - NICV for Median partitioner with Threshold 80 . . 82
B.27.Tarvel Review Ratings - NICV for Uniform partitioner with Threshold 80 . . 83
B.28.Tarvel Review Ratings - NICV for Multi-quantile partitioner with Threshold 80 83
B.29.S1 - CI for Binary partitioner with Threshold 10 84
B.30.S1 - CI for Median partitioner with Threshold 10 84
B.31.S1 - CI for Uniform partitioner with Threshold 10 85
B.32.S1 - CI for Multi-quantile partitioner with Threshold 10 85
B.33.S2 - CI for Binary partitioner with Threshold 10 86
B.34.S2 - CI for Median partitioner with Threshold 10 86
B.35.S2 - CI for Uniform partitioner with Threshold 10 87
B.36.S2 - CI for Multi-quantile partitioner with Threshold 10 87
B.37.S3 - CI for Binary partitioner with Threshold 10 88
B.38.S3 - CI for Median partitioner with Threshold 10 88
B.39.S3 - CI for Uniform partitioner with Threshold 10 89
B.40.S3 - CI for Multi-quantile partitioner with Threshold 10 89
B.41.S4 - CI for Binary partitioner with Threshold 10 90
B.42.S4 - CI for Median partitioner with Threshold 10 90

B.43.S4 - CI for Uniform partitioner with Threshold 10 91
B.44.S4 - CI for Multi-quantile partitioner with Threshold 10 91
B.45.shuttle - CI for Binary partitioner with Threshold 80 92
B.46.shuttle - CI for Median partitioner with Threshold 80 92
B.47.shuttle - CI for Uniform partitioner with Threshold 80 93
B.48.shuttle - CI for Multi-quantile partitioner with Threshold 80 93
B.49.Skin Segmentation - CI for Binary partitioner with Threshold 80 94
B.50.Skin Segmentation - CI for Median partitioner with Threshold 80 94
B.51.Skin Segmentation - CI for Uniform partitioner with Threshold 80 95
B.52.Skin Segmentation - CI for Multi-quantile partitioner with Threshold 80 . . 95
B.53.Tarvel Review Ratings - CI for Binary partitioner with Threshold 80 96
B.54.Tarvel Review Ratings - CI for Median partitioner with Threshold 80 96
B.55.Tarvel Review Ratings - CI for Uniform partitioner with Threshold 80 97
B.56.Tarvel Review Ratings - CI for Multi-quantile partitioner with Threshold 80 97

Índice de Ilustraciones

1.1. Sinopsis privada adaptativa 3

2.1. *k-means* 7
2.2. Dendrograma 7
2.3. DBSCAN 8
2.4. Distribución Gaussiana 8

3.1. Consulta Q en un *quadtree*. 15

4.1. Creación de una sinopsis privada. 16
4.2. Ejemplo particionador binario - tres iteraciones. 21
4.3. Ejemplo de sinopsis privada con particionador binario para S1 22
4.4. Ejemplo particionador mediano - tres iteraciones. 24
4.5. Ejemplo de sinopsis privada con particionador mediano para S1 24
4.6. Ejemplo particionador uniforme - una iteración. 26
4.7. Ejemplo de sinopsis privada con particionador uniforme para S1 27
4.8. Ejemplo particionador multi-cuantil con 3 cuantiles - dos iteraciones. 29
4.9. Ejemplo de sinopsis privada con particionador multi-cuantil para S1 29
4.10. Diagrama de clases 36

5.1. Ejemplo: índice de centroides de 4 39
5.2. Diagrama del marco experimental 41

Capítulo 1

Introducción

1.1. Contexto

Actualmente, estamos generando más datos que nunca en la historia. Estos datos se han convertido en un recurso invaluable para empresas, gobiernos y centros de investigación, facilitando la toma de decisiones informadas, el descubrimiento de patrones y la planificación estratégica. Este fenómeno ha impulsado el desarrollo del área de minería de datos, dedicada a extraer información significativa de grandes volúmenes de datos.

Dentro de los métodos de análisis, la categorización de datos destaca como uno de los más cruciales. Existe una gama de algoritmos diseñados para segmentar la información en grupos distintos. Entre estos, *k-means* es uno de los más conocidos, enfocado en minimizar la varianza dentro de los grupos. El método más popular para implementar *k-means* es el algoritmo de *clustering* de Lloyd [1982]. Este proceso comienza seleccionando k centroides iniciales y luego itera para refinar estos puntos. En cada iteración, los datos se dividen en k grupos basados en la proximidad al centroide más cercano, recalculando luego los centroides como el centro geométrico de cada grupo. Sin embargo, encontrar la partición óptima que minimice la varianza total es un problema NP-Hard, por lo que este algoritmo tiende a encontrar soluciones de óptimo local. Esta no es la única dificultad, también hay que tener en cuenta la privacidad de los datos.

En la era actual, surge una preocupación creciente sobre el manejo de datos sensibles, como información financiera, médica o sociocultural. El algoritmo *k-means* no es una excepción. Existe el riesgo de violación de la privacidad, especialmente cuando se identifican *clusters* pequeños en los que un cambio mínimo puede alterar significativamente el resultado, revelando potencialmente información sobre los elementos individuales. Para mitigar este riesgo, se han desarrollado variantes de *k-means* que integran técnicas de privacidad diferencial, las que dan garantías de privacidad, actualmente la norma de facto en la manipulación de datos sensibles. Un algoritmo se considera diferencialmente privado si, al añadir o eliminar un dato, el resultado obtenido se mantiene sustancialmente inalterado.

Las variantes privadas de *k-means* se dividen en dos categorías : interactivas y no interactivas. Las interactivas modifican cada paso del algoritmo de *clustering* para integrar privacidad diferencial, generalmente mediante la adición de ruido Laplaciano a los cálculos. Por otro lado, las variantes no interactivas aprovechan la propiedad de post-procesamiento de la privacidad diferencial. Esto permite la creación de una *sinopsis privada* que luego se utiliza como entrada para el algoritmo *k-means*; garantizando así que el resultado final mantenga las propiedades y garantías de privacidad diferencial. Para generar una sinopsis privada, primero se realiza una partición del espacio de datos, seguido de un conteo diferencialmente privado de los puntos en cada parte de la partición. La sinopsis privada se conforma entonces por el centro de cada elemento de la partición, acompañado de su respectivo conteo diferencialmente privado, representándose como un punto con un peso específico. Esta sinopsis privada constituye luego la entrada para el algoritmo de *k-means*.

1.2. Problema

En el ámbito del *clustering*, persiste una brecha significativa: la falta de investigaciones que aborden simultáneamente los desafíos de *privacidad*, *procesamiento paralelo* y eficacia en conjuntos de datos de *alta dimensión*.

Esta laguna representa un desafío crítico en una era donde los estándares de privacidad, la complejidad dimensional y el volumen de datos están en constante crecimiento. En un mundo que acumula datos con un número cada vez mayor de características (o *features*), la habilidad para analizar estos datos de manera eficiente y segura se convierte en un diferenciador competitivo clave.

Aunque existen numerosos estudios que satisfacen dos de nuestros tres criterios, lamentablemente, la integración de los tres de manera simultánea sigue siendo esquiva. Esta tesis busca abordar esta brecha. Nuestro objetivo es equilibrar y encontrar un compromiso óptimo entre privacidad, capacidad de cómputo paralelo y eficacia en el manejo de datos de alta dimensión.

1.3. Objetivos

El objetivo principal de este trabajo es diseñar e implementar diversas estrategias para la generación de sinopsis diferencialmente privadas. Estas estrategias serán comparadas en función de su escalabilidad (para adaptarse a *big data*), precisión (para minimizar la pérdida de información en las sinopsis) y su eficacia con datos dispersos.

Los objetivos específicos son:

1. Diseñar algoritmos de creación de sinopsis privadas que mejoren el equilibrio entre privacidad y utilidad para *clustering*.
2. Implementar algoritmos de creación de sinopsis privadas.

3. Evaluar el desempeño de los algoritmos implementados y comparar frente a las alternativas en el estado del arte.

1.4. Solución

Anteriormente, en el contexto del proyecto de colaboración con el Plan Ceibal[1], Federico Olmedo y Matías Toro realizaron una contribución significativa. Diseñaron e implementaron una versión diferencialmente privada no interactiva del algoritmo *k-means*, la llamaremos **DPKFM**.

La creación de una sinopsis privada de **DPKFM** comienza con un conteo inicial de elementos en el conjunto de datos, determinando el nivel de refinamiento de la partición en grillas de primer nivel, como se ilustra en la Figura 1.1. Para cada celda de esta grilla, se realiza un segundo conteo que define la subdivisión en el segundo nivel. Se asume que los puntos dentro de cada celda pueden ser representados por un punto en el centro de esta con un peso equivalente al número de puntos en la celda. Finalmente, la sinopsis privada consiste en un conteo privado para cada celda, acompañado de las coordenadas del centro de la misma, proporcionando una representación simplificada y precisa del conjunto de datos.

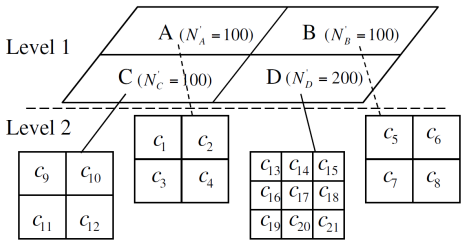


Figura 1.1: Sinopsis privada adaptativa

Sin embargo, esta versión inicial enfrentó desafíos en términos de precisión, especialmente con datos dispersos o de altas dimensiones. A pesar de ofrecer garantías formales de privacidad, el algoritmo tendía a identificar conjuntos de *clusters* diferentes de los que se obtienen con la versión tradicional (no diferencialmente privada) de *k-means*, resultando en una disminución de la precisión.

Este estudio se enfoca en la **generalización** del modelo inicial de F. Olmedo y M. Toro para la generación de sinopsis privadas, ampliando el diseño de **dos** niveles a **múltiples** niveles y adoptando diversas técnicas de partición espacial. Las técnicas de partición incluyen métodos dependientes e independientes de la distribución de datos, destacando la novedosa partición en un número arbitrario de cuantiles. Además, se investigarán métodos de asignación de presupuesto de privacidad, optimizando el algoritmo para el manejo eficiente de conjuntos de datos masivos y dispersos. Estas mejoras tienen como objetivo incrementar significativamente la precisión del algoritmo.

[1]Anonimización escalable de datos masivos y su aplicación al área de analíticas de aprendizaje. Fondo Sectorial de Investigación a partir de Datos FSDA_1_2018_1_1_154620. Agencia Nacional de Investigación e Innovación (ANII), Uruguay. 5/2020- 4/2021

Capítulo 2

Preliminares

2.1. Privacidad diferencial

En esta sección exploraremos en detalle el concepto de privacidad diferencial, sus fundamentos teóricos y cómo está contribuyendo a garantizar un equilibrio entre la utilidad de los datos y la protección de la privacidad de los individuos en un mundo digitalmente conectado.

La privacidad diferencial es un concepto esencial en el campo de la privacidad y la seguridad de datos en la era digital. En un mundo cada vez más interconectado y dependiente de la recopilación y análisis de datos, la privacidad de los individuos se ha convertido en una preocupación crítica. La privacidad diferencial se presenta como una solución innovadora y prometedora para abordar este desafío, al permitir que las organizaciones y los investigadores compartan datos y obtengan información útil sin comprometer la privacidad de las personas involucradas.

Definición 2.1 (ε-privacidad diferencial) *Una función que satisface la privacidad diferencial a menudo se denomina mecanismo. Formalmente, un mecanismo aleatorio M satisface ε-privacidad diferencial si para cualquier par de conjuntos de datos D, D' vecinos (consideramos dos conjuntos de datos D y D' como vecinos si y solo si o $D = D' + t$ o $D' = D + t$, donde $D + t$ denota el conjunto de datos resultante de agregar la tupla t al conjunto de datos D. Usamos $D \simeq D'$ para denotar esto.), y para todo $S \in Range(M)$, se cumple que:*

$$\frac{\Pr(M(D) = S)}{\Pr(M(D') = S)} \leq e^{\varepsilon}$$

El presupuesto de privacidad será denotado por ε, es un parámetro no negativo que controla el nivel de privacidad. Un valor más pequeño de ε implica un mayor nivel de privacidad.

La necesidad de una definición formal de privacidad (Definición 2.1) surge de la creciente complejidad y la cantidad de datos personales que se recopilan, procesan y comparten en el entorno digital. En este contexto, simplemente confiar en políticas de privacidad vagas o en medidas de seguridad insuficientes no es suficiente para proteger a los individuos de los ries-

gos asociados con el mal uso de sus datos. Aquí es donde la privacidad diferencial se presenta como un marco matemáticamente riguroso, proporcionando una garantía fuerte y cuantificable de privacidad. Este enfoque se centra en la idea de *negación plausible*, asegurando que la inclusión o exclusión de un individuo en un conjunto de datos no altera significativamente los resultados del análisis de dichos datos. Esto significa que un atacante, incluso con conocimientos avanzados y acceso a datos adicionales, no puede determinar con certeza si los datos de una persona específica están incluidos en el conjunto de datos analizado. Para alcanzar ε-privacidad diferencial existen múltiples métodos, los que motivan las siguientes definiciones.

Definición 2.2 (Sensibilidad global) *Dada una función de consulta $F : D \to \mathbb{R}^d$ (donde D es un conjunto de datos, $\mathbb{R}^d$ es un vector real de dimensión d), la sensibilidad global se puede expresar como:*

$$\Delta F = \max_{D_1, D_2, D_1 \simeq D_2} ||F(D_1) - F(D_2)||_1$$

donde, $||F(D_1) - F(D_2)||_1$ es la distancia norma 1 del resultado de los conjuntos de datos vecinos D_1 y D_2, lo que indica la sensibilidad de la función de consulta a los cambios de datos. La sensibilidad global solo está relacionada con la función de consulta y no con el conjunto de datos específico.

Definición 2.3 (Mecanismo de Laplace) *Para un conjunto de datos D y una función de mapeo f con la sensibilidad global (Definición 2.2) de Δf, el mecanismo aleatorio $A(D) = f(D) + Y$ proporciona ε privacidad diferencial, donde $Y \sim Lap\left(\frac{\Delta f}{\varepsilon}\right)$ es el ruido de Laplace aleatorio incorporado que obedece la distribución de Laplace con el parámetro $\frac{\Delta f}{\varepsilon}$. La densidad de probabilidad de $Lap\left(\frac{\Delta f}{\varepsilon}\right)$ se puede expresar como:*

$$p(x) = \frac{\varepsilon}{2\Delta f} e^{-\frac{\varepsilon |x|}{\Delta f}}$$

El valor esperado de la distribución de Laplace es 0 y la varianza es $2\left(\frac{\Delta f}{\varepsilon}\right)^2$.

Teorema 2.4 *El mecanismo de Laplace es ε-diferencialmente privado.*

Definición 2.5 (Mecanismo exponencial) *Para un conjunto de datos $D \in \mathcal{D}$, un conjunto $\mathcal{R}$ y una función de utilidad $u : \mathcal{D} \times \mathcal{R} \to \mathbb{R}$ que tiene una sensibilidad global Δu con respecto a $u(\cdot, r)$, con $r \in \mathcal{R}$. El mecanismo exponencial selecciona y produce una salida $r \in \mathcal{R}$, donde la probabilidad de selección de cada r esta dada por:*

$$\Pr[r \text{ es seleccionado}] \propto \exp\left(\frac{\varepsilon u(D, r)}{2\Delta u}\right).$$

Teorema 2.6 *El mecanismo exponencial es ε-diferencialmente privado.*

La privacidad diferencial, permite el uso de datos para análisis y toma de decisiones importantes, mientras protege la confidencialidad individual. Estos análisis suelen estar compuestos

de múltiples pasos, lo que motiva el siguiente teorema.

Teorema 2.7 (Composición secuencial) *Supongamos que existe un conjunto de mecanismos de privacidad diferencial,* $\{M_1, M_2, \ldots, M_n\}$*, y un conjunto de datos arbitrario* D*, donde* M_i $(1 \leq i \leq n)$ *satisface* ε_i*-privacidad diferencial en el conjunto de datos* D*. Entonces, cualquier función* $G(M_1, \ldots, M_n)$ *sobre los mecanismos es* $\sum_{i=1}^{n} \varepsilon_i$ *diferencialmente privada.*

El aspecto secuencial de algunos algoritmos en el marco de la privacidad diferencial juega un papel crucial en la determinación del tamaño del presupuesto de privacidad (ε). En un contexto secuencial, donde se realizan múltiples cálculos o consultas en una serie, el impacto acumulativo en la privacidad debe ser cuidadosamente manejado. Esto generalmente implica la necesidad de asignar presupuestos de privacidad más pequeños para cada operación individual, con el fin de mantener un nivel de privacidad global aceptable. Otra manera de distribuir el presupuesto de privacidad es a través de una división del conjunto de datos, lo que motiva el Teorema 2.8.

Teorema 2.8 (Composición paralela) *Supongamos que el conjunto de datos* D *se puede dividir en* n *partes de subconjuntos independientes y disjuntos,* $\{D_1, D_2, \ldots, D_n\}$*, es decir, una partición de* D*. Existe un conjunto de mecanismos aleatorios,* $\{M_1, M_2, \ldots, M_n\}$*, aplicados a los subconjuntos anteriores respectivamente, donde* $M_i (1 \leq i \leq n)$ *satisface la privacidad diferencial* ε_i *en* D_i*. Entonces, cualquier función* $G(M_1, \ldots, M_n)$ *sobre los mecanismos es* máx$\{\varepsilon_i\}$*-diferencialmente privada.*

El Teorema 2.8 de composición paralela es esencial en la elaboración de sinopsis privadas de datos. Este teorema permite la aplicación de mecanismos de privacidad diferencial de forma independiente a cada subconjunto dentro de una partición. Tal enfoque garantiza la gestión efectiva del presupuesto total de privacidad, previniendo incrementos desmesurados.

Una de las características más valiosas de la privacidad diferencial es el Teorema 2.9, el cual asegura que cualquier manipulación realizada sobre el resultado de un mecanismo de privacidad diferencial preserva sus garantías de privacidad, sin importar la naturaleza de la función aplicada.

Teorema 2.9 (Postprocesamiento) *Sea* $\mathcal{M} : D \to R$ *un mecanismo que es* ε*-diferencialmente privado y sea* $g : R \to H$ *una función cualquiera. Entonces, el mecanismo obtenido mediante la composición* $M(x) = g(\mathcal{M}(x))$ *también es* ε*-diferencialmente privado.*

El postprocesamiento en el contexto de creación de sinopsis juega un papel crucial al permitir la manipulación y mejora de datos protegidos sin comprometer su privacidad. Esta fase es esencial para refinar los resultados obtenidos tras aplicar ruido para asegurar la privacidad, transformándolos en formatos más útiles y manejables. Además, facilita análisis adicionales y la integración de estos resultados en flujos de trabajo de análisis de datos más amplios, manteniendo la utilidad y coherencia de los datos a lo largo de múltiples etapas de procesamiento.

2.2. *Clustering*

El *clustering*, o agrupamiento, es una técnica esencial en el análisis de datos que desempeña un papel fundamental en una amplia gama de aplicaciones, desde la segmentación de clientes en marketing hasta la identificación de patrones en datos biológicos en bioinformática. Esta técnica tiene como objetivo principal la organización de datos en grupos o *clusters*, de tal manera que los elementos dentro de un mismo grupo sean más similares entre sí que con los elementos en otros grupos. El *clustering* es una herramienta poderosa para la exploración y comprensión de datos, ya que puede revelar estructuras ocultas y patrones subyacentes que de otro modo serían difíciles de identificar.

Los algoritmos de *clustering* se dividen en diversas categorías, cada una con enfoques y características particulares para abordar diferentes desafíos en la agrupación de datos. Estas categorías incluyen:

Clustering Particional: Estos algoritmos buscan dividir el conjunto de datos en un número predeterminado de *clusters*, como el conocido algoritmo *k-means*. Sus ventajas incluyen eficiencia computacional y escalabilidad para grandes conjuntos de datos. Sin embargo, requieren que se especifique el número de *clusters* de antemano y pueden ser sensibles a la elección inicial de centroides. En la Figura 2.1 se aprecia el resultado de un algoritmo de *k-means* con $k = 12$ en un conjunto de dos dimensiones además se colorean las regiones de cada *cluster*.

Clustering Jerárquico: Este enfoque crea una jerarquía de *clusters* mediante la unión o división de *clusters* en función de la similitud entre los elementos. Es útil cuando no se conoce de antemano el número óptimo de *clusters*. Sin embargo, puede ser computacionalmente costoso y no siempre es apropiado para grandes conjuntos de datos. En la Figura 2.2 se muestra un *clustering* jerárquico, a la izquierda los puntos y los *cluster* respectivos con respecto al gráfico de la derecha.

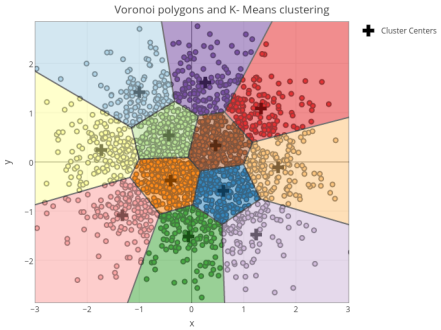


Figura 2.1: *k-means*

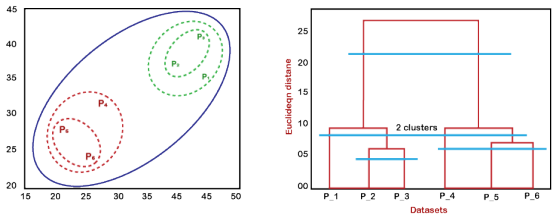


Figura 2.2: Dendrograma

Clustering Basado en Densidad: Algoritmos como **DBSCAN** identifican *clusters* en áreas densamente pobladas de datos y pueden manejar conjuntos de datos con formas irregulares. No requieren especificar previamente el número de *clusters*, pero son sensibles a la elección de parámetros y pueden tener dificultades con datos de diferente densidad. En la Figura 2.3 hay dos *clusters* claramente diferenciados de puntos: uno en azul y otro en verde, ambos en un fondo blanco con puntos grises dispersos alrededor, que probablemente representan datos atípicos o ruido.

Clustering Basado en Modelos: Estos algoritmos intentan ajustar modelos probabilísticos a los datos, como el algoritmo de mezcla de *Gaussianas* (**GMM**). Son útiles cuando los datos se ajustan a distribuciones específicas. Sin embargo, pueden ser computacionalmente costosos y dependen de suposiciones sobre la distribución de los datos. En la Figura 2.4, se observa que la distribución de color azul se ajusta de manera significativa a los datos presentados, mientras que la distribución de color rojo muestra un ajuste más moderado.

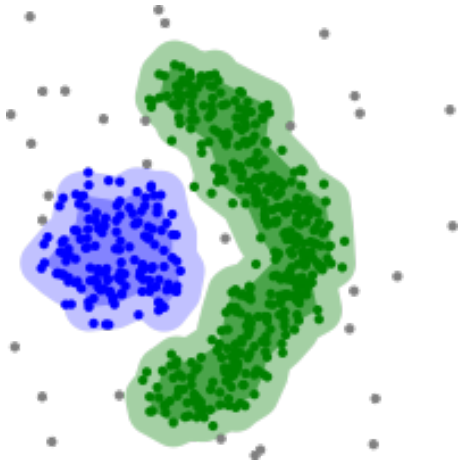


Figura 2.3: DBSCAN

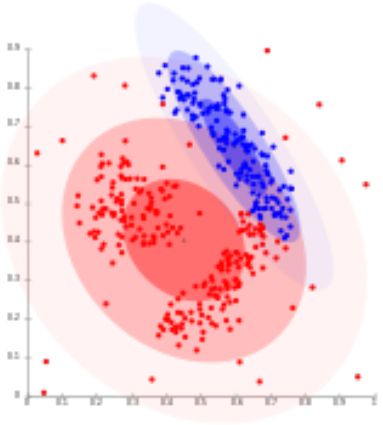


Figura 2.4: Distribución Gaussiana

Cada método presenta ventajas y desafíos únicos, y la selección del algoritmo más conveniente varía según las características específicas de los datos y los objetivos del análisis. Debido a su notable popularidad y versatilidad en una amplia gama de aplicaciones, este estudio opta por utilizar el algoritmo de *k-means*. A continuación, se realiza una exploración exhaustiva de este algoritmo, abarcando los avances más recientes en el área.

2.3. *k-means*

El algoritmo de *k-means* (dentro de la categoría *clustering* particional) busca encontrar una solución al siguiente problema: dado un conjunto de datos $X = \{x_1, x_2, \ldots, x_n\}$ donde cada $x_i \in \mathbb{R}^d$ representa un punto en un espacio d-dimensional, y un número entero k que representa el número de *clusters* a formar, el objetivo es encontrar un conjunto de k centroides $C = \{c_1, c_2, \ldots, c_k\}$, de tal manera que se minimice la suma total de las distancias cuadradas entre cada punto y el centroide más cercano.

La función objetivo a minimizar es:

$$\min_{C} \sum_{i=1}^{n} \min_{c_j \in C} \|x_i - c_j\|^2 \tag{2.1}$$

donde $\|x_i - c_j\|$ es la distancia euclidiana entre el punto x_i y el centroide c_j, el valor de esta función.

La solución mas conocida es el algoritmo de Lloyd (Algoritmo 1), que se remonta a su primera versión propuesta por Lloyd [1982]. El algoritmo de Lloyd parte de la idea de seleccionar inicialmente k puntos aleatorios como centroides iniciales, lo que lo hace relativamente sencillo de entender y de implementar. Sin embargo, su simplicidad conlleva ciertas limitaciones que se han vuelto evidentes con el tiempo.

Algorithm 1 Algoritmo de Lloyd para *k-means*

```
1: Elegir el número de clusters k
2: Inicializar centroides de los clusters de manera aleatoria
3: repeat
4:     for cada punto de datos x_i do
5:         Asignar x_i al centroide más cercano
6:     end for
7:     for cada cluster j do
8:         Recalcular el centroide del cluster j
9:     end for
10: until se cumpla el criterio de convergencia
```

Uno de los aspectos críticos de la primera versión de *k-means* es su sensibilidad a la elección inicial de centroides. Dado que los centroides iniciales se seleccionan aleatoriamente, el algoritmo puede converger a diferentes soluciones en ejecuciones diferentes, lo que dificulta la obtención de un resultado consistente y confiable. Además, aunque *k-means* es paralelizable y puede manejar grandes conjuntos de datos, no ofrece garantías de proximidad al óptimo global, lo que significa que puede quedar atrapado en óptimos locales y no alcanzar la solución globalmente óptima.

Otra limitación significativa del *k-means* de Lloyd es su falta de previsibilidad en cuanto a la eficiencia temporal. El número de iteraciones que el algoritmo necesita para converger a una solución puede variar ampliamente según la elección inicial de centroides y la distribución

de los datos. Esto hace que sea difícil estimar de antemano el tiempo necesario para ejecutar el algoritmo en un conjunto de datos dado, lo que puede ser problemático en aplicaciones donde se requiere una respuesta rápida.

A pesar de estas limitaciones, el *k-means* de Lloyd sigue siendo un algoritmo fundamental en el campo de *clustering*, y ha servido como base para numerosas mejoras y extensiones que abordan sus desafíos. En esta sección, exploraremos en detalle el funcionamiento y las características de *k-means*, así como sus variantes y enfoques modernos que buscan superar las limitaciones mencionadas, proporcionando una visión completa de este algoritmo central en el análisis de datos y el *clustering*.

Para abordar el problema de mínimo local, Arthur y Vassilvitskii [2] propusieron *k-means++* (Algoritmo 2), donde los centroides iniciales se seleccionarían de manera informada, cada nuevo centroide seria seleccionado con probabilidad proporcional a la distancia del centroide más cercano. Con esto se busca inicializar el algoritmo con centroides que estén separados entre ellos y evitar caer en una solución sub-óptima. Esta implementación proporciona una aproximación óptima con un factor de $\mathcal{O}(\log k)$, pero limita el algoritmo a ser secuencial.

Algorithm 2 Inicialización de centroides *k-means++*

1: Elegir el número de *clusters* k
2: $\mathcal{C} \leftarrow$ el primer centroide c_1 seleccionado de manera aleatoria entre el conjunto de puntos
3: **for** $i = 2$ hasta k **do**
4: Calcular la distancia $d(x, \mathcal{C})$ para cada punto de datos x a su centroide más cercano
5: $\phi_X(\mathcal{C}) \leftarrow$ Calcular el costo del *clustering*
6: Seleccionar el siguiente centroide c_i con probabilidad $\frac{d^2(x,\mathcal{C})}{\phi_x(\mathcal{C})}$
7: **end for**
8: Realizar *k-means* estándar con los centroides iniciales seleccionados

Bahmani et al Su et al. [2015] propusieron una versión paralela para la selección de centroides iniciales (Algoritmo 3), garantizando una aproximación $\mathcal{O}(\log k)$ del óptimo. El algoritmo elige inicialmente un punto al azar, calcula el costo total del *clustering* ψ, y luego itera $\log \psi$ veces. En cada iteración, elige candidatos a centroides con probabilidad proporcional a la distancia a los centroides ya elegidos y con un factor de sobre muestreo l. Tras algunas iteraciones, se obtiene un conjunto de candidatos a centroides y, mediante *k-means++*, se seleccionan los k centros iniciales para el algoritmo de Lloyd.

Actualmente la versión implementada en la mayoría de las librerías de *Big Data* utilizan la inicialización propuesta en *k-means‖*. Es este trabajo se aprovecha esta esta implementación y del entorno de calculo distribuido creado en Spark.

Algorithm 3 Inicialización de centroides *k-means*‖ (k,l)

1: $\mathcal{C} \leftarrow$ seleccionar el primer centroide uniformemente del conjunto de puntos
2: $\psi \leftarrow \phi_X(\mathcal{C})$
3: **for** $O(\log \psi)$ veces **do**
4: $\quad \mathcal{C}' \leftarrow$ muestreo de cada $x \in X$ independientemente con probabilidad $p_x = \frac{l \cdot d^2(x,\mathcal{C})}{\phi_x(\mathcal{C})}$
5: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$
6: **end for**
7: Para $x \in \mathcal{C}$, establecer w_x como el número de puntos en X más cercanos a x que a cualquier otro punto en $\mathcal{C}$.
8: Reagrupar los puntos ponderados en $\mathcal{C}$ en k *clusters*.

2.4. Cálculo distribuido

Escalabilidad

En la era de la información actual, los entornos de cálculo distribuido han emergido como una solución vital para manejar, procesar y analizar volúmenes masivos de datos. Estos entornos se caracterizan por la capacidad de distribuir tareas de procesamiento de datos a través de múltiples computadoras interconectadas, conocidas como nodos, en una red. Esta distribución permite que los datos se procesen en paralelo, mejorando significativamente la velocidad en comparación con los sistemas de procesamiento centralizado.

Los entornos de cálculo distribuido son esenciales en situaciones donde los conjuntos de datos son demasiado grandes para ser procesados eficientemente por una sola máquina. Estos entornos son capaces de dividir grandes conjuntos de datos en fragmentos más pequeños, distribuyéndolos a través de múltiples nodos para su procesamiento simultáneo. Además, proporcionan redundancia y tolerancia a fallos, ya que la falla de un nodo individual generalmente no afecta el proceso de cálculo en su conjunto.

La escalabilidad es otra característica clave de los entornos de cálculo distribuido. Conforme aumentan las demandas de procesamiento, se pueden añadir más nodos a la red para mejorar la capacidad de procesamiento. Esto los hace particularmente adecuados para aplicaciones de *big data* y computación en la nube, donde la flexibilidad y la capacidad de manejar cargas de trabajo dinámicas son esenciales.

