國立中興大學資訊科學與工程學系

碩士學位論文

基於 k 鄰近填補法在不完整資料集中找尋近似天際線

Finding Approximate Skyline Set by k-NN Based Imputation under Incomplete Data Set

指導教授： 賈坤芳 Kuen-Fang Jea

研 究 生： 凌政楠 Cheng-Nan Ling

中華民國一百零九年八月

國立中興大學 資訊科學與工程學系

碩士學位論文

題目：

基於 k 鄰近填補法在不完整資料集中找尋近似天際線

Finding Approximate Skyline Set by k-NN Based Imputation under Incomplete Data Set

姓名：凌政楠 學號：7103056065

經 口 試 通 過 特 此 證 明

論文指導教授 賈坤芳

論文考試委員 賈坤芳 王健甫

中華民國 109 年 07 月 27 日

摘要

現今大數據資料分析有一類是考量使用者偏好的相關應用，而天際線查詢演算法是最常被使用於此應用的技術之一。良好的天際線查詢演算法仰賴於完整的輸入資料集，因此解決輸入資料集中因缺失資料而造成資料不完整就成為一個關鍵議題。本研究提出一個基於 k 鄰近點填補缺失資料的方法，儘可能地找到可參考的鄰近點以對缺失值填補新值，當鄰近點不足或是在缺失率高的情況下，則使用採樣法以參考該維度其他無缺失值的鄰近點作為填補新值的依據。本研究以與原天際線的相似程度作為評測我們的方法與原始 k 鄰近點填補法的填補效果比較標準。實驗結果顯示，本研究方法在低缺失率時與原始 k 鄰近點填補法的填補效果相近；當缺失率介於 20%到 70%間，其填補效果較原始 k 鄰近點填補法好 30%至 50%；即使在缺失率高達 80%以上時，與原天際線相似度也高於原始 k 鄰近點填補法 3 到 6 倍。針對解決缺失資料集完整性的議題，本方法面對不同缺失率均具有良好的填補效果。

關鍵字：天際線查詢演算法，缺失資料，k 鄰近點填補法，採樣法

Abstract

In big data analysis, the skyline query algorithm is one of the most commonly used techniques to find optimal decisions satisfying user's preference. A good algorithm for skyline queries relies on the completeness of input data set. Solving the missing data problem, which results in data incompleteness, is however a critical issue. A new imputation method is proposed in this study, which is based on the concept of k-nearest neighbor imputation and consideration of different missing situations simultaneously. The proposed method finds out the nearest neighbors to impute missing data as much as possible. When the available neighbors are too insufficient to be referenced, or at a high rate of data missing, a sampling technique is used to select from the neighbors without missing data. To compare with the original k-nearest neighbor imputation, we adopt the closeness of the skyline set calculated from the imputed data to the original skyline set as the metric of measuring imputation quality. The experiments show that the proposed method has an approximate result to the original k-nearest neighbor imputation at a low missing rate. Furthermore, it outperforms the original k-nearest neighbor imputation from 30% to 50% at the missing rate between 20% and 70%. Even if the missing rate is higher than 80%, the imputation quality of the proposed method can also outperform 3 to 6 times than that of the original k-nearest neighbor imputation. Finally, under any kind of missing situations, the proposed method shows at least 50% approximation of the original skyline set. In sum, the proposed method is effective in solving the missing data problem for skyline query algorithms.

Keywords: skyline query algorithm, missing data, k-nearest neighbor imputation, sampling

目次

表目次

圖目次

第 1 章 簡介

在一個具有 n 個 d 維度資料點的資料集中，若其中某一資料點 p 在所有維度依據某種指標都比另一資料點 q 好時，我們稱該資料點 p 支配資料點 q[12]。在一個資料集中，所有不被任何其他資料點支配的點所形成的集合[4]，被稱為天際線(skyline set)。從資料集中找出天際線的演算法就稱為天際線查詢演算法(skyline query algorithm)。

在現今的大數據資料分析中，天際線查詢演算法在最佳化問題範疇中最常被廣泛地應用在路徑規劃、策略選擇、使用者偏好、多條件排程、多偏好分析與多準則決策等問題上，其中最典型為 Borzsony 於 2001 年的論文內的飯店例子[2]。在生活當中購買房屋時，欲找到的房屋，價格愈低愈好且房屋坪數越大越好。但現實上同時滿足上述兩個條件的房屋並不多，原因是通常坪數大的房屋價格也不低。藉由天際線查詢演算法計算後，最終結果不管是在價格上或是在房屋坪數上都能符合購屋者的期待。

