ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA HỆ THỐNG THÔNG TIN

AN TOÀN & BẢO MẬT HỆ THỐNG THÔNG TIN

DIFFERENTIAL PRIVACY APPLICATIONZ

Lớp: IS335.M11.HTCL

Giảng viên phụ trách: Ths. Hà Lê Hoài Trung

Sinh viên thực hiện:

Đoàn Thục Quyên – 18521320

Dương Bảo Nam – 18521118

TP.HCM – 31/5/2021

Mục lục

A. Lý thuyết về Differential Privacy

- Theo báo cáo cách đây không lâu của Apple, mặt vừa khóc vừa cười và trái tim là 2 biểu tượng cảm xúc emojis được dùng nhiều nhất. Vậy làm sao họ biết được thông tin thống kê này? Có phải họ đã theo dõi những gì mà người dùng chat?

- Câu trả là lời là bằng một kỹ thuật phân tích big data thông minh, Apple vừa có thể thu được số liệu hữu ích, vừa có thể đảm bảo quyền riêng tư của từng người dùng

- Chi tiết hơn, họ đã dùng một kỹ thuật đó là Differential Privacy. Họ đã thêm một số thông tin gây nhiễu để làm dữ liệu trên chiếc điện thoại của người dùng không có ý nghĩa, vô tác dụng. Nhưng sau đó, những dữ liệu bị gây nhiễu này sẽ được kết hợp với dữ liệu gây nhiễu của người khác, từ đó Apple vẫn có thể hiểu được dữ liệu họ muốn lấy nhưng không hề đọc thông tin của từng cá nhân người dùng.

Định nghĩa:

- Differential Privacy là một kỹ thuật toán học bổ sung một lượng ngẫu nhiên có kiểm soát vào tập dữ liệu để ngăn bất kỳ ai lấy thông tin về các cá nhân trong tập dữ liệu. Tính ngẫu nhiên được thêm vào được kiểm soát. Do đó, tập dữ liệu kết quả vẫn đủ chính xác để tạo ra thông tin chi tiết tổng hợp trong khi vẫn duy trì quyền riêng tư của từng người tham gia.

- Differential Privacy sẽ bảo vệ bạn khỏi việc bị lộ bất cứ thông tin cá nhân gì, không chỉ tấn công liên quan đến việc định danh bạn trong một database ẩn danh. Phương pháp này chống lại được linkage attack và các tấn công khác. Ngoài ra, nó cho bạn biết được bạn bị mất cụ thể bao nhiêu riêng tư với phương pháp này, và có cơ chế bảo vệ bạn trong trường hợp kẻ tấn công sử dụng nhiều query để gộp lại. Cuối cùng, các thông tin được lấy ra từ database sẽ vẫn được ẩn danh cho dù kẻ tấn công có xử lý data pro đi thế nào chăng nữa.

Cách hoạt động:

- Differential Privacy đưa ra tham số bị mất quyền riêng tư, thường được ký hiệu là epsilon (ε), vào tập dữ liệu. ε sẽ kiểm soát mức độ nhiễu hoặc độ ngẫu nhiên được thêm vào tập dữ liệu thô.

- Giả sử bạn có một cột trong tập dữ liệu của mình với các câu trả lời "Có" / "Không" từ các cá nhân. Đối với mỗi cá nhân, bạn lật một đồng xu, nếu nó đứng đầu, bạn để nguyên câu trả lời sấp, bạn lật đồng xu lần thứ hai và ghi lại câu trả lời là "Có" nếu đứng đầu và "Không" nếu nghiêng, bất kể câu trả lời thực là gì.

- Quá trình này bổ sung tính ngẫu nhiên cho dữ liệu. Đối với dữ liệu đủ lớn và với thông tin của cơ chế thêm nhiễu, tập dữ liệu vẫn chính xác về các phép đo tổng hợp. Đồng thời, mọi cá nhân trong tập dữ liệu có thể phủ nhận một cách chính đáng câu trả lời thực sự của họ khi đưa ra sự ngẫu nhiên.

- Differential Privacy có thể được chia làm 2 loại là Global Differential Privacy và Locacl Differential Privacy. Trong Local Differential Privacy, độ nhiễu được thêm vào dữ liệu riêng lẻ trước khi nó được đưa vào cơ sở dữ liệu. Trong Global Differential Privacy, độ nhiễu được thêm vào dữ liệu thô sau khi nó được thu thập từ nhiều người dùng.

Ví dụ chi tiết:

- Thí dụ như bạn muốn tiến hành một cuộc khảo sát trước khi bầu lớp trưởng nhằm xác định xem có bao nhiêu người bầu cho ứng cử viên A và B. Khi đó, bạn sẽ gọi những người đi bầu tới, yêu cầu họ bỏ biếu và ghi chép lại đầy đủ trong một cuốn sổ. Tuy nhiên, nếu bảng ghi chép này bị lộ hoặc đánh cắp thì danh sách toàn bộ những người bỏ phiếu cùng lựa chọn của họ sẽ bị lộ. Do đó, với cách làm này thì bạn dù có đạt được mục đích khảo sát nhưng đồng thời lại tạo ra nguy cơ tính riêng tư của nhiều người khác bị xâm hại.

- Bây giờ, hãy nghĩ nếu như người tổ chức khảo sát gọi những người tham gia bầu chọn tới và hỏi họ một câu hỏi khác với việc hỏi thẳng là sẽ chọn ai làm lớp trước. Thí dụ như người tổ chức sẽ yêu cầu người bầu chọn tung đồng xu. Nếu mặt ngửa thì người đi bầu sẽ được yêu cầu nói thật rằng họ sẽ chọn ai làm lớp trưởng. Nếu mặt sấp, họ sẽ được yêu cầu chọn ngẫu nhiên trong số 2 ứng cử viên lớp trưởng và nói tên 1 người. Nói cách khác, đồng xu sấp đồng nghĩa với việc người bầu chọn sẽ chọn A và B theo tỷ lệ 50 - 50. Cuối cùng, cách làm này sẽ giúp người tổ chức cuộc bình chọn sẽ nghe được lựa chọn thật của người bầu chọn với tỷ lệ 75%, 25% còn lại là nghe được lời nói dối. Trong thí dụ này, việc đưa đồng xu vào chính là một cách gây nhiễu dữ liệu gốc và chính người tổ chức cũng không biết được câu trả lời họ nghe là đúng hay sai, chỉ biết được tỷ lệ phần trăm.

- Do đó, cho dù bảng ghi chép kết quả sau cuộc bình chọn lớp trưởng bị lộ ra ngoài thì thông tin cá nhân của mỗi người tham gia bầu chọn vẫn được bảo vệ. Nguyên nhân là do người ta không xác định được ai bỏ phiếu cho ai, mỗi người đều có khả năng trả lời không đáng tin nên người lén đọc dữ liệu cũng không xác định được cái họ đọc chính xác hay không. Tuy nhiên, đối với người tiến hành khảo sát thì họ có thể tính được con số trung bình kết quả bầu chọn bởi chính họ mới là người biết được cách gây nhiễu dữ liệu. Nói cách khác, khi nhìn trên giác độ vĩ mô thì có thể thu được thông tin cần thiết, nhưng khi quan sát vi mô thì không khả dĩ.

B. Privacy Location

1. Khái niệm và sự nguy hiểm của privacy location

- Khái niệm về quyền riêng tư về vị trí có thể được định nghĩa là quyền của các cá nhân quyết định cách thức, thời điểm và mục đích thông tin vị trí của họ có thể được tiết lộ cho các bên khác.

- Việc thiếu bảo vệ quyền riêng tư của vị trí có thể bị kẻ thù lợi dụng để thực hiện các cuộc tấn công khác nhau:

+ Quảng cáo không được yêu cầu, khi vị trí của người dùng có thể bị khai thác mà không có sự đồng ý của họ, để cung cấp quảng cáo về các sản phẩm và dịch vụ có sẵn gần vị trí của người dùng

+ Các cuộc tấn công hoặc quấy rối vật lý, khi vị trí của người dùng có thể cho phép bọn tội phạm thực hiện các cuộc tấn công vật lý đối với các cá nhân cụ thể.

+ Hồ sơ người dùng và theo dõi, khi vị trí của người dùng có thể được sử dụng để suy ra các thông tin nhạy cảm khác, chẳng hạn như tình trạng sức khỏe, thói quen cá nhân hoặc nhiệm vụ chuyên môn, bằng cách liên quan đến các địa điểm hoặc đường dẫn đã ghé thăm.

+ Sự ngược đãi và phân biệt đối xử về chính trị, tôn giáo, tình dục và phân biệt đối xử, khi vị trí của người dùng có thể được sử dụng để làm giảm quyền tự do của các cá nhân và công nghệ di động được sử dụng để xác định và bắt bớ đối thủ.

2. Ý nghĩa của việc bảo vệ Location privacy

- Quyền riêng tư về vị trí bảo vệ vị trí của từng người dùng bằng cách xáo trộn thông tin tương ứng và giảm độ chính xác của thông tin vị trí. Quyền riêng tư về vị trí phù hợp với những môi trường yêu cầu danh tính của người dùng để cung cấp dịch vụ thành công. Một kỹ thuật mà hầu hết các giải pháp khai thác, dù rõ ràng hay ẩn ý, bao gồm việc giảm độ chính xác bằng cách mở rộng vị trí đến mức độ chi tiết thô hơn (từ mét đến hàng trăm mét, từ một khối thành phố đến toàn bộ thị trấn, v.v.).

