淺談人工智慧系統的隱私資訊安全保護機制

Introduction to Privacy-preserving Mechanisms for Artificial Intelligent Systems

王紹睿
Peter Shaojui Wang

隨著人工智慧系統的興起和普及，無所不在的智慧服務讓人們生活更加便利，特別是大數據分析技術的發展，讓工業界和學術界可以觀察到許多過去不容易察覺的趨勢或現象，但同時這些智慧服務所儲存和使用的大量個人資料也讓隱私資訊安全議題受到越來越多的重視。本文探討適用於人工智慧系統特別是大數據分析系統的隱私資訊安全保護機制，回顧這個領域的發展情況及目前熱門技術，並探討常見的使用情境和應用。

With the rise of artificial intelligence, many intelligent services have benefited our life; one of them is Big Data technology, which has helped the industry and the academia get the trends that are not easy to be observed in the past. However, it also has raised the serious privacy concerns while the massive amounts of our personal data are routinely collected in this Big Data era. This paper introduces several privacy-preserving mechanisms for artificial intelligent systems, especially Big Data systems, gives a review, and discusses all possible scenarios of them.

一、前言

近年來人工智慧系統逐漸普及化，包括大數據分析技術等相關技術開始應用到各行各業，促進人們生活更加便利；但在同時，資料隱私安全問題卻也帶來許多隱憂。因此各大廠商例如 Google, Apple 和 Microsoft 等無不投入大量研發資源在隱私安全保護技術上，例如 Google 和 Apple 公司已採用差分隱私技術 (differential privacy) 在他們的 Chrome 瀏覽器以及在 iOS 作業系統的匿名化當機報告上，Google 甚至公開其原始碼 (source code)，而 Microsoft 也應用差分隱私技術在資料庫查詢分析時的隱私保護問題上。本文即在介紹人工智慧特別是大數據分析相關系統的隱私保護機制，包括前面提到的差分隱私技術 (differential privacy) 以及其他各種不同的隱私保護處理技術，並介紹這些技術能夠運用在哪些常見的使用情境之中。

二、隱私保護技術簡介

1. 加隨機亂數雜訊法

加隨機亂數雜訊法 (randomization) 是最直覺也最早開始發展的方法，主要精神是直接對全部資料加少許雜訊，但這個雜訊可以在計算最後被消除，

或者雖無法消除但對計算結果影響很小。最簡單的例子就是直接對全部資料加一個平均值為零的亂數雜訊，若我們要對這擾亂後的全部資料來求平均值的話，會發現亂數雜訊對整體計算結果並無影響。

除此之外，還有一種幾何學上的做法，其基本原理是對全部資料點做幾何旋轉或平移[1]。一個常見的例子是假使我們要求某個資料集裡的資料的群聚分布 (data clustering)，那我們將全部的資料點在幾何空間中做同步的平移或者旋轉，這種擾亂動作並不會影響到資料在幾何意義上的群聚分布，如下圖 2 和 3，全部的資料點都做如圖 1 所述的角度旋轉並不會影響到他們的群聚現象和結果。

在優缺點比較方面，上述兩種方法如果是僅添加能在最後計算結果被消除影響的雜訊或者旋轉平移，攻擊者都很有可能能夠還原出原始資料值[1]，因此系統設計者為了解決這種安全性不夠的問題，常常會更多地使用將對計算結果造成影響且無法還原的亂數雜訊，此舉雖然保證了安全性，卻同時會對計算精準度造成誤差。除此之外，以上作法皆無法對抗資料再識別化攻擊 (re-identification attack)，這也是下一種方法 K 匿名方法 (K-anonymity) 被提出的主因。

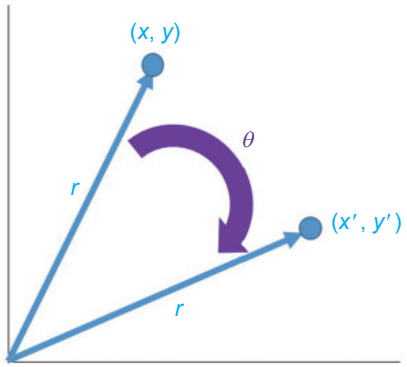

圖 1. 將資料點作旋轉角度 θ。

2. K 匿名方法

K 匿名方法 (K-anonymity) 的發展始自 1998 年資安專家 Latanya Sweeney 還在美國麻省理工學院攻讀博士學位時發現的一個非常著名的隱私安全漏洞：資料再識別化攻擊 (re-identification attack)[2]。在她之前的隱私安全研究大都專注在如何弄亂或刪除資料集中的敏感性資料，例如姓名或身份證字號，但她卻發現即使已經弄亂或甚至已經徹底刪除所謂的敏感性資料，攻擊者仍有可能透過資料集中的其他非敏感性欄位資料與其他外部資料的串連比對，找出原本經過隱私處理的資料集中的每筆資料