Spark y análisis de datos

Apache Spark se ha establecido como uno de los *frameworks* más destacados en el análisis de datos dentro de entornos de cálculo distribuido. Spark ofrece varias características y ventajas que lo hacen ideal para el procesamiento de grandes volúmenes de datos:

- **Rendimiento Optimizado:** Spark está diseñado para ser rápido en el procesamiento de grandes conjuntos de datos. Utiliza una avanzada tecnología de procesamiento en

memoria (*in-memory processing*), lo que significa que puede procesar datos directamente en la memoria RAM de los nodos del *cluster*, reduciendo así la latencia asociada con las operaciones de lectura y escritura en el disco duro.

- **Facilidad de Uso:** Spark proporciona APIs de alto nivel en lenguajes de programación como Python, Java, Scala y R, lo que lo hace accesible para una amplia gama de usuarios, incluyendo ingenieros de datos, científicos de datos y analistas.
- **Capacidad para Manejar Diversos Tipos de Datos:** Spark es versátil en el manejo de diferentes formatos de datos, incluyendo datos estructurados, semi-estructurados y no estructurados. Esto lo hace adecuado para una variedad de aplicaciones de análisis de datos, desde procesamiento de textos hasta análisis de grandes bases de datos.
- **Ecosistema Rico:** Spark se complementa con un conjunto de bibliotecas poderosas, como Spark SQL para consultas de datos, MLlib para machine learning, GraphX para procesamiento de gráficos y Spark Streaming para el procesamiento de datos en tiempo real.
- **Escalabilidad y Tolerancia a Fallos:** Spark está diseñado para escalar de manera eficiente desde un pequeño número de nodos hasta miles de ellos. Además, proporciona una robusta tolerancia a fallos a través de su modelo de ejecución distribuida.

En resumen, Apache Spark se destaca en el ecosistema de cálculo distribuido por su rendimiento, facilidad de uso y capacidad para manejar una amplia gama de aplicaciones de análisis de datos. Su combinación única de velocidad, flexibilidad y un rico conjunto de funcionalidades lo convierte en una herramienta esencial para cualquier organización que busque obtener *insights* valiosos a partir de grandes volúmenes de datos.

Capítulo 3

Estado del arte

3.1. *k-means* diferencialmente privado

En el análisis del estado del arte de *k-means* diferencialmente privado, se distinguen dos enfoques principales: interactivos y no interactivos, ambos empleando privacidad diferencial.

El enfoque interactivo modifica el proceso iterativo del *k-means* añadiendo ruido a las cuentas de puntos y a las sumas de las coordenadas de los puntos en cada *cluster*. Esto da lugar a la obtención de nuevos centroides que son diferencialmente privados. Por otro lado, el enfoque no interactivo se basa en el Teorema 2.9 de post-procesamiento de la privacidad diferencial, asegurando que cualquier manipulación de datos derivados de un proceso diferencialmente privado mantenga la privacidad. Ambas estrategias se han explorado en diversos trabajos, reflejando diferentes métodos para equilibrar la precisión del agrupamiento con la garantía de privacidad.

3.2. Enfoque interactivo

Los enfoques interactivos en *k-means* privado implican la interacción directa con los datos, aplicando mecanismos de privacidad diferencial durante el proceso de agrupación.

En el trabajo de Su et al. [2016a] se propone una variante privada de *k-means* clásico, llamada *DPLloyd*, integrando la privacidad diferencial durante la formación de los *clusters*. Este enfoque busca equilibrar la precisión del *clustering* con la garantía de privacidad, pero enfrenta desafíos en términos de eficiencia computacional y calidad de los *clusters* en datos de alta dimensión o bajo restricciones estrictas de privacidad, es decir, con un presupuesto de privacidad reducido. Por otro lado, Ghazi et al. [2020] examina cómo se pueden obtener ratios de aproximación ajustados para *k-means* bajo privacidad diferencial, sugiriendo un posible equilibrio entre privacidad y precisión de *clustering*. No obstante, el enfoque puede ser computacionalmente intensivo, especialmente con conjuntos de datos grandes o de muchas dimensiones. Finalmente, Su et al. [2017] presenta un enfoque híbrido que une *k-means* con

optimización privada. Este trabajo propone una técnica que mejora la eficiencia del *clustering* manteniendo la privacidad de los datos, aunque puede resultar compleja en escenarios con grandes volúmenes de datos o alta diversidad.

3.3. Enfoque no interactivo

En el enfoque no interactivo de *k-means*, el proceso comienza con la etapa de preprocesamiento diferencialmente privado de los datos, que posteriormente son procesados por el algoritmo de *clustering* elegido. Gracias al Teorema 2.9 de postprocesamiento el resultado obtenido es diferencialmente privado. Dentro del acercamiento no interactivo, se distinguen principalmente dos métodos para la partición del espacio: el uso de **grillas** y de **árboles**. Estas dos acercamientos proporcionan distintas estrategias para organizar y segmentar el espacio de datos, cada una con sus ventajas específicas en términos de eficiencia computacional y precisión en la formación de *clusters*.

División con grillas

En el campo de la privacidad diferencial aplicada a datos geoespaciales en el trabajo de Qardaji et al. [2013] se ha destacado un enfoque que propone un método de grilla adaptativa. Este método coloca una cuadrícula sobre el conjunto de datos y luego divide cada celda según su conteo ruidoso, aprovechando la necesidad de tener una partición de granularidad más fina en regiones densas y, simultáneamente, una partición más gruesa en regiones dispersas. Por otro lado, Su et al. [2016b] proponen **EUGKm**, donde dividen el espacio en un nivel con una grilla uniforme, la granularidad de la división se basa en un análisis estadístico del conjunto de datos. Este análisis logra una división efectiva del espacio de datos, el número de divisiones se calcula como $M = \left(\frac{N\varepsilon}{\theta}\right)^{\frac{2d}{2+d}}$, donde N es la cantidad de puntos, d la dimensión del conjunto de datos, $\theta = 10$ un parámetro seleccionado experimentalmente que ha mostrado ser eficiente en la mayoría de los conjuntos de datos. Es importante destacar que nuestro estudio considera el ruido para el conteo de puntos diferencialmente privado a diferencia del trabajo original de Su et al. [2016b].

División con árboles

Existen amplias investigaciones sobre arboles que preservan la privacidad de sus datos bajo consultas, tal como se refleja en la bibliografía [9, 10, 11, 12, 13]. Estas investigaciones toman métodos tradicionales de indexación espacial, como los *quadtrees* y *k-d trees*[1] , y los

[1]Los **quadtrees** y **kd-trees** son estructuras de datos utilizadas para organizar el espacio en varias dimensiones. Un *quadtree* es una estructura de árbol en la que cada nodo tiene exactamente cuatro hijos. Se utiliza comúnmente en dos dimensiones para dividir un espacio en regiones más pequeñas y eficientes para procesos como la renderización de imágenes y la detección de colisiones. Por otro lado, un *kd-tree* divide el espacio alternando cortes a lo largo de diferentes dimensiones en cada nivel del árbol, creando regiones más pequeñas y manejables para búsquedas eficientes.

adaptan para proporcionar una descripción privada de la distribución de los datos. Cabe destacar que si la creación de estas estructuras es dependiente de los datos, también es necesario privatizar este proceso. Por ejemplo, para el *k-d tree*, se debe calcular la mediana de una manera diferencialmente privada. Una consulta a este tipo de estructura funcionaría de manera similar a sus versiones no privadas. Es decir, descendería a través de los nodos del árbol para encontrar aquellos que contienen la respuesta, pero antes unir el resultados de los nodos, se perturbaría con ruido Laplaciano para asegurar la privacidad de este. Un ejemplo de esto se puede observar en la Figura 3.1, donde se muestra un *quadtree*, el nodo a corresponde a todo el espacio, los nodos b, c, d, y e, corresponden a la primera división del espacio en cuatro cuadrantes y así sucesivamente. Los nodos en rojo son utilizados para responder una consulta Q, en cada nodo hay dos valores: el valor encuadrado corresponde al conteo ruidoso del nodo, y el otro al valor real.

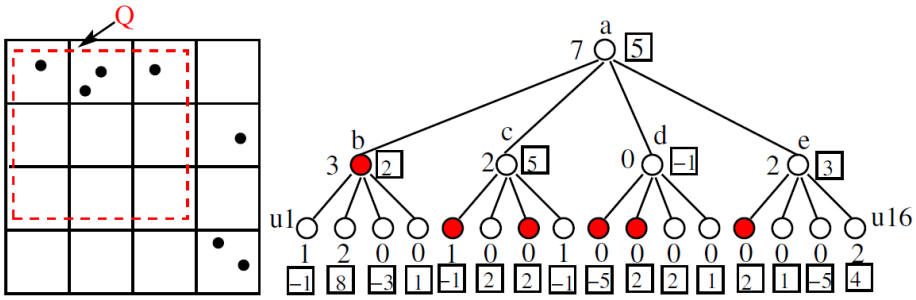


Figura 3.1: Consulta Q en un *quadtree*.

Estos trabajos son de gran relevancia para nuestro estudio, ya que, considerando una estructura como las mencionadas, se podrían utilizar las hojas (o nodos terminales) como la sinopsis diferencialmente privada, que serviría de entrada para nuestro algoritmo de *clustering*. De este modo, podríamos beneficiarnos de los avances en precisión y garantías de privacidad que ofrecen estas estructuras.

Es crucial reconocer que las estructuras indexadas se diseñan para minimizar errores en consultas de bases de datos, a diferencia de nuestro enfoque que busca crear una sinopsis de datos para su posterior publicación y uso con *k-means*. Por lo tanto, aunque nos inspiramos en cómo se construyen estas estructuras, nuestros algoritmos las utilizan de manera muy diferente. Tomaremos las hojas del árbol construido como entrada para el algoritmo *k-means*.

Asignación de presupuesto

En el ámbito de la asignación de presupuestos de privacidad diferencial, se han propuesto varias estrategias innovadoras. Un enfoque propuesto por Wang et al. [2016] incluye la utilización de la sucesión de Fibonacci para la asignación de presupuestos en *quadtrees*, lo cual representa un método interesante en la gestión de datos geoespaciales. Otro trabajo relevante fue el de Yan et al. [2020], donde propone un método aritmético eficiente para la asignación del presupuesto de privacidad diferencial. Este método se enfoca específicamente en la partición y publicación de información de ubicación geográfica utilizando también *quadtrees*, logrando un alto nivel de precisión comparado a otras estrategias.

Capítulo 4

Solución

En este estudio adoptamos un enfoque no interactivo para la creación de sinopsis privadas. Se considera que existe un potencial significativo para mejorar los resultados del estado del arte, en particular considerando extensiones naturales de las técnicas reportadas en la literatura reciente.

Para crear una sinopsis privada, inicialmente se divide el dominio del conjunto de datos en una partición (Definición 4.1). Hacemos la suposición que aglutinar los puntos en el centro de cada partición proporciona una representación lo suficientemente buena del conjunto de datos. En consecuencia, la sinopsis se construye a partir de dos elementos fundamentales: primero, las coordenadas que identifican el centro de cada elemento de la partición, y segundo, un conteo alterado por un ruido Laplaciano. Este último se incorpora para asegurar la privacidad de los datos, evitando que se infiera información sensible de los individuos a partir del conteo en cada partición. A continuación la Figura 4.1 muestra a grandes rasgos la creación de una sinopsis privada, donde en el último paso se agrega el ruido Laplaciano.

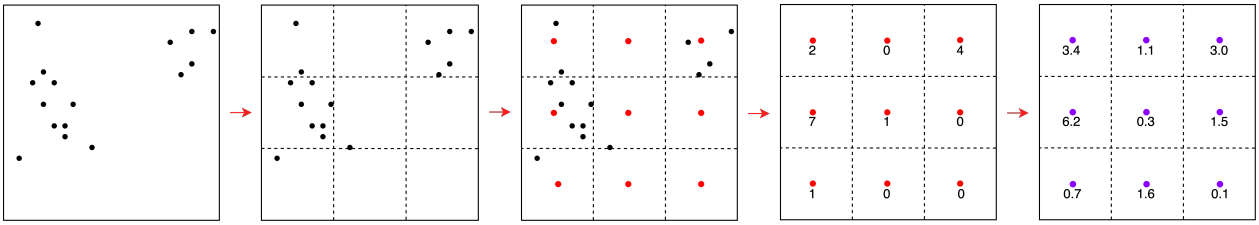


Figura 4.1: Creación de una sinopsis privada.

Definición 4.1 (Partición) *Una partición del conjunto A es un conjunto de uno o más subconjuntos no vacíos de A: $A_1, A_2, A_3, \ldots$, de tal manera que cada elemento de A está en exactamente un conjunto. Simbólicamente,*

(a) $A_1 \cup A_2 \cup A_3 \cup \ldots = A$

(b) Si $i \neq j$ *entonces* $A_i \cap A_j = \emptyset$

Para garantizar la preservación de la privacidad de los datos, también resulta esencial crear la partición de manera diferencialmente privada. Esto implica que, en cada etapa en

la que se requiera el uso de información del conjunto de datos para particionar o dividir el espacio, se debe mantener la privacidad de dichos datos.

La estrategia utilizada para crear la partición del conjunto de datos divide recursivamente los elementos de las particiones (particionamiento recursivo). En nuestra solución, empleamos tres componentes fundamentales: puntos, bloques y un particionador. Los puntos constituyen el conjunto de datos que estamos analizando. Por otro lado, los bloques simbolizan subespacios dentro del espacio total que alberga al conjunto de datos. Finalmente, el particionador es un algoritmo diseñado para generar una partición a partir de un bloque específico y los puntos que este contiene.

La forma en que funciona el particionador es relativamente sencilla. Este recibe como entrada un bloque junto con los puntos que se encuentran en su interior, y como resultado, entrega una lista de nuevos sub-bloques, cada uno con los puntos correspondientes alojados en su interior. El procedimiento comienza con la partición del espacio completo y se procede de manera recursiva en cada sub-bloque de la partición. Este proceso continúa hasta alcanzar un criterio preestablecido, ya sea una profundidad máxima o un umbral mínimo de puntos dentro de los bloques. Esto ofrece ventajas significativas al momento de controlar de manera eficiente el equilibrio entre precisión y privacidad.

Las estrategias para dividir el espacio pueden clasificarse en dos categorías: las que dependen de los datos y las que no. Entre las primeras, encontramos los particionadores como el mediano, el uniforme y el multi-cuantil. Por otro lado, el método que no depende de los datos es el binario. Estas estrategias se detallarán en la sección siguiente.

Las innovaciones propuestas en este trabajo incluyen:

- La creación de sinopsis privadas eficientes, adaptándose a la naturaleza de los datos a tratar. En base a una estrategia recursiva para la partición del espacio, la cual se fundamenta en el uso de particionadores: binario, mediano, uniforme y multi-cuantiles.
- Una estrategia de distribución de presupuesto para la creación de sinopsis privadas. Se basa en el problema de minimización del error de todas las consultas de conteo realizadas en la construcción de la sinopsis privada.

Para comparar nuestros resultados con otros trabajos en el área, la solución explora cómo se divide el espacio en cada nivel, el umbral para detener el paso recursivo, la cantidad de presupuesto de privacidad y las estrategias para distribuir este presupuesto. Así, podemos comparar nuestros resultados con otros trabajos en el área.

Las tecnologías que utilizaremos son Scala, Spark y Docker. Scala es un lenguaje altamente optimizado, con velocidades hasta 10 veces más rápidas que Python en ciertos casos, incluyendo el procesamiento y análisis de datos a gran escala, lo cual es crucial en este contexto. Spark, con su API en Scala, facilita el uso de bibliotecas de *Machine Learning* (ML) que contienen implementaciones de *k-means*, además de proporcionar un entorno para cálculos distribuidos, como los *Resilient Distributed Dataset* (RDD) o los más recientes *DataFrames* (DF). Docker permite la fácil configuración de un entorno de ejecución para Spark, permitiéndonos mover la ejecución de los experimentos de forma local a remota en servidores.

Para la realización de los experimentos, se empleó un contenedor de Docker. Este contenedor estaba configurado con un nodo maestro dotado de 3 GB de memoria RAM y tres nodos trabajadores, cada uno provisto de 4 GB de memoria RAM. Se maximizó la cantidad de nodos virtuales disponibles, alcanzando un total de cuatro. Las imágenes utilizadas en Docker fueron las versiones pre-elaboradas proporcionadas por Bitnami.

En la esta sección, abordaremos con detalle nuestra estrategia de particionamiento recursivo, un componente clave para la división espacial eficaz. Exploraremos en profundidad el funcionamiento de los distintos particionadores por nivel, un aspecto vital de nuestra solución. Luego, profundizaremos en la asignación del presupuesto de privacidad a lo largo de las diferentes fases de nuestro algoritmo. Finalizaremos con una discusión sobre la implementación de los algoritmos, centrándonos en los aspectos técnicos más relevantes.

4.1. Estrategia de particionamiento recursivo

El método de particionamiento recursivo se implementa de esta forma: Iniciamos con un conjunto de puntos, denotado como P, y un bloque B que los incluye, en conjunto con un particionador. En su primera intervención, el particionador divide el bloque y los puntos asociados en subconjuntos (P_i, B_i), donde i varía entre 1 y M y P_i es un conjunto de puntos. Aquí, M representa el número total de divisiones generadas. Cada uno de estos nuevos bloques, con sus respectivos puntos, se emplea como entrada para la fase subsiguiente del algoritmo. Este procedimiento se repite hasta que se alcance una condición de terminación predefinida, que puede ser un número concreto de puntos o una profundidad máxima dentro del algoritmo. Es fundamental enfatizar que todas las decisiones relacionadas con los puntos deben asegurar la privacidad diferencial, por lo tanto el particionador también recibirá un presupuesto de privacidad. Estos cálculos abarcan el conteo de datos para el umbral de parada y la manera en que el particionador segmenta el bloque. Se detallara más en la sección 4.2, específicamente dedicada a los particionadores. Adicionalmente, se proporciona el pseudocódigo en el Algoritmo 4, así como un resumen de la notación en la Tabla 4.1.

Este enfoque recursivo ofrece numerosas ventajas, por un lado su capacidad para segmentar distintas problemáticas. Por ejemplo, aborda cuestiones como el particionamiento óptimo de un nivel y el momento idóneo para detener la iteración recursiva. Su diseño se ajusta con eficacia a las fluctuaciones en la densidad de puntos. Este mecanismo asegura un balance más refinado entre la precisión y la privacidad.

Símbolo	**Descripción**
d	Dimensiones en conjunto de datos.
point	Punto ponderado en $\mathbb{R}^d$.
points	Lista de point.
block	Entidad rectangular en espacio $\mathbb{R}^d$.
$\|P\|$	Cantidad de puntos en un bloque.
threshold	Umbral de parada en algoritmo recursivo.
ε_t	Presupuesto de privacidad para conteo en condición de parada.
ε_{pb}	Presupuesto de privacidad para publicación de datos.
ε_{pt}	Presupuesto de privacidad para partición de datos.
prt	Mecanismo de particionamiento en algoritmo recursivo.
current_depth	Profundidad actual en algoritmo recursivo.
max_depth	Profundidad máxima en algoritmo recursivo.
AD	Divisiones por eje en particionador uniforme.
M	Número de divisiones en particionador uniforme.
m	Número de cuantiles en particionador multi-cuantil.

Tabla 4.1: Resumen de notaciones.

Sin embargo, este enfoque también implica ciertos desafíos. El principal reto es determinar el punto de detención del algoritmo. La mayoría de los trabajos actuales sugieren un umbral fijo, complementado con una búsqueda en grilla para evaluar su eficacia. Otra alternativa es establecer un umbral en función del número de puntos en el conjunto de datos. Esta

opción tiene dos inconvenientes: primero, se debe asignar una parte del presupuesto, aunque sea pequeña, para contar la cantidad de puntos; segundo, incluso si se usa un porcentaje, se debe analizar qué factores optimizan los resultados. Por ejemplo, se podría tener una función umbral como $T(P) = c * (|P| + ruido)$, donde c es una constante a calibrar. El segundo desafío surge de la complejidad computacional de dividir los espacios, dependiente de los particionadores y las dimensiones del conjunto de datos. Un análisis más detallado se presentará en la siguiente sección 4.2. La cantidad de divisiones que pueden generarse aumenta exponencialmente con cada nivel de recursión, lo que hace esencial una cuidadosa consideración de los criterios de parada y los umbrales teóricos del número de divisiones posibles.

Algorithm 4 Partición Recursiva

```
1: function RecursivePartition
2:     Input: block, points, threshold, partitioner, current_depth, max_depth, privacy budget for partition ε_pt, threshold ε_t and publishing ε_pb
3:     if current_depth ≥ max_depth or points.empty then
4:         return Point(block.center, points.noisyCount(ε_pb))
5:     else
6:         if points.noisyCount(ε_t) ≤ threshold then
7:             ε_extra ← Σ_{i>depth} ε_t^i + Σ_{i>depth} ε_pt^i                ▷ ε extra*
8:             return Point(block.center, points.noisyCount(ε_pb + ε_extra))
9:         else
10:            partition ← partitioner(block, points, ε_pt^{current_depth}, current_depth)
11:            return partition.map(RecursivePartition(..., current_depth+1))
12:        end if
13:    end if
14: end function
```

ε **extra***: en el caso que en paso recursivo se detenga debido a el umbral, se reutiliza el presupuesto que estaba destinado para los siguientes niveles de profundidad en la publicación de los datos y conteo de puntos para la condición de parada.

En nuestro algoritmo, el presupuesto de privacidad se utiliza en tres instancias clave:

1. **Publicación** de los datos. Cuando se cumple una condición de parada y los datos deben liberarse para su uso por el algoritmo de *clustering*.
2. **Umbral** de parada. En el conteo de puntos dentro de cada sub-bloque. En cada iteración, es necesario contar los puntos para decidir si se continúa con el paso recursivo, y este conteo requiere privacidad diferencial.
3. **Partición** del espacio. Los particionadores uniforme, mediano y multi-cuantil necesitan presupuesto de privacidad para realizar una partición espacial. El particionador binario es la excepción, ya que en cada iteración divide una dimensión del bloque por la mitad.

La división del presupuesto en tres áreas —publicación, partición y umbral— así como su distribución a lo largo de las iteraciones, representa un desafío interesante. Este problema

ha sido estudiado en la literatura, y en este trabajo proponemos un cálculo que extiende los resultados actuales para algoritmos de particionamiento con un número fijo de divisiones por iteración, por ejemplo los particionadores binario y mediano tienen una cantidad fija de dos elementos por partición.

Finalmente, los particionadores propuestos en este trabajo tienen ventajas y desventajas dependiendo de las distribuciones en los conjuntos de datos. En algunos casos, puede ser más conveniente usar una grilla uniforme en lugar de un particionador binario, o viceversa. Por lo tanto, exploraremos múltiples conjuntos de datos para evaluar y comparar los resultados obtenidos.

4.2. Particionamiento por nivel

En esta sección, se llevará a cabo un análisis detallado de diversos algoritmos de particionamiento por nivel, enfocándose en sus fundamentos teóricos, diseño estructural, gestión del presupuesto de privacidad diferencial, y su análisis general de su complejidad.

4.2.1. Partición binaria

La partición binaria representa una extensión de los *quadtrees* a dimensiones múltiples (d dimensiones). Un *quadtree* es una estructura de datos en árbol utilizada para particionar un espacio bidimensional dividiéndolo en cuatro cuadrantes o regiones. A pesar de su potencial, no se utiliza con frecuencia debido al aumento exponencial en el número de nodos resultante. No obstante, se puede optimizar su aplicación en contextos de alta dimensionalidad mediante un control más refinado de la condición de parada, lo que ayuda a prevenir divisiones innecesarias y permite una adaptación más eficaz de este método. Un ejemplo de tres iteraciones utilizando esta estrategia se ilustra en la Figura 4.2. También se muestra la comparación entre el conjunto S1 y la sinopsis privada creada por el particionador binario en la Figura 4.3.

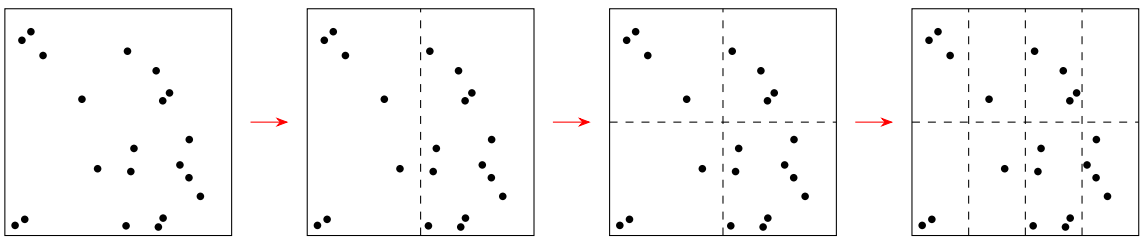
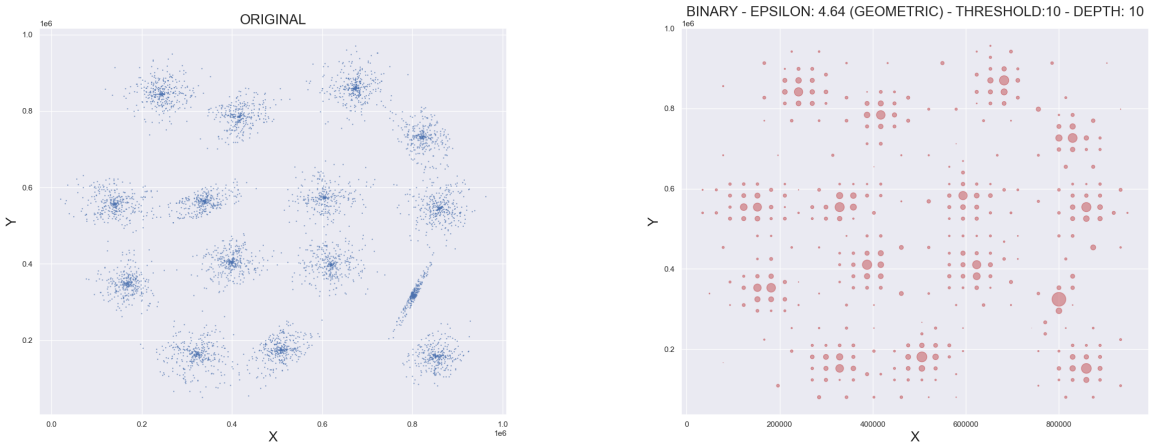


Figura 4.2: Ejemplo particionador binario - tres iteraciones.



(a) S1 original

(b) Particionador binario.

Figura 4.3: Ejemplo de sinopsis privada con particionador binario para S1 .

Diseño de la partición

La implementación de la partición binaria sigue un enfoque similar al del *k-d tree*[1]. En cada iteración se elige consecutivamente uno de los d ejes disponibles. El bloque se divide entonces en **dos mitades iguales** a lo largo de este eje, repitiendo este proceso y alternando entre los diferentes ejes en cada nuevo sub-bloque generado. La partición binaria no depende de los datos para determinar su punto de división, para esto es necesario que los rangos de los datos sean proporcionados por el usuario como entrada.

Esta metodología iterativa garantiza que todas las dimensiones sean consideradas equitativamente, bisecando el espacio sucesivamente a lo largo de un eje seleccionado. Este enfoque genera una estructura de datos jerárquica, optimizando la organización y el acceso a datos en espacios de alta dimensionalidad. Es especialmente útil para la organización espacial de datos geográficos, búsquedas rápidas en bases de datos multidimensionales y procesamiento eficaz de consultas en grandes volúmenes de datos.

El pseudocódigo de la partición binaria se puede encontrar en el Algoritmo 5.

Presupuesto de privacidad

La partición binaria es el único método que no depende de los datos para determinar su punto de división. Es relevante señalar que, para cumplir con la privacidad diferencial, los rangos de los datos deben ser proporcionados por el usuario como entrada.

[1]En un *k-d tree*, se alterna la dimensión de división (como x, y en 2D) en cada nivel del árbol, dividiendo el espacio en regiones más pequeñas. Luego se escoge un punto, usualmente el mediano en la dimensión actual, para dividir el espacio en dos, ayudando a mantener el árbol balanceado.

Algorithm 5 Partición Binaria

```
1: function PARTITION
2:     Input: block, points, privacy bugdet for partitioning ε_pt, current_depth
3:     axis ← depth mod points.head.dim        ▷ Gets axis to divide based on current depth
4:     divider ← block[axis].middle                ▷ Midpoint of the axis boundaries
5:     blockL ← block.updateUpperBound(axis, divider)                    ▷ New block
6:     blockR ← block.updateLowerBound(axis, divider)                    ▷ New block
7:     pointsL ← points.filter( point(axis) ≤ divider)                   ▷ New list of points
8:     pointsR ← points.filter( point(axis) > divider)                   ▷ New list of points
9:     return [(blockL, pointsL), (blockR, pointsR)]      ▷ type : List[(Block, List[Point])]
10: end function
```

Análisis

La complejidad de este algoritmo por nivel no es significativa por bloque individual de división, sino por la cantidad de bloques que se generan en el nivel anterior, lo que resulta en un crecimiento exponencial en cada nivel sucesivo. Sin embargo, los *quadtrees* siguen siendo populares debido a su simplicidad y eficiencia en la división espacial. Se espera que estas características sean igualmente beneficiosas en aplicaciones multidimensionales, especialmente con una gestión efectiva de las condiciones de parada para asegurar la calidad del análisis.

4.2.2. Partición mediana

Este método de particionamiento se deriva de los *k-d trees*, una estructura de datos utilizada para organizar puntos en un espacio k-dimensional. Podemos ver un ejemplo de un particionador mediano (diferencialmente privado) en dos dimensiones en la Figura 4.4. También se muestra la comparación entre el conjunto S1 y la sinopsis privada creada por el particionador mediano en la Figura 4.5.

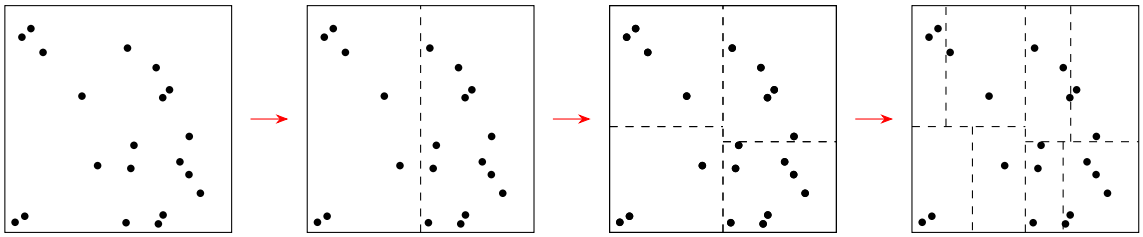
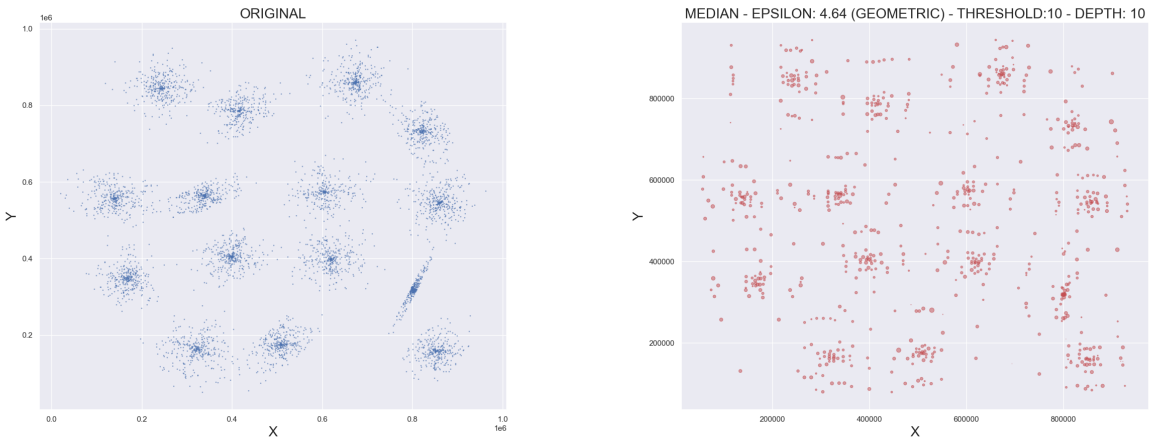


Figura 4.4: Ejemplo particionador mediano - tres iteraciones.