- Quyền riêng tư về danh tính bảo vệ danh tính của những người dùng được liên kết hoặc không thể sử dụng khỏi thông tin vị trí. Với mục đích này, các kỹ thuật bảo vệ nhằm giảm thiểu việc tiết lộ dữ liệu có thể cho phép những kẻ tấn công suy ra danh tính người dùng. Quyền riêng tư về danh tính phù hợp trong các bối cảnh ứng dụng không yêu cầu nhận dạng của người dùng để cung cấp dịch vụ.

3. Một số phương pháp bảo vệ

Vì định nghĩa và yêu cầu về quyền riêng tư của vị trí khác nhau tùy theo tình huống, nên không có kỹ thuật nào có thể giải quyết các yêu cầu của tất cả các danh mục quyền riêng tư của vị trí.

Do đó, trước đây, cộng đồng nghiên cứu, tập trung vào việc cung cấp các giải pháp để bảo vệ quyền riêng tư vị trí của người dùng, đã xác định các kỹ thuật có thể được chia thành ba lớp chính: kỹ thuật dựa trên ẩn danh, dựa trên sự xáo trộn và dựa trên chính sách.

- Có thể dễ dàng nhận thấy rằng các kỹ thuật dựa trên ẩn danh và dựa trên sự xáo trộn có thể được coi là hai loại kép. Và những kỹ thuật này chủ yếu nhằm bảo vệ danh tinh của người sử dụng chứ k thể bảo vệ quyền vị trí cá nhân của người dùng

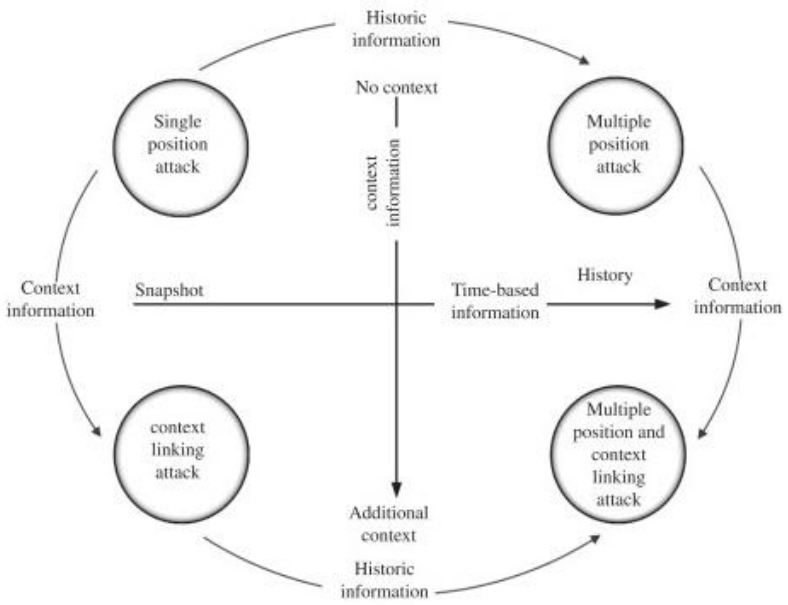
- Các kỹ thuật dựa trên chính sách nói chung phù hợp với tất cả các danh mục quyền riêng tư của vị trí, mặc dù chúng thường khó hiểu và khó quản lý đối với người dùng cuối.

- Bên cạnh đó Beresford và Stajano cũng đã đề xuất ra một phương pháp gọi là vung hỗn hợp. Khi sử dụng kỹ thuật ẩn danh để có thể xáo trộn thông tin của những người trong cùng một khu vực cho nhau và được áp dụng qua k-anonymity

4. Sự tấn công của kẻ thù:

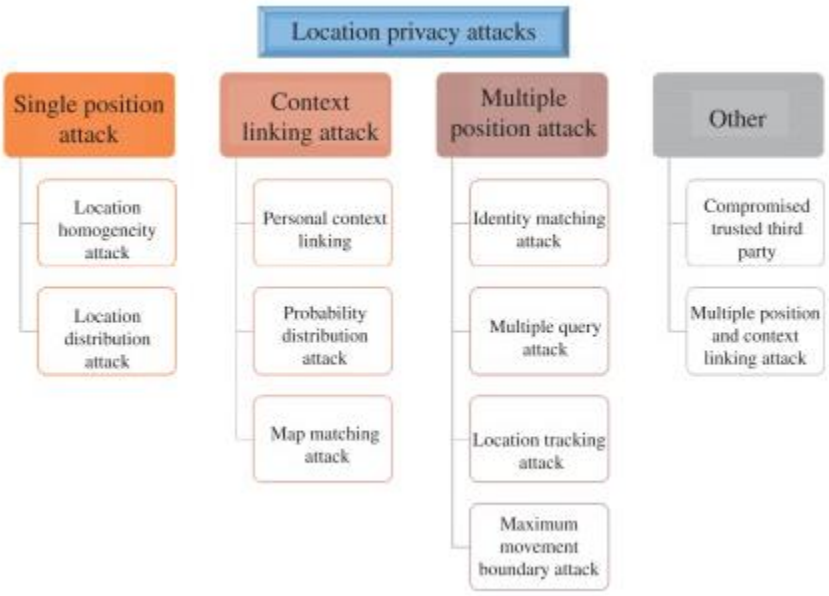
- Kiến thức thông tin tạm thời, trong đó kẻ tấn công có thông tin về ảnh chụp nhanh vị trí người dùng hoặc có quyền truy cập vào nhiều vị trí người dùng được tích lũy trong một khoảng thời gian hoặc có quyền truy cập vào quỹ đạo chuyển động hoàn chỉnh của người dùng.

- Kiến thức thông tin ngữ cảnh, trong đó kẻ tấn công có thông tin ngữ cảnh bổ sung ở trên và ngoài thông tin không gian. Ví dụ: kẻ tấn công có thể đọc danh bạ của người dùng để lấy thông tin về địa chỉ, loại bạn bè, địa điểm yêu thích của người dùng, v.v.



Hình 1 Cuộc tấn công của kẻ thù

➔ Cho thấy kiến thức của kẻ tấn công có thể chuyển đổi giữa không gian tạm thời (dựa trên thời gian) và ngữ cảnh tùy thuộc vào loại thông tin có sẵn cho kẻ tấn công.



Hình 2 Phân loại cuộc tấn công của kẻ thù

5. Location privacy của MCS(Sparse Mobile Crowdsensing)

5.1. Cơ chế bảo vệ của MCS

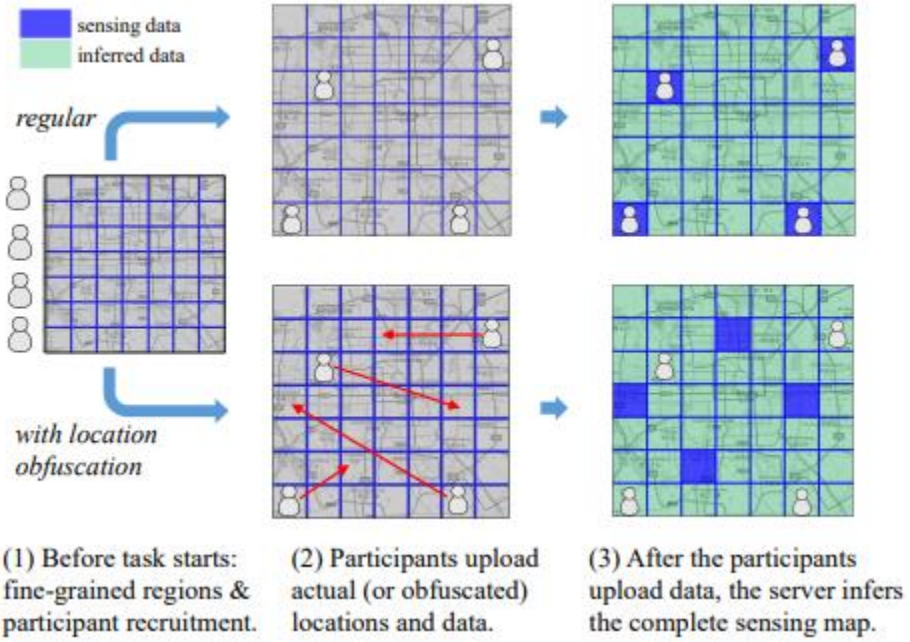
Bởi vì người tham gia phải báo cáo lại dữ liệu cảm biến với các dấu thời gian và vi trí địa lý nên có thể gây ra những rủi ro nghiêm trọng về quyền riêng tư. Qua đó có những nghiên cứu về quyền riêng tư của vị trí và họ đã đề xuất 2 cơ chết bảo vệ :

+ Bảo vệ danh tính của người dùngthông qua ẩn danh, do đó dấu vết vị trí của họ không thểđược liên kết với các cá nhân cụ thể

+ Sử dụng vị trí sự xáo trộn để thay đổi vị trí thực tế của người dùng tiếp xúc với nhà cung cấp dịch vụ

5.2. Sự xáo trộn

-Một trong những cơ chế xáo trộn phổ biến nhất là che đậy là cloaking. Nó đại diện cho vị trí của người dùng dưới dạng được che giấu vùng chứa nhiều ô hạt mịn thay vì một địa điểm hoặc ô cụ thể, Nhưng điều này rất dễ bị kẻ thù phát hiện nếu đối thủ có kiến thức từ trước về phân bố vị trí mục tiêu



Hình 3 Bảo vệ quyền riêng tư bằng xáo trộn

Chú thích:

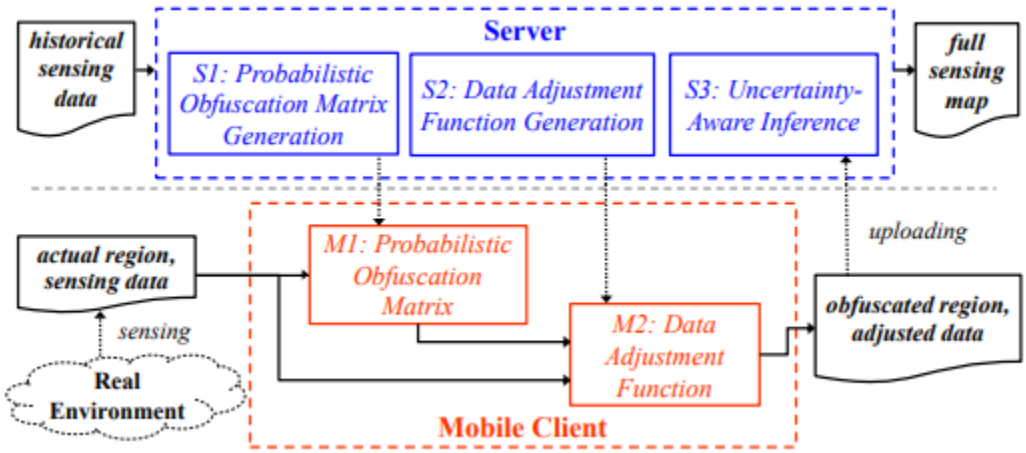
(1) Trước khi nhiệm vụ bắt đầu:vùng hạt mịn &tuyển dụng người tham gia.