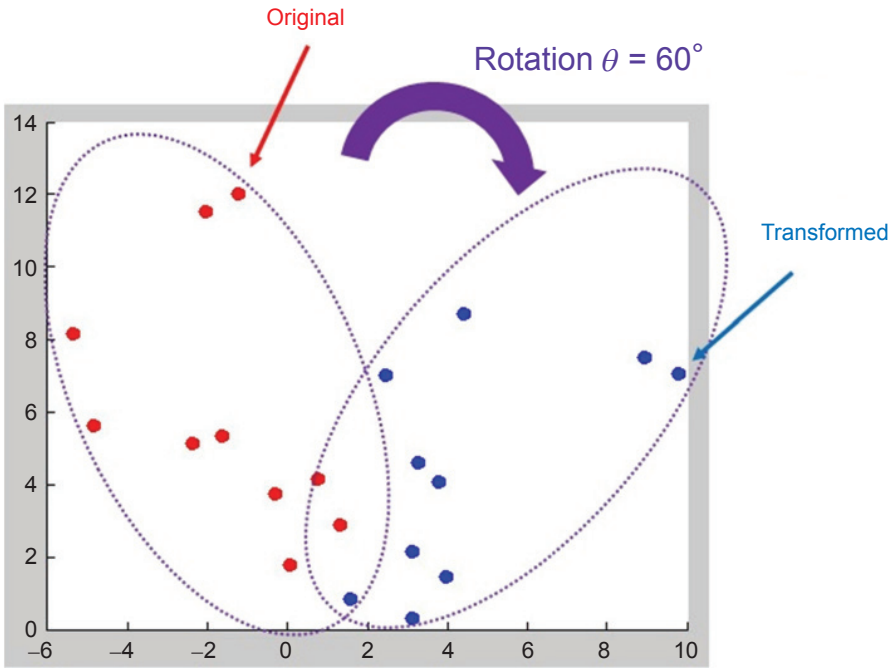

圖 2. 對所有資料點做旋轉 ($\theta = 60°$)。

	X	Y
1	−1.6417	5.3338
2	−1.2333	12.0241
3	0.75100	4.1736
4	−5.3449	8.1687
5	0.0639	1.7738
6	−2.3686	5.1322
7	1.3071	2.8718
8	−2.0504	11.5126
9	−4.8274	5.6395
10	−0.3043	3.7631

	X	Y
1	3.7984	4.0887
2	9.7966	7.0801
3	3.9899	1.4364
4	4.4019	8.7132
5	1.5681	0.8316
6	3.2603	4.6174
7	3.1406	0.3039
8	8.9450	7.5320
9	2.4702	7.0004
10	3.1068	2.1451

圖 3. 針對上圖 x，左表格是原始資料點資料，右表格是對所有資料點做旋轉 600 後的結果。

Name	Birthday	Gender	ZIP	Disease
Mark	1980/1/2	M	12301	cardiopathy
Jenny	1972/4/1	F	14232	hypertension
James	1981/5/7	M	21374	arthritis
Mary	1970/9/9	F	32479	leukemia

圖 4. 醫療記錄資料。

是對應到哪個特定人士。她當年提出的例子非常生動且真實，以致這個方法或類似的改進方法目前仍廣為各國政府 (包括台灣) 採用。當年她就讀博士班所身處的麻州正舉行市長選舉，同時她因為學校研究的關係拿到一份由麻州醫療體系提供的醫療大數據資料，當時這是號稱去隱私化的安全資料：刪除了姓名等的敏感資訊，如圖 4。然而經過檢視之後，Latanya Sweeney 意外地發現這份已刪除姓名的醫療數據資料其剩餘欄位的資訊竟剛好可和當時麻州市長選舉公報上提供的一些資訊彼此吻合，如圖 5，並因此可還原出原本醫療數據資料裡被刪除的姓名 (當然是只有這幾位候選人的姓名)。Latanya Sweeney 甚至由此發現了當年市長選舉當選人的一些特殊醫療記錄並引起廣大的討論。這件事在當年非常轟動，美國政府也因此決定聘請她替政府制訂相關隱私安全標準規範，而因為美國政府的權威性，全世界 (包括台灣) 從那時候深受這套標準的影響至今。雖然這套標準和解決方案在多年後，確切地說是 2006 年，由微軟另一位資安研究員 Cynthia Dwork 發現其安全保護程度上的侷限，並提出更新一代的方法「差分隱私」(differential privacy)，但由於 Latanya Sweeney 提出的解決方案在實作上比較容易和穩定，因此包含台灣在內的各國政府目前在釋出所謂的 Big Data 大數據資料時

Name	Birthday	Gender	ZIP
Mark	1980/1/2	M	12301
Jenny	1972/4/1	F	14232
James	1981/5/7	M	21374
Mary	1970/9/9	F	32479

Birthday	Gender	ZIP	Disease
1980/1/2	M	12301	cardiopathy
1972/4/1	F	14232	hypertension
1981/5/7	M	21374	arthritis
1970/9/9	F	32479	leukemia

圖 5. 左為選舉公報資料，右為刪除姓名後的醫療記錄資料，我們可發現兩者有吻合的欄位。

	Non-Sensitive			Sensitive
	ZIP	Age	Country	Disease
1	12301	22	US	cardiopathy
2	12301	27	RU	cardiopathy
3	12301	24	IN	arthritis
4	12301	28	RU	arthritis
5	14232	51	JP	hypertension
6	14232	53	US	cardiopathy
7	14232	49	RU	arthritis
8	14232	48	RU	arthritis
9	12301	32	RU	hypertension
10	12301	34	JP	hypertension
11	12301	39	IN	hypertension
12	12301	37	RU	hypertension