(a) S1 original

(b) Particionador mediano.

Figura 4.5: Ejemplo de sinopsis privada con particionador mediano para S1 .

Diseño de la partición

La construcción de un *k-d tree* comienza en el nodo raíz, que abarca todo el espacio. Se selecciona una dimensión para la partición y un valor de división —normalmente la mediana del rango actual en esa dimensión— para dividir el bloque en sub-bloques. Este proceso se repite hasta alcanzar un criterio predefinido, como una altura máxima del árbol o un número máximo de puntos de datos por partición.

En nuestra implementación se calcula la mediana utilizando un mecanismo exponencial. Para este caso consideremos la Definición 2.5 junto con una función de utilidad adecuada (Ecuación 4.1).

Notemos que se prefiere un resultado que sea cercano al cuantil real. Sea x un conjunto de datos (ordenado), sea r un posible resultado y sea N el tamaño del conjunto de datos. Considere $\#(Z > r)$ como el número de puntos en x por encima de r. Entonces, la siguiente es una función de utilidad razonable para un resultado r para el cuantil α de x:

$$u(x, r) = \text{máx}(\alpha, (1 - \alpha))N - |(1 - \alpha)\#(Z < r) - \alpha\#(Z > r)|. \tag{4.1}$$

Lema 4.2 *La función de utilidad u mencionada anteriormente tiene una sensibilidad* ℓ_1 *acotada superiormente por* $\text{máx}(1 - \alpha, \alpha)$.

Para calcular la mediana, identificada como el cuantil de orden 0.5, podemos remplazar este valor en las definiciones y lemas ya mencionados, resultando en una probabilidad que es proporcional a:

$$u(x, r) = (N - |\#(Z < r) - \#(Z > r)|)/2 \tag{4.2}$$

$$p_x(r) \propto \exp\left(\varepsilon u(x, r)\right) \tag{4.3}$$

Estos resultados son utilizados en la implementación de los Algoritmos 6 y 7.

Algorithm 6 Partición Mediana

```
1: function PARTITION
2:     Input: block, points, privacy budget ε, current_depth
3:     axis ← depth mod points.head.dim      ▷ Gets axis to divide based on current depth
4:     divider ← MedianDP(points, block, ε_pt)                          ▷ Algirithm 7
5:     blockL ← block.updateUpperBound(axis, divider)                   ▷ New block
6:     blockR ← block.updateLowerBound(axis, divider)                   ▷ New block
7:     pointsL ← points.filter( point(axis) ≤ divider)                  ▷ New list of points
8:     pointsR ← points.filter( point(axis) > divider)                  ▷ New list of points
9:     return [(blockL, pointsL), (blockR, pointsR)]      ▷ type : List[(Block, List[Point])]
10: end function
```

Presupuesto de privacidad

En alto nivel el algoritmo de selección de la mediana utiliza el mecanismo exponencial para seleccionar un representante de los posibles intervalos donde se puede encontrar la mediana.

Algorithm 7 Mediana DP

```
1: function MEDIANDP
2:     Input: block, points, privacy budget ε
3:     probDensity ← points.map(p → exp(εu(p, points)))          ▷ u de 4.1
4:     divider ← SAMPLE(points, probDensity)                     ▷
5:     return divider
6: end function
```

Análisis

El cálculo de la mediana diferencialmente privada presenta una complejidad temporal de orden $O(|P| \log |P|)$. Esto se debe al mecanismo exponencial, que requiere puntos ordenados para determinar la función de utilidad necesaria para el muestreo. Este proceso es el más complejo dentro del algoritmo, superando significativamente cualquier otro cálculo, incluida la separación de puntos por bloque.

La implementación de este particionador es particularmente ventajosa para mitigar sesgos en los resultados, especialmente en presencia de *outliers*. Garantiza una distribución homogénea de los puntos en cada sub-bloque, contribuyendo así a la fiabilidad y precisión del análisis de datos.

4.2.3. Partición uniforme

La estrategia de partición uniforme representa el enfoque más directo e intuitivo para enfrentar este problema. En intentos previos, se ha restringido la aplicación a uno o dos niveles de grillas uniformes. Esta limitación nos impulsa a profundizar en el tema mediante nuestro método recursivo. Dicho método nos facilita la iteración a través de múltiples niveles y simplifica notablemente la implementación. . Un ejemplo de una iteración utilizando esta estrategia se ilustra en la Figura 4.6. También se muestra la comparación entre el conjunto S1 y la sinopsis privada creada por el particionador uniforme en la Figura 4.7.

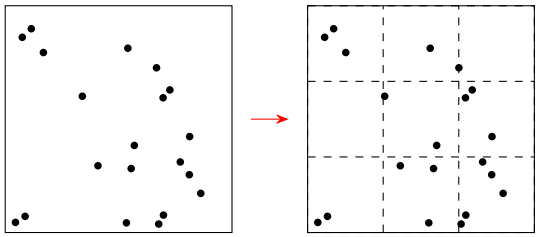


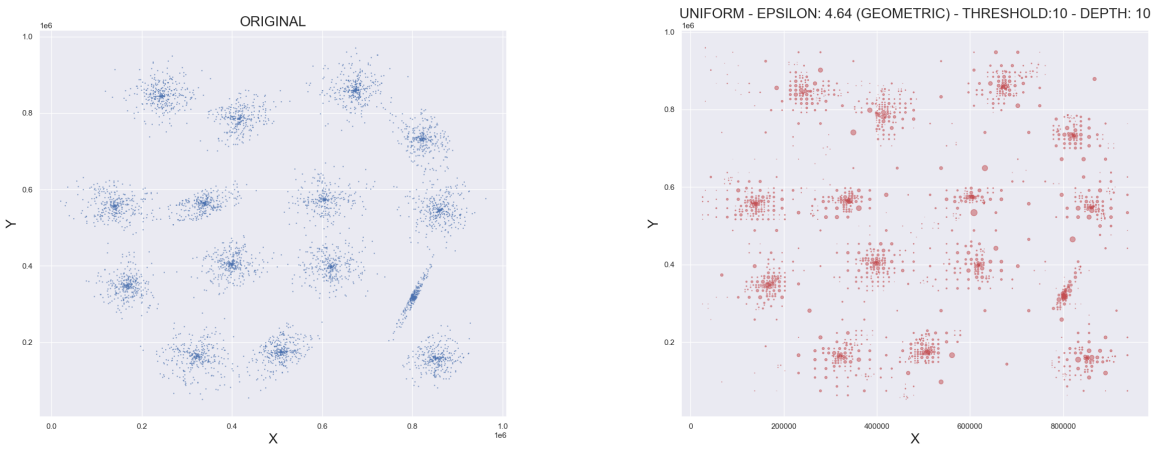
Figura 4.6: Ejemplo particionador uniforme - una iteración.

Diseño de la partición

El dilema central en este contexto es determinar la manera adecuada de dividir cada eje. En investigaciones previas, como en **EUGKm**, la cantidad de bloques en que se debe dividir el espacio se calcula según la ecuación 4.4:

$$M = \left(\frac{|P|\varepsilon}{\theta} \right)^{\frac{2d}{2+d}} \tag{4.4}$$

El valor de $\theta = 10$, tiene un origen experimental y ha demostrado ser eficaz para la mayoría de los conjuntos de datos. Por tanto, la cantidad de divisiones por intervalo se



(a) S1 original

(b) Particionador uniforme.

Figura 4.7: Ejemplo de sinopsis privada con particionador uniforme para S1 .

puede determinar como $AD = M^{1/d}$. En nuestra implementación, se prefiere utilizar $AD = \lfloor \text{mín}\,(M^{\frac{1}{d}}, 10^{\frac{6}{d}}) \rfloor$ para limitar el número de divisiones que el algoritmo debe manejar.

Podemos encontrar el pseudocódigo de la partición uniforme en el Algoritmo 8.

Algorithm 8 Partición Uniforme

```
1:  function PARTITION
2:      Input: block, points, privacy budget for partitioning ε_pt, current_depth
3:      dim ← points.dimension
4:      M ← PARTITIONSIZE(points, dim, ε_pt)                    ▷ Equation 4.4
5:      AD ← ⌊M^(1/d)⌋          ▷ Calculate the division for each axis to have M blocks
6:      if AD == 1 then
7:          return [(block, points)]          ▷ type : List[(Block, List[Point])]
8:      else
9:          blocks = DIVIDE(block, AD)
10:         blocksWithPoints = POINTS2BLOCK(blocks, points)
11:         return blocksWithPoints           ▷ type : List[(Block, List[Point])]
12:     end if
13: end function
```

Presupuesto de privacidad

En relación a la asignación de presupuesto de privacidad, es esencial destinar una fracción para el conteo de puntos en el cálculo de M, tal como se define en la Ecuación 4.4. Cabe destacar que, en la investigación realizada por Su et al. [2015], no se incorporó ruido Laplaciano, lo cual supone una infracción en términos de privacidad diferencial. Aunque las razones de esta omisión no están claras, se hipotetiza que, dada la amplia cantidad de puntos en el conjunto de datos inicial, un porcentaje insignificante del presupuesto de privacidad

podría ser suficiente para obtener una aproximación precisa de la cantidad de puntos. Sin embargo, en nuestro enfoque, dado que abordaremos áreas del espacio con menor densidad de datos, resulta imprescindible la adición de ruido Laplaciano. Por lo tanto, la fórmula se modificaría de la siguiente manera:

$$M = \left(\frac{\varepsilon_{pt}(|P| + \text{Lap}(1/\varepsilon_{pt}))}{\theta} \right)^{\frac{2d}{2+d}} \tag{4.5}$$

$$AD = \lfloor M^{1/d} \rfloor \tag{4.6}$$

Análisis

Tomando en cuenta $|P|$, que representa el número de puntos, y AD^d, que indica la cantidad de bloques creados en un nivel específico de recursión, la complejidad temporal del algoritmo de partición uniforme se establece como $O(|P| + AD^d)$. Al expresar AD en términos de P, esto se convierte en $O(|P| + |P|^{\frac{2d}{2+d}})$. Simplificando, obtenemos $O(|P|^{\frac{2d}{2+d}}) = O(|P|^E)$, donde $E \in [1, 2)$ para conjuntos de datos de al menos dos dimensiones. Es importante destacar que con el incremento de la dimensionalidad, el valor de E se aproxima a 2 y, además, en cada nivel la cantidad promedio de puntos por bloque disminuye de manera inversamente proporcional a la cantidad de divisiones realizadas en el nivel anterior.

Aunque este algoritmo es computacionalmente intensivo (debido al número de divisiones requeridas) su alta precisión en el análisis de datos lo convierte en una herramienta valiosa para nuestra investigación. Este equilibrio entre complejidad y precisión es crucial para el tratamiento eficaz de grandes conjuntos de datos en múltiples dimensiones.

4.2.4. Partición multi-cuantil

La partición multi-cuantil es la extensión natural a una cantidad arbitraria de cuantiles, como por ejemplo, cuartiles, quintiles, deciles o percentiles, en contraste con la mediana. Anteriormente vimos que podíamos dividir el espacio uniformemente, al dividir el espacio entre cuantiles tenemos mas información acerca de la cantidad de puntos que hay en cada sub-bloque. Esto podría ser útil en ciertos contextos. A continuación en la Figura 4.8 podemos ver un ejemplo de un particionador multi-cuantil con 3 cuantiles. También se muestra la comparación entre el conjunto S1 y la sinopsis privada creada por el particionador multi-cuantil en la Figura 4.9.

Diseño de la partición

En el trabajo de Gillenwater et al. [2021] proponen **JointExp** (Algoritmo 10), que permite calcular una cantidad arbitraria (m) de cuantiles para una cantidad fija de presupuesto de

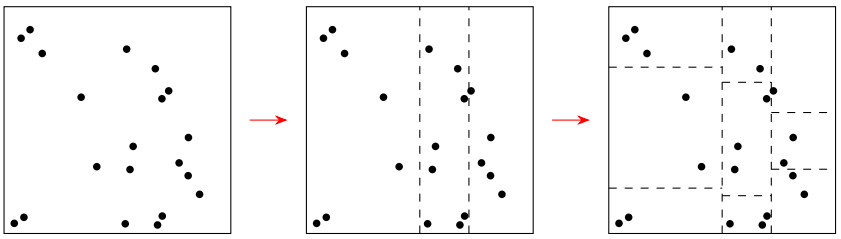
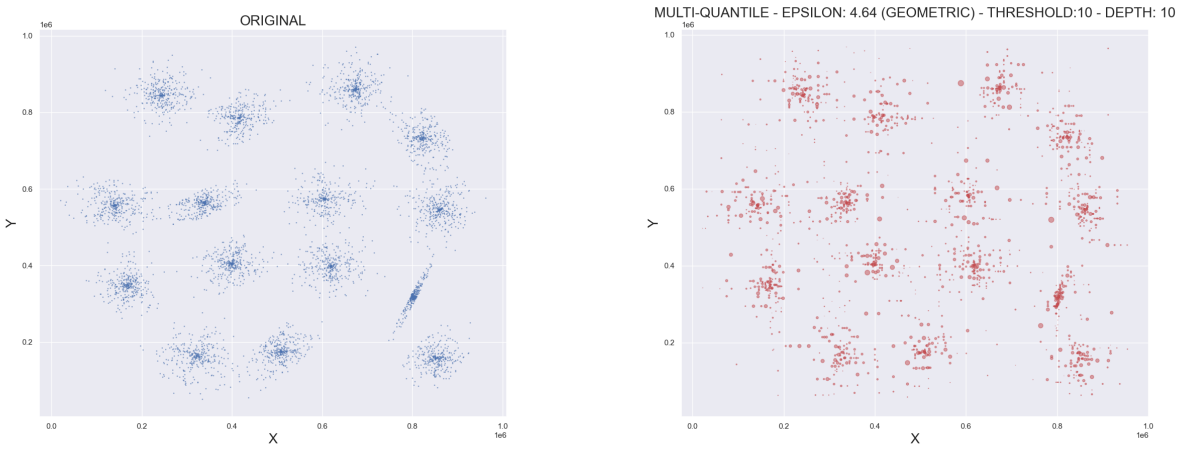


Figura 4.8: Ejemplo particionador multi-cuantil con 3 cuantiles - dos iteraciones.



(a) S1 original

(b) Particionador multi-cuantil.

Figura 4.9: Ejemplo de sinopsis privada con particionador multi-cuantil para S1 .

privacidad. Esto permite dividir el espacio en una cantidad arbitraria de sub-bloques con aproximadamente la misma cantidad de puntos en cada uno de ellos.

Anteriormente para tener m cuantiles se debía desembolsar m veces el presupuesto de privacidad del algoritmo tradicional (Algoritmo 7) para calcular un cuantil.

El algoritmo **JointExp** tiene una complejidad de $O(m|P|\log(|P|)+m^2|P|)$, lo que mejora drásticamente la versión de fuerza bruta en $O(|P|^m)$. Este algoritmo retorna una lista de valores reales que serán los puntos de división para cada dimensión.

Teorema 4.3 ***JointExp*** *cumple con ε-privacidad diferencial, toma un tiempo $O(m|P|\log(|P|)+m^2|P|)$, y utiliza un espacio $O(m^2|P|)$.*

El pseudocódigo para el particionador multi-cuantiles descrito por el Algoritmo 9.

Dada la complejidad del algoritmo **JointExp**, se invita a revisar el trabajo de Gillenwater et al. [2021] para una explicación más detallada.

Algorithm 9 Partición multi-cuantil

function PARTITION
  Input: block, points, privacy budget ε, current_depth
  axis $\leftarrow$ depth mod points.head.dim ▷ Gets axis to divide based on current depth
  dividers $\leftarrow$ JointExp(points, block(axis), quantiles, ε) ▷ Algorithm 10
  intervals $\leftarrow$ [block.lowerBound(axis), dividers, block.upperBound(axis)]
  return split(intervals, block, points) ▷ type : List[(Block, List[Point])]
end function

Algorithm 10 Pseudocódigo de JointExp

function JOINTEXP(X, [a,b], Q, ε)
  Input: Sorted $X = (x_1 \leq \cdots \leq x_n)$ clamped to range $[a, b]$, quantiles $Q = (q_1, \ldots, q_m)$, privacy budget ε
  Set $x_0 = a, x_{n+1} = b$, and $\Delta_{u_Q} = 2$
  Set $I = \{0, ..., n\}, i_0 = 0,$ and $i_{m+1} = n$
  for $i \in I$ **do**
    Set $\alpha(1, i, 1) = \phi(0, i, 1)\tau(i) = \exp\left(-\frac{\varepsilon}{2\Delta_{u_Q}}|i - n_1|\right) \cdot (x_{i+1} - x_i)$
  end for
  for $j = 2, \ldots, m$ **do**
    for $i \in I$ **do**
      Set $\hat{\alpha}(j-1, i) = \sum_{k<j} \alpha(j-1, i, k)$
    end for
    Set $\alpha(j, \cdot, 1) = \tau(\cdot) \times (\phi(\cdot, \cdot, j)^T \cdot \hat{\alpha}(j-1, \cdot)^T)$
    for $k = 2, \ldots, j$ **do**
      for $i \in I$ **do**
        Set $\alpha(j, i, k) = \tau(j)\phi(j, i, j)\alpha(j-1, i, k-1)/k$
      end for
    end for
  end for
  Sample $(i, k) \propto \alpha(m, i, k)\phi(i, n, m+1)$
  Set $i_{m-k+1}, \ldots, i_m = i$, and $j = m - k$
  while $j > 0$ **do**
    Sample $(i, k) \propto \alpha(j, i, k)\phi(i, i_{j+1}, j+1)$
    Set $i_{j-k+1}, \ldots, i_j = i$, and $j = j - k$
  end while
  Output uniform samples $\{o_j \sim_U [x_{i_j}, x_{i_{j+1}})\}_{j=1}^m$ in increasing order
end function

Presupuesto de privacidad

El algoritmo de JointExp utiliza un mecanismo exponencial para seleccionar los cuantiles. Por simplicidad comparte la misma distribución de presupuesto de privacidad que la distribución para el conteo de los puntos. Podría ser beneficioso experimentar a futuro con otras distribuciones, no necesariamente uniforme.

Análisis

La complejidad del algoritmo de partición multi-cuantil se concentra principalmente en el cálculo de los cuantiles, expresado como $O(m|P|\log(|P|) + m^2|P|)$, donde m representa la cantidad de cuantiles. Aunque en nuestros casos m es una constante pequeña, en estudios donde m alcanza valores entre 100 y 1000, su impacto en la complejidad no es despreciable.

Este algoritmo garantiza que aproximadamente un número específico de puntos se distribuya por cada subdivisión. Esta característica es particularmente ventajosa en situaciones con ciertas distribuciones de puntos, ofreciendo una segmentación más equitativa y evitando sesgos en el análisis.

Resumen

La Tabla 4.2 que se presenta a continuación, compendia los ejes comparativos clave de los algoritmos de particionamiento examinados. Esta comparación abarca aspectos como la dependencia de los datos, la complejidad algorítmica y el número de ramificaciones por iteración en cada algoritmo de particionamiento.

Particionador	Dependencia de los datos	Complejidad por iteración	Ramificaciónes
Uniforme	Si - Conteo	$O(\|P\|^{\frac{2d}{2+d}})$	$O(\|P\|^{\frac{2d}{2+d}})$
Binaria	No	$O(\|P\|)$	2
Mediana	Si - Distribución	$O(\|P\|\log\|P\|)$	2
Multi-Cuantil	Si - Distribución	$O(m\|P\|\log(\|P\|) + m^2\|P\|)$	m

Tabla 4.2: Tabla de comparación de particionadores

4.3. Asignación de presupuesto

El dilema de la asignación del presupuesto de privacidad es complejo. Existe un *tradeoff* inherente: al aumentar la privacidad, comprometemos la precisión, y a la inversa, al buscar mayor precisión, la privacidad se ve disminuida. Determinar niveles óptimos de privacidad a lo largo de nuestro algoritmo exige un análisis minucioso y una evaluación precisa de cuándo es apropiado utilizar cada enfoque.

Al abordar la asignación de presupuesto en la partición uniforme, es vital ser cautelosos. Un presupuesto inicial reducido podría llevar a una división excesiva o insuficiente del espacio, influenciando los niveles posteriores del particionamiento. Por lo tanto, es esencial equilibrar estas asignaciones de manera precisa y evaluar su efectividad en función de cada estrategia.

Se debe considerar los presupuestos para las tres partes principales de los algoritmos:

1. ε_{pb}: La **publicación** de datos en las hojas o nodos terminales.
2. ε_{pt}^{i}, $i \in [1..maxdepth]$: La **partición** de bloques en cada nivel.
3. ε_{t}^{i}, $i \in [1..maxdepth]$: El **conteo** ruidoso para la condición de parada en cada nivel.

En este trabajo consideraremos tres tipos de asignación de presupuesto a través de cada iteración recursiva:

1. **Uniforme**: El presupuesto se distribuye uniformemente en cada iteración del algoritmo.
2. **Lineal**: El presupuesto se distribuye siguiendo una progresión lineal en cada iteración del algoritmo.
3. **Geométrico**: El presupuesto se distribuye siguiendo una progresión geométrica en cada iteración del algoritmo.

La primera estrategia de distribución de presupuesto, aunque la más sencilla, no toma en cuenta los diversos objetivos de precisión a lo largo del algoritmo. La segunda estrategia aborda este problema, aunque no garantiza la distribución óptima. En la siguiente sección, demostraremos que la última propuesta surge de la solución a un problema de optimización cuyo objetivo es minimizar el error acumulado por el ruido a lo largo de los niveles del algoritmo.

Además, aplicamos un método sugerido por Shaham et al. [2023] para aumentar la precisión general de la sinopsis. Esta técnica consiste en reasignar el presupuesto inicialmente destinado a los niveles más profundos al proceso de liberación privada de los datos, en caso de que el algoritmo recursivo se detenga antes de alcanzar la máxima profundidad.

4.3.1. Problema de optimización

En el contexto de las estructuras indexadas, las consultas sobre la cantidad de puntos dentro de una región específica se resuelven comúnmente devolviendo el conteo ruidoso de los nodos completamente contenidos en la región consultada. En situaciones donde una región está parcialmente contenida, el proceso se profundiza hacia nodos más específicos hasta que estos estén totalmente contenidos, o bien, bajo una suposición de densidad uniforme, se devuelve una fracción del conteo ruidoso de un nodo, ver Figura 3.1.

El reto de minimizar el error inducido por consultas de conteo en estructuras de árboles diferencialmente privadas no se aplica directamente a este escenario.

En nuestro contexto, la meta es generar una sinopsis con decisiones tomadas de la manera más precisa posible durante su creación. Esto implica, por ejemplo, asignar una cantidad adecuada de presupuesto para el conteo de puntos que se utilizará en la condición de parada, así como destinar la cantidad correcta de presupuesto a los algoritmos de particionamiento para obtener resultados con la máxima precisión. Por lo tanto, consideramos este tipo de consultas en nuestro problema de optimización.

Considerando un algoritmo recursivo con un particionador que en cada etapa ofrece una división recursiva del bloque en F sub-bloques y una profundidad máxima de h, para la profundidad i, se realizarán a lo sumo F^i consultas de conteo. Dado que en cada consulta de conteo se introduce un error relacionado con la distribución de Laplace, y considerando que esta distribución no tiene sesgo pero sí varianza, podemos establecer que $Err(Q_j^i) = Var(Lap(e_i)) = 2/\varepsilon_i^2$. Así, el error total acumulado por las consultas durante la construcción es:

$$Err = \sum_{i=0}^{h} Err(Q^i) = \sum_{i=0}^{h} \sum_{j=1}^{F^i} Err(Q_j^i) = \sum_{i=0}^{h} \sum_{j=1}^{F^i} \frac{2}{\varepsilon_i^2} = \sum_{i=0}^{h} \frac{2}{\varepsilon_i^2} \sum_{j=1}^{F^i} 1 = \sum_{i=0}^{h} \frac{2}{\varepsilon_i^2} F^i \tag{4.7}$$

Esta formulación conduce al siguiente problema de optimización:

$$\begin{aligned} \min_{\varepsilon_i} \quad & \sum_{i=0}^{h} \frac{F^i}{\varepsilon_i^2} \\ \text{sujeto a} \quad & \sum_{i=0}^{l} \varepsilon_i = \varepsilon, \quad \varepsilon_i > 0 \quad \forall i = 0..h \end{aligned} \tag{4.8}$$

Para obtener la solución a través de las condiciones de Karush-Kuhn-Tucker (KKT):

$$L(\varepsilon_1, \ldots, \varepsilon_h, \lambda) = \sum_{i=0}^{h} \frac{F^i}{\varepsilon_i^2} + \lambda \left(\sum_{i=0}^{h} \varepsilon_i - \varepsilon \right) \tag{4.9}$$

$$\Rightarrow \frac{\partial L}{\partial \varepsilon_i} = -\frac{F^i}{\varepsilon_i^3} + \lambda = 0 \tag{4.10}$$

$$\Rightarrow \varepsilon_i = \left(\frac{F^i}{\lambda} \right)^{\frac{1}{3}} \tag{4.11}$$

Remplazando ε_i en las condiciones iniciales nos da que:

$$\varepsilon_i \propto F^{i/3} \tag{4.12}$$

Este enfoque permite una asignación más eficiente del presupuesto de privacidad en cada nivel de la estructura recursiva, optimizando así la precisión de los conteos en presencia de privacidad diferencial.

En este estudio, el presupuesto de privacidad se ha segmentado en tres partes iguales, distribuidas entre la publicación de datos, el cálculo de la condición de parada y los recursos asignados para los particionadores. Es decir, si tenemos un presupuesto total $\varepsilon_{\text{total}} = \varepsilon$ la distribución de presupuesto queda como:

$$\frac{\varepsilon}{3} = \sum_{i=1}^{\text{maxdepth}} \varepsilon_t^i = \sum_{i=1}^{\text{maxdepth}} \varepsilon_{pt}^i = \varepsilon_{pb} \tag{4.13}$$

En el caso del particionador binario, cualquier porción del presupuesto que no se haya utilizado en la fase de partición se redistribuye hacia el presupuesto destinado para la determinación del umbral. Esta estrategia de asignación permite un uso más eficaz y equilibrado de los recursos disponibles, garantizando que cada componente del proceso reciba la atención adecuada en términos de privacidad y eficiencia computacional.

4.4. Implementación y tecnologías

En esta sección expondremos los componentes fundamentales que constituyen nuestra solución, abarcando desde los aspectos más detallados de la implementación hasta los patrones de diseño adoptados. Además, se analizarán las diversas interacciones entre las clases desarrolladas.

Modulos y clases

Para la implementación efectiva de algoritmos recursivos y particiones, fue esencial desarrollar estructuras de datos que permitieran una representación clara y manejable tanto de los datos como de su dominio. A continuación, se describen dos estructuras de datos clave diseñadas para este propósito.

Punto

En el contexto de análisis de datos de alta dimensionalidad, adoptamos una representación basada en puntos en espacios $\mathbb{R}^d$. Cada punto se caracteriza por un conjunto de coordenadas y un peso asociado en $\mathbb{R}$. Esta estructura resulta esencial para garantizar la compatibilidad con los DataFrames o RDDs (*Resilient Distributed Datasets*), proporcionando así una base sólida para la integración con las bibliotecas de Apache Spark. Hemos desarrollado métodos que facilitan la conversión entre distintos formatos y aseguran una sinergia efectiva con el ecosistema de Spark, realzando tanto la flexibilidad como la eficiencia del proceso de análisis de datos.

Bloque

Para representar subespacios rectangulares dentro de $\mathbb{R}^d$, se optó por una metodología que define rectángulos utilizando sus esquinas diagonalmente opuestas, o bien, a través de intervalos para cada dimensión. Esta técnica es crucial para manejar y dividir de manera eficiente el espacio de datos, de acuerdo a los requerimientos de los algoritmos de particionamiento. Implementamos funciones especializadas que simplifican la división y partición de estos bloques, adaptándose dinámicamente a las estrategias de partición.

Particionador

El particionador procesa un conjunto de puntos dentro de un bloque específico, aplicando algoritmos de particionamiento para generar una lista de pares (bloque, lista de puntos). Esta herramienta es capaz de ejecutar todas las formas de particionamiento mencionadas, ofreciendo una gran versatilidad en el manejo de datos.

Estrategia de presupuesto

Bajo un presupuesto de privacidad definido, junto con una cantidad preestablecida de niveles y una estrategia de asignación de presupuesto (lineal, uniforme, geométrica), esta funcionalidad crea la distribución del presupuesto para cada nivel en el algoritmo recursivo. Esta estrategia es fundamental para garantizar la correcta ejecución y eficacia del algoritmo en contextos que requieren privacidad de datos.

Sinopsis privada y recursiva

La sinopsis recursiva implementa el algoritmo de partición recursiva. Ejecuta la partición para el nivel correspondiente, teniendo en cuenta todas las condiciones de parada, y facilita la transición al siguiente nivel de profundidad. Por otro lado, la sinopsis privada actúa como un *wrapper* para la sinopsis recursiva, configurando el entorno con todos los parámetros necesarios para una ejecución eficiente y correcta del algoritmo.

Diseño y diagrama de clases

En nuestro esfuerzo por desarrollar una clase que maneje eficientemente los DataFrame de Spark, hemos aplicado principios de diseño de software para garantizar una configuración flexible y una integración coherente de la lógica de procesamiento de datos.

Como se ilustra en el diagrama de clases en la Figura 4.10, se destacan los elementos clave en términos de ingestión de datos y los patrones de diseño utilizados. La estructura presentada refleja un enfoque sistemático y bien organizado.

El uso del patrón de diseño **Estrategia** es notable en las clases **Particionador** y **EstrategiaPresupuesto**. Este patrón ha sido fundamental para diferenciar y encapsular variados algoritmos de procesamiento de datos, lo que resulta en una mayor flexibilidad y mantenimiento del código. En particular, permite variaciones en las estrategias de particionamiento y asignación de recursos sin alterar la arquitectura principal, facilitando así adaptaciones específicas a distintos contextos de procesamiento.

Este diseño orientado a patrones no solo mejora la modularidad y la escalabilidad del sistema, sino que también proporciona claridad en la estructura del código, permitiendo un enfoque más enfocado y organizado en cada aspecto del proceso de ingestión de datos.