(2) Người tham gia tải lênthực tế (hoặc bị xáo trộn)vị trí và dữ liệu.dữ liệu cảm biến dữ liệu suy luận.

(3) Sau khi những người tham gia tải lên dữ liệu, máy chủ suy luận bản đồ cảm biến hoàn chỉnh

➔ Quyền riêng tư khác biệtđược giới thiệu để đảm bảo rằng người dùng có cơ hộiđược ánh xạ tới một vị trí bị xáo trộn cụ thể từ bất kỳcác địa điểm thực tế là tương tự. Càng giống thìxác suất cho mỗi khu vực thì càng khó để suy ra người dùng 'vị trí ban đầu, dẫn đến bảo vệ quyền riêng tư tốt hơn. Nói cách khác,vị trí của người tham gia có thể được ánh xạ đến một nơi rất xa,miễn là các giá trị cảm biến của hai vị trí gần nhau đầy đủ.

5.3. Khung bảo vệ quyền riêng tư



Nó bao gồm hai tầng - máy chủ và phía khách hàng di động.

Dựa trên dữ liệu cảm nhận lịch sử, phía máy chủ tạo ra một ma trận xáo trộn xác suất (Bước S1)

Và chức năng điều chỉnh dữ liệu (Bước S2) theo cách ngoại tuyến.Ma trận mã hóa xác suất làm xáo trộn bất kỳvùng này sang vùng khác. Qua đó có thể bảo vệ vị trí của người dùng quyền riêng tư bằng cách lựa chọn cẩn thận các xác suất, có thể khiến không thể suy luận chính xác một khu vực thực tế từ đối tác khó hiểu của nó, ngay cả khi đối thủ biết ma trận xáo trộn. Chức năng điều chỉnh dữ liệu được sử dụng để giảm độ không chắc chắn của dữ liệu do nhiễu vùng. Nó được học bằng cách phân tích mối tương quan giữa hai dữ liệu cảm nhận của khu vực trong nhật ký lịch sử.

Sau khi tải xuống trước cả ma trận xáo trộn và chức năng điều chỉnh dữ liệu cho điện thoại di động của họ mật mã có thể thực hiện nhiệm vụ cảm biến trên các máy khách di động như sau.

Đầu tiên, mỗi điện thoại di động cảm nhận được vị trí. Sau đó, dựa trên ma trận xáo trộn xác suất,nó ánh xạ vùng liên kết sang vùng khác (Bước M1).

Sau đó, chức năng điều chỉnh dữ liệu sẽ thay đổi bản gốccảm nhận dữ liệu để phù hợp với các thuộc tính của vùng bị xáo trộn(Bước M2). Sau đó, khách hàng di động tải lên bản sửa đổivùng và dữ liệu đến máy chủ.

Sau đó máy chủ cung cấp thông tin đầy đủ cảm nhận bản đồ từ tất cả các khu vực bị xáo trộn nói chung, chứa một mức độ không chắc chắn nhất định so vớ idữ liệu thực tế (Bước S3).

C. Histogram

Kí hiệu E = epsilon

Quyền riêng tư khác biệt (DP) mong muốn đưa ra kết quả của các truy vấn thống kê về dữ liệu nhạy cảm, với đảm bảo quyền riêng tư mạnh mẽ chống lại những kẻ thù có kiến thức nền tùy ý. Các nghiên cứu hiện tại về quyền riêng tư khác biệt chủ yếu tập trung vào các tổng hợp số đếm. Histogram là một công cụ phân tích quan trọng để hiển thị sự phân bố của một biến ngẫu nhiên, ví dụ: kích thước hóa đơn bệnh viện cho một số bệnh nhân nhất định. So với các tổng hợp có kết quả hoàn toàn là số, thì truy vấn biểu đồ vốn

đã phức tạp hơn, vì nó cũng phải xác định kết cấu, tức là, phạm vi của các thùng. Như chúng ta chứng minh trong bài báo, biểu đồ tuân thủ DP với các thùng mịn hơn thực sự có thể dẫn đến độ chính xác thấp hơn đáng kể so với biểu đồ thô hơn, vì biểu đồ trước đây yêu cầu nhiễu loạn mạnh hơn để đáp ứng DP. Hơn nữa, bản thân cấu trúc biểu đồ có thể tiết lộ thông tin nhạy cảm, điều này càng làm phức tạp thêm vấn đề. Ở phần này, chúng ta giới thiệu hai cơ chế mới, đó là NoiseFirst và StructureFirst, để tính toán histogram tuân thủ DP. Sự khác nhau giữa 2 cơ chế là thứ tự bước thêm nhiễu và bước tính toán cấu trúc histogram. Ngoài ra, NoiseFirst cải thiện độ chính xác của DP-complaint histogram tính toán bằng phương pháp naïve.

Mỗi cơ chế được đề xuất, chúng ta thiết kế các thuật toán để tính toán cấu trúc biểu đồ tối ưu (optimal histogram structure) với hai mục tiêu khác nhau: giảm thiểu sai số bình phương trung bình và sai số tuyệt đối trung bình, tương ứng (minimizing the mean square error and the mean absolute error, tương ứng).

Tiến thêm một bước nữa, chúng ta mở rộng cả hai cơ chế để trả lời các truy vấn phạm vi tùy ý. Các thử nghiệm mở rộng, sử dụng một số tập dữ liệu thực, xác nhận rằng hai đề xuất của chúng ta tạo ra các câu trả lời truy vấn có độ chính xác cao và luôn hoạt động tốt hơn các đối thủ cạnh tranh hiện có.

1. Giới thiệu

Các kỹ thuật kỹ thuật số đã cho phép các tổ chức khác nhau dễ dàng thu thập lượng lớn thông tin cá nhân, chẳng hạn như hồ sơ y tế, lịch sử tìm kiếm trên web, v.v. Phân tích trên những dữ liệu đó có thể dẫn đến những hiểu biết sâu sắc có giá trị, bao gồm những hiểu biết mới về bệnh tật và hành vi tiêu dùng điển hình trong cộng đồng . Tuy nhiên, những lo ngại về quyền riêng tư hiện đang trở thành một rào cản lớn đối với những phân tích như vậy, theo hai khía cạnh.

- Đầu tiên, nó làm tăng khó khăn cho các nhà phân tích dữ liệu của bên thứ ba trong việc truy cập dữ liệu đầu vào của họ. Ví dụ, các nhà nghiên cứu y tế thường được yêu

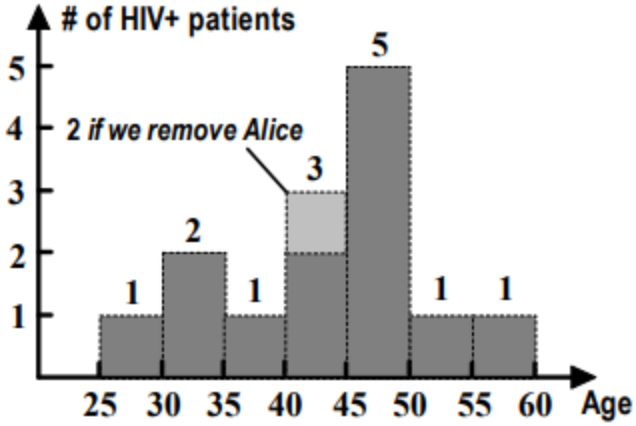
cầu phải có được sự chấp thuận của hội đồng đánh giá thể chế tương ứng của họ, điều này rất tẻ nhạt và tốn thời gian, trước khi họ có thể xem xét dữ liệu họ cần.

- Thứ hai, những lo ngại về quyền riêng tư làm phức tạp thêm việc công bố kết quả phân tích. Một ví dụ đáng chú ý là cơ sở dữ liệu dbGaP1, chứa các kết quả nghiên cứu di truyền. Kết quả như vậy đã từng là được công bố rộng rãi, cho đến khi một bài báo gần đây mô tả một kết quả suy đoán liệu một người có tham gia vào một nghiên cứu nhất định (ví dụ, trên bệnh nhân tiểu đường) từ kết quả của nó hay không; sau đó, việc truy cập vào các kết quả đó được kiểm soát chặt chẽ. Hơn nữa, một phiên bản tăng cường của cuộc tấn công này đe dọa việc xuất bản bất kỳ tài liệu nghiên cứu nào về các nghiên cứu liên kết toàn bộ hệ gen, hiện đang là một lĩnh vực tích cực trong nghiên cứu y sinh học.