	Non-Sensitive			Sensitive
	ZIP	Age	Country	Disease
1	123**	< 30	*	cardiopathy
2	123**	< 30	*	cardiopathy
3	123**	< 30	*	arthritis
4	123**	< 30	*	arthritis
5	1423*	> 40	*	hypertension
6	1423*	> 40	*	cardiopathy
7	1423*	> 40	*	arthritis
8	1423*	> 40	*	arthritis
9	123**	3*	*	hypertension
10	123**	3*	*	hypertension
11	123**	3*	*	hypertension
12	123**	3*	*	hypertension

圖 6. 此為 4-anonymity 的例子，當 K = 4，表示攻擊者無法在所有可能的 4 筆資料組合內區分出單筆隱私資料代表的對象，左圖是原始資料，右圖是其符合 4-anonymity 的變形資料。

仍常常採用其方法，而我們平時見到各種所謂的有經過隱私處理的資料 (例如將原始資料部分內容用星號或其他符號代替) 其實也大都深受此方法的影響。

在 Latanya Sweeney 這套 K 匿名方法 (K-anonymity) 中，目標是找到一資料遮蔽方法可以讓攻擊者無法在所有可能的 4 筆資料組合內區分出單筆隱私資料代表的對象。所以 K 值越大，代表隱私保護程度越強，但相對地資料被破壞的程度也越大。關於這裡說的資料遮蔽方法，Latanya Sweeney 提出兩種方法: 第一種是使用泛化法 (generalization) 將資料轉為更廣的區間，例如年齡資料「23歲」改成 [20-25] 區間，第二種是使用抑制法 (suppression) 將資料部分內容用星號 "*" 代替，例如姓名資料「王小明」改成「王*明」，依照 Latanya Sweeney 的理論，對於大部分資料使用前述兩種方法的組合應該都能找到符合 K-anonymity 標準的隱私保護方法。例如下圖 6，左邊是一般資料，右邊是其符合 4-anonymity 隱私標準的處理結果 (在此 K = 4)。

3. 差分隱私

在前文提到 2006 年微軟一位資安研究員 Cynthia Dwork 發現 K-anonymity (及其之後衍伸出的類似方法) 在安全保護程度上的侷限，主要是沒有清楚定義隱私保護在統計學上的意義，以致於 K-anonymity 及類似方法均只能不斷針對新出現的隱私漏洞予以填補增添安全性，因此 Cynthia Dwork 經過深入研究後，提出一套被公認是在統計學上比較完整的隱私保護定義，稱作「差分隱私」(differential Privacy)[3]。

差分隱私的定義可以用一個很簡化的例子來說明，假設今天總統生病住進 A 醫院，但因為隱私原因不願向外界透漏他住的是台灣的哪間醫院，而八卦雜誌記者 (在此以系統角度來說可假設為攻擊者) 卻希望能夠知道該醫院資訊以搶得獨家新聞。雖然全台灣有非常多的醫院，但八卦雜誌記者卻有可能找到一個辨別方式：哪一間醫院的維安標準突然升級很多，那就有很大的機率是因為總統住進了這醫院。而站在總統府方面 (防衛者) 來說，隱私保護成功的關鍵就是是否能夠找到一套方法，讓 A 醫院接受總統住院和無接受總統住院這兩種情況以

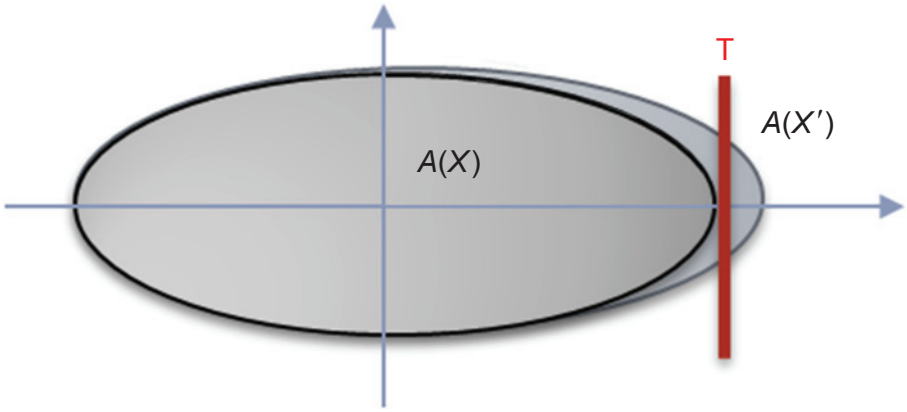

圖 7. 差分隱私的關鍵是如何找到一個足夠好的擾亂方法 A 可以使得 $A(X)$ 和 $A(X')$ 的機率分布範圍非常近似以至於攻擊者無法找到一個門檻標準 T 可以區分兩者。