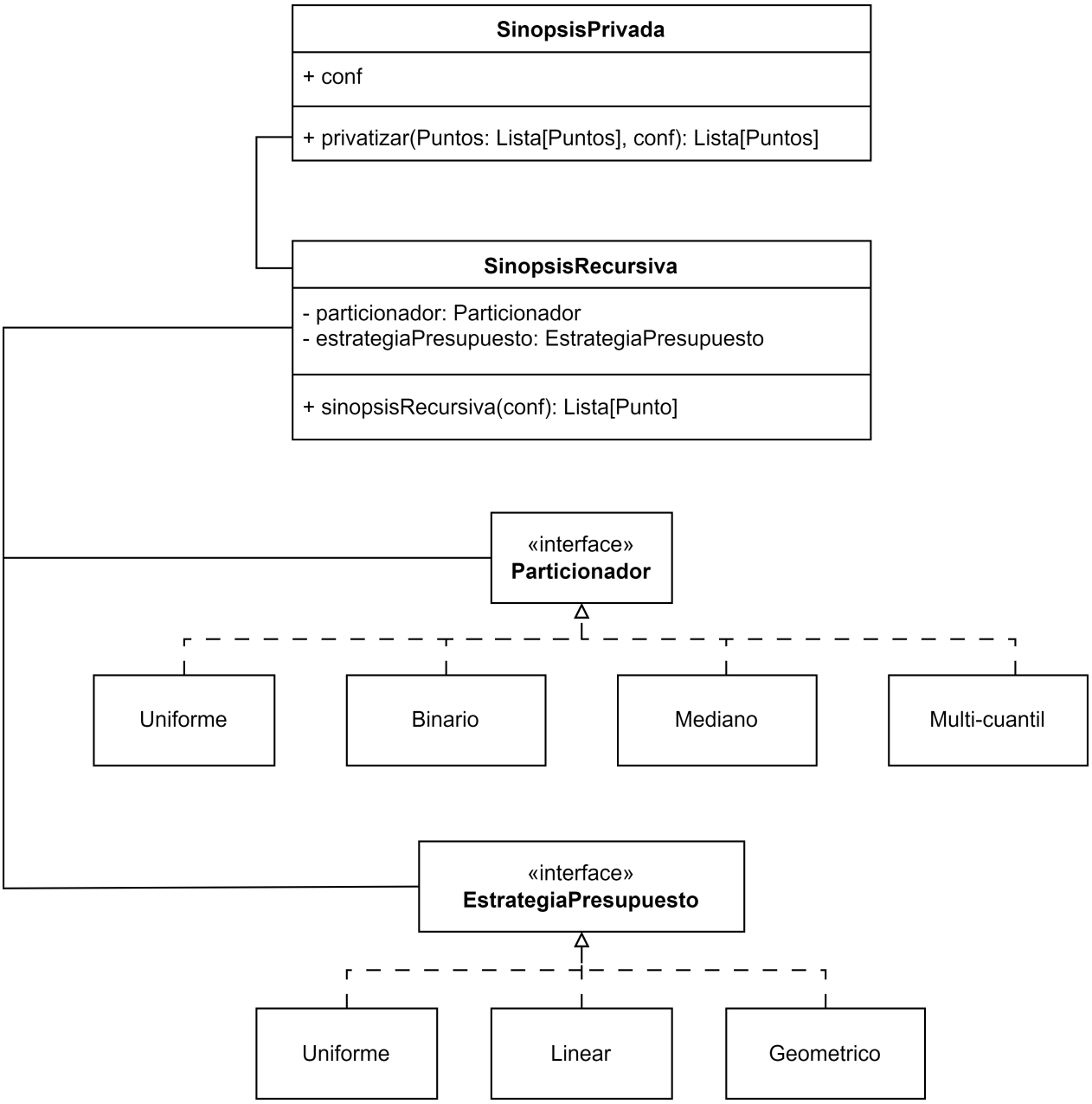


Figura 4.10: Diagrama de clases

Capítulo 5

Evaluación

5.1. Conjuntos de datos

La elección adecuada de bases de datos constituye un pilar fundamental en nuestro estudio sobre *clustering* diferencialmente privado y *clustering* tradicional. Para este fin, hemos realizado una revisión meticulosa y detallada de las publicaciones más relevantes en estas áreas, con el objetivo de identificar conjuntos de datos que se alineen estrechamente con nuestros objetivos de investigación.

Los conjuntos de datos seleccionados, detallados en la Tabla 5.1, han sido escogidos por presentar estructuras de *clusters* distintivas y poseer atributos que son críticos para la obtención de resultados significativos y replicables. Estas bases de datos no solo reflejan una variedad de escenarios en los que el *clustering* es aplicable, sino que también ofrecen la complejidad y las peculiaridades necesarias para un análisis profundo y exhaustivo.

Además, estos conjuntos de datos desempeñan un rol crucial en la creación de *benchmarks* para evaluar el desempeño de algoritmos en diferentes facetas de *clustering*. Esto incluye la aplicación en contextos de alta dimensionalidad y en escenarios donde se requiere la privacidad diferencial. La inclusión de estas bases de datos nos permite no solo comparar el rendimiento de nuestros algoritmos propuestos contra estándares establecidos, sino también explorar su efectividad en entornos desafiantes y variados, proporcionando así una comprensión más amplia de su aplicabilidad y robustez.

En nuestro estudio, experimentamos con un total de ocho conjuntos de datos, divididos en dos categorías: cuatro sintéticos y cuatro reales.

Los conjuntos de datos sintéticos -S Sets- comparten características comunes: todos son bidimensionales, constan de 5000 filas y presentan 15 agrupaciones (*clusters*) de distribución gaussiana.

En cuanto a los conjuntos de datos reales, se describen a continuación:

1. Reseñas de Viajes (Tarvel review ratings): Este conjunto incluye calificaciones prome-

dio, en una escala de 1 a 5, para diversas atracciones turísticas en Europa, abarcando 24 categorías distintas como parques, restaurantes, galerías de arte, discotecas, entre otras. Se identifican principalmente cinco *clusters* y consta de 5456 filas.

2. Transbordador (Shuttle): Compuesto por 9 atributos numéricos, este conjunto de datos destaca por la presencia significativa de valores atípicos (*outliers*), lo que lo hace ideal para evaluar la eficacia de nuestros algoritmos. Contiene siete *clusters*, uno de los cuales representa el 80 % del total de los datos, con un total de 58000 filas.

3. Segmentación de Piel: Este conjunto se centra en tres atributos construidos sobre el espacio de color BGR (Azul, Verde, Rojo) y se generó utilizando texturas de piel de rostros humanos, abarcando una amplia diversidad de edades, géneros y razas. Se distinguen dos *clusters* principales: piel y no piel, y cuenta con 245057 filas.

4. Metro PT: Recoge datos de un tren de metro en operación, incluyendo mediciones de presión, temperatura, corriente del motor y de las válvulas de entrada de aire de la Unidad de Producción de Aire (APU) de un compresor. Este conjunto está compuesto por 15 variables continuas, cinco *clusters* y un total de 1516948 filas.

A continuación, presentamos una tabla 5.1 resumen de los conjuntos de datos utilizados:

Conjunto de datos	**k**	**# Dimensiones**	**# Filas**	**Referencia**
S Sets	15	2	5000	Fränti and Sieranoja [2018]
Reseñas de Viajes	5	24	5456	Renjith [2018]
Transbordador	7	9	58000	Catlett [2006]
Segmentación de Piel	2	3	245057	Bhatt and Dhall [2012]
Metro PT	5	15	1516948	Davari et al. [2023]

Tabla 5.1: Descripción de los conjuntos de datos

5.2. Métricas

La eficacia de un algoritmo de *clustering* depende en gran medida de su capacidad para identificar agrupaciones significativas y diferenciables en los datos. Para evaluar y comparar el rendimiento de estos algoritmos, se utilizan diversas métricas. En esta sección, nos enfocaremos en dos métricas cruciales: la varianza normalizada entre *clusters* y el índice de centroides.

Varianza normalizada entre *clusters*

La varianza normalizada entre *clusters* (NICV, por sus siglas en inglés) se presenta como una métrica esencial para evaluar la eficacia del *clustering*, reflejando la suma de las distancias al cuadrado entre cada punto y su centroide más cercano, ofreciendo una medida directa y relevante del costo promedio del *clustering*.

Dado un conjunto de datos de d-dimensiones $D = \{x^1, x^2, \ldots, x^N\}$, y su partición en k subconjuntos $O = \{O^1, O^2, \ldots, O^k\}$, la NICV se define como:

$$\text{NICV} = \frac{1}{N} \sum_{j=1}^{k} \sum_{x \in O^j} \left\| x - c^j \right\|^2 . \tag{5.1}$$

Esta métrica, directamente relacionada con el costo promedio del *clustering*, proporciona una indicación clara de cómo la calidad varía en función de los hiperparámetros. Cuanto menor sea el valor mejor sera el resultado del *clustering*.

Índice de centroides

Además de la meta primordial de minimizar el costo asociado al proceso de agrupamiento, un objetivo adicional es garantizar que los *clusters* generados guarden una similitud sustancial con aquellos identificados en estudios de referencia, idealmente manifestando una correspondencia uno a uno.

El índice de centroides (CI), una métrica externa utilizada para evaluar la calidad del agrupamiento, es decir, compara los resultados de agrupación con un conjunto predefinido de centros de referencia. Esta métrica es crucial para determinar la concordancia entre los centros de referencia establecidos y los resultados obtenidos del proceso de agrupamiento. Por ejemplo, en la Figura 5.1, se muestra un caso en el que se logra un CI de 4. En esta figura se resaltan áreas con centroides adicionales, identificados por flechas rojas, y zonas que carecen de centroides, marcadas con flechas azules.

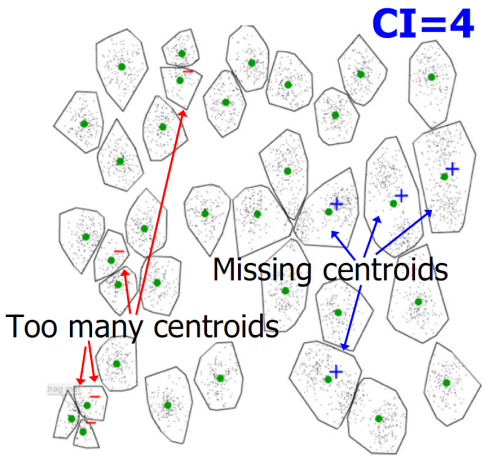


Figura 5.1: Ejemplo: índice de centroides de 4

El cálculo del CI implica la comparación de las similitudes entre pares de puntos de datos dentro del mismo grupo, tanto en la partición de agrupamiento obtenida como en la partición de referencia. Un valor bajo del CI indica una mayor concordancia y similitud entre ambas particiones, sugiriendo una mayor calidad en el resultado del agrupamiento.

Para calcular el índice de centroides, se consideran dos conjuntos de centroides $C = \{c_1, c_2, \ldots, c_k\}$ y $C' = \{c'_1, c'_2, \ldots, c'_k\}$ (puntos verdes en la Figura 5.1). Se establecen mapeos de vecinos más cercanos ($C \to C'$) de la siguiente forma:

$$q_i = \arg \min_{1 \leq j \leq k} \|c_i - c'_j\|^2 \quad \forall i \in [1, k]$$

Para cada centroide calculado c'_j con $j \in [1, k]$, se examina cuántos centroides de referencia c_i lo consideran como el más cercano ($q_i = j$). Nos interesamos especialmente en aquellos centroides a los cuales no se les ha asignado un centroide de referencia. Se define un *centroide huérfano* de c'_j de la siguiente manera:

$$\text{huérfano}(c'_j) = \begin{cases} 1 & \text{si } q_i \neq j \quad \forall i \\ 0 & \text{de lo contrario} \end{cases}$$

Por lo tanto, la disimilitud entre C y C' se mide por el número de centroides huérfanos:

$$CI(C, C') = \sum_{j=1}^{k} \text{huérfano}(c'_j)$$

En la Figura 5.1, señalados con flechas rojas se muestran las regiones con un mayor número de centroides de lo esperado. Cada centro (en verde) que no sea el más cercano al centro de referencia se considerará un centro huérfano. Cabe señalar que los centros de referencia no se muestran en la figura; sin embargo, es posible estimar su ubicación a partir de los puntos. Por lo tanto, el CI se define como la cantidad total de centros huérfanos.

Estas métricas ofrecen una visión complementaria sobre el desempeño de los algoritmos de particionamiento, resultando esenciales para el análisis de nuestros algoritmos.

5.3. Marco experimental

La comparación de algoritmos no deterministas presenta desafíos significativos debido a la variabilidad de los resultados incluso en condiciones experimentales idénticas. Nuestra investigación se enfoca en la interacción entre la sinopsis diferencialmente privada y el algoritmo *k-means*, con un énfasis particular en la calidad y consistencia de ambos componentes dentro del marco experimental.

En el proceso de creación de la sinopsis, que incorpora elementos no deterministas, se generan $R_s = 10$ sinopsis para cada configuración definida por la tupla de parámetros (particionador, presupuesto de privacidad, asignación de presupuesto, umbral, profundidad máxima). Posteriormente, se evalúa la eficacia del *clustering* en los datos privatizados ejecutando *k-means* sobre las sinopsis generadas, repitiéndose este proceso $R_k = 5$ veces. Esta frecuencia de repetición se determinó experimentalmente considerando la estabilidad del algoritmo de

Parámetros	Espacio de prueba
Particionadores	multi-cuantil, uniforme, binario, mediano
Estrategias de Presupuesto	geométrico, uniforme, lineal
Epsilones	0.01, 0.02, 0.04, 0.1, 0.21, 0.46, 1.0, 2.15, 4.64, 10.0
Profundidades Máximas	2, 4, 6, 8, 10, 12
Umbrales	10, 20, 40, 80

Tabla 5.2: Espacio de configuraciones

k-means. Luego se selecciona el conjunto de centroides con el mejor valor de NICV con respecto a la sinopsis (para no violar la privacidad), finalmente se calcula el NICV con respecto a los datos originales. Adicionalmente, se calcula el CI de los centroides seleccionados en comparación con aquellos obtenidos mediante el *baseline* no privado *k-means++* Bahmani et al. [2012]. Finalmente, se promedian los resultados de todas las sinopsis correspondientes a la misma configuración de parámetros. Se puede apreciar un diagrama de este proceso en la Figura 5.2.

En este estudio, se exploraron diversas configuraciones, resultantes de la combinación de múltiples variables: particionadores, estrategias de presupuesto, epsilons, profundidades máximas y umbrales. La cantidad total de configuraciones únicas se calculó como el producto de las cantidades de opciones disponibles en cada variable, resultando en $|\text{particionadores}| \cdot |\text{estrategias de presupuesto}| \cdot |\text{epsilons}| \cdot |\text{profundidades máximas}| \cdot |\text{umbrales}| = 4 \cdot 3 \cdot 10 \cdot 6 \cdot 4 = 2,880$. Para cada configuración, se realizaron $R_s = 10$ sinopsis privadas, dando un total de 28,800 abstracciones. Además, se ejecutaron $R_k = 5$ instancias de *k-means* para cada sinopsis, lo que nos da un total de 144,000 ejecuciones de *k-means* para cada conjunto de datos. El espacio de configuraciones se puede apreciar en la Tabla 5.2

Dada la intensidad computacional de estos experimentos, no fue factible realizarlos en un equipo personal. Por ello, se empleó un servidor proporcionado por el Instituto Milenio Fundamentos de los Datos (IMFD). Este servidor estaba equipado con el sistema operativo Ubuntu 22.04, 4 núcleos físicos, contaba con 16GB de memoria RAM, lo que aseguró la eficiencia y fiabilidad de las ejecuciones experimentales.

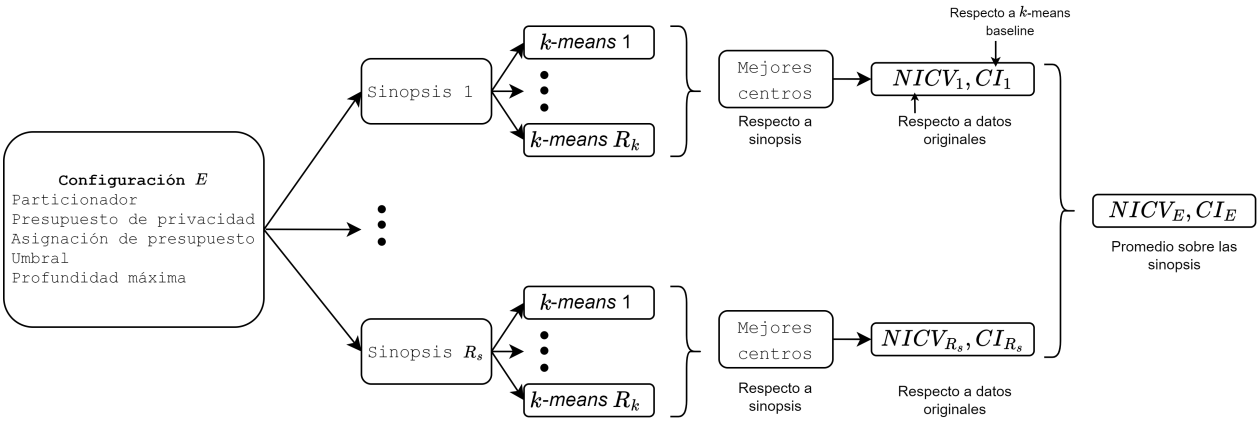


Figura 5.2: Diagrama del marco experimental

En nuestros experimentos se compararán cuatro tipos de particionadores, múltiples niveles

de presupuesto de privacidad, tres estrategias de asignación de presupuesto, varios umbrales y diferentes profundidades máximas. Determinados conjuntos de datos se someterán a pruebas adicionales en función de características específicas, como el número de dimensiones.

5.4. *Baseline* y *benchmarks*

La elección de *benchmarks* y *baselines* adecuados es esencial para evaluar de manera efectiva las mejoras propuestas en nuestro estudio. En este contexto, hemos identificado un *baseline* y dos *benchmarks* que toman fuerte relevancia en nuestro contexto.

Consideraremos como *baseline* no privado la implementación de *k-means* en Spark por Bahmani et al. [2012]. Para poder comparar efectivamente se considerara el mejor valor de *NICV* de las R_k ejecuciones.

El primer *benchmark*, **DPKFM**, se basa en el trabajo de Matías Toro y Federico Olmedo en el Plan Ceibal, donde desarrollaron una sinopsis privada de dos niveles con un particionador uniforme y sin condiciones de parada. Esta sinopsis se utiliza como entrada para *kmeans*. En la implementación disponible cada sinopsis privada implica una ejecución de *k-means* sobre ella, evaluaremos el promedio de los valores R_k de *NICV*.

El segundo *benchmark*, **EUGKm**, introduce la creación de sinopsis privadas mediante un particionador uniforme de un nivel. Este enfoque, directo pero eficiente, se destaca en la abstracción de datos y en la provisión de garantías de privacidad. Representa un hito significativo en el ámbito del *clustering* diferencialmente privado. **EUGKm** es una de las variantes más comúnmente empleadas, sobresaliendo por su sofisticado análisis de los errores inducidos por el ruido laplaciano y el presupuesto de privacidad disponible. Este análisis es crucial para determinar la manera adecuada de dividir el espacio de la partición.

La integración de estos *benchmarks* nos permitirá realizar una evaluación integral y comparativa de nuestras propuestas, situando nuestros resultados en el contexto de contribuciones previas y avances recientes en el campo del *clustering* diferencialmente privado.

5.5. Resultados

En esta sección, presentamos los resultados de los experimentos realizados para la creación de sinopsis privadas en múltiples conjuntos de datos. Estos conjuntos nos permiten evaluar la calidad de nuestros algoritmos desde diversas perspectivas. La organización de esta sección sigue el orden de los conjuntos de datos: S1, S2, S3, S4, Shuttle (Transbordador), Skin Segmentation (Segmenctación de piel) y Travel Review Ratings (Reseñas de viaje).

Aunque la Tabla 5.2 destaca la configuración de los experimentos planificados, finalmente solo se ejecutó una fracción de estos. Esto se debe a que, a medida que se obtenían resultados, algunas conclusiones comenzaron a esclarecerse, lo que llevó a la decisión de excluir ciertos parámetros de las configuraciones iniciales. Por ejemplo, en conjuntos de datos reales con más de 50,000 puntos, el parámetro del umbral perdió relevancia (la diferencia entre 10 y 80 era marginal), lo que nos llevó a presentar resultados utilizando un umbral de 80 puntos para estos casos.

Los gráficos y tablas que ilustran los resultados de los experimentos se incluyen en el Anexo A y Anexo B.

Se compararán los resultados de nuestros algoritmos con los *benchmarks*, y se dirá que una configuración es mejor si presenta mejores (en este caso menores) valores de las métricas en la mayoría de los presupuestos de privacidad.

Resultados del Conjunto de Datos S1

El conjunto de datos S1, de carácter sintético y bidimensional, consta de 5000 puntos y se divide en 15 *clusters*. A continuación, presentamos los resultados más destacados para cada tipo de particionador, enfatizando las configuraciones que ofrecieron el mejor desempeño.

Gracias a los gráficos, podemos observar que los resultados para S1 son significativamente mejores cuando el umbral es menor. Por este motivo, en la presentación de resultados subsiguiente (para los conjuntos S) haremos referencia únicamente a los experimentos con un umbral establecido en 10.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV):** La configuración que muestra mejores resultados es aquella con presupuesto uniforme, con profundidad **8**.
- **Índice de Centroides (CI):** La configuración que muestra mejores resultados es aquella con presupuesto uniforme y profundidad **12**.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones que muestran mejores resultados son aquellas con presupuesto uniforme y con profundidad **10**, junto con las que tienen presupuesto geométrico y profundidad **10** y **12**.
- **Índice de Centroides (CI):** Las configuraciones con presupuesto geométrico y uniforme, y profundidades **10** y **12** muestran los mejores resultados con respecto a la métrica CI.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV):** Pocas configuraciones superan los benchmarks consistentemente. Aquella que mostró mejores resultados fue la configuración con presupuesto geométrico y profundidad de **10**.
- **Índice de Centroides (CI):** La configuración que mostró mejores resultados fue la uniforme con profundidad 6, le siguieron de cerca las geométricas con profundidades 2 y 4.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** La configuración que mostró mejores resultados fue la uniforme con profundidad **8**, muy de cerca le sigue la con profundidad 6, así mismo aquellas con presupuesto geométrico y las mismas profundidades.

- **Índice de Centroides (CI):** Las configuraciones con presupuesto uniforme y profundidad mayor a 6 mostraron los mejores resultados. Le siguen las configuraciones con presupuesto geométrico.

Es importante mencionar que los valores superiores a los *benchmarks* suelen repetirse para las mismas configuraciones y presupuestos de privacidad.

Además, al realizar una comparación entre los distintos particionadores, se destaca el **particionador multi-cuantil** por presentar las métricas más bajas (y por ende, más favorables) tanto para NICV como para CI. En segundo lugar, se encuentra el particionador uniforme.

Resultados del Conjunto de Datos S2

El conjunto de datos S2, de carácter sintético y bidimensional, consta de 5000 puntos y se divide en 15 *clusters*. A continuación, presentamos los resultados más destacados para cada tipo de particionador, enfatizando las configuraciones que ofrecieron el mejor desempeño.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** La configuración que muestra mejores resultados es aquella con presupuesto uniforme y profundidad **12**.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV):** Solo una configuración pudo superar algún presupuesto de privacidad, y fue la geométrica con profundidad **10**.
- **Índice de Centroides (CI):** Las configuraciones que presentaron mejores resultados fueron las geométricas con profundidades **10** y **12**.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** La mejor configuración fue la geométrica con profundidad **4**, seguida por la de profundidad **6** del mismo tipo de presupuesto.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** Para las configuraciones con presupuesto uniforme y geométrico, con profundidades entre **6** y **12**, se obtuvieron resultados medianamente pobres con respecto a los *benchmarks*.
- **Índice de Centroides (CI):** Para las configuraciones con presupuesto uniforme y geométrico, con profundidades entre **6** y **12**, se obtuvieron resultados similares y mejores que los *benchmarks* para altos presupuestos de privacidad.

Además, al realizar una comparación entre los distintos particionadores, se destacan los particionadores **uniforme** y **multi-cuantil**, el primero obteniendo mejores valores de NICV y el segundo mejores valores de CI.

Resultados del Conjunto de Datos S3

El conjunto de datos S3, de carácter sintético y bidimensional, consta de 5000 puntos y se divide en 15 *clusters*. A continuación, presentamos los resultados más destacados para cada tipo de particionador, enfatizando las configuraciones que ofrecieron el mejor desempeño.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** La configuración que muestra mejores resultados es aquella con presupuesto geométrico y profundidad **10**.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV):** La configuración con presupuesto geométrico y profundidad **12** presentó las mejores métricas.
- **Índice de Centroides (CI):** Las configuraciones que presentaron mejores resultados fueron las geométricas y uniformes con profundidad **4**.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** La mejor configuración fue la geométrica con profundidad **4**.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones con presupuesto uniforme y geométrico obtuvieron resultados similares para las profundidades mayores a **6**.
- **Índice de Centroides (CI):** Las configuraciones con presupuesto uniforme y geométrico con profundidad **8** obtuvieron las mejores métricas.

Además, al realizar una comparación entre los distintos particionadores, se destacan los particionadores **uniforme** y **multi-cuantil**, el primero obteniendo mejores valores de NICV y el segundo mejores valores de CI.

Resultados del Conjunto de Datos S4

El conjunto de datos S4, de carácter sintético y bidimensional, consta de 5000 puntos y se divide en 15 *clusters*. A continuación, presentamos los resultados más destacados para cada tipo de particionador, enfatizando las configuraciones que ofrecieron el mejor desempeño.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** Las configuraciones que presentan mejores resultados son aquellas con presupuesto geométrico y uniforme con profundidades **10** y **12**.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV):** Ninguna configuración tiene mejores valores de NICV que los *benchmarks*.
- **Índice de Centroides (CI):** Las configuraciones que presentan mejores resultados son aquellas con presupuesto geométrico y uniforme con profundidades **10** y **12**.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV):** La configuración con mejores resultados es la con presupuesto geométrico y profundidad **4**, seguida por aquella con profundidad **6** del mismo tipo de presupuesto.
- **Índice de Centroides (CI):** Las configuraciones que presentan mejores resultados son aquellas con presupuesto geométrico y profundidades **6** y **8**.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones con presupuesto uniforme y geométrico obtuvieron resultados similares para las profundidades de **6** y **8**.
- **Índice de Centroides (CI):** La configuración con mejores métricas fue la con presupuesto uniforme y profundidad **8**.

Además, al realizar una comparación entre los distintos particionadores, el particionador binario obtuvo buenas métricas (NICV y CI) para la configuración con presupuesto uniforme y profundidad **12**. Sin embargo, el **particionador uniforme** alcanzó mejores métricas que el particionador multi-cuantil y binario, tanto en NICV como en CI.

Resultados del Conjunto de Datos Shuttle (Transbordador)

Compuesto por 9 atributos numéricos, este conjunto de datos destaca por la presencia significativa de valores atípicos (*outliers*), lo que lo hace ideal para evaluar la eficacia de nuestros algoritmos. Contiene siete *clusters*, uno de los cuales representa el 80 % del total de los datos, con un total de 58000 filas.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** Los valores muestran poca variación respecto a los diferentes tipos y cantidades de presupuesto de privacidad, así como en las profundidades alcanzadas.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV) e Índice de Centroides (CI):** Las configuraciones exhiben un comportamiento errático. Podríamos decir que no existe una configuración que se destaque significativamente sobre las demás.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV):** La mayoría de las configuraciones supera los valores del *benchmark*, especialmente aquellas que emplean un presupuesto geométrico.
- **Índice de Centroides (CI):** La mayoría de las configuraciones registran valores de CI entre 3 y 4.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones con presupuesto uniforme y geométrico muestran resultados similares para profundidades superiores a **6**.
- **Índice de Centroides (CI):** Los valores tienden a oscilar cerca de 3.

El **particionador uniforme** presenta los mejores valores para las métricas de NICV en este conjunto de datos.

Resultados del Conjunto de Datos Skin Segmentation (Segmentación de Piel)

Este conjunto se centra en tres atributos construidos sobre el espacio de color BGR (Azul, Verde, Rojo) y fue generado utilizando texturas de piel de rostros humanos, abarcando una amplia diversidad de edades, géneros y razas. Se distinguen dos clústeres principales: piel y no piel, y cuenta con 245,057 filas.

Para este conjunto de datos se muestran los resultados con un umbral de 80, dado que se observó que no había una diferencia significativa entre 10 y 80 para los conjuntos con más datos.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones que presentan mejores resultados son aquellas con presupuesto geométrico y uniforme con profundidad **12**, seguidas por aquellas con profundidad **10**.
- **Índice de Centroides (CI):** Todas las configuraciones obtienen un CI de 0.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones con mejores valores son las uniformes y geométricas con profundidad **12**.
- **Índice de Centroides (CI):** Todas las configuraciones obtienen un CI de 0, a excepción de las dos con profundidad 2.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV):** La mayoría de las configuraciones obtienen valores cercanos a 0 para presupuestos mayores a 0.2.
- **Índice de Centroides (CI):** Todas las configuraciones obtienen un CI de 0.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones con presupuesto uniforme y geométrico obtuvieron resultados similares para las profundidades mayores a **6**, alcanzando un promedio de 0.1 para presupuestos mayores a 0.21.
- **Índice de Centroides (CI):** Todas las configuraciones obtienen un CI de 0.

El **particionador uniforme** presenta mejores valores cuando hay más presupuesto de privacidad. Por otro lado, el **particionador multi-cuantil** muestra mejores valores para presupuestos más bajos; ambos tienen los mismos resultados para los valores de CI.

Resultados del Conjunto de Datos Travel Reviews Ratings (Reseñas de Viajes)

Este conjunto incluye calificaciones promedio, en una escala de 1 a 5, para diversas atracciones turísticas en Europa, abarcando 24 categorías distintas como parques, restaurantes, galerías de arte, discotecas, entre otras. Se identifican principalmente cinco *clusters* y consta de 5456 filas.

Para este conjunto de datos se muestran los resultados con un umbral de 80, dado que se observó que no había una diferencia significativa entre 10 y 80 para los conjuntos con más valores.

Particionador Binario:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones que obtuvieron mejores métricas fueron las con presupuesto uniforme y geométrico con profundidad de **12**.
- **Índice de Centroides (CI):** Los valores de CI son bastante bajos, rondan entre 1 y 2. La que obtuvo mejores métricas fue la configuración con presupuesto geométrico y profundidad de **10**.

Particionador Mediano:

- **Varianza entre *clusters* normalizada (NICV):** Las configuraciones con mejores valores son aquellas con presupuesto geométrico y profundidad mayor a 8.
- **Índice de Centroides (CI):** Los valores de CI son bastante bajos, rondan entre 1 y 2, más cercanos a 2. La que obtuvo mejores métricas fue la configuración con presupuesto geométrico y profundidad de **10**.

Particionador Uniforme:

- **Varianza entre *clusters* normalizada (NICV):** Obtiene el mismo valor que el *benchmark*.
- **Índice de Centroides (CI):** Todas las configuraciones obtienen un CI de -1, lo que significa que no se crearon más puntos que la cantidad de *clusters* del conjunto de datos.

Particionador Multi-cuantil:

- **Varianza entre *clusters* normalizada (NICV):** La mejor configuración fue la geométrica con profundidad de **12**.
- **Índice de Centroides (CI):** La configuración con mejores valores es la geométrica con profundidad de **4**.

Todos los particionadores presentan valores mucho mayores que el *baseline*, pero mejores que el *benchmark*. En este conjunto de datos, el particionador con mejores métricas NICV y CI es el **particionador binario**.

Resultados del Conjunto de Datos Metro PT

Los algoritmos de particionamiento experimentaron dificultades al momento de procesar el conjunto de datos de Metro PT. Esto se debió principalmente a la gran cantidad de datos presentes, lo que causó errores de desbordamiento del *stack* de memoria en Java. Por ejemplo, en el caso del particionador uniforme, debido a la numerosa cantidad de puntos y a las dimensiones de los elementos de la partición, la cantidad de subdivisiones que se tenían que realizar era tan grande que no cabían en la memoria. En cuanto a los otros particionadores, se produjeron errores en los cálculos de métricas, lo que impidió analizar la calidad de las sinopsis privadas creadas.

5.6. Interpretación de resultados

Influencia del presupuesto en la precisión

La relación entre el presupuesto de privacidad y la exactitud de los resultados es directa: un mayor presupuesto de privacidad conduce a resultados más precisos. Este presupuesto influye en la definición de la grilla del particionador uniforme y en la precisión de los métodos de particionamiento espacial, los cuales dependen de la ubicación de los puntos.

Impacto del umbral en la calidad de los resultados

Hemos observado que en los conjuntos de datos S, las diferencias son significativas entre los umbrales de 10 y 80, mientras que en otros conjuntos, estas diferencias son mínimas. Esto sugiere la existencia de un umbral a partir del cual los incrementos en precisión se tornan marginales, dependiendo de factores como la cantidad de puntos, su densidad y distribución.

Efectos de la profundidad de particionamiento

La profundidad de particionamiento varía su impacto según el tipo de particionador utilizado. Los particionadores binario y mediano muestran mejoras con mayores profundidades, a diferencia de los particionadores uniforme y multi-cuantil, que tienden a estabilizarse entre las profundidades 6 y 8. La necesidad de una mayor profundidad para los primeros se debe a su menor número inicial de divisiones comparado con los últimos.