Khái niệm về quyền riêng tư khác biệt (DP): giải quyết các vấn đề trên bằng cách đưa một lượng nhỏ nhiễu ngẫu nhiên vào các kết quả thống kê. DP đang nhanh chóng trở nên phổ biến, bởi vì nó cung cấp các đảm bảo quyền riêng tư nghiêm ngặt chống lại những kẻ thù có thông tin cơ bản tùy ý. Công việc này tập trung vào việc tính toán biểu đồ tuân thủ DP, đây là một công cụ phổ biến để trình bày phân phối của một biến ngẫu nhiên.

Name	Age	HIV+
Alice	42	Yes
Bob	31	Yes
Carol	32	Yes
Dave	36	No
Ellen	43	Yes
Frank	41	Yes
Grace	26	Yes
...	...	...

(a) Example sensitive data



(b) Unperturbed histogram

Fig. 1 Example sensitive dataset and its corresponding histogram

Hinh 1(a): cho thấy những bản ghi về những dữ liệu nhạy cảm về những bệnh nhân dương tính với HIV.

Hình 1(b): minh họa biểu đồ của nó cho thấy được sự phân bố theo tuổi của bệnh nhân đó.

Ứng dụng DP vào biểu đồ cần đảm bảo rằng việc thay đổi hoặc xóa bất kỳ bản ghi nào khỏi cơ sở dữ liệu tác động không đáng kể đến biểu đồ đầu ra. Điều này có nghĩa là đối thủ không thể suy luận liệu một bệnh nhân cụ thể (giả sử Alice) có bị nhiễm HIV hay không, ngay cả khi họ biết tình trạng nhiễm HIV của tất cả các bệnh nhân còn lại trong cơ sở dữ liệu.

Biểu đồ có cấu trúc chia tập hợp các truy vấn có phạm vi số đếm rời rạc cho mỗi bin. Phương pháp hiện đại này gọi là cơ chế Laplace (LM) làm xáo trộn đầu ra để thỏa mãn DP.

Đầu tiên, LM xác định độ nhạy cảm Δ cho những số đếm này, là những thay đổi tối đa có thể xảy ra trong kết quả truy vấn nếu chúng ta xóa một bản ghi khỏi (hoặc thêm một bản ghi vào) cơ sở dữ liệu.

Trong ví dụ trên, ta có Δ = 1, vì mỗi bệnh nhân ảnh hưởng đến giá trị của chính xác một ô trong biểu đồ nhiều nhất là 1. Ví dụ: loại bỏ Alice làm giảm số lượng bệnh nhân HIV+ ở độ tuổi 40-45 xuống 1. Sau đó, LM thêm vào mỗi bin một giá trị ngẫu nhiên tuân theo phân phối Laplace với giá trị trung bình mean = 0 và tỷ lệ scale Δ / ε, trong đó ε là một tham số cho biết mức độ riêng tư (level of privacy). Ví dụ, khi ε = 1, tiếng ồn được thêm vào mỗi thùng có phương sai = 2(variance), phương sai này bao hàm những tác động (tức là 1) của bất kỳ cá nhân nào trong cơ sở dữ liệu.

Độ chính xác của biểu đồ tuân thủ DP phụ thuộc nhiều vào cấu trúc của nó. Biểu đồ thô hơn đôi khi có thể dẫn đến độ chính xác cao hơn biểu đồ mịn hơn, như được minh họa trong ví dụ bên dưới.

Ví dụ 1:

Hình 2 (a) cho thấy một biểu đồ khác của tập dữ liệu trong Hình 1 (a) với 3 bin lần lượt là 25-40, 40-50 và 50-60. Chúng ta sử dụng thuật ngữ “dải đơn vị độ dài” (unit-length range) để tính trung bình là dải tương ứng với bin ban đầu trong biểu đồ ở Hình 1 (b), ví dụ: 25-30. Trong biểu đồ ở Hình 2 (a), mỗi bin bao gồm nhiều phạm vi độ dài đơn vị và các số trên đầu mỗi thùng tương ứng vớibần tiện số lượng của mỗi phạm vi độ dài đơn vị, ví dụ: 1,33 ở trên phạm vi 25-30 được tính bằng cách chia toàn bộ số bệnh nhân (tức là 4) trong thùng 25-40 bằng số dãy đơn vị chiều dài mà thùng đó bao gồm (tức là 3). Như chúng ta chứng minh ở phần sau của bài báo, việc lấy trung bình như vậy làm giảm lượng nhiễu trong đó. Cụ thể, tiếng ồn Laplace được thêm vào mỗi phạm vi chiều dài đơn vị bên trong 1 bin được che bởi b, như vậy phạm vi có tỉ lệ 1/b · E, so với tỉ lệ 1/ ε trong biểu đồ của Hình 1 (b).

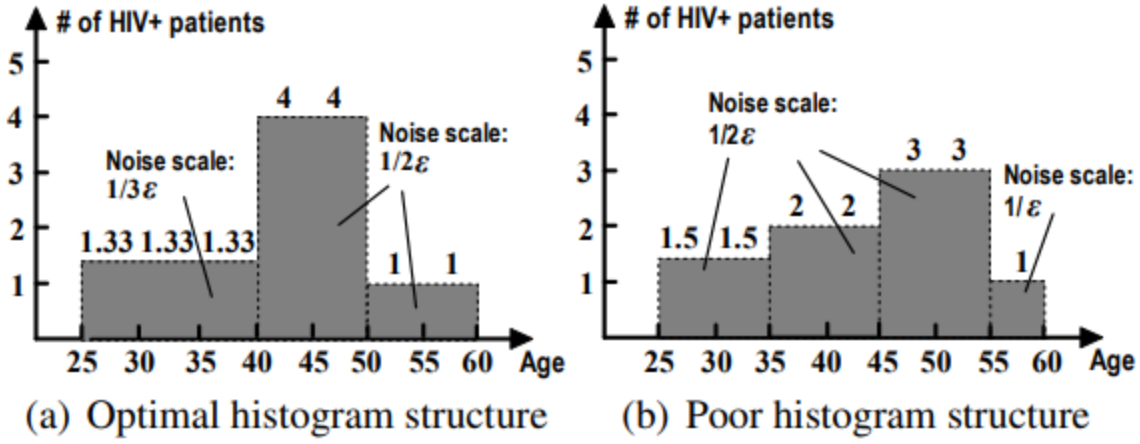


Fig. 2 Impact of histogram structures (k=3) on noise scales

Ví dụ trên chứng minh rằng một biểu đồ thô hơn có thể dẫn đến lượng nhiễu được thêm vào mỗi thùng ít hơn. Tuy nhiên, việc sử dụng các bin lớn như vậy gây mất thông tin. Chất lượng của cấu trúc histogram dựa vào việc cân bằng giữa mất thông tin và giảm error.

Hình 2(b), ví dụ, cho thấy một biểu đồ khác cho cùng một tập dữ liệu trong Ví dụ 1, dễ dạng thấy độ chính xác kém vì (i) nó kết hợp các phạm vi độ dài đơn vị với các giá trị rất khác nhau, ví dụ: phạm vi 35-40 và 40-45 (1 và 3), dẫn đến mất mát thông tin cao; và (ii) bin rất nhỏ, như bin 55-60, yêu cầu tỷ lệ độ nhiễu cao (1/ ε), đặc biệt khi giá trị ε nhỏ. Ví dụ này ngụ ý rằng cấu trúc tốt nhất phụ thuộc vào việc phân phối dữ liệu cũng như ε.

Một sự phức tạp nữa là nếu chúng ta xây dựng cấu trúc biểu đồ tối ưu trên chuỗi số ban đầu, như được hiển thị trong Ví dụ 1, bản thân cấu trúc tối ưu có thể tiết lộ thông tin nhạy cảm. Điều này là do việc xóa bản ghi khỏi cơ sở dữ liệu gốc có thể khiến cấu trúc tối ưu thay đổi, có thể bị kẻ thù lợi dụng để suy ra thông tin nhạy cảm. Do đó, việc chọn cấu trúc tốt nhất với kỹ thuật xây dựng biểu đồ hiện có vi phạm DP, bất kể lượng nhiễu được đưa vào số đếm.

Đối mặt với những thách thức này, đề xuất hai giải pháp hiệu quả để tính toán biểu đồ tuân thủ DP, đó là NoiseFirst và StructureFirst.

- **NoiseFirst**: thêm nhiễu sau đó xác định cấu trúc histogram.
- **StructureFrist**: tạo cấu trúc histogram sau đó thêm nhiễu.

Chúng ta thảo luận cả hai cách tiếp cận ở trên dưới hai chiến lược xây dựng biểu đồ chung, tức là biểu đồ trung bình và biểu đồ trung vị. Biểu đồ trung bình áp dụng trung bình của tất cả số lượng trong mỗi ngăn biểu đồ làm số lượng đại diện. Trong biểu đồ trung vị, các trung điểm được sử dụng thay thế cho phương tiện để tóm tắt tập hợp số lượng cho tất cả các thùng biểu đồ. Mặc dù biểu đồ trung vị hiếm khi được sử dụng trong cơ sở dữ liệu thông thường, chúng ta cho thấy rằng nó phù hợp hơn với sự riêng tư khác biệt, do đó cải thiện độ chính xác của biểu đồ

Hơn nữa, chúng ta điều chỉnh biểu đồ DP để trả lời đối với các câu truy vấn phạm vi số đếm tùy ý. Đối với các truy vấn như vậy, NoiseFirst đạt được độ chính xác tốt hơn cho các phạm vi ngắn, trong khi Structure-First phù hợp hơn cho các phạm vi dài hơn. NoiseFirst và StructureFirst cho ra biểu đồ có độ chính xác cao và vượt trội hơn đáng kể so với các phương pháp hiện có cho các truy vấn đếm phạm vi.