外在行為結果來說 A 醫院看起來表現完全一致，也就是說能夠讓八卦雜誌記者 (攻擊者) 無論用任何辨別方式皆無法判斷總統是否住進 A 醫院。以上即為一個簡單的差分隱私定義說明，若以較嚴謹的數學來說明，如下式 1 和圖 7，假設 X 和 X' 為僅差一元素(總統)存在的資料集合 (病人集合)，A 為一基於 X 和 X' 的隨機演算法 (醫院為照顧病人而產生的各種隨機外在表現)，S 為 A 演算後的所有可能結果 (醫院為照顧病人而產生的所有可能的隨機外在表現)，Pr 為前述 A 隨機結果的機率分布範圍(醫院為照顧病人而產生的各種隨機外在表現的機率分布範圍)，e^{ϵ} 為一極小值，也就是說，若能找到一個很好的方法 A，能夠讓僅相差一元素的 X 和 X' 兩資料夾經過 A 的運算處理後，其機率分布差異非常微小，也就是 $A(X)$ 和 $A(X')$ 產生的結果看起來幾乎一致，以致攻擊者找不到任何一種辨別方式 T 能夠區分哪個是由 X 還是 X' 產生的結果，也就是達到了隱私保護的效果。

$$\frac{Pr\left[A(X)\in S\right]}{Pr\left[A(X')\in S\right]}\le e^{\epsilon} \qquad (1)$$

差分隱私可說是目前最高的隱私標準，但由於他本質上只是一個統計學的定義，並沒有明確說包括演算法等應該一定要怎麼做，因此在實務上也最難應用實作。例如，在實務上依照不同應用情境，差分隱私工具還可再分為交互式 (interactive) 和非交互式 (non-interactive) 兩種類型的：

- 交互式 (interactive)：在此應用情境中，使用者提出一個查詢 (query)，資料庫系統返回一個擾亂後的結果給他。這其實比較符合差分隱私最一開始提出的定義。目前大部分使用 differential privacy 的工具屬此。
- 非交互式 (non-interactive)：資料擁有者直接釋出一個擾亂後的資料集給使用者自由使用。就類似前面提的使用 K 匿名方法 (K-anonymity) 這派方法從一個原始資料集產出一個擾亂後的資料集給別人使用。如果要從差分隱私的角度來說明，就是這個擾亂後的資料集可以應付所有未來所有可能的查詢 (query)，仍能夠維持差分隱私標準。

4. 安全多方計算

安全多方計算 (secure multi-party computation) 是適用於協同運算情境(例如協同式資料探勘系統)，解決一組互不信任的參與方之間如何保護彼此隱私的問題。這個領域一開始是由第一位獲得圖靈獎 (Turing Award) 的華人姚期智提出的百萬富翁問題 (Yao's millionaire problem) 所引領出來的[4]。姚期智的百萬富翁問題是假如有兩個富翁他們能否在不揭露彼此財富數目的情況下知道誰比較有錢。

以下舉一個更簡化的例子來說明安全多方計算的基本概念 (不考慮成員之間彼此串通的問題)。如

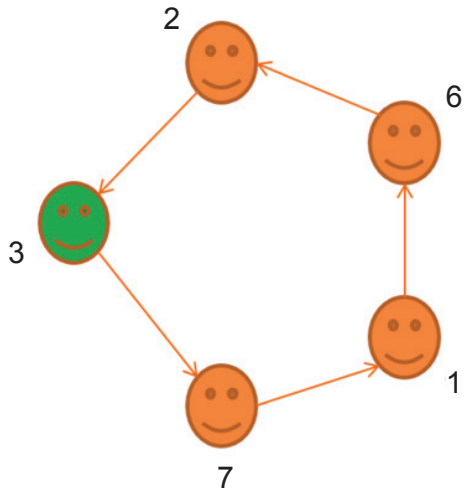

圖 8. 假設有五個人手上各有一些錢 (如圖中的數字)。

下圖 8，假設有五個人 (如圖中五個人臉) 手上各有一些錢 (如圖中的數字)，他們能否在不揭露彼此錢財數目的情況下知道所有人財產的總和呢？一個方法是首先任選一個人當作發起者(如圖中綠色人臉)，他自己想一個隨機數字 (例如 100) 然後和手上的錢財數字 (3) 做相加，將數字和 (103) 傳給下一個人，而下一個人因為看到的數字 (103) 是被隨機數字 (100) 擾亂後的結果所以無法知道前一位的真實錢財數字是多少，然後他就盲目將自己手上的錢財數目 (7) 和這個擾亂後的數字 (103) 做相加，並將總和傳給下一位，接著所有的下一位都做類似的動作，如下圖 9，直到最後一位將自己手上的錢財數目和前面的總和相加後，將最終總和傳回給當初的第一位發起者。而第一位發起者因為隨機數字 (100) 是他自己想的，所以他知道這個別人不知道的數字，他就將最終總和 (119) 減去這個隨機數字 (100)，就能得到所有人的財富總和數值 (19)，如下圖 10。在上述過程中所有人 (包括發起者) 都不知道彼此的財富數字，但經過這樣的運算協定後仍然可以得到所有人的總財富數目。

這個領域的方法已經被應用在包括分散式支持向量機 (distributed support vector machine) 等演算法系統[5]，例如在分散式支持向量機方法中，可以將核函數值 (kernel) 當作中介值，利用這個安全多方計算方法 (secure multi-party computation) 將所有