目前天際線查詢演算法中[15], [18]，以 block-nested-loops(BNL)為例，每一個資料點必須與其他資料點逐一比較每一個維度(屬性)值的大小，在確定各點之間的支配關係後，方能決定哪些資料點可被納入天際線。任何天際線查詢演算法都有共同的假設：輸入資料集不能有缺失值的存在。若在確定支配關係的過程中，某資料點在某個或某些維度具有缺失值，造成該維度的值無法被比較進而導致無法確定該點與其他點的支配關係，這種情形將使得天際線查詢演算法無法執行。然而在現實生活上中，蒐集到資料難免會面臨到資料不齊全的狀況[11], [26]，例如在蒐集過程中不慎或某些因素致使資料遺失[5]，導致蒐集到的資料不完整。

為了解決天際線查詢演算法無法運用於不完整資料集的問題[27]，本論文針對不完整資料集提出新的填補缺失值技術，使所有缺失值都有可參考的新值，形成一個新的完整資料集，讓天際線查詢演算法可以順利執行。在過去填補缺失值填補研究中， k 鄰近點填補法是最常見且簡單的方法。該填補法針對具有缺失值(假設在維度 i)的資料點尋找其 k 個鄰近資料點，以這 k 個點在維度 i 的平均值作為填補的新值。k 鄰近點填補法具有兩個缺點：其一是在填補過程中計算含有缺失值的資料點與其他點的歐氏距離時，尋找到的距離最短的 k 鄰近點可能不是真正的 k 鄰近點，並且不同鄰近點均給予相同的權重值也不一定合理；其二是若鄰近點不足 k 個，原始 k 鄰近點填補法並不會積極地尋找剩下可參考的鄰近點，使得填補效果趨近於單一數值的填補法，這種情況會隨著缺失值比例愈大而愈趨嚴重。過去研究對 k 鄰近點填補法並沒有針對不同缺失值(佔原資料集的)比例之填補效果有太多深入的分析。

為了解決上述 k 鄰近點填補法所面臨的問題，本研究提出了一個 sk-NN imputation 演算法賦予不同鄰近點合理權重值的方法以及挑選鄰近點的採樣機制。為彰顯愈鄰近的點對填補缺失值有較高的影響力，我們改以鄰近點與含有缺

失值資料點距離的倒數作為填補缺失值的新權重值，藉此改善 k 鄰近點填補法中對不同鄰近點皆賦予相同權重值之不合理性。我們並且透過適當的採樣機制儘可能地挑選出足夠的鄰近點以計算填補值，改善了 k 鄰近點填補法面臨鄰近點不足時退化為以單一數值填補的問題。

本研究進行兩個模擬實驗。首先觀察原始 k 鄰近點填補法在不同 k 值與缺失值比例下對原天際線所造成的乖離狀況，然後比較本方法與各填補法在填補缺失資料後所產生的天際線與原天際線的相似度。實驗結果顯示，在 k 值足夠大時，本研究方法在缺失值比例低時所填補的效果幾乎與原始 k 鄰近填補法相同甚至更好；在缺失值比例較高時，本研究方法所得到天際線與原天際線相似度至少維持 50%，表示超過一半的天際線經過本研究的填補方法被找回來。

本研究的主要貢獻在於改善原始 k 鄰近點填補法對鄰近點不合理的權重分配，並且提出鄰近點的採樣方法以解決無法找到足夠可參考的鄰近點進行填補的問題。這兩點改進使得針對高缺失值比例的資料集執行天際線查詢演算法時，仍然可以找到與原天際線相似的近似天際線。

本論文後續的章節如下: 第二章敘述相關研究，第三章描述問題與方法、第四章顯示實驗結果與分析，以及第五章是結論與未來研究方向。

第 2 章 相關研究

本論文相關研究有兩個面向：資料缺失類型(types of missing value)與缺失值處理方法(missing value handling)。

2.1 資料缺失類型

資料集中常見的缺失值類型分別有隨機缺失類型(missing at random，MAR)、完全隨機缺失類型(missing completely at random，MCAR)以及完全非隨機缺失類型(missing not at random，MNAR)三種[8], [27]。

首先，若缺失資料發生的機率與資料集內的其他變數有關，但與缺失值本身的數值大小無關，則稱此一類型的缺失值為隨機缺失(MAR)類型。例如問卷中若有包含心情沮喪的議題則較容易缺少男性族群的資料，這種情形會讓人誤以為男性族群與該議題看似乎有相關性。但這可能只是因為男性比較不願意填寫此種議題的問卷，事實上是與男性族群毫無關係，此類型的缺失值就被歸類為隨機缺失值。若缺失資料類型屬於 MAR 類型時，則對該資料集內所含有的缺失值可以被納入考量或者經過缺失值的相關處理方法解決缺失值的問題。可選擇的方法包括將具有缺失值的資料紀錄刪除或是針對該缺失值賦予一個合理可供參考新的值，其他原本無缺失的數值仍可被拿來作為後續資料分析用途。