2. Xây dựng biểu đồ

Với một dãy liên tiếp các n số đếm cho trước trong tập D chứa các giá trị {x1, x2, ..., xn} và một tham số k, một biểu đồ H hợp nhất các số đếm bên cạnh vào k bin trong histogram H bao gồm các giá trị các bin {B1, B2,...,Bk} và sử dụng một số đếm đại diện cho mỗi bin. Mỗi bin Bj = (lj, rj, cj) chứa đựng một đoạn [lj, rj] là tập con của đoạn [1,n], một số đếm cj xấp xỉ các số đếm trong D rơi vào đoạn [lj, rj] (ví dụ: giá trị x(i), với i lớn hơn hoặc bằng lj và bé hơn hoặc bằng rj). Các bin trong một biểu đồ phải được phân chia và chưa chung quy tất cả các số đếm trong D.

Từ đó, một biểu đồ sử dụng các số đếm ít hơn tập chứa các giá trị liên tiếp D chắc chắn sẽ dẫn đến lỗi. Lỗi này thường được đo lường bởi độ đo "Sum of Squared Error"-(SSE) trong biểu đồ H và các số đếm ban đầu trong tập D, như sau:

- Một dãy liên tiếp các n số đếm trong tập D = {x1, x2, ..., xn}

- Tham số k,

- Một biểu đồ H hợp nhất các số đếm bên cạnh vào k bin trong histogram H

- Các giá trị các bin {B1,B2,...,Bk}

- Mỗi bin Bj = (lj, rj, cj) :

  + đoạn [lj, rj] là tập con của đoạn [1, n],

  + một số đếm cj xấp xỉ các số đếm trong D rơi vào đoạn [lj, rj] (ví dụ: giá trị x(i), với i lớn hơn hoặc bằng lj và bé hơn hoặc bằng rj).

- Các bin trong một biểu đồ phải được phân chia và chưa chung quy tất cả các số đếm trong D.

- Từ đó, một biểu đồ sử dụng các số đếm ít hơn tập chứa các giá trị liên tiếp D chắc chắn sẽ dẫn đến lỗi.

Lỗi này thường được đo lường bởi độ đo "Sum of Squared Error"-(SSE) trong biểu đồ H và các số đếm ban đầu trong tập D, như sau:

$$SSE(H, D) = \sum_{j} \sum_{l_j \le i \le r_j} (c_j - x_i)^2.$$

SSE(H, D) (SSE của H và D) cũng có thể được diễn giải như là sum of squared error trong tất cả các câu truy vấn phạm vi số đếm có dải đơn vị chiều dài. Cho trước cấu trúc của biểu đồ (ví dụ: đoạn [lj, rj] cho mỗi bin Bj, giá trị tối ưu nhất của cj đối với Bj là làm tối thiểu SSE(H,D) là lấy trung bình của các số đếm trong [lj,rj], ví dụ như là cj = $\frac{\sum_{i=l_j}^{r_j} x_i}{r_j - l_j + 1}$.

Dựa vào đó, vấn đề của xây dựng biểu đồ thông thường nhắm tới việc xác định cấu trúc tối ưu của biểu đồ H chứa đựng k bin (k là một tham số cho trước) làm tối thiểu SSE(H,D).

Hình 3 liệt kê các kết quả trung gian của quá trình lập trình động, sử dụng chuỗi dữ liệu trong Hình 1 (b) và đặt k = 3.

Mỗi mục trong cột i và hàng k kí hiệu là T(i,k), đại diện là error nhỏ nhất của bất kì histogram với k bin bao gồm những chuỗi số đếm đầu là Di = {x1, . . . , xi}. SSE(D,l,r) kí hiệu cho sum of square error phát sinh bởi việc hợp những chuỗi số đếm từng phần {xl , . . . , xr} vào từng bin. Số đếm trung bình cho việc kết hợp này là: $\bar{x}(l,r) = \sum_{i=l}^{r} x_i/(r-l+1)$.

Vì vậy SSE(D, l, r) = $\sum_{i=l}^{r}(x_i - \bar{x}(l,r))^2$.

i=1	2	3	4	5	6	7	
0	0.5	0.67	2.75	11.2	12.8	14	*k*=1
	0	0.5	0.67	2.67	8.67	11.2	2
		0	0.5	0.67	2.67	2.67	3

Fig. 3 Building the optimal histogram for the dataset in Figure 1

Thuật toán chương trình động trong tính đệ quy giá trị SSE tối thiểu, ví dụ như T(n, k) cho việc xây dựng biểu đồ tối ưu, sử dụng phương trình sau:

$$T(i,k) = \min_{k-1 \le j \le i-1} (T(j,k-1) + SSE(D, j+1, i))$$

Cho giá trị SSE tối thiểu, chúng ta có thể xác định được câu trúc tối ưu của đồ thị bằng việc tìm lại những sự lựa chọn ranh giới tối ưu của bin (cho ta thấy ở ô màu xám trong hình 3). Trong ví dụ này, biểu đồ tối ưu là $H* = \{(1, 3, 1.33), (4, 5, 4), (6, 7, 1.0)\}$

- **Differential Privacy**

Cho chuỗi số đếm D = {x1, x2, . . . , xn}

D' là chuỗi lân cận của D nếu D' khác D duy nhất chỉ 1 số đếm. Về mặt hình thức, tồn tại một số nguyên 1 <= m <= n, như là D' = {x1, x2, . . . , xm−1, xm ± 1, xm+1, . . . , xn}

Cơ chế xuất bản biểu đồ thỏa E-DP nếu nó xuất ra một biểu đồ ngẫu nhiên H:

- ∀D, D', H: Pr(Q(D) = H) ≤ e^E × Pr(Q(D')=H)
- D, D' kí hiệu cho 2 chuỗi lân cận tùy ý
- Pr(Q(D) = H) kí hiệu cho xác xuất mà Q xuất ra H với đầu vào là D.

Cơ chế đầu tiên và được sử dụng phổ biến nhất cho sự riêng tư khác biệt là cơ chế Laplace, dựa trên khái niệm độ nhạy cảm. Đặc biệt, độ nhạy Δ của truy vấn (ví dụ: một câu truy vấn biểu đồ) được xác định bằng khoảng cách L1 tối đa giữa kết quả trả về của câu truy vấn Q trên 2 cơ sở dữ liệu lận cận bất kì D và D':

$$\Delta = \max_{D,D'} \|Q(D) - Q(D')\|_1.$$

Dwork chứng minh rằng DP có thể đạt được bởi việc gây rối mỗi lần xuất của Q bởi việc nhiễu ngẫu nhiên độc lập mà nó theo phân bố Laplace trung bình 0 (zero-mean Laplace distribution) với tỷ lệ của b = Δ/E.

Trong bài toán này, giải pháp đơn giản là thêm độ nhiễu Laplace cho mỗi số đếm ở mỗi bin cho biểu đồ tối ưu không thỏa DP, bởi vì cấu trúc của biểu đồ tối ưu dựa trên dữ liệu gốc. Hậu quả là, đối thủ có thể suy ra dữ liệu nhạy cảm dựa trên cấu trúc tối ưu của

biểu đồ. Ví dụ, xem xét lại ví dụ trong hình 1(b) và giả sử rằng đối thủ biết tất cả những bệnh nhân AIDS ngoài trừ Alice. Nếu Alice không phải bệnh nhân HIV+, ở đây chỉ có 2 bệnh trong bin 40-45 tuổi, điều này dẫn đến biểu đồ tối ưu 3 bin khác như hình 4. Vì cấu trúc tối ưu của biểu đồ được xuất bản là một cái được hiện thị trong hình 2(a), đối thủ có thể suy ra Alice phải là bệnh nhân HIV+, dẫn đến vi phạm quyền riêng tư.

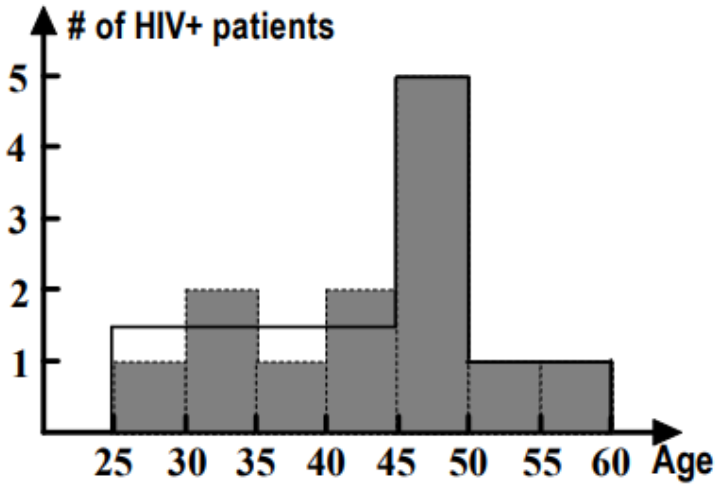


Fig. 4 Optimal histogram when Alice is excluded from the dataset

Table 1 Summary of frequent notations

Δ	query sensitivity
$D = \{x_1, \ldots, x_i, \ldots, x_n\}$	a count sequence
$D' = \{x_1, \ldots, x_i \pm 1, \ldots, x_n\}$	a neighboring count sequence of D
$\tilde{D} = \{\tilde{x}_1, \ldots, \tilde{x}_n\}$	a noisy count sequence of D
$D_i = \{x_1, \ldots, x_i\}$	a partial count sequence of D
$B_i = (l_i, r_i, c_i)$	the i^{th} histogram bin with left boundary l_i, right boundary r_i and representative value c_i
H_k^*	the optimal k-bin histogram on D
$\tilde{H}_k^*$	the optimal k-bin histogram on the noisy count sequence $\tilde{D}$
$H^*(D_i, k)$	the optimal k-bin histogram on the partial count sequence D_i
$SSE(D, l, r)/SAE(D, l, r)$	the SSE/SAE error if we merge a partial count sequence $\{x_l, \ldots, x_r\}$ into a single bin
$SSE(H, D)/SAE(H, D)$	the SSE/SAE error if we build histogram H on D