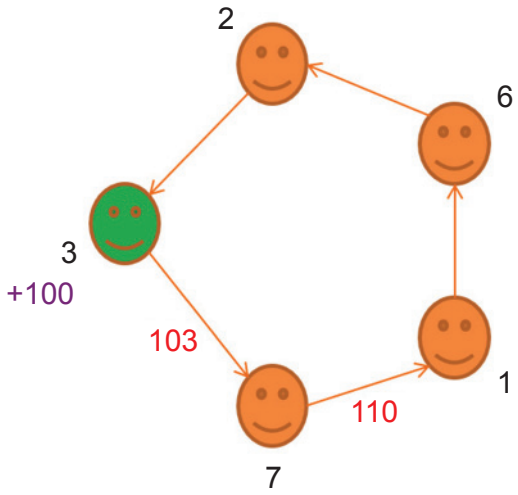

圖 9. 發起者想一個隨機數字 100 並和手上的錢財數字 3 做相加，將數字和 103 傳給下一個人，而下一個人再將自己的數字 7 和這個擾亂後的數字 103 做相加，再將數字和 110 傳給下一個人。

協同運算的成員的核函數值 (kernel) 取得總和 (sum of kernel values) 繼續完成支持向量機運算，而在過程中不用洩漏彼此成員的核函數值 (因為根據研究[5]，當核函數值被洩漏是有可能被攻擊還原出原始資料值的)。

除此之外，這個領域的方法有一個額外的好處是通常能夠在不喪失資料精準度的情況下保護資料

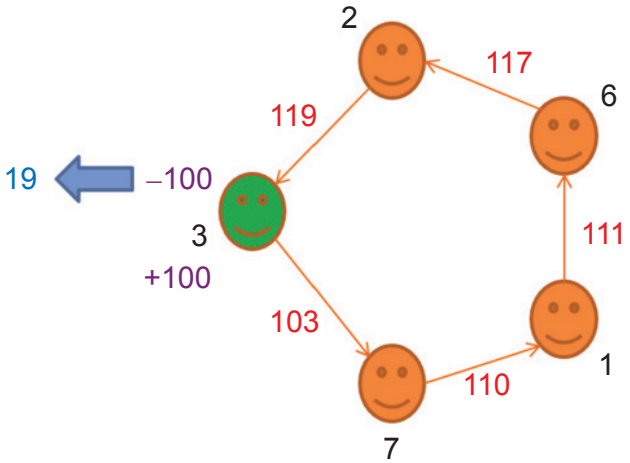

圖 10. 最後一位將最終總和 119 傳回給發起者。而發起者減去自己想的隨機數字 100，就能得到所有人的財富總和數目 19。

隱私，這對於某些要求高精準度的人工智慧資料探勘演算法系統而言是十分吸引人的。

5. 同態加密法

同態加密法 (homomorphic encryption) 是提供原文加密成密文後仍舊能在密文層次做運算的方法，例如經過這種加密方法做的密文可以直接彼此做加減乘除，計算的結果在解密後會恰等於是原本明文做相同的加減乘除後的結果。這種特性特別適用於雲端運算 (cloud computing) 的隱私保護，因為只要讓用戶先在自己的電腦上對資料做這種加密，然後將加密後的密文上傳到雲端，雲端服務提供者就可以在不用知道明文資料的狀態下直接對密文做數學運算，然後將運算結果傳回給用戶電腦，最後因為只有用戶才有解密金鑰，所以用戶就在自己的電腦上做解密，得到真正的運算結果明文。而在這過程中雲端服務完全不知道資料真正的值是什麼，他看到的只有一堆密文。

這個領域最早可追溯自 1978 年 RSA 加密標準被提出後，RSA 發明人就曾提到過，但當年只能做部分同態加密 (partially homomorphic encryption)，也就是只能選擇對密文選對加法或乘法其中之一的方法來做運算[6]。第一個全同態加密 (fully homomorphic encryption) 要等到 2009 年由 IBM 的 Gentry 提出[7]，當年這是件轟動密碼學界的大事，因為若能證明加法和乘法都能做同態加密，表示所有的數學運算都能適用：原因是所有的數學運算都能夠轉化為以布林運算 (Boolean operation) 來表示，而在布林運算中，所有的運算都能轉化為 AND 和 XOR 兩種邏輯閘 (Gate) 的組合，而 AND 和 XOR 正分別對應數學中的乘法和加法；因此，只要能夠同時對加法和乘法都能做同態加密，也就是所謂的全同態加密，則所有的數學運算皆能適用。

雖然在 2009 年 Gentry 首次提出一個數學上正確的全同態加密方法，但是在實作上卻會讓系統運作非常緩慢，例如做一個簡單加法可能要花幾個小時，因此非常地不實用。雖然在隨後的幾年時間內，後繼研究人員不斷將這個數學運算效能提高，但是這離真正可用的程度尚有一段距離。

三、常見使用情境和工具

對於人工智慧特別是大數據分析的系統流程來說，一般有三個可能進行隱私保護處理的位置，分別是資料蒐集階段、演算法本身、系統輸出部分，如下圖 11 所示。這是因為根據不同使用情境，導致加雜訊／進行隱私處理的位置不同，以下將接著介紹幾種常見的使用情境，並說明各個情境可能的解決技術及常用工具。

1. 非交互式資料釋出

- [使用情境]

  非交互式資料釋出 (non-interactive publishing) 的使用情境是資料擁有者直接釋出一個擾亂後的資料集給外界使用者自由使用。例如 K 匿名方法