完全隨機缺失(MCAR)類型是指出現缺失值的變數其缺失機率與資料集內其它變數都沒有任何相關性，例如問卷上的回答錯誤、忘記填值、資料遺失等皆屬此一類型。若缺失資料類型屬於 MCAR，可以在無缺失值的資料筆數足夠多時刪除部分具有缺失值的資料。

若缺失情形不屬於其他兩者，則被歸屬於完全非隨機缺失(MNAR)類型。此類型的缺失值與其他維度具有相關性，屬於此類型的缺失資料會表現出某一種資料特性，故此缺失類型完全不可忽略，也不宜用任何方式異動缺失值。例如薪資調查問卷時，高薪資與低薪資族群因為不想透漏實際薪資而拒絕填寫，造成資料缺失情況，進而導致影響資料集真實性。這類缺失類型完全不可以忽略含有缺失值的任何一筆資料，且不建議擅自刪除那些含有缺失值的資料點，以防止對資料集的特徵做出作錯誤判斷與誤導。

2.2 缺失值處理方法

一般在做資料處理的時候，假如資料集中某些資料欄位的值不存在，這種情況被稱為資料缺失，這些沒有資料的欄位具有缺失值。舉例來說，表 2.1 為一個含有若干個缺失值的資料集，並以橫線-表示具有缺失值的欄位。

表 2.1 含有缺失值的資料集

id	age	gender	height(cm)	weight(kg)
1	-	Female	-	53
2	18	Male	180	75
3	-	-	174	67
4	21	Female	158	47

面臨資料缺失時，為了讓整體資料集可以做後續的統計、分析及決策等工作，我們必須先對缺失值做一些處理，無論是忽視不管、移除含有缺失值的資料或者賦予替代的值等，都是處理缺失值的可能策略。缺失值處理方法可分為三種，分別是丟棄法(dropout)[10]、填補法(imputation)[3], [6], [14]與 k 鄰近點填補法，其中 k 鄰近點填補法[28]是效果較好的方法，以下分別說明之。

2.2.1 丟棄法

丟棄法[10]是根據不同的目的以及資料缺失類型，將缺失值從原資料集中移除，使得資料集中不包含任何的缺失值。丟棄法的優點是透過簡單移除缺失值的方式達到保有完整資料集。然而，丟棄法的缺點是可能會因為過度刪除而出現喪失資訊的問題(將於下面詳細說明)。

丟棄法主要分成兩類，一類是將含有缺失值的整筆資料刪除，即刪除資料列，另一類是將含有缺失值的資料屬性刪除，也就是刪除缺失值所在的維度，稱為刪除維度。以下分別介紹刪除資料列與刪除維度這兩類方法。

刪除資料列的方法就是將含有缺失值的那筆資料從資料集內刪除，也稱為單筆去除法(listwise deletion)[17]。若缺失值的缺失資料類型為完全隨機缺失類型(MCAR)且缺失值比例不大時，比較適合採用此法來移除缺失值。其理由是在上述的情況下刪除缺失值後的資料集與刪除前的資料集相比，兩者的資料分布不會相差太多；換句話說，在 MCAR 的情況下刪除資料列並不會造成資料集的偏差(bias)。然而，刪除資料列也有其缺陷，就是當缺失值比例較高的時候，資料集內可用的完整資料不足。舉例來說，表 2.2 為表 2.1 刪除資料列後的結果，從原先 4 筆資料減為只剩 2 筆資料，在表 2.2 中的兩筆資料皆為完整資料。

表 2.2 刪除資料列

id	age	gender	height(cm)	weight(kg)
2	18	Male	180	75
4	21	Female	158	47

刪除維度法就是將含有缺失值的維度從資料集刪除。若資料集內缺失值的類型為不完全隨機缺失類型(MNAR)，且在某個維度 d 中資料缺失的情況非常嚴重時，適合使用刪除維度法。刪除維度法在資料處理的角度而言就相當於將原本的資料集降維，優點是可以減少演算法查詢的時間。其缺點是可能會刪除太多的維度，若代表性很高的維度被刪除，會導致剩下的維度無法充分表現資料集的特徵。例如資料集內具有分類標籤(label)性質集中於某一維度時，刪除該維度將使對應到的分類資料大量失去特徵訊息，可能造成刪除後的資料做分類預測時不準確。舉例來說，表 2.3 為表 2.1 刪除含有缺失值的三個維度後(age、gender 與 height)的結果。

表 2.3 刪除維度

id	weight(kg)
1	53
2	75
3	67
4	47

2.2.2 填補法

填補法是尋找資料集內不含缺失值的資料點，參考這些資料點的值賦予缺失值一個合理值。填補法與丟棄法相同之處是經過填補後仍然可以保持資料集的完整性；不同之處是經過填補法填補後，資料集的資料個數、維度數都與填補前一致，不會如同丟棄法有喪失資料個數或維度的問題。