Hình 1 Bảng chú thích kí hiệu

3. NoiseFirst

Gồm hai bước:

Bước 1: Tính toán differentially private histogram với độ chi tiết tốt nhất, với chiều dài mỗi bín sử dụng cơ chế Laplace. Rõ ràng, độ nhạy của bước này là 1, vì thêm hoặc xóa bất kỳ bản ghi nào có thể thay đổi số lượng của một bin nhiều nhất là 1. Do đó, nó đủ để thêm noise Laplace với độ lớn 1/E vào mỗi bin để thỏa E-ĐP. Kết quả là 1 noisy sequence Dˆ = {xˆ1, . . . , xˆn}

Bước 2: Biểu đồ trung bình, trong đó mỗi bin xuất ra giá trị trung bình của tập hợp số lượng trong đó. NoiseFirst tính toán cấu trúc biểu đồ tối ưu dựa trên noisy count sequence Dˆ, sử dụng thuật toán lập trình động(dynamic programming algorithm). Mã giả của NoiseFirst được liệt kê trong Thuật toán 1. Rõ ràng, NoiseFirst có thể được sử dụng như một bước xử lý sau để tối ưu hóa biểu đồ Dˆ đã xuất bản được tính toán bởi LM, bằng cách hợp nhất các số nhiễu liền kề

Algorithm 1 Mean-NoiseFirst (count sequence D, the number of bins k, privacy guarantee ϵ)

1: Generate a new database $\hat{D}$ by adding independent $Lap(\frac{1}{\epsilon})$ on every count $x_i \in D$.
2: Build the optimal k-bin histogram on $\hat{D}$, denoted as $\hat{H}_k^*$, with dynamic programming method.
3: Return histogram $\hat{H}_k^* = \{(l_1, r_1, c_1), \ldots, (l_k, r_k, c_k)\}$, in which every c_j is the mean of the noisy count subsequence $\{\hat{x}_{l_j}, \ldots, \hat{x}_{r_j}\}$.

Vì NoiseFirst tính toán cấu trúc biểu đồ dựa trên số lượng nhiễu, một câu hỏi tự nhiên là liệu biểu đồ được tính có thực sự có chất lượng cao hay không. Để trả lời điều này, trong phần còn lại của phần này, chúng ta tiến hành phân tích lý thuyết về tổng sai số bình phương (SSE) dự kiến do NoiseFirst gây ra. Mục tiêu chính trong phân tích là tìm ra mối liên hệ giữa lỗi của biểu đồ tối ưu được xây dựng trên noisy count sequence D^ và lỗi của cùng một biểu đồ được xây dựng trên original count sequence D.

Đầu tiên, chúng ta phân tích tác động của việc hợp nhất các số đếm nhiễu liên tiếp (consecutive noisy counts) vào một bin. 2 lemma xác định sai số dự kiến của 1 merged bin trên noisy count sequence và original count sequence.

Lemma 1:

Một subsequence của các số đếm {xl , xl+1, . . . , xr}

- {xˆl , . . . , xˆr} : là noisy counts sau bước 1 của Algorithm 1
- Nếu (l,r,c) là bin kết quả bởi việc merge all số đếm thì squared error của bin liên quan đến {xˆl , . . . , xˆr} là: E(Error((l, r, c), {xˆl , . . . , xˆr}) = SSE(D, l, r)+2(r − l)/E^2

Lemma ở trên đánh giá SSE của một bin đơn được xây dựng trên noisy count sequence D^.

Lemma sau chỉ ra cách ước tính sai số (error) của một bin đơn liên quan đến chuỗi số ban đầu (original count sequence) D.

Lemma 2:

Một subsequence của các số đếm {xl , xl+1, . . . , xr}

- {xˆl , . . . , xˆr} : là noisy counts sau bước 1 của Algorithm 1
- Nếu (l,r,c) là bin kết quả bởi việc merge all số đếm thì squared error của bin liên quan đến {xl , . . . , xr} là: $\mathbb{E}\left(Error(\{(l,r,c\},\{x_l,\ldots,x_r\})\right) = SSE(D,l,r) + \frac{2}{\epsilon^2}$.

Lemma 2 cho ta thấy error của histogram với single-bin là SSE(D,l,r) + 2/E^2 . Nó ngụ ý rằng độ chính xác của histogram thô nhất (single-bin) thì chính xác hơn histogram được làm mịn nhất (unit-bin), khi E là đủ nhỏ,

$$\epsilon < \sqrt{\frac{2(r-l)}{SSE(D,l,r)}}$$

Hk : histogram có k bin cùng cấu trúc như H^k* nhưng khác nhau số đếm trong các bin.

Thay vì sử dụng noisy counts {xˆlj , . . . , xˆrj }, Hk tính số đếm trung bình cho bin Bj sử dụng số đếm gốc (xlj + . . . + xrj)/(rj − lj + 1) $(x_{l_j} + \ldots + x_{r_j})/(r_j - l_j + 1)$ thì $Error(H_k, D) = \sum_{j=1}^{k} SSE(D, l_j, r_j)$.

Theorem 1:

Cho Hk và H^k*, sum of squared error của histogram trên Dˆ và D lần lượt là:

$$\mathbb{E}(Error(\hat{H}_k^*, \hat{D})) = Error(H_k, D) + \frac{2(n-k)}{\epsilon^2}$$
$$\mathbb{E}(Error(\hat{H}_k^*, D)) = Error(H_k, D) + \frac{2k}{\epsilon^2}$$

Vì k cố định trước khi chạy NoiseFrist, theorem ở trên chỉ ra chúng tối ưu hóa histogram bằng cách làm nhỏ $Error(\hat{H}_k^*, \hat{D})$ dẫn đến làm nhỏ $Error(\hat{H}_k^*, D)$:

Nó vẫn còn để làm rõ cách chọn một giá trị thích hợp cho tham số k. Vì dữ liệu nhiễu đã đáp ứngε-DP, chúng ta chỉ thực hiện thuật toán n lần, với các giá trị tham số k = 1,. . . , nvà trả về k tốt nhất để làm giảm thiểu expected SSE. Để tính toán expected SSE, mặc dù không thể đánh giá trực tiếp lỗi dự kiến trên D, chúng ta có thể tận dụng lại Theorem 1 một lần nữa. Nếu H^k * là histogram tối ưu trả về bởi thuật toán 1 với k bins (k = 1, … , n), kết quả cuối cùng histogram H^k* được chọn dựa trên mục tiêu tối ưu hóa:

$$\hat{H}_{k^*}^* = \arg\min_{\hat{H}_k^*} \mathbb{E}\left(Error(\hat{H}_k^*, \hat{D}) - \frac{2n-4k}{\epsilon^2}\right)$$

Algorithm 2: chi tiết việc chọn tham số tối ưu k cho NoiseFirst.

Algorithm 2 ComputeOptimalK (noisy count sequence $\hat{D}$ with $|\hat{D}| = n$, privacy guarantee ϵ)

```
for each i from 1 up to n do
    T̂SSE[i, 1] := SSE(D, 1, i)
    for each j from 2 to n do
        T̂SSE[i, j] := +∞
        for each l := j − 1 down to 1 do
            T̂SSE[i, j] := min (T̂SSE[i, j],
                               T̂SSE[l, j − 1] + SSE(D̂, l + 1, i))
minErr = +∞
for each k from 1 up to n do
    Err = T̂SSE[n, k] − (2n − 4k)/ε²
    if Err < minErr then
        minErr = Err
        k* = k
Return k*
```

Bằng việc tính noisy counts, noise Laplace trung bình 0 (the zero-mean Laplace noise) sẽ được thêm vào cho mỗi số đếm được làm mịn lại, vì vậy NoiseFirst cải thiện được sộ chính xác so với giải pháp LM. Nói cách khác, NoiseFirst có cùng sộ nhạy cảm như LM, và tiêm độ nhiễu Lap(1/E) cho mỗi số đếm trong suốt bước 1. Như đã trình bày ở vd 1 của section 1, độ nhạy cảm của histogram có thể tiếp tục giảm, nếu chúng ta tiêm độ nhiễu sau xây dựng histogram – StructureFrist.

- **Thêm nhiễu**

Thêm độ nhiễu Laplace cho mỗi số đếm ở mỗi bin cho biểu đồ không thỏa DP, bởi vì cấu trúc của biểu đồ dựa trên dữ liệu gốc.