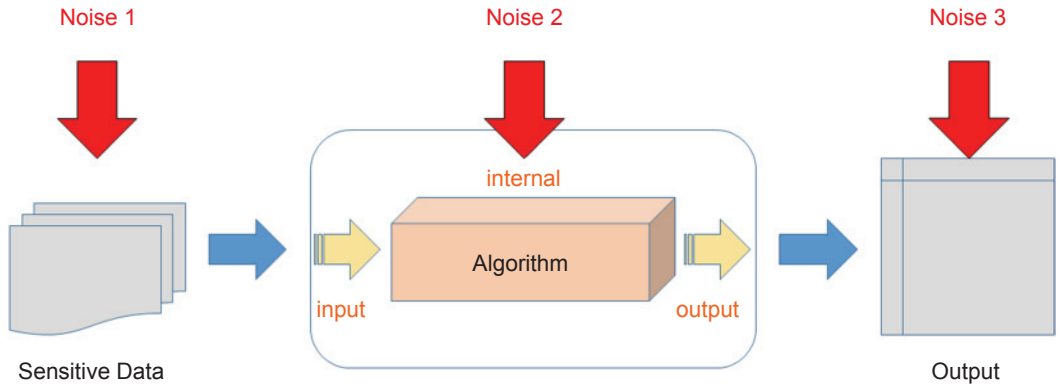

圖 11. 常見需隱私保護情境：資料蒐集階段、演算法本身、系統輸出部分。

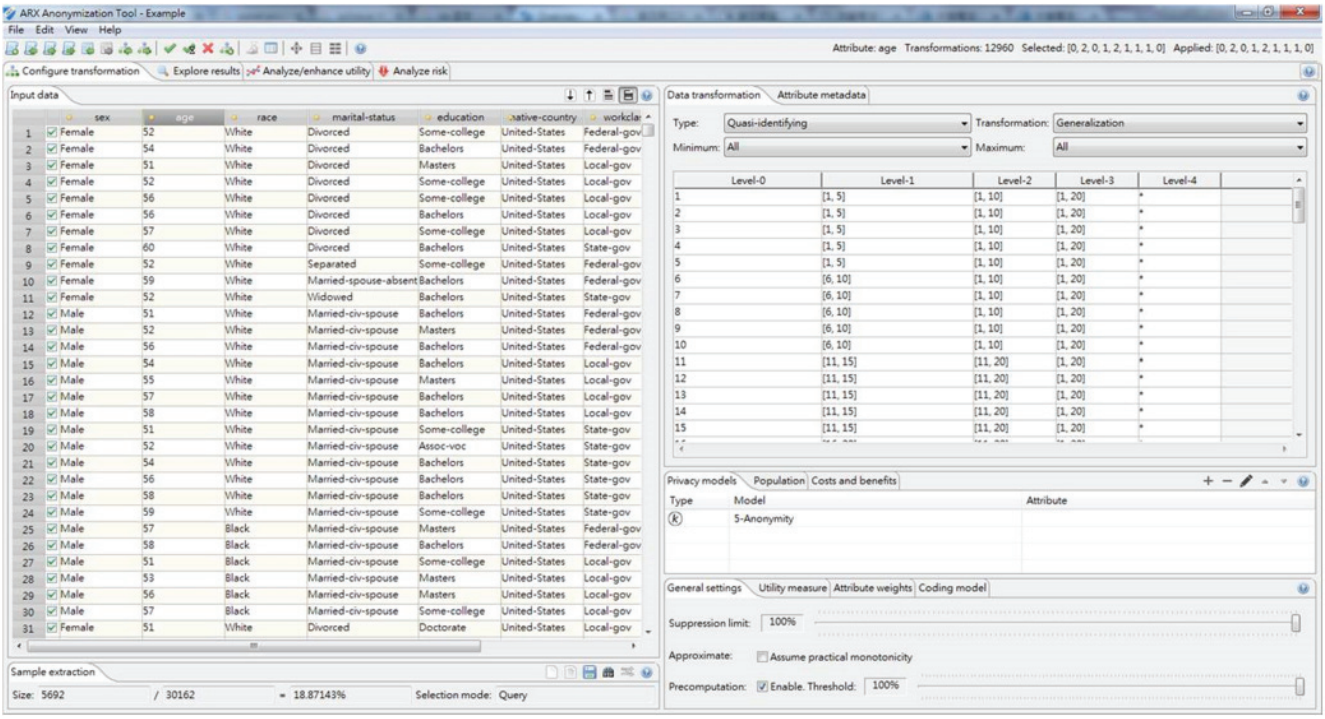

圖 12. 德國慕尼黑大學開放原始碼軟體 ARX 介面。

(K-anonymity) 或差分隱私 (differential privacy) 可從一個原始資料集產出一個擾亂後的資料集結果給別人使用。

- [加雜訊位置]
  資料收集階段 (Noise 1)。先加亂數到輸入資料，然後才開始跑演算法。
- [優缺點]
  優點是自由度大、應用範圍廣；缺點是有時會被少部分專家質疑隱私保護安全性。
- [常用工具]
  德國著名的慕尼黑工業大學曾發展一套名為 "ARX" 的開放原始碼工具 (open source library) [8, 9]，為近年來此領域使用最廣的工具之一，其採用 Apache 2.0 開放原始碼標準。其被廣泛使用的原因除了提供開放原始碼的軟體介面 (API) 供程式開發者呼叫，同時提供強大的圖形化操作介面 (GUI) 供操作，如下圖 12。目前此工具除了支援 K-anonymity, 此 library 也支援比較新的 l-diversity, t-closeness, δ-presence, differential privacy 等資料隱私保護標準。只是隱私保護技術若真的要從學術或實驗室環境應用到商業上，其實需要依照細部使用環境做不同程度的客製化，例如如果要應用此工具在自己的使用環境來實現差分隱私 (differential privacy) 很可能會發現仍有許多難題待克服。
- [備註]
  以差分隱私領域來說，非交互式資料釋出 (non-interactive publishing) 這類型方法和工具比較少。因為一開始差分隱私的定義是以交互式為主來發展的 (執行一個查詢 query，返回弄亂後的結果，讓使用者無法區分是否存在特定一筆私人資料)，而且當初就是有一些學者認為資料釋出 (data publishing) 這件事 (之前 K 匿名方法著重的目標) 永遠不會安全，所以才會發展有交互式的差分隱私理論。但是，後來外界慢慢發現資料釋出 (data publishing) 在一些真實應用上仍有其需要，而且差分隱私理論如果只適用於交互式環境非常不實用 (大多數真實世界案例都無法應用)，所以才又發展符合差分隱私精神的非交互式資料釋出 (non-interactive publishing) 以及後面說的用戶端隱私 (local privacy) 和演算法修改 (algorithm modification) 等方法。

2. 交互式查詢－回答系統

• [使用情境]