根據填補後產生一組或是多組資料集，填補法可分為單一填補法(single imputation)與多重填補法(multiple imputations，MI)[1], [6], [23]兩類。兩者最明顯的差別在於，單一填補法經過單次填補完後只會產生一組完整資料集，之後便以這組資料集進行後續的統計與分析；而多重填補法則可以視為以不同的單一填補法，經過多次地填補後，產生多組不同的完整資料集，再將這些多組填補結果合併後做後續的統計與分析。

填補法的優點在於不影響輸入統計模型、資料集分布以及避免喪失資料集的表現特徵，參考無缺失值的資料點來填補缺失值較有可參考性且預期會有比較好的效果。然而，填補法的缺點就是無法保證填補缺失的值是否正確，且填補缺失值後，可能會影響原本資料集的內容而造成資料的失真。

2.2.2.1 單一填補法

單一填補法[3], [14]填補的原理敘述如下。假設資料集 S 中有一個資料點 p 在維度 d 有缺失值，單一填補法會根據一套規則尋找一群資料點集合 Q (Q 包含於 S)，將 Q 中的點在維度 d 上的值經計算後得到一個合理值，並以該合理值填補資料點 p 在維度 d 上的缺失值。依照尋找 Q 的規則之不同，單一填補法又細分為固定數值填補法、平均值填補法(mean imputation or mean substitution)、熱卡填補法(hot deck imputation)[9]、冷卡填補法(cold deck imputation)[21]、迴歸填補法(regression imputation)與最鄰近填補法(nearest neighbor method)[10]，以下依序分別說明。

首先，最簡單的填補方式就是固定數值填補法。此填補法的規則是對相同維度的缺失值賦予同一個固定數值，例如在維度 d 有缺失值，蒐集所有資料點在維度 d 上的數值，計算這些值的眾數、平均值、中位數、極大值或極小值等等[14]，以此計算值填補缺失值。此方法的優點是計算量不大，填補過程最簡單，而且不受任何缺失值類型與資料分布影響。但填補固定數值的問題在於，當缺失值數量較多時，由於同一維度下的全部缺失值都被填補相同的計算值，這樣填補方式會降低資料點間的差異性，使得資料集內有太多的重複值，造成填補的效果很差。因此除非缺失值數量不多，注重資料差異性區別的資料集並不建議以固定數值填補缺失值。

平均值填補法是固定數值填補法中的一個特例，填補的方式依據資料集中所有在缺失值所在的維度 d 上的值，計算其平均值以填補缺失值。當缺失值比例變高時，此方法雖然不會改變整體資料集中的平均值，但經過填補後的維度會因為填補的值均相同，使得變異數會變小，因此平均值填補法也會影響資料點的分布。

熱卡填補法與冷卡填補法這兩種單一填補法參考的規則大致相同。熱卡填補法也稱為 last observation carried forward (LOCF)，此法記錄每一個資料點出現的頻率，參考出現頻率最高的資料點 q 之值做填補。舉例來說，資料點 p 在維度 d 上有缺失值，熱卡填補法會尋找出現頻率最高的資料點 q，以 q 在維度 d 的值來填補缺失值。而冷卡填補法與熱卡填補法作法完全相反，尋找出現頻率最低的資料點填補缺失值。

迴歸填補法是針對連續型資料有缺失值的填補方法。此類方法假設連續型資料中的缺失值與其他維度有一定的相關性且缺失值類型為 MAR，此時缺失值可以經由迴歸計算該相關維度之值後填補缺失值。此種填補法會讓填補值之間的變異數降低，導致填補後的值都會落在迴歸線上，使得資料分布太過規律而缺乏資料多樣性，所以還需要再加上一個隨機誤差作為最終缺失值的填補值。迴歸填補法的缺點是當缺失值比例很高時容易影響資料集的分布，因此這類方法只適合在

缺失值比例較小時使用。

最後一種單一填補法為最鄰近填補法，由 Fix 與 Hodges 在 1951 提出[22]。若資料點 p 在維度 d 上有缺失值，透過計算其他點與 p 的距離，依距離值由小到大排序找出與 p 距離值最小的資料點 q，稱 q 為最鄰近點。以資料點 q 在維度 d 的值填補 p 的在維度 d 的缺失值。如果最鄰近點不只一個，則將最鄰近的那些點在維度 d 的值取平均值來填補缺失值。在找尋鄰近點的過程中，需要計算任意兩筆資料點的距離(distance)，若距離越小則可以推測兩資料點的值也會愈接近。根據不同的應用而採用的計算距離方式也有所差異。另一類 k 鄰近填補法屬於最鄰近填補法其中的一種，本論文是基於 k 鄰近點填補法上做改善，因此另立一小節(於 2.2.2.3 節)詳細說明 k 鄰近點填補法。