Variaciones según el tipo de presupuesto de privacidad

El presupuesto uniforme sobresale con bajos niveles de privacidad frente al presupuesto geométrico, pero este último prevalece con presupuestos más altos. El enfoque del presupuesto uniforme en los primeros niveles de particionamiento lo hace más efectivo a niveles bajos de privacidad. El particionador lineal fue descartado por la falta de una heurística adecuada para definir la ecuación de la recta que definiria la distribución de presupuesto, resultando en resultados subóptimos.

Desempeño de los particionadores

Los particionadores uniforme y *multiquantile* se destacaron con los conjuntos de datos S por su capacidad de refinar el espacio adecuadamente, revelando estructuras de *clusters*. En particular, el particionador uniforme mostró excelentes resultados con el conjunto de datos *shuttle*, superando a los *benchmarks* establecidos, aunque sin alcanzar el *baseline* no privado. En contraste, todos los particionadores lograron buenos resultados con el conjunto

de datos *skin segmentation*, beneficiándose de la menor complejidad de sus *clusters*. Para el conjunto de datos *travel review ratings*, de mayor dimensionalidad, el particionador binario fue el que presento mejores resultados, aunque aún distante del *baseline* no privado, situación posiblemente mejorable con un incremento en la profundidad máxima.

Rendimiento inferior en algunos particionadores

En ciertos conjuntos de datos como S, *shuttle* y *skin segmentation*, los particionadores binario y mediano podrían necesitar mayores profundidades para capturar adecuadamente la forma y precisión de los datos. Específicamente, el particionador uniforme enfrentó desafíos en *travel review ratings* debido a su fórmula de división, inadecuada para altas dimensiones, lo que resultó en una sinopsis privada ineficaz.

Desafíos técnicos y optimización

Los desafíos incluyeron manejar grandes volúmenes de datos y evitar cuellos de botella en la conversión y procesamiento de datos. Las optimizaciones en la implementación **DPKFM** mejoraron la complejidad teórica y los tiempos de ejecución, especialmente en la creación de la grilla uniforme y la asignación de puntos, lo que facilitó la experimentación y el análisis de errores.

Capítulo 6

Conclusión

En este estudio, hemos analizado diversas estrategias de particionamiento espacial para la creación de sinopsis diferencialmente privadas. Hemos comparado estas técnicas con los estándares de referencia actuales y, aunque no todas superaron dichos benchmarks, hemos encontrado que las estrategias de particionamiento multi-cuantil y uniforme son especialmente prometedoras con la selección apropiada de hiperparámetros.

Hemos evaluado la efectividad de los particionadores en función de la dimensionalidad del conjunto de datos. Los resultados indican que, para conjuntos de alta dimensionalidad, los particionadores binarios o medianos con profunda segmentación son más adecuados, dado que los enfoques uniforme y multi-cuantil pueden llevar a una excesiva fragmentación, lo que aminora la velocidad de los algoritmos. Por el contrario, para conjuntos de datos de baja dimensionalidad, los particionadores uniforme y multi-cuantiles proporcionan una representación más efectiva.

En términos del impacto del volumen de datos en el rendimiento de los particionadores, hemos observado que el modelo uniforme se ve significativamente afectado con grandes volúmenes de datos debido a su dependencia directa del tamaño del conjunto para calcular las divisiones necesarias. En contraste, el particionador multi-cuantil se ve mucho menos afectado, aunque no son despreciables las operaciones de ordenamiento que requiere, mientras que los modelos binarios y medianos son menos susceptibles a variaciones en la cantidad de datos.

La selección de un particionador apropiado puede beneficiarse de la información sobre el *clustering* de los datos. Cuando se conoce la disposición espacial de los *clusters*, el particionador multi-cuantil es preferible debido a su habilidad para separar eficazmente los grupos según su ubicación. Sin embargo, en ausencia de información detallada sobre la distribución de los datos, el particionador uniforme resulta ser una herramienta útil para una exploración exhaustiva del espacio, garantizando la preservación de las distribuciones de los puntos.

En conclusión, los resultados obtenidos con los particionadores uniforme y multi-cuantil son alentadores y demuestran su capacidad para mantener la privacidad y la precisión en línea con los *benchmarks* actuales, cumpliendo así con nuestros objetivos iniciales.

6.1. Trabajo futuro

Este trabajo establece una base sólida para investigaciones futuras en el ámbito del *clustering* diferencialmente privado y la partición de datos. Hemos identificado varias direcciones prometedoras que pueden impulsar significativamente este campo. Una de ellas es la exploración de la combinación de diferentes tipos de particionadores. Esta aproximación podría mejorar la precisión del *clustering* al adaptarse de manera más eficiente a las variadas distribuciones de datos en conjuntos de datos de alta dimensionalidad.

Otro enfoque crucial es la búsqueda de una distribución óptima del presupuesto de privacidad (*epsilon*). Esto es particularmente desafiante debido a la necesidad de equilibrar la precisión de los resultados con las garantías de privacidad. La asignación adecuada de *epsilon* en diferentes partes del algoritmo puede resultar en mejoras significativas tanto en la eficiencia como en la efectividad del proceso de *clustering*.

Además, se sugiere investigar la aplicación de otros algoritmos de *clustering* en el contexto de la privacidad diferencial. Estos algoritmos podrían proporcionar perspectivas únicas y soluciones innovadoras a los desafíos inherentes al *clustering* bajo restricciones de privacidad.

La optimización del código es otra área de interés. Mejorar la eficiencia del algoritmo no solo acelerará el proceso de *clustering*, sino que también lo hará más aplicable en situaciones del mundo real donde el tiempo de respuesta es un factor crítico.

Finalmente, la implementación de umbrales automáticos y la determinación automática de la profundidad en los algoritmos de *clustering* pueden proporcionar una mejora considerable en la usabilidad y adaptabilidad del proceso de *clustering*. Estas características permitirían que los algoritmos sean más flexibles y eficientes, ajustándose dinámicamente a las características del conjunto de datos.

En conjunto, estas direcciones representan oportunidades emocionantes para el avance del campo de la privacidad diferencial y el *clustering* de datos. La investigación futura en estas áreas no solo mejorará la teoría subyacente, sino que también tendrá un impacto práctico significativo en la aplicación de técnicas de *clustering* en entornos sensibles a la privacidad.

Bibliografía

S. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28 (2):129–137, 1982. doi: 10.1109/TIT.1982.1056489.

David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, page 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering, 2015.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, CODASPY '16, page 26–37, New York, NY, USA, 2016a. Association for Computing Machinery. ISBN 9781450339353. doi: 10.1145/2857705.2857708. URL `https://doi.org/10.1145/2857705.2857708`.

Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, Min Lyu, and Hongxia Jin. Differentially private k-means clustering and a hybrid approach to private optimization. *ACM Trans. Priv. Secur.*, 20(4), oct 2017. ISSN 2471-2566. doi: 10.1145/3133201. URL `https://doi.org/10.1145/3133201`.

Wahbeh Qardaji, Weining Yang, and Ninghui Li. Differentially private grids for geospatial data. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 757–768, 2013. doi: 10.1109/ICDE.2013.6544872.

Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. Differentially private k-means clustering. In *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, CODASPY '16, page 26–37, New York, NY, USA, 2016b. Association for Computing Machinery. ISBN 9781450339353. doi: 10.1145/2857705.2857708. URL `https://doi.org/10.1145/2857705.2857708`.

Yonghui Xiao, Li Xiong, and Chun Yuan. Differentially private data release through multidimensional partitioning. In Willem Jonker and Milan Petković, editors, *Secure Data*

Management, pages 150–168, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-15546-8.

Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. Differentially private spatial decompositions. In *2012 IEEE 28th International Conference on Data Engineering*, pages 20–31, 2012. doi: 10.1109/ICDE.2012.16.

Sina Shaham, Gabriel Ghinita, Ritesh Ahuja, John Krumm, and Cyrus Shahabi. Htf: Homogeneous tree framework for differentially private release of large geospatial datasets with self-tuning structure height. *ACM Trans. Spatial Algorithms Syst.*, 9(4), nov 2023. ISSN 2374-0353. doi: 10.1145/3569087. URL `https://doi.org/10.1145/3569087`.

Wahbeh Qardaji and Ninghui Li. Recursive partitioning and summarization: A practical framework for differentially private data publishing. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, page 38–39, New York, NY, USA, 2012a. Association for Computing Machinery. ISBN 9781450316484. doi: 10.1145/2414456.2414477. URL `https://doi.org/10.1145/2414456.2414477`.

Jun Zhang, Xiaokui Xiao, and Xing Xie. Privtree: A differentially private algorithm for hierarchical decompositions. In *Proceedings of the 2016 International Conference on Management of Data*, SIGMOD '16, page 155–170, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450335317. doi: 10.1145/2882903.2882928. URL `https://doi.org/10.1145/2882903.2882928`.

Jun Wang, Shubo Liu, Yongkai Li, Hui Cao, and Mengjun Liu. Differentially private spatial decompositions for geospatial point data. *China Communications*, 13(4):97–107, 2016. doi: 10.1109/CC.2016.7464127.

Yan Yan, Xin Gao, Adnan Mahmood, Yang Zhang, Shuang Wang, and Quan Z. Sheng. An arithmetic differential privacy budget allocation method for the partitioning and publishing of location information. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 1395–1401, 2020. doi: 10.1109/TrustCom50675.2020.00188.

Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. Differentially private quantiles. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 3713–3722. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/gillenwater21a.html`.

Pasi Fränti and Sami Sieranoja. K-means properties on six clustering benchmark datasets, 2018. URL `http://cs.uef.fi/sipu/datasets/`.

Shini Renjith. Tarvel Review Ratings. UCI Machine Learning Repository, 2018. DOI: https://doi.org/10.24432/C5C31Q.

Jason Catlett. Statlog (Shuttle). UCI Machine Learning Repository, 2006. DOI: https://doi.org/10.24432/C5WS31.

Rajen Bhatt and Abhinav Dhall. Skin Segmentation. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C5T30C.

Narjes Davari, Bruno Veloso, Rita Ribeiro, and Joao Gama. MetroPT-3 Dataset. UCI Machine Learning Repository, 2023. DOI: https://doi.org/10.24432/C5VW3R.

Bahman Bahmani, Benjamin Moseley, Andrea Vattani, Ravi Kumar, and Sergei Vassilvitskii. Scalable k-means++. *Proc. VLDB Endow.*, 5(7):622–633, mar 2012. ISSN 2150-8097. doi: 10.14778/2180912.2180915. URL `https://doi.org/10.14778/2180912.2180915`.

Wahbeh Qardaji and Ninghui Li. Recursive partitioning and summarization: A practical framework for differentially private data publishing. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, ASIACCS '12, page 38–39, New York, NY, USA, 2012b. Association for Computing Machinery. ISBN 9781450316484. doi: 10.1145/2414456.2414477. URL `https://doi.org/10.1145/2414456.2414477`.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

Anexo A

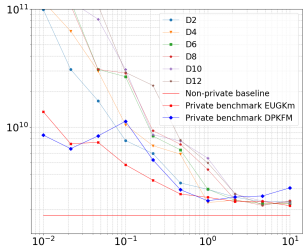
Gráficos

En esta sección se encuentran todos los gráficos que destilan de la siguiente tabla de configuraciones.

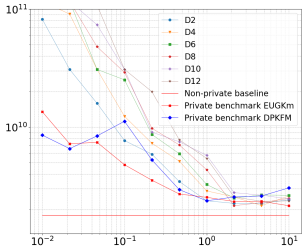
Parámetros	Espacio de prueba
Particionadores	multi-cuantil, uniforme, binario, mediano
Estrategias de Presupuesto	geométrico, uniforme
Epsilones	0.01, 0.02, 0.04, 0.1, 0.21, 0.46, 1.0, 2.15, 4.64, 10.0
Profundidades Máximas	2, 4, 6, 8, 10, 12
Umbrales	10, 80

Tabla A.1: Espacio de configuraciones de resultados

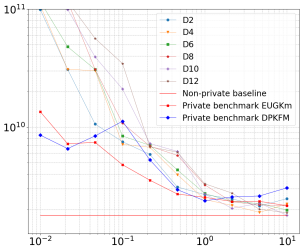
Experimentos sobre el conjunto de datos: S1



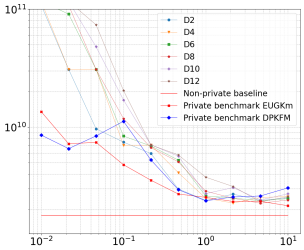
(a) umb: 10, ε: uni



(b) umb: 80, ε: uni

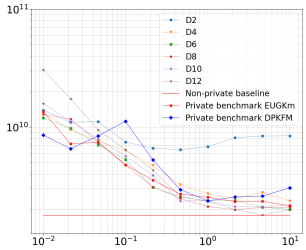


(c) umb: 10, ε: geo

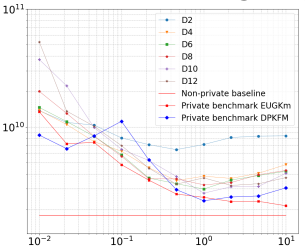


(d) umb: 80, ε: geo

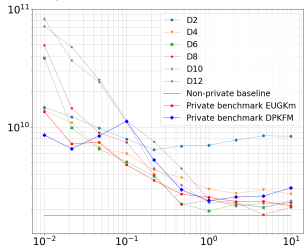
Uniforme



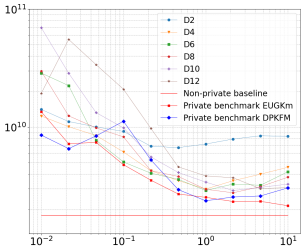
(e) umb: 10, ε: uni



(f) umb: 80, ε: uni

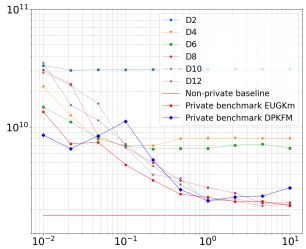


(g) umb: 10, ε: geo

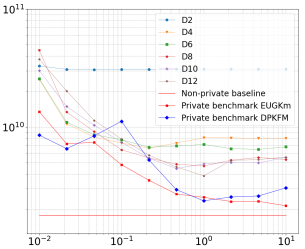


(h) umb: 80, ε: geo

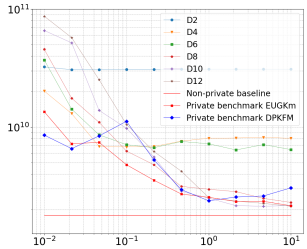
Multi-cuantil



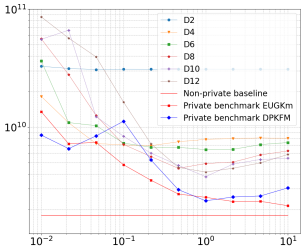
(i) umb: 10, ε: uni



(j) umb: 80, ε: uni

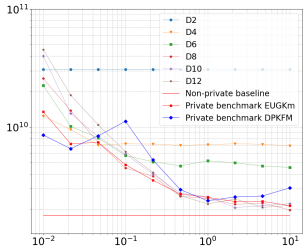


(k) umb: 10, ε: geo

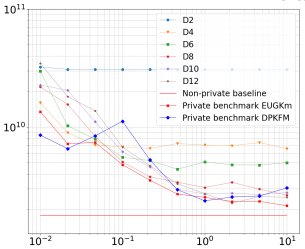


(l) umb: 80, ε: geo

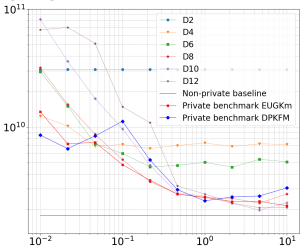
Mediano



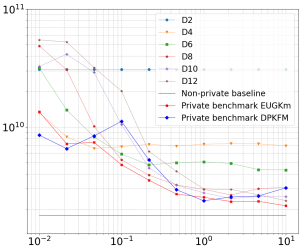
(m) umb: 10, ε: uni



(n) umb: 80, ε: uni



(ñ) umb: 10, ε: geo

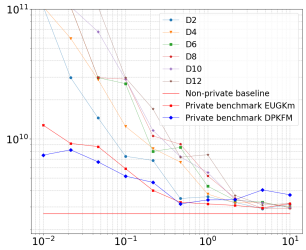


(o) umb: 80, ε: geo

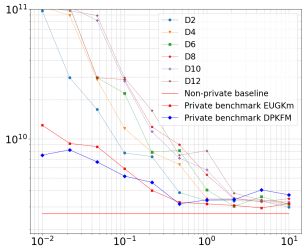
Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

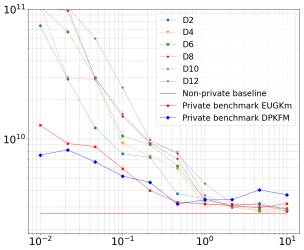
Experimentos sobre el conjunto de datos: S2



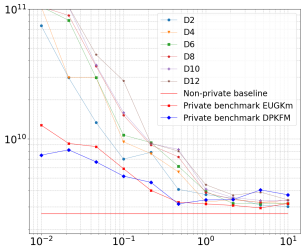
(a) umb: 10, ε: uni



(b) umb: 80, ε: uni

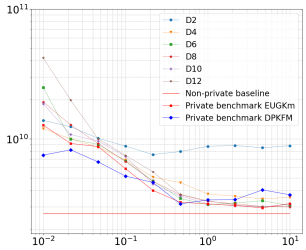


(c) umb: 10, ε: geo

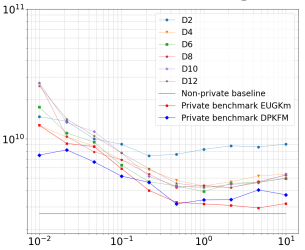


(d) umb: 80, ε: geo

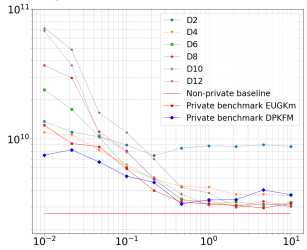
Uniforme



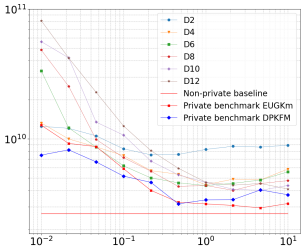
(e) umb: 10, ε: uni



(f) umb: 80, ε: uni

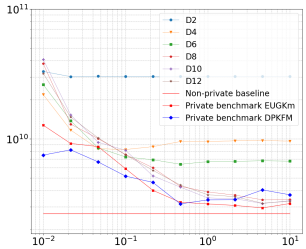


(g) umb: 10, ε: geo

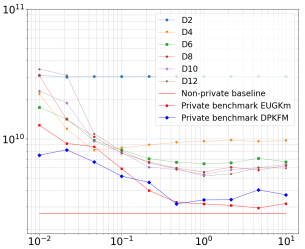


(h) umb: 80, ε: geo

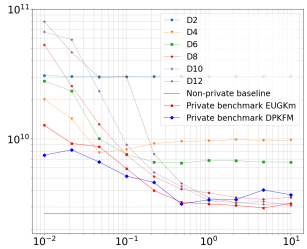
Multi-cuantil



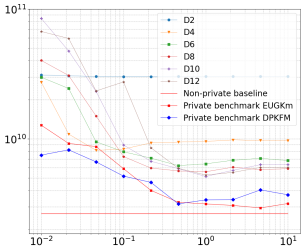
(i) umb: 10, ε: uni



(j) umb: 80, ε: uni

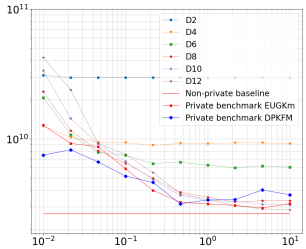


(k) umb: 10, ε: geo

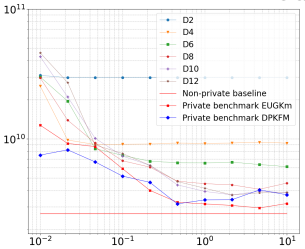


(l) umb: 80, ε: geo

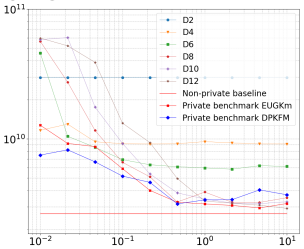
Mediano



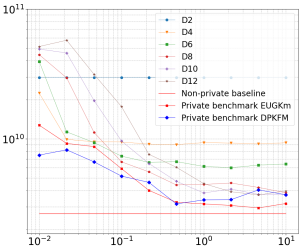
(m) umb: 10, ε: uni



(n) umb: 80, ε: uni



(ñ) umb: 10, ε: geo

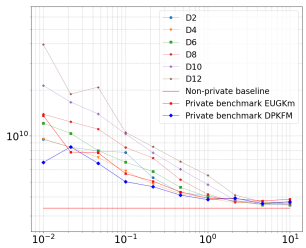


(o) umb: 80, ε: geo

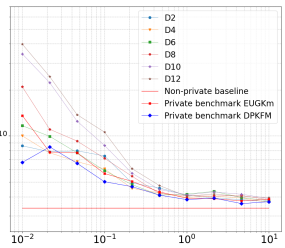
Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

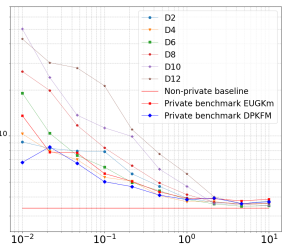
Experimentos sobre el conjunto de datos: S3



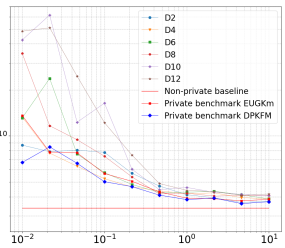
(a) umb: 10, ε: uni



(b) umb: 80, ε: uni

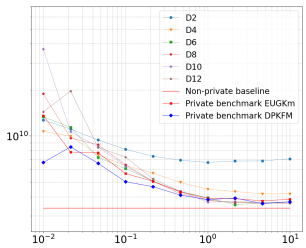


(c) umb: 10, ε: geo

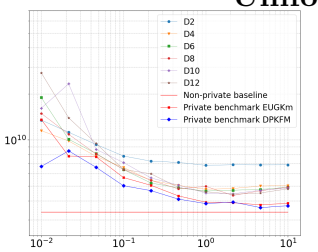


(d) umb: 80, ε: geo

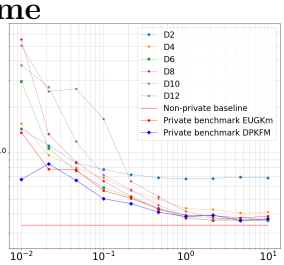
Uniforme



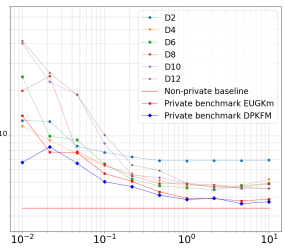
(e) umb: 10, ε: uni



(f) umb: 80, ε: uni

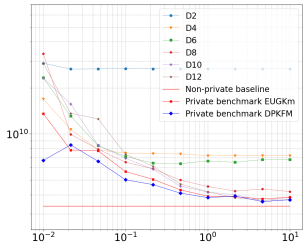


(g) umb: 10, ε: geo

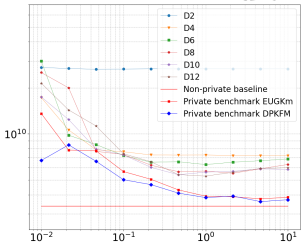


(h) umb: 80, ε: geo

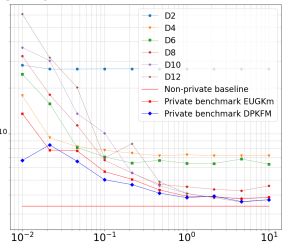
Multi-cuantil



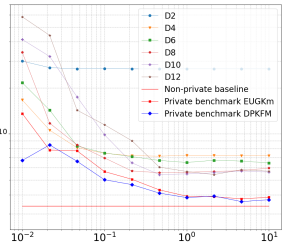
(i) umb: 10, ε: uni



(j) umb: 80, ε: uni

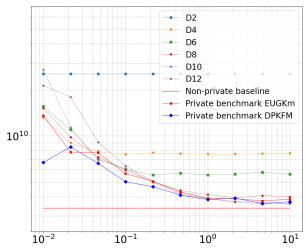


(k) umb: 10, ε: geo

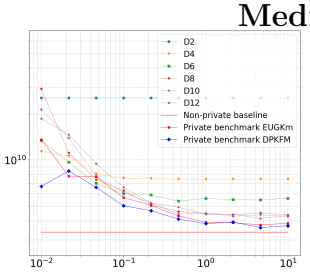


(l) umb: 80, ε: geo

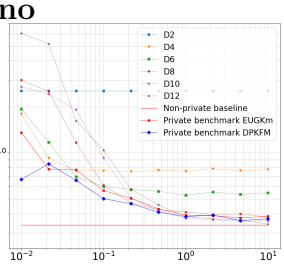
Mediano



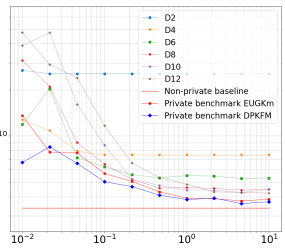
(m) umb: 10, ε: uni



(n) umb: 80, ε: uni



(ñ) umb: 10, ε: geo

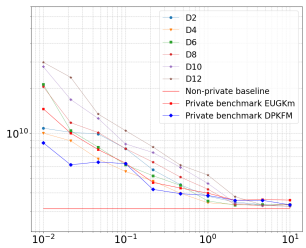


(o) umb: 80, ε: geo

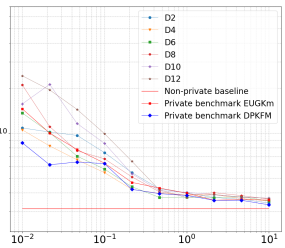
Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

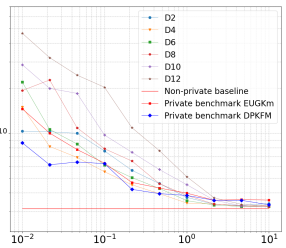
Experimentos sobre el conjunto de datos: S4



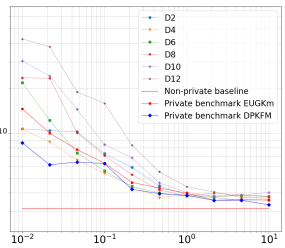
(a) umb: 10, ε: uni



(b) umb: 80, ε: uni

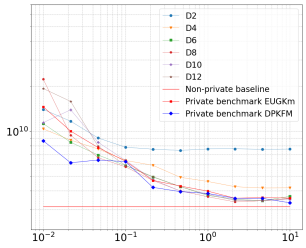


(c) umb: 10, ε: geo

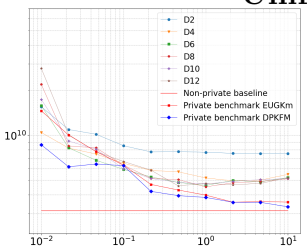


(d) umb: 80, ε: geo

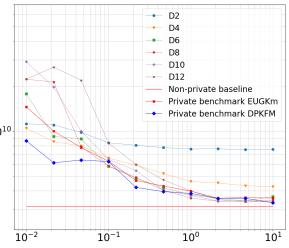
Uniform



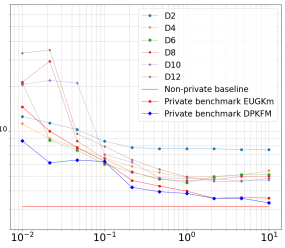
(e) umb: 10, ε: uni



(f) umb: 80, ε: uni

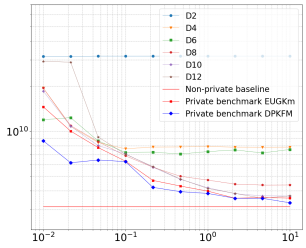


(g) umb: 10, ε: geo

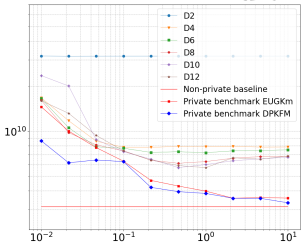


(h) umb: 80, ε: geo

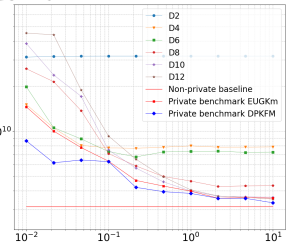
Multi-cuantil



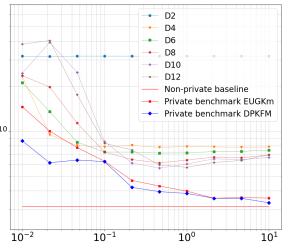
(i) umb: 10, ε: uni



(j) umb: 80, ε: uni

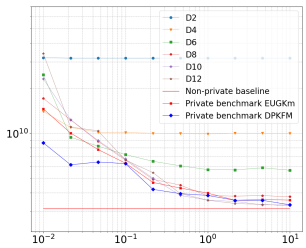


(k) umb: 10, ε: geo

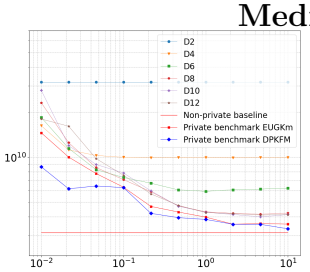


(l) umb: 80, ε: geo

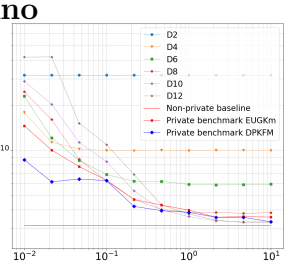
Mediano



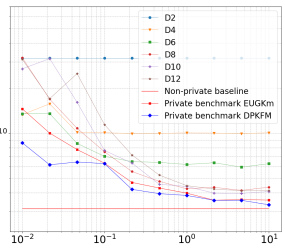
(m) umb: 10, ε: uni



(n) umb: 80, ε: uni



(ñ) umb: 10, ε: geo

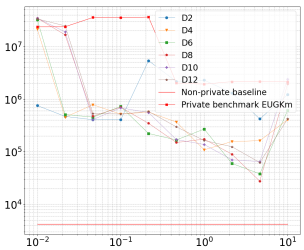


(o) umb: 80, ε: geo

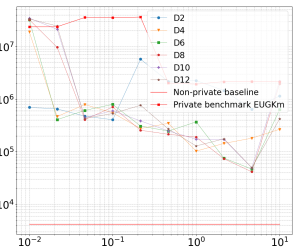
Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

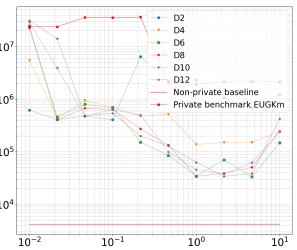
Experimentos sobre el conjunto de datos: Transbordador—Shuttle