交互式查詢：回答系統 (interactive query-answer) 的使用情境是假設在使用者和資料庫軟體中間存在一個可信第三方軟體或服務 (trusted third party)，使用者原本要查詢資料庫的行為改成必須要透過此第三方軟體才能查詢資料庫內容，而對此第三方軟體下達查詢指令的方法類似目前常用的 SQL 資料庫指令，接著第三方軟體將指令翻譯成真正相對應但是包含去除隱私部分的 SQL 指令給資料庫軟體，並將資料庫軟體回傳的結果再做一次去除隱私內容的作業，最後將此去除隱私內容的結果回傳給使用者，如下圖 13。

• [加雜訊位置]

輸出結果 (Noise 3)。先跑演算法，然後才加亂數到演算法的輸出結果。

• [優缺點]

優點是安全 (其實這是差分隱私 differential privacy 最原始的構想)；缺點是許多真實應用並不允許這個情境 (這也是一開始差分隱私 differential privacy 被許多人認為不符實際需求的原因)。

• [對應的工具]

美國的賓州大學有做出一個相關的開放原始碼軟體 "Fuzz" [10, 11]，此工具的作業系統環境限制為Linux，使用此工具可實作前述可信第三方軟體，使用上會限制所有可用查詢指令 (query) 為其提供的安全查詢指令，若程式開發者要客製化需要自己重寫所有可能用到的安全查詢指令函數 (包括從最基本的 SQL 指令到較高階的 K-means 等)。

3. 用戶端隱私

• [使用情境]

用戶端隱私 (local privacy) 在所有可能的隱私保護情境中這可能是目前存在最高商用化和實用化程度解決方案產品的情境領域，包括 Google 和 Apple 公司已採用此法在他們的 Chrome 瀏覽器和 iOS 作業系統，並對外廣為宣傳。這種使用情境基本上是用在「蒐集」用戶意見調查資料 (類似「問卷調查」這種情境才可以)，例如瀏覽器或作業系統要傳送用戶端電腦的匿名化當機報告給軟體開發商。整體流程可想成伺服器端 (server) 問用戶端 (client) 或使用者一個問題 (例如是否瀏覽過某色情網站的)，用戶端將弄亂自己的回答資料 (例如匿名化處理後的資料)，然後上傳給伺服器端。在此使用情境中不需存在所謂的可信第三方軟體或服務 (trusted third party)。

• [加雜訊位置]

資料收集階段 (Noise 1)。特別是針對「蒐集」使用者資料這個動作過程。

• [優缺點]

優點是簡單、已被各大科技公司採用 (已被用在 Google Chrome 和 Apple iOS)；缺點是應用很受限 (蒐集用戶意見)，其他應用並不適合。

• [對應的工具]

Google 和 Apple 是使用前面提到的差分隱私 (differential privacy) 技術在匿名化當機報告的方法，Google 並公開其原始碼 (source code)，名稱叫做 "PAPPOR" [12, 13]。以下簡介 Google 發表的 RAPPOR 軟體方法，Apple 雖沒公布方法細節，但一般相信和 Google 的方法原理類似。目前 RAPPOR 軟體的作業系統環境限