2.2.2.2 多重填補法

多重填補法首次在 1978 年被 Rubin 提出來[20]。多重填補法主要有三個過程：第一步，對每一個缺失值被重複填補 m 次以產生 m 個完整資料集。第二步，將 m 個完整資料集都使用針對完整資料集進行統計分析。第三步，將 m 個來自各完整資料集的結果，以評分函數選擇一個最合理的值或合併所有值，產生最終的結果作為填補缺失值。根據 Rubin 的建議是，當 m 過大時並不會有更好的填補效果，因此建議 m 的範圍落在 3 到 10 之間。

多重填補法的優點為挑選填補值具有多樣性。由於對相同缺失值填補很多次，產生多組的填補候選值，可以從中挑選其一，有比較多種的選擇，因此採隨機抽取其中一種作為填補值會比單一填補法更具多樣性，避免每次計算後的填補值都是同一填補值。

然而多重填補法仰賴於資料集上模擬分布模型[19]。遇到缺失資料時，此法根據模型的分布給予一群可能為該缺失值的候選解集合，並在陸續填補過程中調整資料集的分布、變異數以及信賴區間等，因此在填補過程中會需要極大量的計算且在實務上不容易進行模擬。由於生成多重填補值的過程與所花費時間比起單一填補法複雜許多，因此多重填補法不常被採用。

2.2.2.3 k 鄰近點填補法

k 鄰近點填補法(k-nearest neighbor imputation，k-NN imputation)[8], [16]是很實用的一種填補缺失值的方法。此方法從資料集中透過距離計算，找出 k 個與缺失資料最近的資料點，這 k 個點稱為鄰近點，取這些鄰近點的值並且計算其平均值，最後用此平均值來填補缺失值。常用距離的度量方法為歐氏距離(Euclidean distance)。

k 鄰近點填補法的填補過程詳述如下。假設資料集中存在某資料點 p 在任意

維度 d 上含有缺失值。在填補值之前，首先計算所有資料點與點 p 的距離值，接著將這些距離值由小到大排序後，蒐集那些距離 p 最近的 k 個點，這 k 個點也被稱為資料點 p 的 k 個鄰近點。接著取 k 個鄰近點同樣在維度 d 上的值，將這 k 個值相加之後求其算術平均值作為最終填補資料點 p 在維度 d 上的缺失值。

k 鄰近點填補法的優點如下。由於上述 k 鄰近點填補法在填補過程中並未牽涉到維度大小限制、資料類型，在不考慮計算量下，k 鄰近點填補法可以輕易地推廣至任意維度且幾乎適用於各種資料類型[27]，如連續型資料(continuous data)、離散型資料(discrete data)、有序型資料(ordinal)，甚至是分類型資料(categorical data)[28]。k 鄰近點填補法是最為常見且被認為效果比較好的補值法，且比起填補用固定數值，例如平均數、中位數、極值、眾數等的填補方法，k 鄰近點填補法更準確許多，出原因是 k 鄰近點填補法會同時參照其他與該缺失值相鄰點去預測一個更合理的填補值。

然而，k 鄰近點填補法也存在某些缺點。其中一個缺點是 k 鄰近點填補法參考鄰近點的值時，在無資料預處理的前提下，如果挑選到的鄰近點含有偏差值，可能無法對缺失值填補較合理的值。另一個缺點是，k 鄰近點填補法的填補過程會計算所有資料點之間的距離，距離的計算量會受到維度多寡與資料點的個數影響，其計算量在多維度資料集且含有大量的資料點時，所耗費的時間更是嚴重。

第 3 章 問題與方法

本章 3.1 節說明研究動機，3.2 節敘述問題定義，3.3 節提出問題分析，3.4 節提出 sk-NN imputation 演算法，最後 3.5 節闡述在不完整資料集中如何以原天際線評斷各填補法的表現優劣。

3.1 研究動機

尋找天際線時需要針對每二筆資料比對所有維度的值，也就是說每一筆資料的每一個維度都必須有值存在方能比對，因此資料集的完整性在天際線查詢演算法中就成為了必要條件。由於現實生活中有許多不可抗拒之因素使得取得資料集的過程難免會遇到欄位裡的值無法完備，期待蒐集到的資料每一維度都沒有缺失值是不切實際的。

為了保證資料集的完整性，解決缺失值最直覺採用的策略為丟棄法。丟棄法有兩種刪除資料的方式，分別為刪除資料列以及刪除維度。如果不完整資料集中缺失值的比例越高，刪除缺失值所在的資料列會讓所剩資料數量不足，而刪除維度則會喪失原資料集所表現的特徵。反之，採取填補法不僅不會喪失原資料集特徵，還可以保證資料點個數與原始資料集一致。在考量資料集的完整性與天際線查詢演算法的適用性，填補法會比丟棄法來得更適合。