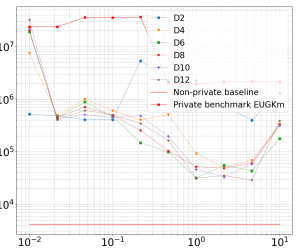
(a) umb: 10, ε: uni



(b) umb: 80, ε: uni

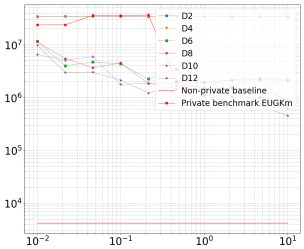


(c) umb: 10, ε: geo

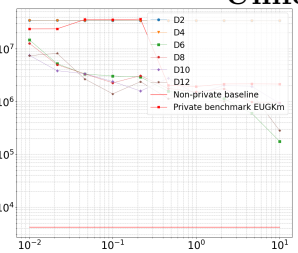


(d) umb: 80, ε: geo

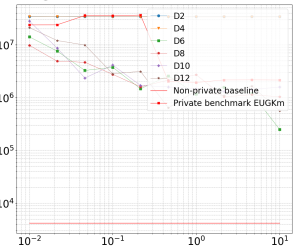
Uniforme



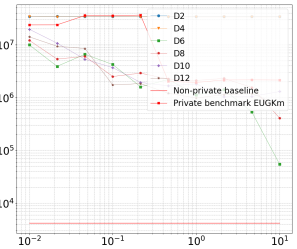
(e) umb: 10, ε: uni



(f) umb: 80, ε: uni

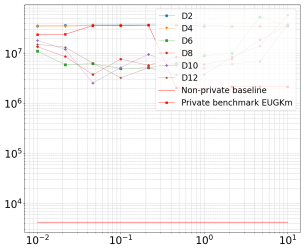


(g) umb: 10, ε: geo

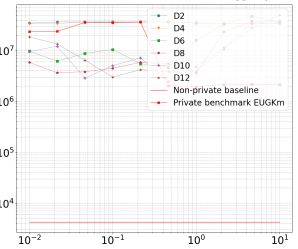


(h) umb: 80, ε: geo

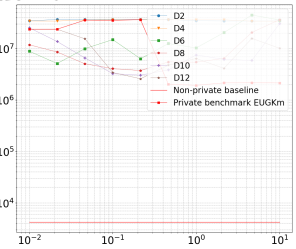
Multi-cuantil



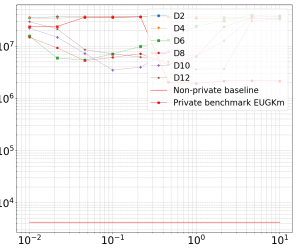
(i) umb: 10, ε: uni



(j) umb: 80, ε: uni

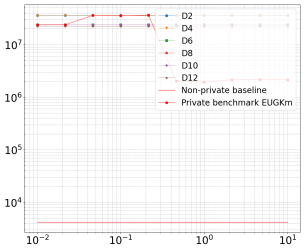


(k) umb: 10, ε: geo

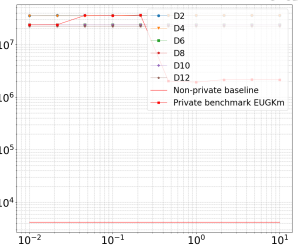


(l) umb: 80, ε: geo

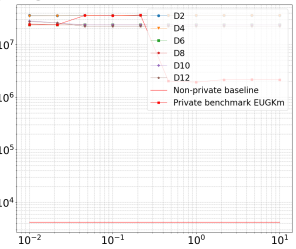
Mediano



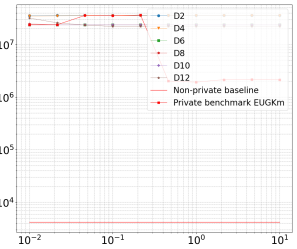
(m) umb: 10, ε: uni



(n) umb: 80, ε: uni



(ñ) umb: 10, ε: geo

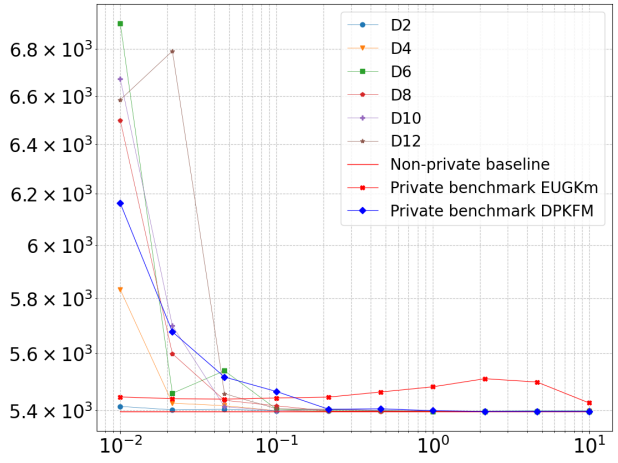


(o) umb: 80, ε: geo

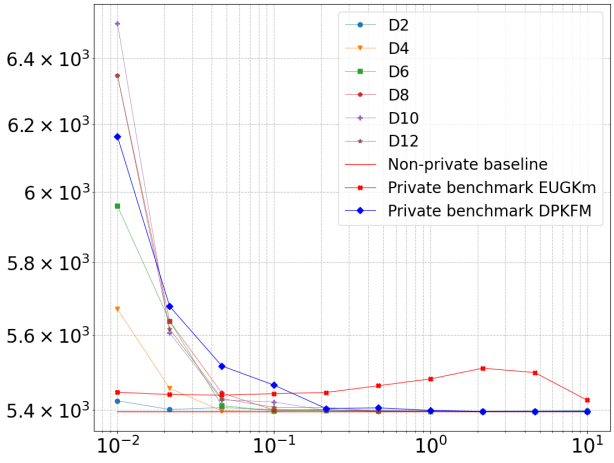
Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

Experimentos sobre el conjunto de datos: Skin segmentation

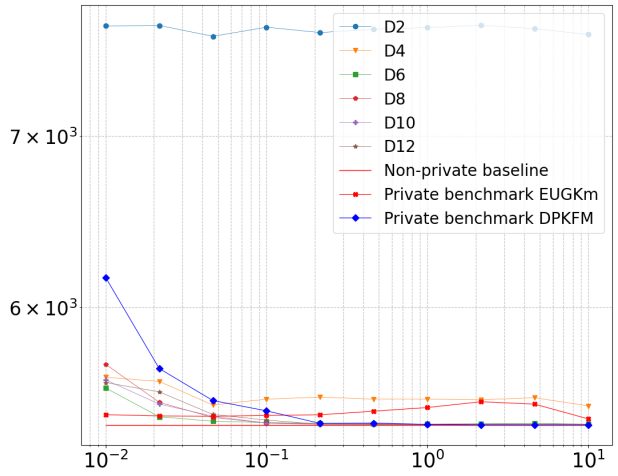


(a) umb: 80, ε: uni

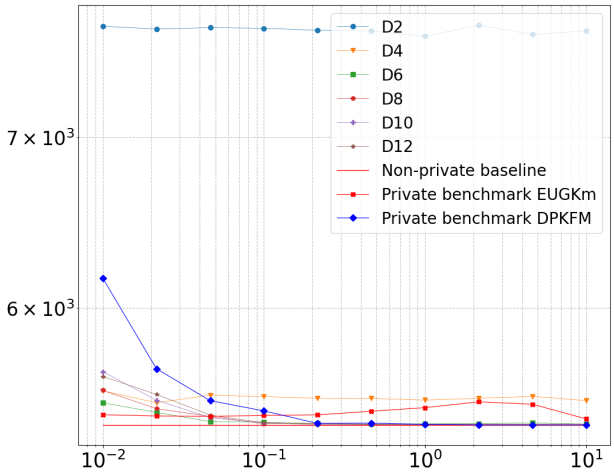


(b) umb: 80, ε: geo

Uniforme



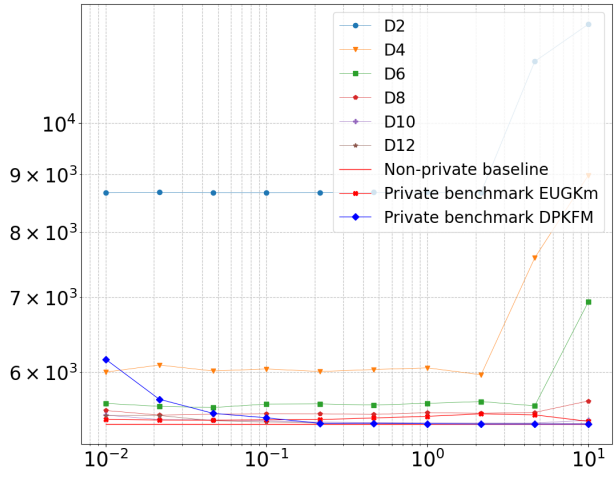
(c) umb: 80, ε: uni



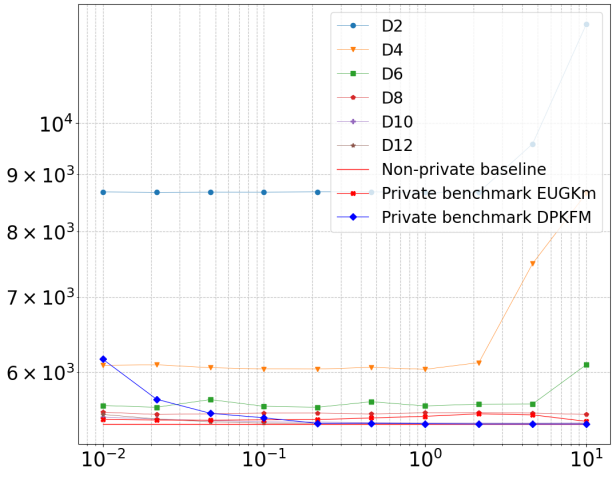
(d) umb: 80, ε: geo

Multi-cuantil

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

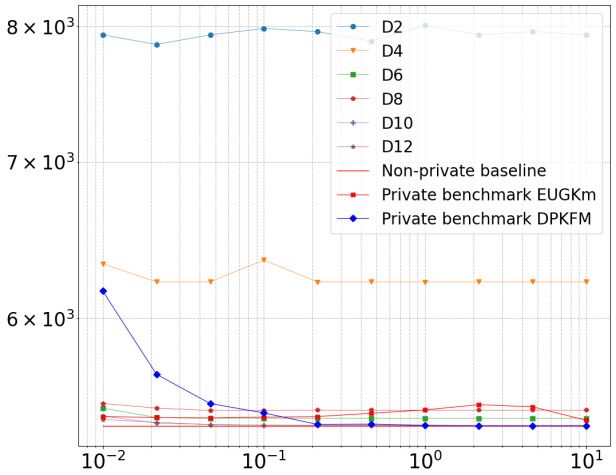


(a) umb: 80, ε: uni

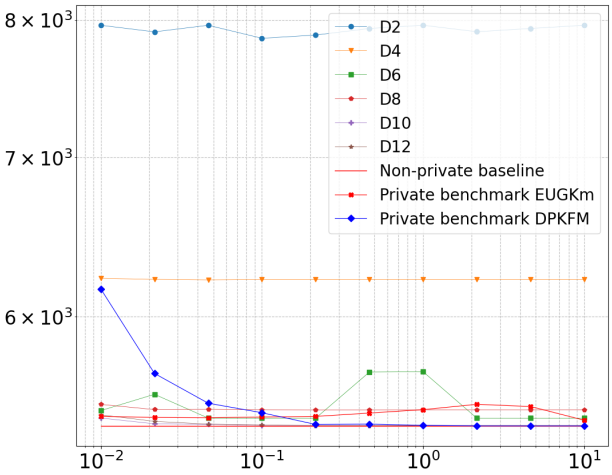


(b) umb: 80, ε: geo

Mediano



(c) umb: 80, ε: uni

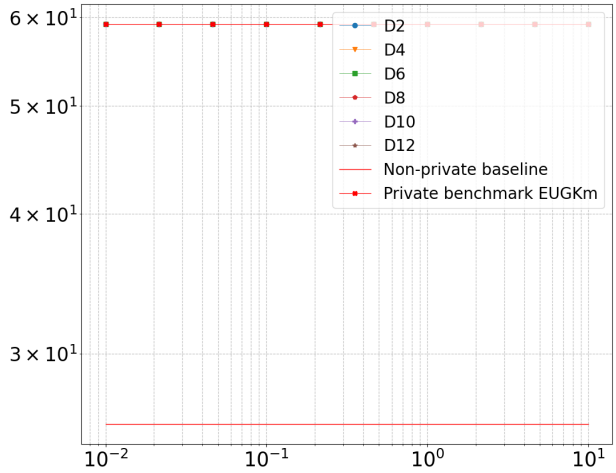


(d) umb: 80, ε: geo

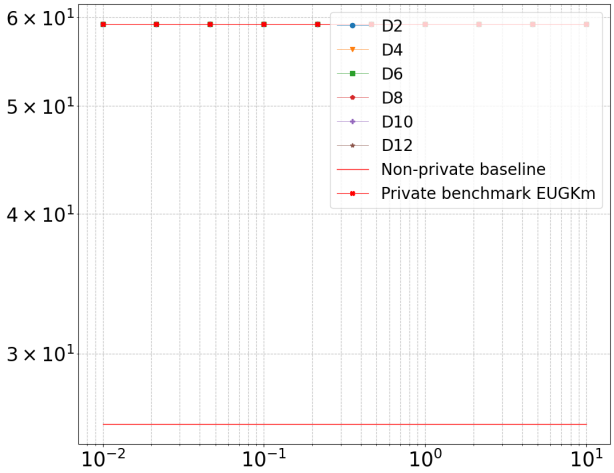
Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

Experimentos sobre el conjunto de datos: Tarvel reviews rating

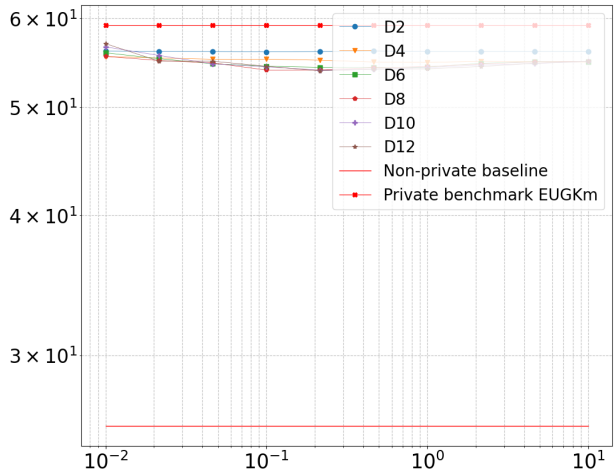


(a) umb: 80, ε: uni

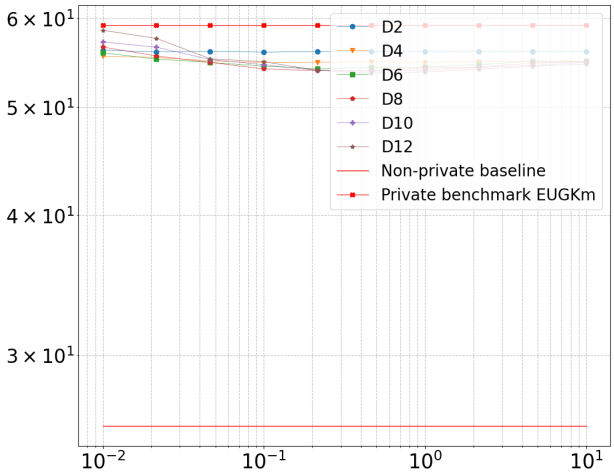


(b) umb: 80, ε: geo

Uniforme



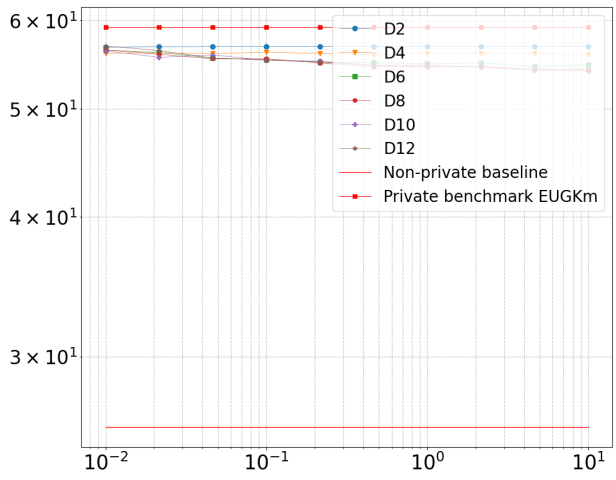
(c) umb: 80, ε: uni



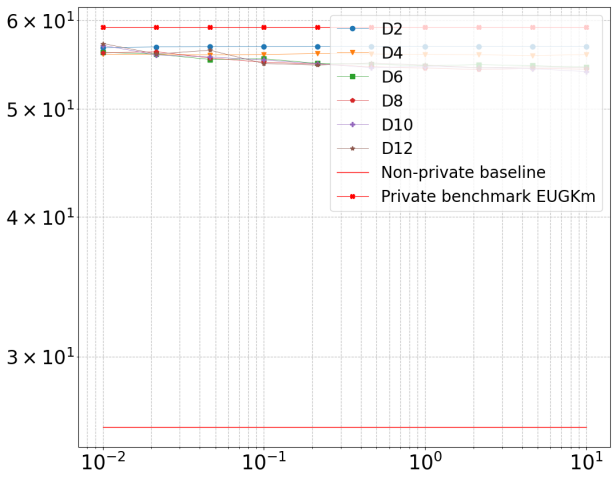
(d) umb: 80, ε: geo

Multi-cuantil

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

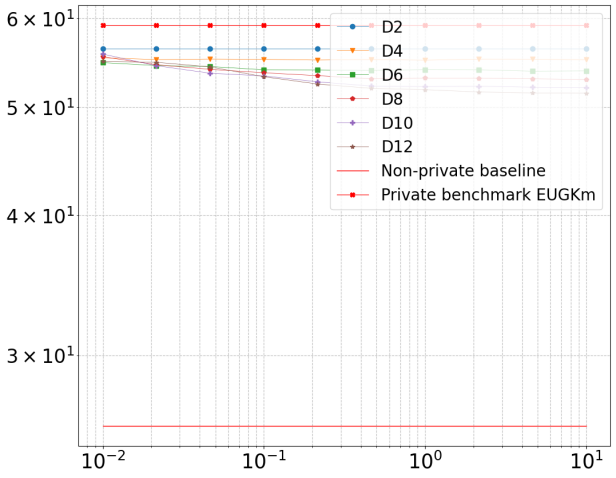


(a) umb: 80, ε: uni

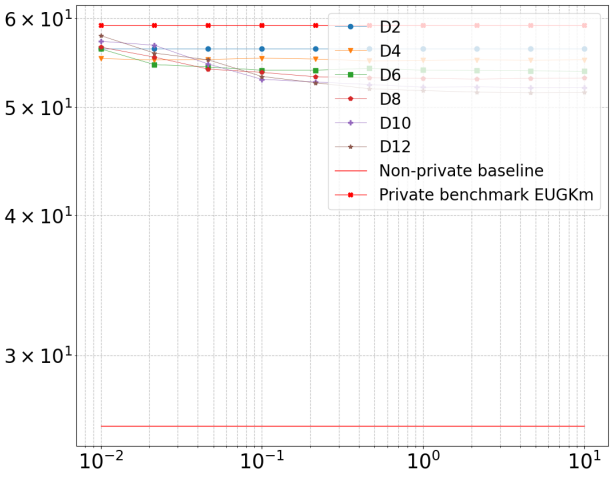


(b) umb: 80, ε: geo

Mediano



(c) umb: 80, ε: uni



(d) umb: 80, ε: geo

Binario

Eje x: ε en escala logarítmica. **Eje y**: NICV en escala logarítmica.

Anexo B

Tablas

S1 - NICV

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	656.1	379.3	1626.3	595.0	1162.2	1346.9	2143.2	2443.8	1626.3	595.3	1553.6	1681.4	4491.9	3640.3
0.02	304.3	268.0	1626.3	432.9	468.1	672.8	631.0	947.0	1626.3	472.3	742.7	771.1	1921.9	3813.3
0.05	316.2	371.5	1626.3	294.0	354.3	316.7	365.1	486.9	1626.3	285.3	304.4	387.4	877.7	2771.7
0.10	168.8	527.6	1626.3	304.7	221.3	152.1	231.2	246.3	1626.3	298.7	234.8	197.0	441.3	731.5
0.21	98.7	195.4	1626.3	302.1	184.9	114.5	123.3	131.3	1626.3	268.1	158.4	93.4	166.1	511.7
0.46	51.8	65.3	1626.3	292.3	163.0	45.3	54.5	47.0	1626.3	291.2	165.4	51.4	54.7	77.8
1.00	42.3	33.0	1626.3	297.5	190.2	39.0	43.6	25.3	1626.3	310.7	181.7	41.2	45.4	51.8
2.15	31.1	43.3	1626.3	304.6	179.0	24.3	16.5	37.7	1626.3	284.2	156.7	39.4	26.5	27.3
4.64	31.6	46.2	1626.3	300.7	162.6	27.3	20.9	17.0	1626.3	303.9	198.8	27.1	11.0	16.1
10.0	20.7	71.2	1626.3	289.2	155.1	11.3	25.8	21.4	1626.3	299.3	184.5	52.1	28.3	16.7

Tabla B.1: S1 - NICV for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	656.1	379.3	1763.4	1131.2	729.2	1603.8	1523.6	1862.7	1712.9	1032.4	1975.8	2451.8	3588.7	4773.0
0.02	304.3	268.0	1601.6	601.6	519.1	1195.5	1167.3	760.4	1615.8	629.2	694.9	884.9	2799.4	3108.2
0.05	316.2	371.5	1612.1	337.6	350.9	370.4	784.7	541.1	1613.7	284.0	381.6	514.7	676.3	1313.1
0.10	168.8	527.6	1616.3	291.4	287.7	277.7	279.4	303.4	1622.4	280.2	297.7	250.9	447.4	489.6
0.21	98.7	195.4	1630.1	288.6	265.1	179.7	121.6	162.4	1629.3	283.6	274.7	168.7	213.1	248.1
0.46	51.8	65.3	1634.8	345.2	268.1	97.6	106.9	83.3	1633.0	326.2	323.1	76.1	60.4	136.8
1.00	42.3	33.0	1633.1	349.6	269.8	72.3	30.9	36.7	1634.6	350.9	304.4	66.6	40.0	41.9
2.15	31.1	43.3	1633.6	352.7	294.0	54.4	38.7	32.0	1633.3	352.9	258.6	58.9	20.9	32.2
4.64	31.6	46.2	1633.5	349.1	301.3	27.4	25.5	20.4	1633.6	357.5	298.4	39.3	19.7	26.4
10.0	20.7	71.2	1633.3	348.6	270.7	28.8	25.0	24.9	1633.4	350.7	261.1	28.8	21.1	19.5

Tabla B.2: S1 - NICV for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	656.1	379.3	5457.6	6415.4	6415.4	6415.4	6415.4	6415.4	5457.6	5936.5	6415.4	6415.4	6415.4	6415.4
0.02	304.3	268.0	1626.3	3541.9	6415.4	5936.5	6415.4	6415.4	1626.3	1626.3	2584.1	6415.4	5457.6	6415.4
0.05	316.2	371.5	836.2	1568.4	1626.3	1626.3	4499.8	5936.5	494.7	1586.3	1626.3	1626.3	2105.2	3063.0
0.10	168.8	527.6	331.4	486.8	1399.1	1506.5	1626.3	1528.3	321.7	300.5	370.6	505.0	1077.9	1836.5
0.21	98.7	195.4	234.2	289.5	370.4	422.0	383.4	1161.5	228.7	277.6	292.6	285.5	306.7	281.4
0.46	51.8	65.3	88.5	230.4	259.6	300.8	327.0	334.9	74.1	121.7	143.2	222.8	249.0	243.6
1.00	42.3	33.0	66.2	28.5	67.0	145.4	206.0	178.9	50.1	36.3	53.3	81.6	51.7	85.6
2.15	31.1	43.3	42.6	34.6	34.2	35.3	51.3	52.7	38.1	21.4	31.2	30.1	14.7	54.4
4.64	31.6	46.2	30.3	25.9	25.8	21.6	32.2	22.7	20.2	6.0	26.0	22.7	26.8	12.7
10.0	20.7	71.2	30.9	26.4	25.6	33.2	27.6	30.8	38.7	25.2	11.2	21.0	0.4	5.4

Tabla B.3: S1 - NICV for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	656.1	379.3	679.1	640.7	569.8	618.2	784.7	1615.2	721.4	675.9	2059.6	2672.7	4529.9	3915.6
0.02	304.3	268.0	516.5	431.1	448.2	554.7	527.0	871.0	582.9	507.5	452.0	711.2	1955.3	2570.3
0.05	316.2	371.5	522.3	327.0	296.2	320.6	342.6	429.4	449.5	276.0	266.0	403.0	1248.4	1315.1
0.10	168.8	527.6	319.0	254.8	197.6	165.2	218.1	260.9	343.4	233.6	184.5	315.2	521.7	538.3
0.21	98.7	195.4	270.5	164.6	73.2	74.6	119.2	142.3	260.4	141.8	115.1	122.2	150.1	321.9
0.46	51.8	65.3	260.8	82.6	43.5	39.1	32.3	44.8	286.9	108.5	25.2	23.5	81.1	150.6
1.00	42.3	33.0	282.7	52.4	32.7	18.9	33.0	31.6	293.1	67.0	9.2	34.1	28.1	37.2
2.15	31.1	43.3	354.8	38.9	34.0	11.8	18.8	12.2	334.9	53.8	24.0	28.3	26.9	19.8
4.64	31.6	46.2	370.5	56.3	17.3	17.1	17.6	1.0	373.4	65.4	16.9	1.3	17.1	26.8
10.0	20.7	71.2	372.3	32.6	12.3	17.4	22.5	11.5	368.6	51.8	29.0	17.1	33.6	27.6

Tabla B.4: S1 - NICV for Multi-quantile partitioner with Threshold 10

S2 - NICV

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	379.4	182.2	1061.5	370.8	681.9	772.3	1163.5	1495.3	1014.2	338.0	1626.6	2025.4	2088.2	2149.7
0.02	247.0	209.8	1014.2	286.0	304.7	335.1	440.9	800.3	1014.2	388.8	293.8	928.8	2168.5	1860.0
0.05	228.2	150.2	1014.2	253.9	196.6	204.2	244.2	248.5	1014.2	257.6	223.9	336.5	561.0	1362.7
0.10	121.9	94.5	1014.2	251.3	181.8	144.2	152.8	186.0	1014.2	245.5	160.3	148.3	247.1	397.8
0.21	51.0	73.9	1014.2	235.3	143.0	87.2	83.0	107.4	1014.2	243.6	137.8	91.9	102.0	252.4
0.46	22.2	18.7	1014.2	247.6	149.1	46.1	38.2	42.1	1014.2	245.2	129.5	22.8	44.4	86.3
1.00	18.7	27.4	1014.2	245.6	135.8	33.6	25.8	28.3	1014.2	258.4	124.9	46.4	29.6	24.8
2.15	15.6	28.1	1014.2	251.0	126.1	22.9	24.4	15.2	1014.2	250.2	120.0	22.2	20.4	20.2
4.64	10.5	52.2	1014.2	250.5	132.3	25.5	18.1	8.6	1014.2	244.2	133.8	22.0	19.3	16.0
10.0	19.6	39.5	1014.2	244.2	127.8	26.0	17.0	7.2	1014.2	243.6	131.0	32.5	23.3	9.7

Tabla B.5: S2 - NICV for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	379.4	182.2	1143.8	724.9	883.4	1328.0	1438.8	1101.6	1056.1	657.6	950.4	1897.9	2410.7	2917.4
0.02	247.0	209.8	1034.8	340.1	420.0	388.8	476.7	454.3	1038.5	437.1	780.3	859.7	2083.4	1648.6
0.05	228.2	150.2	1044.5	216.2	221.3	278.6	254.5	283.1	1036.3	194.9	277.7	388.3	777.5	1018.6
0.10	121.9	94.5	1037.1	211.4	173.2	185.3	207.5	183.6	1035.1	211.2	185.7	183.0	239.4	1033.4
0.21	51.0	73.9	1037.8	228.0	159.9	125.5	95.0	115.5	1038.3	246.2	149.4	93.5	107.4	188.1
0.46	22.2	18.7	1037.6	258.7	139.9	62.1	60.0	65.8	1037.6	259.6	145.4	54.2	62.4	71.5
1.00	18.7	27.4	1037.3	258.4	151.1	46.8	32.0	39.7	1037.2	263.6	157.0	44.2	25.7	32.4
2.15	15.6	28.1	1037.5	263.8	153.0	38.7	30.1	35.0	1037.6	270.9	155.5	33.3	21.3	21.5
4.64	10.5	52.2	1037.5	265.9	154.8	27.6	19.5	20.1	1037.5	266.1	149.6	28.8	23.9	17.8
10.0	19.6	39.5	1037.6	261.5	153.6	28.1	25.8	24.7	1037.6	268.4	149.5	33.2	15.3	20.0

Tabla B.6: S2 - NICV for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	379.4	182.2	3823.5	3823.5	3823.5	3823.5	3823.5	3823.5	2699.8	3823.5	3823.5	3823.5	3823.5	3823.5
0.02	247.0	209.8	1014.2	2137.9	3823.5	3823.5	3823.5	3823.5	986.5	1014.2	2418.9	3542.6	3823.5	3823.5
0.05	228.2	150.2	445.8	977.1	1014.2	1014.2	2418.9	3823.5	357.7	977.1	1014.2	1014.2	1014.2	2137.9
0.10	121.9	94.5	176.5	369.7	902.5	985.3	985.3	1014.2	189.3	248.3	295.7	460.4	485.9	835.9
0.21	51.0	73.9	156.7	218.1	201.6	296.4	338.0	540.3	171.6	177.2	241.1	248.8	252.0	269.9
0.46	22.2	18.7	30.7	149.4	223.6	242.4	174.1	172.2	42.6	120.8	132.5	164.6	198.4	189.5
1.00	18.7	27.4	34.8	41.4	62.5	95.2	107.3	184.0	29.7	17.5	30.1	25.6	38.6	70.4
2.15	15.6	28.1	22.0	21.1	26.5	31.3	27.6	38.0	14.4	15.4	12.4	17.6	11.7	19.5
4.64	10.5	52.2	8.8	21.5	22.7	9.6	20.9	16.6	20.0	6.7	4.6	16.7	14.2	16.4
10.0	19.6	39.5	17.6	12.5	10.8	9.7	11.6	19.8	7.1	10.1	4.5	4.4	8.5	9.9

Tabla B.7: S2 - NICV for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	379.4	182.2	422.5	351.3	837.9	624.6	602.3	1486.1	413.8	320.0	799.5	1285.9	2463.6	2581.7
0.02	247.0	209.8	366.9	275.9	274.8	382.6	306.1	648.6	324.5	303.5	535.3	1012.9	1288.0	1735.7
0.05	228.2	150.2	280.4	225.5	241.9	247.1	262.6	277.2	290.5	208.1	291.9	328.0	303.0	499.8
0.10	121.9	94.5	230.9	158.7	154.2	151.8	175.9	180.6	237.4	138.1	126.5	204.5	199.0	322.0
0.21	51.0	73.9	184.8	90.7	78.8	77.1	68.6	109.6	179.7	84.6	84.9	84.5	89.9	163.4
0.46	22.2	18.7	200.7	73.0	30.7	39.7	36.6	40.5	218.9	62.1	29.4	27.2	41.8	59.5
1.00	18.7	27.4	229.8	41.8	17.2	24.1	19.0	25.5	232.9	59.2	18.3	24.7	16.3	44.2
2.15	15.6	28.1	234.5	36.0	20.5	19.2	14.8	16.8	229.2	39.8	23.3	13.7	16.4	11.5
4.64	10.5	52.2	221.7	29.9	25.7	13.4	13.0	13.0	238.7	41.0	15.9	24.2	17.4	20.1
10.0	19.6	39.5	233.2	33.3	15.3	12.4	17.6	14.0	229.5	37.1	21.6	13.1	16.2	13.1