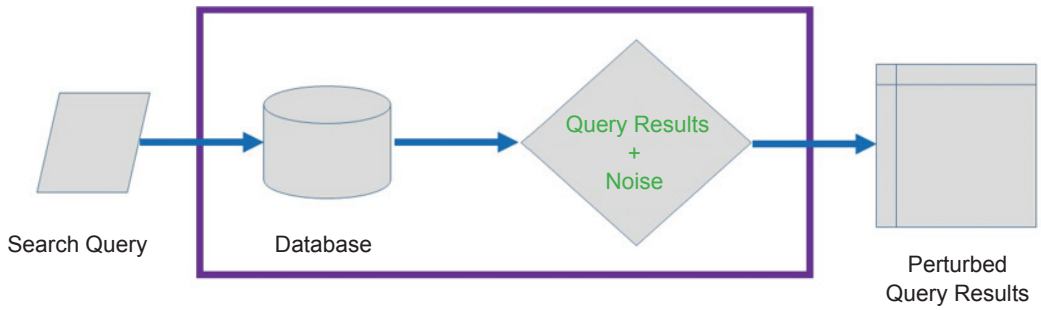

圖13. 使用者須透過可信任第三方軟體來查詢資料庫並得到去除隱私內容的查詢結果。

制為 Linux。RAPPOR 的運作原理是用隨機化回答技術 (randomized response) 來實作差分隱私 (differential privacy) 的標準，一個簡單的例子為：Google 伺服器問用戶端瀏覽器 (以下以「使用者」表示) 一個問題「是否瀏覽過色情網站」，如果使用者有瀏覽過則回 Yes，如果沒有則回 No；但除此之外，使用者手上還有一個硬幣 (或公正的亂數產生器)，使用者在回答 Server 問題前先擲一次硬幣，如果反面朝上則永遠答 Yes (混淆用)，正面朝上則回答真實答案。在此情況下，會有一半的使用者因為擲硬幣的關係直接回答 Yes，另外一半的使用者則回答真實答案，所以以統計學來說，將回答 "No" 的人數 " 乘以 2" 即可得到真實答案是 No 的人數。

在歷史發展上，其實在 1965 年 randomized response 理論就已被提出 (一開始 1965 年 S. L. Warner 等人初次提出[14]，後來 1969 年 B. G. Greenberg 等有再修改[15]), 並且被大量用在統計領域 (保護隱私的統計方法)，後來 Google 發現這個方法實作恰可以符合差分隱私 (differential privacy)，所以就重新包裝應用在他們某一版本的 Chrome 瀏覽器的當機報告統計。過兩年後 Apple 也應用在他們的 iOS，目前看來也是用在當機報告統計。

4. 演算法修改

- [使用情境]
  演算法修改 (algorithm modification) 的使用情境是直接修改演算法本身，然後資料擁有者以此新 data mining 演算法做運算，完成一次具隱私保護性的資料分析任務
- [加雜訊位置]
  演算法本身 (Noise 2)。例如在目標函數 (objective function) 加入雜訊，此舉可改變原本最佳化問題邊界值 (optimization surface)，如下式 2，使得部分原本符合的資料變成不符合 (或者相反)，來達成隱私保護如差分隱私的效果。
- [優缺點]
  優點是可直接針對要做的資料探勘 (data mining) 演算法內部做修改；缺點是只能適用於該資料探勘演算法類型的問題。

$$\underset{\omega}{\operatorname{argmin}}\left\{\frac{1}{\mathrm{n}}\sum_{i=1}^{n}L(y_i\omega^{\mathrm{T}}x_i)+\frac{1}{2}\lambda\|\omega\|^2+noise\right. \qquad (2)$$

四、結論

由以上提到的各種隱私保護機制及常見的使用情境內容，我們可以知道一個人工智慧特別是大數據分析系統有各個層面的隱私保護議題和相關的技術工具，若要妥善保護自己的大數據系統，首先必須深入了解自己的系統需求和平台特性，找出最需要注意隱私資料保護的環節，使用相對應的技術和工具，在不影響系統整體運作的前提下，達成保護用戶個人資料隱私的目標。

參考文獻

1. S. R. M. Oliveira and O. R. Zaiane, *Journal of Information and Data Management*, **1** (1), 37 (2010).
2. L. Sweeney, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, **10** (5), 571 (2002).
3. Cynthia Dwork, "Differential privacy", *Automata, languages and programming*, Venice: Springer, (1) 2006.
4. A. C. Yao, "Protocols for secure computations", *23rd Annual Symposium on Foundations of Computer Science*, 160 (1982).
5. P. S. Wang, F. Lai, H.-C. Hsiao, and J-L. Wu, *IEEE Access*, **4**, 2244 (2016).
6. R. L. Rivest, L. Adleman, and M. L. Dertouzos, *Foundations of Secure Computation*, **4** (11), 169 (1978).
7. C. Gentry, "Fully homomorphic encryption using ideal lattices", *STOC '09 Proceedings of the forty-first annual ACM symposium on Theory of computing*, 169 (2009).
8. N. Li, W. H. Qardaji, and D. Su, "Provably private data anonymization: Or, k-anonymity meets differential privacy", , *Arxiv*, 2011.
9. Please refer to the web site: http://arx.deidentifier.org/downloads/
10. A. Haeberlen, B. C. Pierce, and A. Narayan, "Differential Privacy Under Fire", *USENIX Security Symposium,* 33 (2011).
11. Please refer to the web site: http://privacy.cis.upenn.edu/software.html
12. Ú. Erlingsson, P. Vasyl, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," *Proceedings of the ACM SIGSAC conference on computer and*

communications security, 1054 (2014).
13. Please refer to the web site: https://github.com/google/rappor
14. S. L. Warner, *Journal of American Statistical Association*, **60** (309), 63 (1965).
15. B. Greenberg, A. Abul-Ela, W. Simmons, and D. Horvitz, *Journal of the American Statistical Association*, **64** (326), 520 (1969)

王紹睿先生為國立臺灣大學資訊工程研究所博士，現為中華電信研究院資安所研究員，國立政治大學資訊科學系兼任助理教授。

Peter Shaojui Wang received his Ph.D. in computer science from National Taiwan University. He is currently a research fellow at Chunghwa Telecom Laboratories and an adjunct assistant professor in the Department of Computer Science at National Chengchi University.