針對不完整資料集如何執行天際線查詢演算法的問題上，本研究採取填補法填補缺失值。從填補後的效果來看，k 鄰近點填補法是填補法當中表現比較好的[24]。k 鄰近點填補法首先針對維度 d 有缺失值的資料點找出其 k 個鄰近資料點，取得這些鄰近點的維度 d 之值，再計算這些 d 維度值的算術平均作為新填補值，這使得被填補值相較於以一般填補固定數值的方法更具有參考性。然而 k 鄰近點填補法也有其缺點，其一是當計算含有缺失值之資料點與其他資料點的歐氏距離(必須先忽視含有缺失值之維度)會不準確而導致找錯鄰近點，其二是 k 鄰近點填補法遇到可參考的鄰近點不足時會退化為填補固定數值[7]。

對一個含有缺失值的資料點 p 而言，尋找鄰近點的過程中，k 鄰近點填補法先計算所有資料點與 p 的距離，再根據距離值由小到大排序。排序的意義代表著各資料點分別對 p 的影響程度，但是 k 鄰近點填補法卻將 p 的所有 k 個鄰近點視為相同的權重值[15]。此做法與先前排序的意義，期待找出最有參考價值的鄰近點以填補該缺失值的概念相違背且不合理[27]，也就是未考慮鄰近程度對填補值的影響力。因此本研究針 k 鄰近點填補法的缺點以及未考慮鄰近程度的影響力問題分別提出鄰近差別權重分配與新的選擇鄰近點機制以改善 k 鄰近點填補法。

3.2 問題定義

本研究要解決的問題定義如下: 在不完整資料集中，如何改善原始 k 鄰近點填補法填補缺失值，使填補後的完整資料集具有最近似的天際線?

本研究假設不完整資料集中，缺失值的缺失類型為 2.1 節中提到的完全隨機缺失類型(MCAR)，即含有缺失值的屬性與其他欄位屬性無相關性。為驗證填補效果，我們將填補後的完整資料集計算近似天際線，並與原先無缺失資料集的天際線比較其差異，以此差異作為衡量近似天際線的相似程度。若相似程度愈高，代表該填補法的填補效果愈好。

3.3 問題分析

所有計算距離公式[8], [16]中，最普遍常見的歐氏距離計算方法[13]是採資料集中兩兩資料點相對應維度的差值平方和再取平方根。若是至少一個維度具有缺失值，則在計算歐氏距離時並不會採計具有缺失值的維度，此計算方式是最廣為主流的算法[24]。由此計算方式可看出一個潛在的問題 : 具有缺失值的兩資料點其距離計算所得之值可能會誤導此二資料點之間的實際距離，今舉例說明之。

如圖 3.1 所示，A、B、C 三個二維資料點，其座標分別為(1, 1)、(2, 8)、(3, 3)，。在沒有任何缺失值情況下，按照傳統歐氏距離的計算: AB 距離 $\overline{AB}$ 應該為$\sqrt{(1-2)^2+(1-8)^2}=\sqrt{50}$ 且 $\overline{AC}$ 應該為 $\sqrt{(1-3)^2+(1-3)^2}=\sqrt{8}$ ，故 $\overline{AB} > \overline{AC}$。但若將點 A 中的 y 座標設為缺失值時，則含有缺失值後的歐氏距離會將有缺失值的維度捨棄而不列入計算，新的 AB 距離 $\overline{AB}'$ 為 $\sqrt{(1-2)^2}=1$，而新的 AC 距離 $\overline{AC}'$ 則為$\sqrt{(1-3)^2}=2$ ，使得 $\overline{AB}' < \overline{AC}'$。此時新的距離會讓原本為了避免誤算不納入缺失值的機制反而錯估了距離的實際值，間接導致了大小順序上誤判的結果，這就是 k 鄰近點填補法在有缺失情況下只單依靠距離大小決定鄰近參考點所可能會陷入的誤區，最終與其原目的相違背。