Tabla B.8: S2 - NICV for Multi-quantile partitioner with Threshold 10

S3 - NICV

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	302.4	98.4	655.1	287.5	362.1	348.4	700.8	532.8	655.1	431.2	474.5	787.2	700.2	1692.5
0.02	132.0	150.9	655.1	165.2	224.8	189.8	235.5	431.2	655.1	173.6	247.9	645.6	619.3	1433.5
0.05	129.5	96.0	655.1	133.4	108.4	114.9	107.8	168.7	655.1	125.9	106.3	246.0	468.3	379.5
0.10	67.9	48.4	655.1	123.8	78.7	76.7	82.0	88.4	655.1	126.0	79.1	82.1	175.4	208.4
0.21	49.5	37.9	655.1	128.9	64.6	48.5	47.4	49.5	655.1	123.5	70.7	39.3	50.5	71.7
0.46	27.5	21.4	655.1	125.8	69.0	30.4	21.6	24.6	655.1	127.3	65.9	26.0	23.4	35.3
1.00	15.9	13.4	655.1	124.5	64.6	22.2	15.2	15.1	655.1	124.6	56.6	21.5	10.6	13.8
2.15	14.9	16.1	655.1	124.5	66.3	16.7	10.4	9.0	655.1	133.0	64.1	17.4	8.2	9.6
4.64	11.4	6.8	655.1	126.1	71.2	20.5	8.8	6.9	655.1	126.4	58.5	18.0	6.1	7.6
10.0	13.9	9.8	655.1	126.3	66.3	18.3	6.7	7.4	655.1	129.9	62.4	12.9	6.6	1.3

Tabla B.9: S3 - NICV for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	302.4	98.4	757.9	400.1	588.5	888.6	585.9	756.9	733.0	427.7	628.5	855.6	985.4	1700.5
0.02	132.0	150.9	686.1	217.2	287.2	194.0	365.1	303.9	685.9	179.9	367.0	434.5	792.6	833.4
0.05	129.5	96.0	688.0	132.5	148.1	135.1	149.3	271.2	686.7	144.5	141.2	236.8	302.4	497.1
0.10	67.9	48.4	690.3	121.9	110.7	93.4	105.2	117.7	687.9	131.4	109.6	99.2	199.8	105.9
0.21	49.5	37.9	688.7	119.5	90.9	75.3	73.8	82.2	688.0	121.3	91.0	64.3	64.9	155.5
0.46	27.5	21.4	688.4	118.7	89.2	47.7	34.8	39.2	689.0	112.9	99.0	38.5	35.8	44.2
1.00	15.9	13.4	688.7	113.5	97.1	34.1	23.1	23.9	688.4	116.4	90.3	33.6	21.0	20.6
2.15	14.9	16.1	688.3	113.5	93.1	25.2	18.3	13.3	688.4	113.4	89.6	28.6	13.3	13.3
4.64	11.4	6.8	688.2	113.2	101.0	29.1	8.7	8.8	688.3	113.4	102.7	25.8	12.6	6.0
10.0	13.9	9.8	688.2	113.3	100.9	24.3	14.1	10.0	688.2	113.3	88.0	35.3	13.5	9.8

Tabla B.10: S3 - NICV for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	302.4	98.4	181.1	182.3	257.9	310.0	532.1	1075.9	170.3	205.4	464.7	681.6	1383.1	1175.5
0.02	132.0	150.9	149.5	150.2	207.6	265.9	392.7	455.9	145.3	135.2	208.3	486.6	614.9	791.9
0.05	129.5	96.0	136.3	113.7	137.4	228.1	312.7	516.5	136.3	107.2	120.7	247.2	304.7	724.4
0.10	67.9	48.4	130.9	74.5	99.2	148.9	205.2	213.0	133.8	58.7	84.9	147.4	234.4	529.8
0.21	49.5	37.9	57.8	43.3	73.2	112.3	130.1	151.5	67.3	49.6	46.9	89.0	193.8	226.0
0.46	27.5	21.4	26.0	24.0	36.6	53.2	79.4	101.3	38.6	20.6	29.6	45.7	79.7	126.1
1.00	15.9	13.4	20.1	16.0	18.4	23.0	42.2	63.5	15.9	9.9	13.7	22.4	38.7	67.4
2.15	14.9	16.1	10.2	8.1	11.3	9.2	12.3	21.5	8.7	10.1	6.2	9.4	6.9	18.2
4.64	11.4	6.8	9.8	8.2	4.7	6.3	5.9	9.4	6.5	5.0	2.6	6.6	5.4	6.9
10.0	13.9	9.8	7.0	9.9	4.7	8.2	4.7	5.1	11.1	7.3	3.5	7.5	3.9	6.9

Tabla B.11: S3 - NICV for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	302.4	98.4	276.3	217.1	294.9	458.5	993.5	324.8	325.2	357.9	768.2	1533.4	1009.6	1392.5
0.02	132.0	150.9	227.8	194.3	236.7	186.2	215.0	481.9	230.1	185.5	216.4	292.5	699.5	648.1
0.05	129.5	96.0	178.6	122.1	114.1	159.4	151.0	149.0	155.2	136.0	126.4	158.0	251.9	676.5
0.10	67.9	48.4	141.8	89.4	81.2	91.6	87.7	115.2	130.8	102.7	75.8	93.6	112.6	394.7
0.21	49.5	37.9	118.0	69.1	49.8	49.7	49.1	55.0	113.6	66.7	52.2	53.7	70.0	93.2
0.46	27.5	21.4	105.4	47.0	28.2	23.8	28.3	28.5	103.8	47.7	26.5	29.3	34.7	53.1
1.00	15.9	13.4	98.6	32.5	16.5	12.1	9.1	12.9	100.3	27.9	13.5	10.4	15.2	23.0
2.15	14.9	16.1	102.8	28.1	5.1	8.3	10.6	8.5	101.7	25.8	6.7	7.6	11.1	11.7
4.64	11.4	6.8	103.2	23.6	8.1	7.5	7.9	8.2	105.5	19.2	7.2	11.2	7.2	8.9
10.0	13.9	9.8	109.5	23.7	9.6	7.5	7.8	7.8	103.5	20.8	6.6	8.4	7.8	8.0

Tabla B.12: S3 - NICV for Multi-quantile partitioner with Threshold 10

S4 - NICV

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	363.1	174.5	913.4	344.6	682.3	444.0	634.5	985.5	909.4	467.6	629.4	681.5	820.8	1232.1
0.02	218.7	95.7	909.4	247.7	199.0	290.5	291.1	249.7	909.4	259.7	284.5	409.4	545.1	1235.8
0.05	147.5	104.4	909.4	223.1	160.8	180.9	184.4	230.0	909.4	223.5	172.3	176.4	259.6	382.3
0.10	100.7	99.3	909.4	221.8	129.0	110.4	114.1	110.3	909.4	217.1	119.1	97.9	167.0	247.2
0.21	49.2	34.1	909.4	217.6	106.5	56.6	59.3	73.7	909.4	216.0	97.5	49.6	70.3	119.0
0.46	36.9	25.6	909.4	216.8	92.7	43.2	22.1	25.9	909.4	217.9	97.1	27.8	27.0	37.9
1.00	26.6	22.2	909.4	214.2	81.8	21.7	13.7	13.5	909.4	215.5	88.1	22.6	14.6	19.2
2.15	13.5	13.4	909.4	218.1	80.8	21.0	11.4	7.9	909.4	217.6	86.5	22.4	7.3	8.4
4.64	14.7	13.3	909.4	217.9	86.9	21.6	5.7	6.4	909.4	218.1	87.1	20.5	5.6	4.8
10.0	13.7	5.8	909.4	217.4	80.0	20.2	5.8	4.3	909.4	217.5	88.7	21.8	5.8	4.9

Tabla B.13: S4 - NICV for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	363.1	174.5	906.3	517.7	279.5	518.0	489.6	833.3	894.5	380.1	531.3	731.1	1127.8	1340.3
0.02	218.7	95.7	905.6	242.1	291.9	247.0	243.9	818.7	904.8	233.9	235.9	581.3	655.5	1299.3
0.05	147.5	104.4	909.7	167.6	174.0	171.2	154.9	191.5	909.1	155.8	182.6	336.9	444.4	501.0
0.10	100.7	99.3	909.5	142.7	129.5	117.8	124.5	123.3	909.7	146.2	132.8	125.6	138.4	195.6
0.21	49.2	34.1	909.6	149.4	130.7	82.8	82.9	84.7	909.7	144.6	114.3	86.5	80.5	106.9
0.46	36.9	25.6	909.6	149.7	124.0	59.1	51.9	52.5	909.7	148.9	131.3	58.5	46.0	57.9
1.00	26.6	22.2	909.6	151.1	132.6	50.0	33.3	31.9	909.7	153.7	133.6	47.8	26.8	29.1
2.15	13.5	13.4	909.6	149.1	138.3	41.0	21.8	22.2	909.6	148.9	134.4	36.0	17.1	17.5
4.64	14.7	13.3	909.6	148.1	127.8	39.0	17.6	13.4	909.6	149.2	129.5	37.1	16.1	15.1
10.0	13.7	5.8	909.6	149.1	140.1	39.2	17.4	17.8	909.6	149.9	130.5	37.9	15.3	12.0

Tabla B.14: S4 - NICV for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	363.1	174.5	243.8	218.5	575.3	550.2	785.2	846.5	228.4	375.3	598.0	513.2	812.2	1380.1
0.02	218.7	95.7	222.3	183.1	231.4	274.0	434.4	651.2	225.4	159.4	237.1	621.4	533.4	916.6
0.05	147.5	104.4	214.7	114.4	157.1	221.7	299.8	327.4	217.7	119.2	169.5	245.8	487.1	676.9
0.10	100.7	99.3	151.4	76.6	95.9	151.1	169.9	231.2	142.2	76.0	95.4	150.6	209.3	545.2
0.21	49.2	34.1	81.3	52.7	65.5	103.9	136.4	157.6	79.8	43.9	60.7	107.6	137.3	246.5
0.46	36.9	25.6	44.2	26.4	42.8	62.1	88.6	94.9	46.8	25.6	36.1	46.6	82.1	143.6
1.00	26.6	22.2	20.8	9.2	12.3	34.6	47.2	67.4	18.7	8.7	12.9	22.3	44.2	63.1
2.15	13.5	13.4	10.3	6.3	5.5	6.4	8.8	19.8	6.6	7.0	5.5	7.0	13.6	17.8
4.64	14.7	13.3	6.1	4.9	6.3	4.4	4.6	5.6	4.8	3.9	3.0	3.8	4.1	6.5
10.0	13.7	5.8	6.0	6.2	5.8	6.5	3.4	3.5	7.9	5.1	2.9	2.6	3.8	3.6

Tabla B.15: S4 - NICV for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	363.1	174.5	344.6	229.6	257.4	607.2	263.0	513.0	256.8	232.8	464.8	607.2	826.5	608.4
0.02	218.7	95.7	270.9	177.4	168.5	199.1	342.0	402.9	249.8	171.1	193.6	577.8	528.1	751.8
0.05	147.5	104.4	186.8	142.0	120.7	111.7	168.4	152.7	212.8	153.4	183.1	150.5	218.9	596.9
0.10	100.7	99.3	148.8	103.0	87.4	83.9	103.3	88.1	166.1	110.5	85.5	86.3	106.5	165.4
0.21	49.2	34.1	141.1	88.3	57.2	49.1	51.2	58.2	156.5	88.8	54.7	56.4	73.8	91.1
0.46	36.9	25.6	137.0	56.6	35.7	27.1	26.8	35.8	147.2	65.8	33.2	28.5	28.1	50.8
1.00	26.6	22.2	142.3	46.9	18.4	16.1	21.4	19.8	142.0	46.7	19.5	13.7	18.5	27.0
2.15	13.5	13.4	142.5	35.5	11.7	7.2	10.8	9.8	142.6	43.4	12.9	8.5	8.6	11.0
4.64	14.7	13.3	141.3	32.9	9.2	9.2	9.1	9.4	140.0	37.6	8.8	8.8	9.1	7.1
10.0	13.7	5.8	141.9	33.3	16.8	12.9	11.4	13.3	140.5	35.6	16.5	10.1	6.8	8.5

Tabla B.16: S4 - NICV for Multi-quantile partitioner with Threshold 10

Shuttle - NICV

			Binary										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	562665.7	841628.6	832998.6	559820.2	559680.8	559544.0	528158.0	839020.1	825345.4	559810.9	585797.0	559531.3	756261.4
0.02	564169.5	846845.8	832995.8	559819.4	559664.8	559523.3	524662.0	846845.8	831468.2	559821.6	559676.3	559508.8	613486.8
0.05	846770.8	846845.8	831469.0	559831.4	559664.6	559506.7	524661.6	846845.8	836058.6	559821.8	559675.3	559518.4	554273.1
0.10	839020.7	846845.8	836060.9	559832.5	559676.8	559516.3	524651.4	846845.8	836059.5	559833.8	559675.7	559508.2	524659.2
0.21	862278.6	846845.8	836061.1	559832.3	559675.6	559517.6	524658.7	846845.8	836060.3	559833.7	559676.3	559517.8	524658.8
0.46	48012.4	846845.8	836061.0	559833.5	559675.5	559518.2	524658.4	846845.8	836060.0	559834.7	559676.1	559518.3	524658.3
1.00	45701.3	846845.8	836060.9	559834.3	559676.8	559518.0	524658.2	846845.8	836060.6	559834.2	559676.6	559518.9	524658.0
2.15	50887.5	846845.8	836060.1	559833.9	559677.2	559518.4	524658.1	846845.8	836060.3	559834.0	559676.4	559518.2	524658.4
4.64	51239.9	846845.8	836060.8	559832.6	559676.0	559519.0	524658.4	846845.8	836060.2	559834.3	559676.2	559518.5	524658.9
10.0	50749.9	846845.8	836060.2	559834.4	559676.1	559518.1	524658.8	846845.8	836061.0	559834.5	559675.9	559518.2	524658.2

Tabla B.17: Shuttle - NICV for Binary partitioner with Threshold 80

			Median										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	562665.7	821213.1	808781.4	230033.2	139553.8	226528.7	439676.9	827359.9	816094.4	371276.9	350855.4	526213.7	697766.9
0.02	564169.5	864777.9	809662.1	147113.8	86637.7	288562.5	320278.2	865804.9	810938.9	140121.8	219368.9	350815.3	506470.7
0.05	846770.8	869292.1	860205.0	208041.0	90942.5	68532.2	152938.8	869300.1	854741.4	128003.0	124397.6	171151.9	202485.7
0.10	839020.7	869303.8	858609.8	248957.9	106855.4	119868.1	71285.8	869303.9	858510.9	167648.4	144200.0	82729.2	168211.6
0.21	862278.6	869304.7	858458.5	130488.1	138416.5	169836.3	99803.6	869304.2	858331.7	234323.9	169153.6	93251.3	146123.6
0.46	48012.4	869299.1	858392.3	109725.9	130497.8	58421.3	91056.3	869302.3	858634.5	306279.5	104981.4	185339.9	120352.3
1.00	45701.3	819310.8	858638.7	377856.6	86057.4	114384.2	94756.7	817709.5	858636.3	572528.6	149460.2	151100.4	86091.2
2.15	50887.5	816916.6	814064.2	779492.5	551661.2	272629.9	250658.4	815848.1	858635.9	718752.2	553305.6	299917.9	86868.8
4.64	51239.9	815433.2	808076.8	1129084.5	890520.9	820126.7	664705.8	815773.2	807067.2	934476.4	850599.3	831598.3	723889.1
10.0	50749.9	817559.7	808971.3	857474.3	855123.9	909050.1	1185591.7	813722.0	807480.6	858703.8	908310.9	777033.7	783385.7

Tabla B.18: Shuttle - NICV for Median partitioner with Threshold 80

			Uniform										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	562665.7	16447.7	451322.9	768554.5	768557.6	820824.0	794690.9	12183.6	174683.2	451718.5	475689.3	504536.7	768561.2
0.02	564169.5	15345.0	10979.1	9548.9	229510.1	504461.0	582869.9	11358.3	11546.0	10482.2	10420.6	9546.5	9547.3
0.05	846770.8	11171.8	18609.8	14409.3	9608.7	10426.8	10131.9	9539.9	23518.1	20828.3	16689.1	11976.5	14424.9
0.10	839020.7	9519.8	12872.3	18785.5	16683.3	14238.1	12537.9	9520.3	14157.0	10993.5	11513.7	10470.0	11907.7
0.21	862278.6	136390.2	6461.1	7215.6	5973.9	9049.9	18055.8	127754.2	9490.5	3433.3	5961.7	11342.6	8077.5
0.46	48012.4	51308.3	8138.7	5759.7	4999.6	5800.9	6328.3	51635.0	11934.0	2261.0	2362.4	4562.8	3824.5
1.00	45701.3	53312.3	2344.9	8669.0	4382.1	4012.2	2957.9	53069.3	2078.0	649.6	1138.5	1003.4	647.2
2.15	50887.5	34458.3	3314.6	1694.5	1645.0	3965.0	4079.1	17338.1	1104.5	1215.3	1058.7	683.6	737.4
4.64	51239.9	15159.4	4202.6	1010.6	884.5	1078.8	1082.6	9340.8	1518.5	935.3	1339.4	1285.6	587.0
10.0	50749.9	27127.7	6237.3	14865.1	45898.6	52131.2	9883.3	30665.5	7740.6	4106.8	7717.0	7343.2	9017.9

Tabla B.19: Shuttle - NICV for Uniform partitioner with Threshold 80

			Multi-quantile										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	562665.7	815419.4	805174.2	350460.5	299650.0	178032.6	175193.7	812595.1	807458.8	235335.0	287178.1	464731.5	334175.4
0.02	564169.5	815675.2	807854.9	123236.1	117972.7	90964.8	193666.3	816878.3	801928.6	92380.1	126278.1	253761.4	221486.3
0.05	846770.8	813505.6	801826.8	78600.6	78900.6	79610.5	63138.7	813901.6	806356.9	156051.0	143186.1	125682.8	200872.7
0.10	839020.7	812436.1	806789.0	72116.5	53447.8	56610.7	33046.7	812414.1	804169.1	99971.1	59276.9	87407.6	41036.8
0.21	862278.6	810061.1	801845.8	68795.5	73086.9	37380.9	56317.5	810103.3	803553.4	37769.0	68627.9	45662.6	43410.8
0.46	48012.4	810038.7	798825.4	36926.8	41896.7	65339.7	26871.4	810045.5	798817.9	52141.6	52939.7	39310.6	28521.8
1.00	45701.3	810039.3	798668.9	30857.4	35920.6	32288.9	30924.7	810037.4	798827.3	26577.8	49805.3	42958.2	23103.7
2.15	50887.5	810050.0	798631.0	26894.4	41321.7	48024.4	23918.3	810047.1	798628.9	33348.7	54363.8	24365.6	26855.2
4.64	51239.9	810046.5	798631.4	14066.0	23063.4	45818.5	33474.4	810046.4	798637.6	12563.4	36774.2	24789.6	28037.9
10.0	50749.9	810047.2	798626.6	4081.3	22271.0	21723.5	6602.8	810047.4	798620.6	1195.2	9600.7	30694.5	16958.4

Tabla B.20: Shuttle - NICV for Multi-quantile partitioner with Threshold 80

Skin segmentation - NICV

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.9	14.2	47.0	17.3	1.8	2.2	1.0	0.6	47.5	15.4	1.5	2.1	0.8	1.1
0.02	0.8	5.2	45.6	15.3	0.8	1.8	0.3	0.4	46.5	15.3	3.1	1.6	0.2	0.4
0.05	0.8	2.2	47.0	15.3	0.8	1.5	0.1	0.1	47.5	15.2	0.8	1.6	0.1	0.2
0.10	0.9	1.3	47.9	17.7	0.8	1.5	0.1	0.1	45.6	15.2	0.8	1.6	0.1	0.1
0.21	0.9	0.1	47.5	15.2	0.8	1.6	0.1	0.0	46.1	15.2	0.8	1.6	0.1	0.0
0.46	1.3	0.2	46.1	15.2	0.8	1.6	0.1	0.1	47.0	15.2	5.4	1.6	0.1	0.0
1.00	1.6	0.1	48.4	15.2	0.8	1.6	0.1	0.0	47.5	15.2	5.4	1.6	0.1	0.0
2.15	2.1	0.0	47.0	15.2	0.8	1.6	0.1	0.0	46.5	15.2	0.8	1.6	0.1	0.0
4.64	1.9	0.0	47.5	15.2	0.8	1.6	0.1	0.0	47.0	15.2	0.8	1.6	0.1	0.0
10.0	0.6	0.0	47.0	15.2	0.8	1.6	0.1	0.0	47.5	15.2	0.8	1.6	0.1	0.0

Tabla B.21: Skin Segmentation - NICV for Binary partitioner with Threshold 80

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.9	14.2	60.7	11.2	4.3	2.8	1.8	1.8	60.8	12.8	3.9	2.5	1.4	2.0
0.02	0.8	5.2	60.8	12.9	3.7	1.9	0.9	1.8	60.7	12.9	3.5	2.0	1.0	1.0
0.05	0.8	2.2	60.7	11.5	3.4	2.0	0.8	0.7	60.8	12.2	5.1	2.1	0.7	0.4
0.10	0.9	1.3	60.7	12.0	4.2	2.1	0.6	0.4	60.8	12.0	3.7	2.3	0.5	0.2
0.21	0.9	0.1	60.7	11.4	4.2	2.1	0.4	0.2	60.9	11.9	3.5	2.3	0.4	0.2
0.46	1.3	0.2	60.8	11.8	4.0	2.0	0.3	0.2	60.8	12.3	4.7	2.1	0.3	0.1
1.00	1.6	0.1	60.8	12.2	4.3	2.4	0.3	0.1	60.7	11.9	3.8	2.3	0.2	0.1
2.15	2.1	0.0	60.7	10.6	4.7	2.2	0.2	0.1	60.8	13.4	4.2	2.4	0.2	0.1
4.64	1.9	0.0	110.2	40.6	3.8	2.4	0.2	0.1	77.5	38.9	4.2	2.3	0.2	0.1
10.0	0.6	0.0	126.9	66.5	28.5	4.8	0.8	0.1	126.8	60.4	12.9	2.0	0.2	0.1

Tabla B.22: Skin Segmentation - NICV for Median partitioner with Threshold 80

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.9	14.2	0.3	8.1	28.1	20.4	23.6	22.0	0.5	5.1	10.4	17.6	20.6	17.6
0.02	0.8	5.2	0.1	0.5	1.2	3.8	5.6	25.9	0.1	1.1	4.5	4.5	3.9	4.1
0.05	0.8	2.2	0.1	0.4	2.6	0.7	0.3	1.1	0.2	0.0	0.3	0.9	0.6	0.6
0.10	0.9	1.3	0.1	0.1	0.2	0.4	0.1	0.2	0.1	0.1	0.0	0.1	0.5	0.2
0.21	0.9	0.1	0.0	0.0	0.1	0.0	0.2	0.1	0.1	0.0	0.1	0.1	0.1	0.2
0.46	1.3	0.2	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.1	0.0
1.00	1.6	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2.15	2.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4.64	1.9	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
10.0	0.6	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tabla B.23: Skin Segmentation - NICV for Uniform partitioner with Threshold 80

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.9	14.2	43.2	4.4	3.4	5.6	4.2	3.9	43.4	3.1	2.0	3.2	4.9	4.5
0.02	0.8	5.2	43.3	4.0	0.7	2.1	1.9	3.0	43.1	2.0	1.2	1.5	2.3	2.8
0.05	0.8	2.2	41.9	1.8	0.3	0.7	0.8	1.0	43.3	2.7	0.3	0.8	0.7	0.9
0.10	0.9	1.3	43.1	2.4	0.2	0.2	0.2	0.4	43.2	2.6	0.3	0.1	0.2	0.2
0.21	0.9	0.1	42.4	2.5	0.1	0.1	0.1	0.1	42.9	2.4	0.1	0.1	0.1	0.1
0.46	1.3	0.2	42.8	2.4	0.1	0.1	0.0	0.0	42.8	2.4	0.1	0.1	0.1	0.0
1.00	1.6	0.1	43.0	2.4	0.1	0.1	0.0	0.1	42.1	2.3	0.1	0.1	0.0	0.1
2.15	2.1	0.0	43.3	2.3	0.1	0.1	0.1	0.0	43.6	2.4	0.1	0.1	0.0	0.0
4.64	1.9	0.0	42.9	2.5	0.1	0.1	0.1	0.1	42.4	2.6	0.2	0.1	0.1	0.1
10.0	0.6	0.0	42.1	1.7	0.1	0.1	0.1	0.1	42.8	2.2	0.1	0.1	0.1	0.1

Tabla B.24: Skin Segmentation - NICV for Multi-quantile partitioner with Threshold 80

Tarvel review ratings - NICV

			Binary										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	128.1	117.3	113.4	111.2	113.8	115.0	111.8	117.3	113.0	117.3	118.1	120.7	123.3
0.02	128.1	117.3	112.5	110.0	110.3	109.9	111.4	117.3	112.4	110.4	113.6	118.9	115.4
0.05	128.1	117.3	112.8	109.5	108.3	106.7	109.4	117.3	112.6	109.3	108.4	110.6	112.5
0.10	128.1	117.3	112.7	108.1	106.9	105.5	105.2	117.3	113.2	108.0	107.0	104.1	105.3
0.21	128.1	117.3	112.3	108.1	105.6	103.0	102.1	117.3	112.8	108.0	105.2	103.0	102.6
0.46	128.1	117.3	112.6	107.9	104.3	101.3	100.3	117.3	112.0	108.7	104.8	101.9	100.2
1.00	128.1	117.3	112.3	108.2	104.6	101.1	99.8	117.3	112.2	108.0	104.5	100.8	99.4
2.15	128.1	117.3	112.8	108.1	104.5	101.2	98.8	117.3	112.4	107.8	104.2	101.0	98.8
4.64	128.1	117.3	112.7	107.5	104.3	100.8	98.5	117.3	112.3	107.7	104.5	100.7	98.6
10.0	128.1	117.3	112.5	107.7	104.0	100.7	98.3	117.3	112.4	107.5	104.7	100.7	98.7

Tabla B.25: Tarvel Review Ratings - NICV for Binary partitioner with Threshold 80

			Median										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	128.1	118.9	116.2	117.6	117.7	117.2	119.5	118.7	115.7	116.6	116.4	119.7	120.6
0.02	128.1	119.1	116.4	116.5	115.7	114.4	117.4	119.0	115.7	115.8	116.9	115.4	115.9
0.05	128.1	119.2	116.0	114.1	113.7	115.4	114.1	119.2	115.4	113.4	114.1	114.5	117.5
0.10	128.1	119.3	116.6	113.3	113.7	113.1	113.1	119.3	115.7	113.7	112.2	113.2	111.7
0.21	128.1	119.3	116.1	112.3	111.9	112.7	112.3	119.3	116.0	111.7	111.4	111.5	111.0
0.46	128.1	119.3	115.6	112.0	110.2	110.7	110.9	119.3	115.8	111.4	110.2	109.8	111.9
1.00	128.1	119.3	116.2	111.5	110.1	110.4	111.0	119.3	115.7	110.7	109.7	110.4	111.0
2.15	128.1	119.3	116.2	111.9	110.1	110.4	110.0	119.3	115.5	111.2	109.1	109.9	109.8
4.64	128.1	119.3	116.2	110.4	109.2	109.3	108.7	119.3	115.1	110.7	109.6	109.4	110.2
10.0	128.1	119.3	115.5	111.1	108.3	109.3	108.9	119.3	115.6	110.1	109.2	108.1	109.9

Tabla B.26: Tarvel Review Ratings - NICV for Median partitioner with Threshold 80

			Uniform										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
0.02	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
0.05	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
0.10	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
0.21	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
0.46	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
1.00	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
2.15	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
4.64	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1
10.0	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1	128.1

Tabla B.27: Tarvel Review Ratings - NICV for Uniform partitioner with Threshold 80

			Multi-quantile										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	128.1	116.3	113.9	115.5	114.0	118.2	119.6	116.7	114.0	115.6	118.1	120.4	125.8
0.02	128.1	116.2	113.2	112.9	112.1	114.4	111.9	115.9	113.5	112.8	114.1	118.0	122.0
0.05	128.1	116.1	112.7	110.7	111.1	110.7	111.8	116.2	112.0	111.3	111.3	112.6	112.9
0.10	128.1	115.9	112.7	109.6	108.0	109.4	109.6	115.8	111.2	109.7	108.5	110.3	111.6
0.21	128.1	116.1	112.3	109.2	108.1	108.0	107.7	116.1	111.4	108.5	107.8	107.9	107.9
0.46	128.1	116.4	111.5	109.1	109.0	108.4	108.4	116.2	111.6	109.1	107.8	107.5	106.7
1.00	128.1	116.2	111.3	109.3	109.5	109.1	108.5	116.0	111.2	109.3	108.9	108.0	107.2
2.15	128.1	116.1	111.8	110.9	110.6	110.2	109.5	116.2	111.7	110.3	109.3	109.1	108.4
4.64	128.1	116.2	111.6	111.6	111.6	110.8	111.0	116.1	111.8	111.8	111.0	110.1	109.7
10.0	128.1	116.2	111.5	111.7	111.7	111.8	111.8	116.2	111.7	111.7	111.6	111.3	110.6

Tabla B.28: Tarvel Review Ratings - NICV for Multi-quantile partitioner with Threshold 80

S1 - CI

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	2.25	-1.00	1.50	4.25	4.86	5.17	4.50	-1.00	2.00	3.50	4.83	5.50	6.00
0.02	1.80	2.00	-1.00	1.67	3.00	3.67	4.30	4.44	-1.00	2.33	2.25	3.56	6.17	5.33
0.05	2.20	1.50	-1.00	1.25	2.20	1.80	2.60	3.30	-1.00	1.60	2.30	2.40	3.62	5.00
0.10	1.80	3.00	-1.00	2.00	1.50	1.10	2.00	1.80	-1.00	1.70	1.60	1.60	2.22	3.67
0.21	1.20	1.75	-1.00	1.90	1.40	1.30	1.10	1.10	-1.00	1.40	1.40	0.90	1.50	3.00
0.46	0.90	1.00	-1.00	1.70	1.30	0.60	0.70	0.60	-1.00	1.80	1.80	0.70	0.80	0.80
1.00	0.70	0.50	-1.00	1.80	1.60	0.60	0.70	0.40	-1.00	1.90	1.60	0.70	0.80	0.70
2.15	0.60	0.75	-1.00	1.90	1.60	0.40	0.30	0.70	-1.00	1.60	1.20	0.70	0.50	0.50
4.64	0.60	0.75	-1.00	1.80	1.60	0.50	0.40	0.30	-1.00	1.80	1.80	0.50	0.20	0.30
10.0	0.40	1.25	-1.00	1.60	1.40	0.20	0.50	0.40	-1.00	1.70	1.60	1.00	0.50	0.30

Tabla B.29: S1 - CI for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	2.25	-1.00	-1.00	4.14	4.00	4.00	5.50	-1.00	5.50	2.50	6.00	6.00	7.00
0.02	1.80	2.00	-1.00	3.00	2.40	3.56	3.43	3.62	-1.00	2.83	3.33	4.22	4.25	3.00
0.05	2.20	1.50	-1.00	1.67	1.80	2.00	3.00	3.11	-1.00	1.20	2.30	2.50	3.50	4.50
0.10	1.80	3.00	-1.00	1.40	2.30	1.90	2.00	2.20	-1.00	1.20	2.00	1.30	2.60	3.30
0.21	1.20	1.75	-1.00	1.90	1.90	1.40	0.90	1.00	-1.00	1.40	1.70	1.30	1.30	1.60
0.46	0.90	1.00	-1.00	1.70	1.70	1.00	0.70	0.70	-1.00	1.30	1.90	0.70	0.60	0.80
1.00	0.70	0.50	-1.00	1.30	1.90	0.80	0.30	0.40	-1.00	1.50	2.50	0.90	0.50	0.40
2.15	0.60	0.75	-1.00	1.20	2.40	0.70	0.60	0.50	-1.00	1.20	1.90	0.80	0.30	0.50
4.64	0.60	0.75	-1.00	1.00	2.40	0.30	0.40	0.30	-1.00	1.40	2.40	0.50	0.30	0.40
10.0	0.40	1.25	-1.00	1.10	2.00	0.30	0.40	0.40	-1.00	1.00	2.00	0.30	0.30	0.30