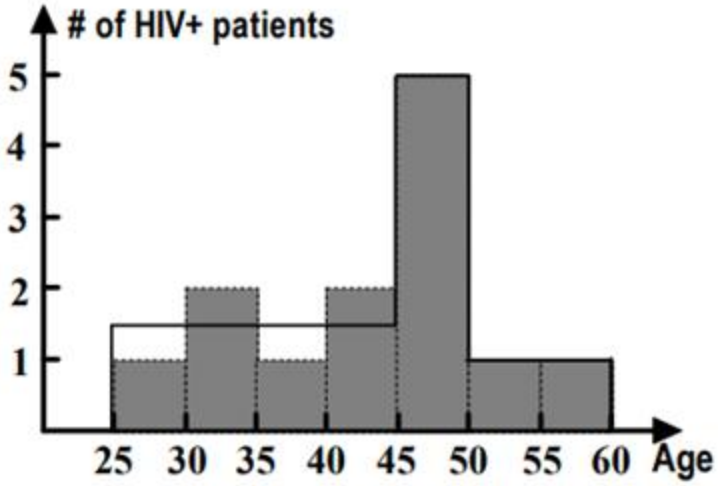


Fig. 4 Optimal histogram when Alice is excluded from the dataset

Trong bài toán này, giải pháp đơn giản là thêm độ nhiễu Laplace cho mỗi số đếm ở mỗi bin cho biểu đồ tối ưu không thỏa DP, bởi vì cấu trúc của biểu đồ tối ưu dựa trên dữ liệu gốc. Hậu quả là, đối thủ có thể suy ra dữ liệu nhạy cảm dựa trên cấu trúc tối ưu của biểu đồ. Ví dụ, xem xét lại ví dụ trong hình 1(b) và giả sử rằng đối thủ biết tất cả những bệnh nhân AIDS ngoài trừ Alice. Nếu Alice không phải bệnh nhân HIV+, ở đây chỉ có 2 bệnh trong bin 40-45 tuổi, điều này dẫn đến biểu đồ tối ưu 3 bin khác như hình 4. Vì cấu trúc tối ưu của biểu đồ được xuất bản là một cái được hiện thị trong hình 2(a), đối thủ có thể suy ra Alice phải là bệnh nhân HIV+, dẫn đến vi phạm quyền riêng tư.

4. StructureFirst

Không giống như NoiseFirst, thuật toán StructureFirst tính toán cấu trúc biểu đồ tối ưu (optimal histogram structure) trên chuỗi đếm ban đầu (optimal histogram structure), trước khi thêm nhiễu Laplace (Laplace noise) vào mỗi lần đếm (count). -> cấu trúc biểu đồ tối ưu (optimal histogram structure) rất nhạy cảm -> Cần privacy budget E để bảo vệ.

- Di: chỉ count sequence từng phần Di = {x1, x2, . . . , xi}
- H^* (Di , j) : chỉ optimal histogram với bins j trên Di.

Thuật toán 3

Nó xây dựng biểu đồ tối ưu có k bins dựa trên chuỗi số gốc D. Tất cả các kết quả trung gian được lưu trữ trong một bảng, như được thực hiện trong Hình 3. Để bảo vệ cấu trúc của biểu đồ tối ưu, StructureFirst di chuyển ngẫu nhiên ranh giới giữa các bin của biểu đồ. Khi chọn ranh giới giữa bin Bj và bin Bj + 1, cụ thể là rj ở đây, rj được đặt về vị trí q ∈ [1, |D| + 1] với xác suất là tỷ số sau:

$$\exp\left\{-\frac{\epsilon_1 E_{SSE}(q,j,r_{j+1})}{2(k-1)(2F+1)}\right\}$$

Algorithm 3 Mean-StructureFirst (count sequence D with $|D| = n$, the number of bins k, privacy guarantee ϵ, count upper bound F)

1: Partition ϵ into two parts ϵ_1 and ϵ_2 that $\epsilon_1 + \epsilon_2 = \epsilon$.
2: Build the optimal histogram H_k^* on D and keep all intermediate results, i.e., $Error(H^*(D_i, j), D_i)$ for all $1 \le i \le n$ and $1 \le j \le k$
3: Set $r_k = n + 1$
4: **for** each j from $k-1$ down to 1 **do**
5: **for** each q from j up to $(r_{j+1} - 1)$ **do**
6: Calculate $E_{SSE}(q, j, r_{j+1})$ $= Error(H^*(D_q, j), D_q) + SSE(D, q+1, r_{j+1})$
7: set $r_j = q\,(k-1 \le q < r_{j+1})$ with a probability proportional to $\exp\left\{-\frac{\epsilon_1 E_{SSE}(q,j,r_{j+1})}{2(k-1)(2F+1)}\right\}$
8: Set $l_{j+1} = r_j + 1$
9: Calculate the average count c_j for every bin with interval $[l_j, r_j]$.
10: Add Laplace noise to the average counts as $\hat{c}_j = c_j + Lap\left(\frac{1}{\epsilon_2(r_j - l_j + 1)}\right)$
11: Return histogram $\{(l_1, r_1, \hat{c}_1), \ldots, (l_k, r_k, \hat{c}_k)\}$

D. Datamining

Chúng ta xem vấn đề khai thác dữ liệu với quyền riêng tư chính thức đảm bảo, được cung cấp giao diện truy cập dữ liệu dựa trên khuôn khổ quyền riêng tư hấp dẫn khác nhau. Sự riêng tư khác biệt yêu cầu điều đó tính toán không nhạy cảm với những thay đổi trong bất kỳ thay đổi cụ thể nào trong bản ghi của dividual, do đó hạn chế rò rỉ dữ

liệu qua kết quả. Giao diện bảo vệ quyền riêng tư đảm bảo quyền truy cập dữ liệu an toàn theo từng giai đoạn và không yêu cầu từ người khai thác dữ liệu bất kỳ chuyên môn nào về quyền riêng tư. Tuy nhiên, khi chúng ta hiển thị trong bài báo, một cách sử dụng ngây thơ của giao diện để xây dựng quyền riêng tư bảo vệ các thuật toán khai thác dữ liệu có thể dẫn đến kết quả khai thác dữ liệu kém hơn. Chúng ta giải quyết vấn đề này bằng cách xem xét quyền riêng tư và các yêu cầu thuật toán đồng thời, tập trung vào cảm ứng cây quyết định như một ứng dụng sam sung. Cơ chế bảo mật có ảnh hưởng sâu sắc đến hiệu suất của các phương pháp được chọn bởi dữ liệu thợ mỏ. Chúng ta chứng minh rằng sự lựa chọn này có thể làm cho sự khác biệt giữa bộ phân loại chính xác và bộ phân loại hoàn toàn một cái vô dụng. Hơn nữa, một thuật toán cải tiến có thể đạt được cùng mức độ chính xác và quyền riêng tư như cách đề cập ngây thơ nhưng với mức độ học ít hơn mẫu.

1. Cách hoạt động

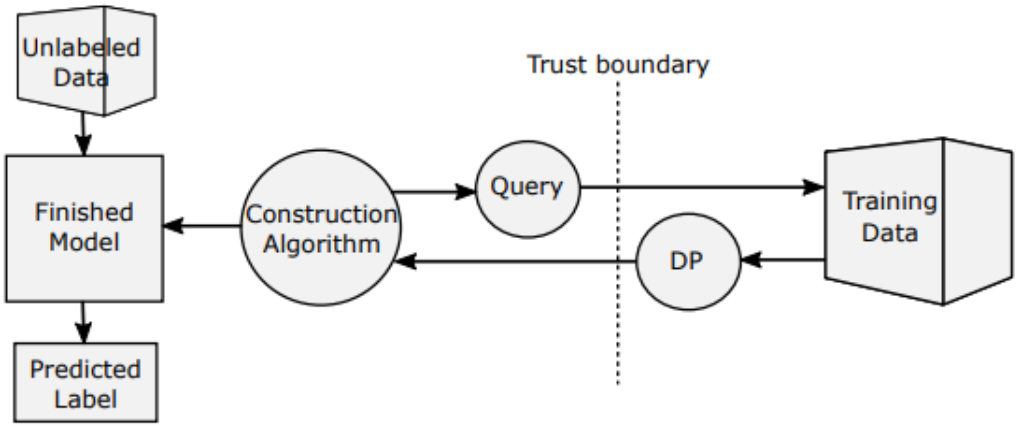


FIGURE 2.4: High-level representation of the analyst's interface with private data, using differential privacy (DP).

Hình thể hiện đường đi của thuật toán khai thác dữ liệu DP; một thuật toán gửi một truy vấn tới tập dữ liệu, tập dữ liệu sẽ tính toán câu trả lời cho truy vấn và sau đó dùng cơ chế DP sửa đổi câu trả lời theo cách bảo vệ quyền riêng tư của mỗi cá nhân trong tập dữ liệu.

Đối với bất kỳ hành động truy vấn f nào, giá trị của ε bé hơn hoặc bằng lượng người chủ tập dữ liệu x cung cấp cho nhà phân tích. Chúng tôi định nghĩa số lượng này là tổng privacy budget β.

Mỗi lần người dùng truy vấn tập dữ liệu, một lượng β được sử dụng tùy thuộc vào mức độ xâm phạm của truy vấn đối với quyền riêng tư cá cá nhân được mô tả trong tập dữ liệu và mức độ chính xác của kết quả trả về của câu truy vấn mà nhà phân tích mong muốn. Khi tất cả β được sử dụng, nhà phân tích sẽ mất quyền truy cập vào dữ liệu vĩnh viễn.

2. Privacy budget β

Một phần của privacy budget β cần được sử dụng bất cứ khi nào dữ liệu cá nhân được truy vấn. Điều này có nghĩa là điều quan trọng là phải xác định chính xác dữ liệu cần được truy vấn để xây dựng cây quyết định. Số lần dữ liệu cần được truy vấn càng ít thì sử dụng càng ít privacy budget.

3. Differentially-Private Decision Trees

Trong phần này, chúng tôi đề xuất một số chiến lược để xây dựng mô hình cây quyết định riêng tư khác biệt. Trước khi chứng minh cách chúng tôi cải thiện tính năng hiện đại, trước tiên, chúng tôi bối cảnh hóa công việc của mình theo tính năng hiện đại nhất trong phần này.