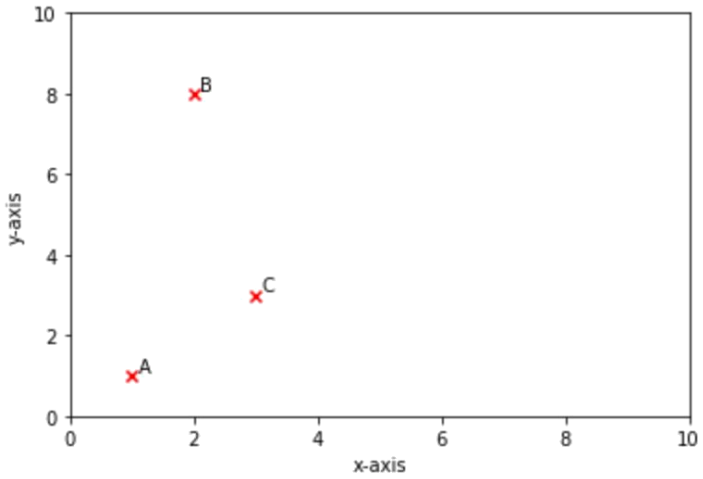

圖 3.1 NaN-Euclidean distance

k 鄰近點填補法的另一個問題在於，當缺失值愈高時， k 值很大意味著鄰近點仍存有非缺失值的機會並不大，而此時 k 鄰近點填補法在無法找到滿足 k 個鄰近點情況下，選擇不從剩下的鄰近點補足並從缺，這樣的現象尤其當存在非缺失值很稀少時更為嚴峻，導致 k 鄰近點填補法會幾乎用同一數值填補回去，如此便會與只填補單一數值(同一維度的平均數、眾數、極大值、極小值)無異，填補後找尋天際線時又會因為該維度幾乎都是同一數值，更容易形成有如該維度直接被刪除一樣而產生無意義地比較結果。

鑒於以上分析，本論文提出新演算法除了在缺失值比例不高時填補效果能與原始 k 鄰近點演算法相近，並且在缺失值比例較高時也能夠改善原始 k 鄰近點填補法的不足。有別於 k 鄰近點填補法對鄰近點不足而選擇從缺不補，本研究方法在缺失值所在的維度上，尋找其他沒有缺失值的點，從這些點採樣其中 k 個點，計算它們在該維度的平均值，最後以該平均值填補原缺失值。其目的是不讓鄰近點的不足而計算不準確，導致填補值後產生的天際線與原天際線乖離太大。

3.4 sk-NN imputation 演算法

本節先說明本論文提出的演算法內會運用到的符號與其定義，如表 3.1 所示，然後提出本研究的演算法 sk-NN imputation。

已知一個不完整資料集 C 含有若干個缺失值，常數 k為可參考鄰近點的上限個數，n 與 m 分別為 C 的資料點個數與 C 的維度個數。c_i 表示 C 中第 i 筆資料點，且 v_{id} 為該資料點 c_i 於維度 d 的值。

D 為不完整資料集 C 中任兩資料點 c_i 與 c_j 之間距離的距離矩陣，且其中 d_{ij} 為任兩資料點 c_i 與 c_j 之歐氏距離。W 是一個資料點 c_i 在每一個維度 d 的值 v_{id} 相對其他任意相異資料點 c_j 的權重矩陣，ω_{ij}代表兩資料點 c_i 與 c_j

之間的權重值。t 是一個決定權重值的函數，用來決定某一缺失值的被參考維度值 v_{id} 的權重值。若 t 值為 $uniform$，則所有權重值 ω_{ij} 均被設為 1；若 t 值為 $distance$，則 ω_{ij} 會以 D 中任兩資料點 c_i 與 c_j 之間距離 d_{ij} 的倒數作為該權重值。

N 是一個由元素 N_i 有序串列所構成的集合。每一個 N_i 記錄 k 個鄰近 c_i 的資料點 p_i^h，而 p_i^h 的順序則是依照資料點 c_j 與 c_i 的距離值 d_{ij}，由小到大排序。p_i^h 表示 c_i 所有的鄰近點中，d_{ij} 不為 0 的且與 c_i 第 h 個最接近的資料點 c_j。

最後說明填補過程中使用到的符號，若輸入資料集 C 中某 v_{id} 為缺失值，則本論文填補法將會賦予該缺失值一個新值 v'_{id} 填入原缺失欄位。mask 為一個長度 k 的陣列，記錄最鄰近c_i 的 k 個鄰近點在維度d是否出現缺失值。若鄰近點p_i^h在維度d有缺失值，則 mask[h]標註為 True，否則標註為 False。最終，演算法輸出一個原始缺失值均已被填補的資料集 $\hat{C}$。

表 3.1 sk-NN imputation 演算法符號定義表

符號	說明
n	資料點個數，共 n 個資料點
m	資料點維度個數，共 m 個維度
incomplete data set C, $C = \{c_i \mid c_i = (v_{i1}, v_{i2} \ldots v_{id} \ldots, v_{im}),\ 1 \le i \le n, 1 \le d \le m\}$	不完整輸入資料集，大小為 $n * m$
$k, k \in N$	常數 k，代表填補缺失值需參考的鄰近點個數
v_{id}	資料集內第 i 筆資料在維度 d 的值
c_i	C 的第 i 筆資料
D	距離值矩陣，紀錄全部的d_{ij}
d_{ij}	資料點 c_i 與 c_j 之間的歐氏距離，記作 d_{ij} 且 $d_{ij} = d_{ji}$
W	權重值矩陣
$\omega_{ij} = \begin{cases} 1, & t: uniform \\ 1/d_{ij}, & t: distance \end{cases}$	兩資料點 c_i 與 c_j 之間距離的權重值，其中 $i \neq j$
$t = \{uniform, distance\}$	權重值決定函數 t
p_i^h	c_i 的所有鄰近點中，d_{ij} 不為 0 且第 h 個鄰近c_i的資料點c_j，h 的次序以d_{ij}的值由小到大排序所決定，h∈[1,k]
$N_i = \langle p_i^1, p_i^2, \ldots p_i^h \ldots . p_i^k \rangle$	由最鄰近 c_i 的 k 個資料點 p_i^h所構成