Tabla B.30: S1 - CI for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	2.25	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.02	1.80	2.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.05	2.20	1.50	1.00	-1.00	-1.00	-1.00	-1.00	-1.00	1.50	-1.00	-1.00	-1.00	-1.00	-1.00
0.10	1.80	3.00	1.60	1.00	-1.00	-1.00	-1.00	-1.00	1.50	1.56	1.50	-1.00	-1.00	-1.00
0.21	1.20	1.75	1.80	1.50	2.14	2.00	2.00	2.00	1.30	1.40	1.40	1.30	2.00	1.50
0.46	0.90	1.00	1.20	1.70	1.80	1.40	1.40	1.67	0.70	1.20	0.80	1.60	1.90	1.90
1.00	0.70	0.50	1.20	0.40	0.70	1.00	1.60	1.30	0.90	0.50	0.70	1.10	0.60	1.10
2.15	0.60	0.75	0.80	0.60	0.60	0.50	0.70	0.50	0.70	0.40	0.50	0.50	0.20	0.80
4.64	0.60	0.75	0.60	0.50	0.50	0.40	0.60	0.40	0.40	0.10	0.50	0.40	0.50	0.20
10.0	0.40	1.25	0.60	0.50	0.50	0.60	0.50	0.60	0.70	0.50	0.20	0.40	0.00	0.10

Tabla B.31: S1 - CI for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	2.25	4.22	3.70	3.10	3.60	3.60	5.00	2.50	3.80	4.38	4.86	7.00	3.67
0.02	1.80	2.00	3.12	2.70	2.40	3.10	3.20	4.11	3.30	3.00	3.10	3.70	4.50	4.71
0.05	2.20	1.50	3.10	2.40	1.60	1.80	2.30	2.80	2.67	1.70	1.90	2.50	3.78	4.11
0.10	1.80	3.00	1.70	2.00	1.70	1.00	1.80	1.70	2.10	1.70	1.40	2.20	2.50	2.90
0.21	1.20	1.75	1.50	1.10	0.50	0.80	1.10	1.00	1.20	1.30	1.10	1.30	1.30	2.30
0.46	0.90	1.00	1.70	0.80	0.60	0.50	0.30	0.50	2.00	1.00	0.20	0.20	0.80	1.30
1.00	0.70	0.50	1.70	0.70	0.50	0.30	0.50	0.50	1.50	0.80	0.10	0.50	0.40	0.40
2.15	0.60	0.75	2.40	0.50	0.60	0.20	0.30	0.20	2.50	0.70	0.40	0.50	0.40	0.30
4.64	0.60	0.75	3.20	0.80	0.30	0.30	0.30	0.00	3.30	0.90	0.30	0.00	0.30	0.50
10.0	0.40	1.25	3.80	0.40	0.20	0.30	0.40	0.20	3.70	0.70	0.50	0.30	0.60	0.50

Tabla B.32: S1 - CI for Multi-quantile partitioner with Threshold 10

S2 - CI

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	1.75	-1.00	-1.00	4.17	5.40	4.17	4.50	-1.00	2.00	6.00	6.50	5.33	6.50
0.02	2.20	2.25	-1.00	2.00	3.33	3.40	4.22	3.71	-1.00	2.20	3.20	5.00	4.50	4.60
0.05	2.20	1.25	-1.00	2.56	2.10	2.70	2.90	3.00	-1.00	2.33	2.90	4.00	4.33	3.83
0.10	2.10	1.75	-1.00	2.10	2.30	1.60	2.10	2.20	-1.00	2.30	2.20	2.00	2.80	4.22
0.21	0.90	1.50	-1.00	2.00	2.30	1.60	1.40	1.60	-1.00	2.10	2.30	1.50	1.70	2.44
0.46	0.80	0.25	-1.00	2.10	2.10	1.20	0.90	0.80	-1.00	2.20	2.10	0.50	1.10	1.60
1.00	0.70	0.75	-1.00	2.00	2.20	0.80	0.90	0.80	-1.00	2.00	2.10	1.40	0.90	0.60
2.15	0.60	1.00	-1.00	2.20	2.30	0.80	0.90	0.50	-1.00	2.00	2.00	0.60	0.70	0.70
4.64	0.50	1.25	-1.00	2.30	2.10	0.80	0.70	0.40	-1.00	2.00	2.10	0.80	0.70	0.60
10.0	0.80	1.00	-1.00	2.10	1.80	0.90	0.60	0.20	-1.00	2.00	1.70	1.20	0.80	0.40

Tabla B.33: S2 - CI for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	1.75	-1.00	-1.00	3.67	4.80	6.50	5.00	-1.00	-1.00	3.83	5.00	6.00	4.00
0.02	2.20	2.25	-1.00	4.67	4.56	3.10	4.50	4.00	-1.00	1.00	3.86	4.17	5.75	5.50
0.05	2.20	1.25	-1.00	2.00	2.40	3.40	3.20	3.00	-1.00	1.90	3.00	4.00	4.12	5.50
0.10	2.10	1.75	-1.00	2.00	1.70	2.40	2.30	2.50	-1.00	1.80	1.70	2.00	3.00	3.29
0.21	0.90	1.50	-1.00	2.30	1.60	1.70	1.30	1.50	-1.00	2.40	1.80	1.00	1.40	2.30
0.46	0.80	0.25	-1.00	2.70	1.70	0.80	1.00	1.10	-1.00	2.60	1.50	0.90	0.70	1.30
1.00	0.70	0.75	-1.00	2.90	1.60	0.80	0.70	1.00	-1.00	2.70	1.60	1.10	0.60	0.70
2.15	0.60	1.00	-1.00	2.80	1.50	0.80	0.90	1.10	-1.00	2.90	1.70	0.80	0.70	0.70
4.64	0.50	1.25	-1.00	3.00	1.50	0.60	0.60	0.60	-1.00	3.00	1.40	0.70	0.90	0.50
10.0	0.80	1.00	-1.00	2.50	1.60	0.70	0.80	0.90	-1.00	3.10	1.70	0.80	0.50	0.70

Tabla B.34: S2 - CI for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	1.75	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.02	2.20	2.25	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.05	2.20	1.25	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	3.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.10	2.10	1.75	1.60	-1.00	-1.00	-1.00	-1.00	-1.00	1.70	2.20	2.00	-1.00	-1.00	-1.00
0.21	0.90	1.50	1.50	1.60	1.43	2.50	2.00	-1.00	2.40	1.60	1.80	2.10	1.89	2.11
0.46	0.80	0.25	0.80	1.70	2.10	2.30	1.60	1.30	0.70	2.10	1.60	2.00	2.70	2.40
1.00	0.70	0.75	1.10	1.20	1.10	1.30	1.60	1.90	1.00	0.40	0.80	0.60	0.80	1.20
2.15	0.60	1.00	0.90	0.80	0.70	1.00	0.60	0.70	0.50	0.60	0.50	0.60	0.30	0.60
4.64	0.50	1.25	0.40	0.70	1.00	0.30	0.70	0.60	0.90	0.30	0.20	0.60	0.50	0.70
10.0	0.80	1.00	0.70	0.50	0.50	0.40	0.40	0.80	0.30	0.40	0.20	0.20	0.40	0.40

Tabla B.35: S2 - CI for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	1.75	4.83	3.90	4.22	5.10	4.60	4.29	4.00	3.50	4.56	3.86	3.25	6.25
0.02	2.20	2.25	4.43	3.50	3.80	3.60	2.70	5.10	2.86	3.70	3.33	5.44	5.50	5.57
0.05	2.20	1.25	3.22	2.80	2.90	2.90	2.90	3.00	3.25	2.40	2.50	3.10	4.00	4.22
0.10	2.10	1.75	2.33	2.10	2.10	2.10	2.30	2.50	2.60	2.00	1.90	2.00	2.30	2.90
0.21	0.90	1.50	1.50	1.20	1.20	1.40	1.10	1.50	1.70	1.40	1.30	1.10	1.40	2.60
0.46	0.80	0.25	1.60	1.40	0.60	0.80	0.70	0.90	1.70	1.10	0.70	0.60	0.80	0.80
1.00	0.70	0.75	1.40	0.70	0.50	0.70	0.60	0.80	1.60	1.50	0.60	0.80	0.40	0.70
2.15	0.60	1.00	1.60	0.90	0.60	0.70	0.50	0.60	1.90	1.00	0.80	0.50	0.50	0.30
4.64	0.50	1.25	2.00	0.80	0.90	0.40	0.40	0.40	1.50	1.20	0.60	0.90	0.60	0.80
10.0	0.80	1.00	1.50	0.90	0.40	0.50	0.60	0.50	1.90	1.00	0.80	0.50	0.60	0.50

Tabla B.36: S2 - CI for Multi-quantile partitioner with Threshold 10

S3 - CI

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.00	-1.00	3.00	5.00	5.14	5.29	5.67	-1.00	2.50	4.80	6.67	5.80	5.00
0.02	2.20	3.50	-1.00	2.20	4.33	4.00	4.60	5.89	-1.00	2.50	3.56	5.86	5.00	5.67
0.05	4.10	3.00	-1.00	2.22	3.10	2.90	2.80	3.50	-1.00	2.10	3.00	4.20	5.00	5.62
0.10	3.30	1.75	-1.00	2.10	2.80	2.40	2.30	2.50	-1.00	2.20	2.90	2.30	3.10	4.30
0.21	2.60	1.75	-1.00	2.40	2.70	2.10	1.70	1.80	-1.00	2.20	3.20	1.80	1.60	2.00
0.46	1.80	1.00	-1.00	2.20	3.40	1.60	0.80	1.40	-1.00	2.20	3.00	1.40	0.80	1.00
1.00	1.00	0.75	-1.00	2.10	2.60	1.30	1.00	1.00	-1.00	2.10	2.70	1.30	0.70	0.80
2.15	1.00	1.25	-1.00	2.20	3.50	0.90	0.70	0.70	-1.00	2.50	3.00	1.10	0.60	0.70
4.64	0.80	0.50	-1.00	2.30	3.10	1.30	0.70	0.50	-1.00	2.40	3.00	1.10	0.50	0.60
10.0	1.00	0.75	-1.00	2.20	3.20	1.10	0.50	0.60	-1.00	2.40	2.80	0.80	0.50	0.10

Tabla B.37: S3 - CI for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.00	-1.00	-1.00	4.50	6.25	6.00	6.86	-1.00	5.00	4.67	5.25	6.00	6.00
0.02	2.20	3.50	-1.00	4.00	5.33	4.44	5.62	5.11	-1.00	4.40	4.50	5.44	5.20	5.20
0.05	4.10	3.00	-1.00	1.88	3.60	3.60	4.10	4.56	-1.00	2.67	3.80	4.20	4.67	5.75
0.10	3.30	1.75	-1.00	1.50	2.50	2.50	2.70	2.70	-1.00	1.50	2.90	3.00	3.40	3.00
0.21	2.60	1.75	-1.00	1.00	2.50	2.50	2.30	2.20	-1.00	1.40	2.70	2.20	1.90	3.20
0.46	1.80	1.00	-1.00	1.00	3.30	1.80	1.10	1.40	-1.00	1.00	2.70	1.50	1.40	1.30
1.00	1.00	0.75	-1.00	1.00	3.10	1.40	1.00	1.00	-1.00	1.00	3.00	1.70	0.90	0.70
2.15	1.00	1.25	-1.00	1.00	3.40	0.90	1.00	0.60	-1.00	1.00	3.00	1.50	0.80	0.60
4.64	0.80	0.50	-1.00	1.00	3.30	1.40	0.40	0.50	-1.00	1.00	3.10	1.10	0.80	0.20
10.0	1.00	0.75	-1.00	1.00	3.00	0.80	0.90	0.60	-1.00	1.00	2.60	1.70	0.80	0.50

Tabla B.38: S3 - CI for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.00	2.80	4.00	5.50	5.40	6.90	6.86	2.86	4.60	5.33	5.00	6.60	8.00
0.02	2.20	3.50	2.90	3.90	4.80	5.20	5.50	6.00	2.60	3.70	4.70	5.67	6.33	6.44
0.05	4.10	3.00	3.20	3.70	3.60	4.80	5.10	6.40	2.50	3.70	3.70	5.30	5.10	7.33
0.10	3.30	1.75	4.20	2.20	3.00	4.10	3.80	4.30	4.30	2.30	2.80	4.20	4.40	5.89
0.21	2.60	1.75	2.10	1.80	3.10	2.90	4.00	4.00	2.90	2.20	1.80	3.50	4.60	5.10
0.46	1.80	1.00	1.30	1.20	1.70	1.90	2.80	2.70	2.10	1.10	1.30	1.80	2.40	3.60
1.00	1.00	0.75	1.40	1.10	1.00	1.10	1.70	2.50	0.90	0.60	0.90	1.10	1.90	2.20
2.15	1.00	1.25	0.80	0.60	0.90	0.70	0.70	1.10	0.70	0.80	0.50	0.70	0.40	1.10
4.64	0.80	0.50	0.80	0.70	0.40	0.50	0.50	0.80	0.50	0.40	0.20	0.50	0.40	0.50
10.0	1.00	0.75	0.60	0.80	0.40	0.70	0.40	0.40	0.90	0.60	0.30	0.60	0.30	0.50

Tabla B.39: S3 - CI for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.00	5.83	5.40	5.30	4.56	6.43	5.90	6.25	6.00	5.25	4.67	5.50	6.20
0.02	2.20	3.50	5.00	4.10	4.60	4.10	4.00	4.89	4.57	4.20	4.40	4.60	4.75	6.00
0.05	4.10	3.00	4.22	3.70	3.40	4.00	4.20	3.70	3.11	3.70	3.70	3.00	4.50	5.25
0.10	3.30	1.75	3.10	2.80	2.50	2.50	2.70	2.50	2.80	3.30	2.30	2.70	2.90	3.67
0.21	2.60	1.75	2.20	2.50	1.80	2.00	1.60	1.70	2.10	2.40	2.00	1.70	2.30	2.80
0.46	1.80	1.00	2.10	1.70	1.20	1.00	1.20	1.20	1.90	1.90	1.10	1.30	1.30	1.30
1.00	1.00	0.75	1.90	1.30	1.00	0.60	0.30	0.60	1.80	1.00	0.80	0.50	0.60	0.90
2.15	1.00	1.25	2.00	1.20	0.20	0.60	0.60	0.60	2.00	1.10	0.40	0.50	0.70	0.60
4.64	0.80	0.50	2.00	1.00	0.50	0.50	0.60	0.60	2.20	0.60	0.50	0.90	0.50	0.60
10.0	1.00	0.75	2.30	1.10	0.60	0.50	0.50	0.50	2.00	0.90	0.30	0.60	0.60	0.60

Tabla B.40: S3 - CI for Multi-quantile partitioner with Threshold 10

S4 - CI

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.75	-1.00	4.00	5.50	5.50	5.50	5.50	-1.00	-1.00	5.50	6.00	6.75	8.00
0.02	4.40	2.75	-1.00	4.60	4.80	5.22	4.88	5.10	-1.00	4.33	5.00	5.75	6.14	5.50
0.05	5.60	2.50	-1.00	4.33	4.40	4.30	4.20	4.80	-1.00	4.50	4.50	4.50	4.90	5.88
0.10	3.90	2.50	-1.00	4.40	3.80	4.00	3.70	3.20	-1.00	4.40	3.80	2.90	4.30	4.56
0.21	2.80	1.25	-1.00	4.20	3.60	2.90	2.30	2.90	-1.00	4.20	4.10	2.30	2.60	3.30
0.46	2.40	1.25	-1.00	4.40	3.30	2.40	1.60	1.50	-1.00	4.40	3.90	1.80	1.80	1.60
1.00	2.20	2.00	-1.00	4.10	3.30	1.70	1.30	1.30	-1.00	4.30	3.50	1.70	1.20	1.20
2.15	1.50	1.50	-1.00	4.40	2.90	1.70	1.10	0.90	-1.00	4.40	3.30	1.60	0.80	0.80
4.64	1.40	1.25	-1.00	4.60	3.50	2.00	0.70	0.80	-1.00	4.50	3.60	1.70	0.70	0.60
10.0	1.40	0.50	-1.00	4.40	3.00	1.70	0.70	0.50	-1.00	4.40	3.20	1.60	0.80	0.70

Tabla B.41: S4 - CI for Binary partitioner with Threshold 10

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.75	-1.00	5.00	5.71	5.80	5.33	7.67	-1.00	6.00	5.80	6.00	4.00	8.00
0.02	4.40	2.75	-1.00	5.00	5.25	5.60	5.40	6.20	-1.00	5.20	5.89	4.50	5.67	7.00
0.05	5.60	2.50	-1.00	4.33	5.10	4.40	4.60	4.50	-1.00	4.00	4.00	4.11	6.22	6.14
0.10	3.90	2.50	-1.00	3.40	3.50	3.40	4.10	3.20	-1.00	3.50	3.50	3.70	3.80	5.00
0.21	2.80	1.25	-1.00	3.20	3.00	2.90	2.20	2.90	-1.00	3.60	3.10	3.20	2.20	2.90
0.46	2.40	1.25	-1.00	3.30	2.70	1.70	1.30	1.70	-1.00	3.40	2.50	2.20	1.90	1.60
1.00	2.20	2.00	-1.00	3.70	2.50	1.70	1.30	1.30	-1.00	3.90	2.90	2.10	1.50	1.00
2.15	1.50	1.50	-1.00	3.60	2.70	1.90	1.00	1.00	-1.00	3.60	2.80	1.70	1.00	0.80
4.64	1.40	1.25	-1.00	3.60	2.50	2.10	1.30	0.80	-1.00	3.70	2.30	2.20	1.20	1.00
10.0	1.40	0.50	-1.00	3.50	3.10	2.00	1.30	1.30	-1.00	3.80	2.90	2.20	1.00	0.70

Tabla B.42: S4 - CI for Median partitioner with Threshold 10

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.75	4.62	5.40	6.00	6.56	7.14	7.67	4.50	5.22	5.75	6.78	6.88	7.83
0.02	4.40	2.75	4.40	5.10	5.60	5.70	5.90	8.20	4.60	4.20	5.60	6.25	7.20	8.38
0.05	5.60	2.50	4.80	4.00	4.70	5.70	5.80	6.50	4.40	3.80	5.20	5.30	6.33	7.50
0.10	3.90	2.50	5.10	3.30	3.90	4.80	5.20	5.10	4.90	3.60	3.60	4.60	5.70	6.44
0.21	2.80	1.25	3.70	2.90	3.10	4.00	4.50	4.20	3.40	2.60	3.20	4.30	4.00	6.10
0.46	2.40	1.25	2.30	1.70	2.30	2.80	3.60	3.60	2.40	1.90	2.20	2.40	2.80	5.10
1.00	2.20	2.00	1.70	0.70	1.10	2.30	2.10	2.90	1.70	0.90	1.10	1.60	2.50	3.00
2.15	1.50	1.50	1.10	0.80	0.50	0.60	0.70	1.40	0.80	0.90	0.70	0.80	1.40	1.40
4.64	1.40	1.25	0.80	0.60	0.90	0.70	0.60	0.80	0.60	0.60	0.40	0.50	0.60	0.80
10.0	1.40	0.50	0.90	0.80	0.70	0.90	0.50	0.40	0.90	0.70	0.40	0.30	0.50	0.50

Tabla B.43: S4 - CI for Uniform partitioner with Threshold 10

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	3.75	7.71	6.30	6.50	5.86	5.60	6.12	6.40	6.40	5.67	6.75	7.83	5.86
0.02	4.40	2.75	6.12	5.10	4.80	4.60	5.11	5.11	6.00	4.80	5.60	5.75	4.88	7.17
0.05	5.60	2.50	5.20	4.20	3.70	3.90	4.80	4.10	5.00	4.90	4.30	4.00	4.60	6.00
0.10	3.90	2.50	4.30	3.60	3.50	2.50	3.40	2.60	4.40	4.60	3.00	3.30	3.30	4.80
0.21	2.80	1.25	3.20	3.50	2.30	1.90	2.40	2.20	3.80	3.70	2.20	2.20	2.80	3.20
0.46	2.40	1.25	3.00	2.30	1.70	1.30	1.30	1.50	3.40	2.80	1.80	1.30	1.10	1.70
1.00	2.20	2.00	3.30	1.80	1.10	1.20	1.20	1.30	3.30	2.10	1.40	0.80	1.00	1.40
2.15	1.50	1.50	3.70	1.90	1.00	0.50	1.00	0.70	3.90	1.80	0.90	0.40	0.70	0.60
4.64	1.40	1.25	4.00	1.60	0.70	0.80	0.80	0.80	3.60	2.20	0.50	0.80	0.80	0.50
10.0	1.40	0.50	4.00	1.90	1.50	1.10	1.10	1.20	3.80	2.00	1.20	0.80	0.60	0.80

Tabla B.44: S4 - CI for Multi-quantile partitioner with Threshold 10

Shuttle - CI

			Binary										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	4.00	-1.00	3.56	4.00	3.80	3.70	3.20	-1.00	3.83	4.00	3.78	3.70	3.80
0.02	4.00	-1.00	3.38	3.90	3.70	3.20	3.30	-1.00	3.50	4.00	3.20	3.50	3.43
0.05	2.70	-1.00	3.44	3.70	3.60	3.40	3.00	-1.00	3.50	3.90	3.40	3.30	3.33
0.10	2.90	-1.00	3.38	3.60	3.10	3.30	3.00	-1.00	3.33	3.50	3.20	3.30	3.20
0.21	2.70	-1.00	3.20	3.40	3.00	2.80	3.10	-1.00	3.10	3.60	3.30	2.90	3.10
0.46	3.30	-1.00	3.56	3.40	2.50	2.70	3.00	-1.00	3.11	3.40	2.90	3.10	3.00
1.00	3.40	-1.00	3.25	3.30	3.00	3.00	3.00	-1.00	3.00	3.30	2.80	2.90	2.90
2.15	3.20	-1.00	3.00	3.40	3.20	2.70	2.90	-1.00	3.12	3.40	2.90	2.70	2.90
4.64	3.20	-1.00	3.11	3.10	2.80	3.00	3.00	-1.00	3.11	3.30	2.70	3.00	2.90
10.0	3.20	-1.00	3.11	3.40	2.70	2.60	2.80	-1.00	3.14	3.40	2.60	2.80	3.00

Tabla B.45: shuttle - CI for Binary partitioner with Threshold 80

			Median										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	4.00	-1.00	3.40	3.30	3.40	3.50	3.70	-1.00	3.50	3.00	3.40	3.43	3.67
0.02	4.00	-1.00	3.30	3.40	3.40	3.00	3.20	-1.00	3.00	3.10	3.10	3.30	2.86
0.05	2.70	-1.00	4.20	2.70	3.50	3.10	3.40	-1.00	4.00	3.10	3.30	3.50	3.30
0.10	2.90	-1.00	4.00	3.10	3.20	3.20	3.30	-1.00	3.80	3.00	3.50	3.50	3.60
0.21	2.70	-1.00	3.90	3.30	3.30	2.90	3.10	-1.00	3.90	3.50	3.30	3.50	3.50
0.46	3.30	-1.00	3.80	3.50	3.40	3.40	3.30	-1.00	3.80	3.40	3.40	3.30	3.20
1.00	3.40	-1.00	3.90	3.20	3.40	3.50	3.50	-1.00	3.80	3.20	3.40	3.60	3.50
2.15	3.20	-1.00	3.00	3.00	3.40	3.60	3.80	-1.00	3.60	3.20	3.30	3.70	3.60
4.64	3.20	-1.00	3.80	2.60	3.50	3.30	3.70	-1.00	3.90	3.60	3.10	3.10	3.60
10.0	3.20	-1.00	4.00	1.70	2.70	2.20	2.60	-1.00	4.00	2.00	2.50	3.40	3.50

Tabla B.46: shuttle - CI for Median partitioner with Threshold 80

			Uniform										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	4.00	3.30	3.90	4.00	4.00	-1.00	4.00	3.10	3.30	3.90	3.78	3.90	4.00
0.02	4.00	3.30	3.10	3.00	3.40	3.90	3.86	3.10	3.20	3.10	3.10	3.00	3.00
0.05	2.70	3.10	3.40	3.20	3.00	3.10	3.10	3.00	3.30	3.50	3.30	3.30	3.20
0.10	2.90	3.00	3.40	3.20	3.30	3.20	3.20	3.00	3.10	3.20	3.40	3.50	3.40
0.21	2.70	4.00	3.00	2.80	3.40	3.50	3.40	4.00	3.00	3.50	2.90	2.90	3.00
0.46	3.30	4.00	4.00	3.10	3.40	3.00	3.10	4.00	3.20	3.40	3.00	3.20	3.30
1.00	3.40	4.00	3.80	4.00	4.00	4.00	3.30	4.00	3.80	3.40	3.50	3.20	3.10
2.15	3.20	3.50	3.90	4.00	4.00	4.00	3.90	3.30	4.00	4.00	4.00	3.10	3.20
4.64	3.20	3.70	3.70	4.00	4.00	4.00	4.00	3.60	3.90	3.90	4.00	4.00	3.70
10.0	3.20	3.60	3.90	3.90	4.00	4.00	3.90	3.50	4.00	3.20	3.00	2.90	2.90

Tabla B.47: shuttle - CI for Uniform partitioner with Threshold 80

			Multi-quantile										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	4.00	4.00	3.30	3.00	3.40	3.60	3.30	4.00	3.60	3.44	3.00	3.78	3.50
0.02	4.00	4.00	3.40	3.40	3.10	3.40	3.10	4.00	3.40	3.50	3.60	3.67	3.40
0.05	2.70	4.00	3.70	3.30	3.10	3.30	3.20	4.00	3.30	3.00	3.30	3.20	3.10
0.10	2.90	4.00	3.60	3.00	3.20	3.40	3.60	4.00	3.40	3.30	2.90	3.40	3.50
0.21	2.70	4.00	3.10	2.70	2.90	3.00	3.40	4.00	3.30	3.30	3.10	3.10	3.50
0.46	3.30	4.00	3.30	3.30	3.30	2.70	3.70	4.00	3.50	3.00	3.10	3.00	3.20
1.00	3.40	3.90	3.00	3.50	3.30	3.10	3.40	3.90	3.00	3.10	3.40	3.00	3.70
2.15	3.20	3.90	3.00	3.40	3.10	3.50	3.50	3.50	3.00	3.20	3.00	3.20	3.20
4.64	3.20	3.40	3.00	3.50	3.40	3.50	2.90	3.00	3.00	3.10	3.10	3.20	3.40
10.0	3.20	3.80	3.00	3.40	3.30	3.20	3.50	3.40	3.00	3.90	3.70	3.40	3.30

Tabla B.48: shuttle - CI for Multi-quantile partitioner with Threshold 80

Skin segmentation - CI

			Binary											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla B.49: Skin Segmentation - CI for Binary partitioner with Threshold 80

			Median											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.64	0.00	0.00	0.40	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10.0	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00

Tabla B.50: Skin Segmentation - CI for Median partitioner with Threshold 80

			Uniform											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla B.51: Skin Segmentation - CI for Uniform partitioner with Threshold 80

			Multi-quantile											
			Uniform						Geometric					
Epsilon	EUGKM	DPKFM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.05	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10.0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Tabla B.52: Skin Segmentation - CI for Multi-quantile partitioner with Threshold 80

Tarvel review ratings - CI

			Binary										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	-1.00	1.70	2.00	2.12	2.14	1.90	-1.00	1.50	2.00	2.20	1.75	2.00
0.02	-1.00	-1.00	1.40	1.60	1.90	1.90	2.00	-1.00	1.40	1.60	1.86	2.00	2.25
0.05	-1.00	-1.00	1.60	1.70	1.70	1.50	2.10	-1.00	1.40	1.80	1.40	1.78	1.89
0.10	-1.00	-1.00	1.40	1.40	1.50	1.50	1.50	-1.00	1.60	1.90	2.00	1.20	1.40
0.21	-1.00	-1.00	1.60	1.80	1.90	1.80	1.80	-1.00	1.20	1.60	1.70	1.30	1.70
0.46	-1.00	-1.00	1.30	1.80	1.20	1.20	1.50	-1.00	1.40	1.50	1.20	1.50	1.70
1.00	-1.00	-1.00	1.10	1.50	1.10	1.20	1.40	-1.00	1.40	1.50	1.20	1.20	1.20
2.15	-1.00	-1.00	1.30	1.60	1.30	1.60	1.50	-1.00	1.30	1.60	1.20	1.10	1.30
4.64	-1.00	-1.00	1.40	1.60	1.10	1.10	1.10	-1.00	1.40	1.50	1.40	0.80	1.60
10.0	-1.00	-1.00	1.20	1.70	1.30	0.70	1.30	-1.00	1.10	1.80	1.50	0.90	1.10

Tabla B.53: Tarvel Review Ratings - CI for Binary partitioner with Threshold 80

			Median										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	-1.00	1.90	2.00	2.56	2.33	2.75	-1.00	2.10	2.40	2.60	3.14	2.67
0.02	-1.00	-1.00	2.00	2.00	1.90	2.20	2.89	-1.00	1.70	2.11	2.67	2.60	2.50
0.05	-1.00	-1.00	1.80	1.50	1.50	2.10	1.80	-1.00	1.70	1.70	1.90	1.44	2.38
0.10	-1.00	-1.00	1.90	1.70	2.00	1.60	1.90	-1.00	1.60	1.90	1.50	1.80	1.80
0.21	-1.00	-1.00	1.50	2.00	2.00	1.60	1.80	-1.00	1.80	1.90	1.80	1.70	1.70
0.46	-1.00	-1.00	1.90	1.70	1.60	1.60	1.50	-1.00	2.00	1.80	1.80	1.70	1.80
1.00	-1.00	-1.00	1.70	1.80	1.80	1.80	1.70	-1.00	1.70	1.90	1.90	1.40	1.40
2.15	-1.00	-1.00	1.80	1.90	1.90	1.70	1.90	-1.00	1.90	1.80	1.80	1.80	1.60
4.64	-1.00	-1.00	1.70	2.00	2.00	1.80	1.60	-1.00	2.00	1.90	1.70	1.80	1.90
10.0	-1.00	-1.00	2.00	1.80	1.80	1.70	1.90	-1.00	1.80	1.90	1.70	2.00	1.70

Tabla B.54: Tarvel Review Ratings - CI for Median partitioner with Threshold 80

			Uniform										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.02	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.05	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.10	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.21	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
0.46	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
2.15	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
4.64	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00
10.0	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00	-1.00

Tabla B.55: Tarvel Review Ratings - CI for Uniform partitioner with Threshold 80

			Multi-quantile										
			Uniform					Geometric					
Epsilon	EUGKM	D2	D4	D6	D8	D10	D12	D2	D4	D6	D8	D10	D12
0.01	-1.00	2.80	1.90	2.60	2.30	2.75	3.12	2.60	2.20	2.70	2.62	2.60	3.00
0.02	-1.00	2.80	2.10	2.40	2.90	2.40	2.00	2.70	1.90	2.20	2.44	2.14	3.00
0.05	-1.00	2.40	1.60	2.10	2.10	1.40	1.90	2.80	1.30	2.00	2.20	2.20	2.30
0.10	-1.00	2.80	1.60	1.40	1.40	1.70	1.90	3.10	1.60	2.00	1.50	2.10	2.20
0.21	-1.00	2.50	1.50	1.70	1.90	2.00	1.20	2.70	1.30	1.80	1.90	1.90	1.70
0.46	-1.00	2.30	1.30	1.50	1.40	1.80	1.20	2.30	1.40	1.50	1.60	1.60	1.60
1.00	-1.00	2.20	1.10	1.60	1.70	1.30	1.30	2.20	1.50	1.50	1.20	1.30	1.60
2.15	-1.00	2.10	1.30	1.40	1.30	1.30	1.90	2.30	1.10	1.10	1.20	1.70	1.50
4.64	-1.00	2.10	1.20	1.20	1.20	1.20	1.40	2.10	1.10	1.30	1.30	1.40	1.30
10.0	-1.00	2.00	1.40	1.30	1.20	1.00	1.10	2.00	1.00	1.00	1.20	1.10	1.40

Tabla B.56: Tarvel Review Ratings - CI for Multi-quantile partitioner with Threshold 80