Một số công trình trước đây đã khảo sát xung đột giữa khai thác dữ liệu và quyền riêng tư khác biệt. Chúng tôi tập trung vào cây quyết định nói riêng, đi sâu hơn vào sự phức tạp của các thuật toán cây quyết định riêng tư khác biệt. Sarwate và Chaudhuri [2013] đã cung cấp tổng quan chung về học máy với quyền riêng tư khác biệt, thảo luận ngắn gọn về phân loại, hồi quy, giảm kích thước, chuỗi thời gian, lọc và một số "khối xây dựng" cơ bản của quyền riêng tư khác biệt, chẳng hạn như như sự khác biệt giữa đầu vào, đầu ra và thêm nhiễu vào mục tiêu, cơ chế hàm mũ và số liệu thống kê khác biệt-riêng tư mạnh mẽ. Ji và cộng sự. [2014] tập trung nhiều hơn vào các thuật toán khai thác dữ liệu vi phân-riêng tư cụ thể, cung cấp tổng quan về công việc được thực hiện

với mô hình Bayes ngây thơ, hồi quy tuyến tính, máy vectơ hỗ trợ tuyến tính [Rubinstein và cộng sự, 2012], hồi quy logistic, máy vectơ hỗ trợ hạt nhân [Jain và Thakurta, 2013], cây quyết định, lập trình lồi trực tuyến, phân cụm [Chen và cộng sự, 2015], lựa chọn đối tượng, PCA [Chaudhuri và cộng sự, 2013], và ước tính thống kê.

4. Truy cập dữ liệu Differentially-Private

Một phần của ngân sách bảo mật β cần được chi tiêu bất cứ khi nào dữ liệu cá nhân được truy vấn. Điều này có nghĩa là điều quan trọng là phải xác định thời điểm chính xác dữ liệu cần được truy vấn theo thứ tự để xây dựng cây quyết định. Số lần dữ liệu cần được truy vấn càng ít thì ngân sách chúng ta cần chi tiêu càng ít. Đôi khi dữ liệu sẽ không nhất thiết phải được truy vấn, nhưng làm như vậy sẽ vẫn đáng giá với chi phí bảo mật do trình phân loại hoạt động tốt hơn nhờ nó.

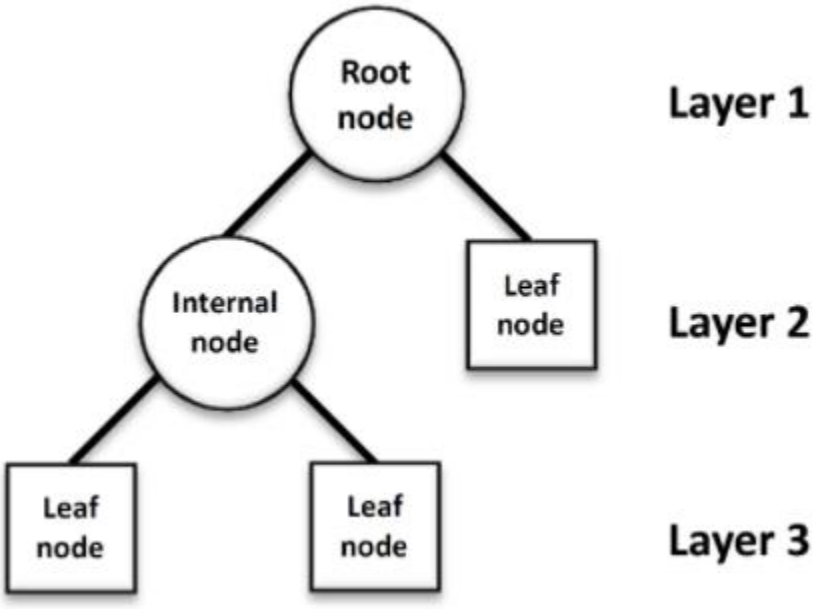
Những truy vấn nào nên được coi là “bắt buộc” phụ thuộc vào một đặc điểm chính của thuật toán xây dựng cây: liệu nó xây dựng cây một cách tham lam hay ngẫu nhiên.

Thuật toán “tham lam” (greedily), là một thuật toán sử dụng một hàm mục tiêu cục bộ trong mỗi nút; theo kinh nghiệm tìm ra thuộc tính tốt nhất để phân chia tập hợp con dữ liệu cục bộ chứa trong một nút. Bất kể hàm mục tiêu nào được sử dụng (có thể là information gain, chỉ mục gini(gini index) hoặc bất kỳ chức năng nào khác), nó yêu cầu truy vấn dữ liệu cục bộ ít nhất một lần. Chúng tôi gọi những điều này là "truy vấn không phải lá", vì chúng được thực hiện trong mỗi nút không phải của cây.

Truy vấn bắt buộc khác cho cây tham lam (greedily tree) cũng giống như truy vấn bắt buộc duy nhất cho cây ngẫu nhiên(random tree).

Thuật toán "ngẫu nhiên", chúng tôi đề cập đến các thuật toán không sử dụng một hàm mục tiêu trong mỗi nút và thay vào đó chọn các thuộc tính một cách ngẫu nhiên. Truy vấn bắt buộc mà cả hai danh mục cây quyết định chia sẻ là một truy vấn cho phép dự đoán các nhãn lớp cho dữ liệu không được gắn nhãn, không được gắn nhãn. Trong hầu hết các trường hợp, điều này có dạng truy vấn phân phối số lượng lớp trong mỗi nút lá,

với nhãn lớp phổ biến nhất (tức là đa số) được sử dụng làm nhãn dự đoán cho các bản ghi trong tương lai. Chúng tôi gọi những truy vấn này là "truy vấn lá".



- **Non-Leaf Queries**

Quyết định phân chia thuộc tính nào cho mỗi node bằng để tối ưu hóa khả năng phân biệt các nhãn lớp của nó là cốt lõi của thuật toán cây quyết định.

Các thuật toán non-private truyền thống có thể được chuyển đổi để đạt được differential privacy, bằng cách rephrasing các hàm phân tách theo các truy vấn.

Blum đã làm Information Gain differentially-private bằng cách chia nhỏ nó thành hai truy vấn đếm cho mỗi thuộc tính a, và sau đó làm cho các truy vấn đếm phân biệt-riêng tư với cơ chế Laplace:

Query 1: $n_i^{v,c} + Lap(1/\epsilon); \forall c \in C, \forall v \in A$ and

Query 2: $n_i^v + Lap(1/\epsilon); \forall v \in A.$

+ ni : số lượng records trên node i,

+ ni^v : số lượng records trên node i với giá trị v (thuộc nhãn phân lớp c)

Đầu ra của DP sẽ không ảnh hướng đến vi phạm quyền riêng tư trong qua trình tiền xử lí nên không cần thiết phải làm gì thêm để đáp ứng DP trong non-leaf nodes.

Nếu chúng tôi coi tổng ngân sách bảo mật của người dùng là β, thì hai truy vấn được liệt kê ở trên chỉ có thể nhận được một phần nhỏ của β mỗi truy vấn. Khi truy vấn bất kỳ thuộc tính A nhất định nào với các truy vấn trên, số đếm cho mỗi tổ hợp giá trị v và nhãn c liên quan đến một tập con rời rạc của xi (tức là dữ liệu trong nút i), cho phép Truy vấn 1 được tạo song song ∀c ∈ C, ∀v ∈ A. Tuy nhiên, truy vấn 2 sử dụng cùng dữ liệu với truy vấn đầu tiên và do đó, chi phí bảo mật của nó được tính bằng chi phí của truy vấn đầu tiến. Điều này sau đó cần được lặp lại cho mọi thuộc tính A, ở mọi cấp độ của cây (các nút anh chị em chứa các tập con rời rạc của x và có thể được tạo song song).

Mỗi truy vấn trong thuật toán có một phần trong tổng privacy budget β bằng:

$$\epsilon = \frac{\beta}{2md}$$

Trong đó:

- m là số thuộc tính

- d là độ sâu của cây (bao gồm cả node root và node lá)

Friedman và Schuster [2010] đề xuất xây dựng cây quyết định bằng cách truy vấn tập dữ liệu hai lần tại mỗi node (dựa trên cơ chế lũy thừa - Exponential mechanism)

Query 1: Đếm có bao nhiêu bản ghi trong nút

Query 2: Tìm thuộc tính tốt nhất để tách các bản ghi.

Điều này có nghĩa là đối với tổng ngân sách bảo mật β nhất định, mỗi truy vấn được gửi đến tập dữ liệu có một phần ngân sách bằng:

$$\epsilon = \frac{\beta}{2d}$$

Hình thể hiện đường đi của thuật toán khai thác dữ liệu khác biệt-riêng tư; một thuật toán gửi một truy vấn tới tập dữ liệu, tập dữ liệu sẽ tính toán câu trả lời cho truy vấn và sau đó cơ chế riêng tư khác biệt sửa đổi câu trả lời theo cách bảo vệ quyền riêng tư của mỗi cá nhân trong tập dữ liệu.

- **Leaf Queries**

Thông tin được yêu cầu từ dữ liệu trong một nút lá khá khác với thông tin được yêu cầu trong một nút không phải là lá. Thay vì tìm thuộc tính tốt nhất để phân vùng dữ liệu, mục đích của nút lá là để dự đoán nhãn lớp của các bản ghi không được gắn nhãn.

Có thể xem truy vấn là một truy vấn đơn được gửi song song cho tất cả các nút lá j, trả về biểu đồ của lớp đếm trong các nút đó:

$$\left\{n_j^c + Lap(1/\epsilon); \forall c \in C\right\}; \forall j$$

Những số đếm mỗi phân lớp của mỗi node lá sẽ được thực hiện DP bằng cách thêm Lap(1/ε).

Phân công công việc

Sinh viên	**Công việc**
Đoàn Thục Quyên - 18521320	Tìm hiểu về Differential Privacy, làm báo cáo và trình bày Histogram, Data Mining
Dương Bảo Nam - 18521118	Tìm hiểu về Differential Privacy, làm báo cáo và trình bày Location Privacy