	的有序集合，i∈[1,n], h∈[1,k]
$N = \{N_i \mid i \in [1,n]\}$	元素 N_i所構成的集合
v'_{ij}	缺失值被填補的新值
mask[h]	長度為 k 的陣列，陣列中的值為 True 或 False。mask[h] 對應 N_i 內的第 h 個點 p_i^h，若p_i^h在維度 d 有缺失值，則 mask[h] 為 True，否則為 False。
imputed data set $\hat{C}$	缺失值被填補後的資料集

本論文所提出的 sk-NN imputation 演算法顯示於圖 3.2，其輸入參數為一不完整資料集C、常數 k、以及決定權重值的函數t。

執行過程依序為：step1 與 step2 載入不完整資料集C並初始化填補資料集$\hat{C}$。step3 先初始化距離矩陣，step3-1 與 step3-2 計算任兩資料點之間包含相對應維度有缺失值的歐氏距離。step4 初始化權重值矩陣後，根據 t 來決定計算任兩資料點之間的權重值，若 t 為 uniform 則如同 k 鄰近點填補法給予相同權重，若 t 為 distance 則給予差別權重方式為兩點之間的距離倒數，其背後意義為兩點距離愈大對彼此的影響力愈小。step5 列出每一筆資料點其所有鄰近點並儲存於 N_i 中，並且依照與該點 c_i 的歐氏距離值由小到大排序。step6 遍歷輸入資料集中所有缺失值，並分別檢視其鄰近點以填補新值，填補新值是透過執行副程式 Impute_Process()(示於圖 3.3)所完成。step7 回傳填補後的新完整資料集$\hat{C}$，結束 sk-NN imputation 演算法。

Algorithm sk-NN imputation (C, k, t) {
Input : incomplete data set C, constant k, weighting type t
Output : imputed data set $\hat{C}$
Method :
step 1. load incomplete data set C
step 2. initialize imputed data set $\hat{C}$ to C
step 3. // compute distance matrix D
  3-1. **for** each d_{ij} in distance matrix D **do**
  3-2. d_{ij} ← Euclidean distance of data samples c_i and c_j
  end for
step 4. // compute weight matrix W
  4-1. **for** each ω_{ij} in weight matrix W **do**
  4-2. $\omega_{ij} = \begin{cases} 1, t: uniform \\ 1/d_{ij}, t: distance \end{cases}$
  end for

step 5. // find all nearest neighbors of c_i, store in N_i
  5-1. create and initialize an empty N_i, with size k
  5-2. **for** each c_i in C **do**
  5-3. retrieve all d_{ij} between any pair of the c_i and c_j from D
  5-4. sort d_{ij} in ascending order
  5-5. assign c_j to N_i as the h^{th} nearest neighbor (i.e., p_i^h), according to the ascending order of d_{ij}
  5-6. **repeat**
  5-7. **if** $d_{ij} \neq 0$ **then**
  5-8. append p_i^h to N_i
  end if
  5-9. **until** all p_i^h are inserted into N_i
  end for
step 6. // search all missing values in $\hat{C}$, then impute new value back into the missing position in $\hat{C}$
  6-1. **for** each v_{id} in C **do**
  6-2. **if** v_{id} is missing **then**
  6-3. **call** procedure Impute_Process(i, d, k)
  end if
  end for
step 7. **return** $\hat{C}$ as imputed data set
}

圖 3.2 sk-NN imputation 演算法

接下來說明 Procedure Impute_Process()的填補過程，圖 3.3 為計算填補值的 pseudocode。首先 step1 初始化代表鄰近點缺失狀況的 mask array。step2 檢查每一個缺失值的鄰近點在該維度的值是否也為缺失值，若為缺失值則在 mask array 中標記 True，否則標記為 False。step3 利用 mask array 檢視所有在維度 d 有缺失值的資料點其所有鄰近點在維度 d 是否全部都是缺失值；若 mask 的值皆為 True，則判定可參考的鄰近點在相同維度也都是缺失值。此時就會觸發採樣機制去參考相同維度非缺失值的點(不一定為鄰近點，因為鄰近點已經不具參考性)。step4 將採樣點在該維度的平均值填補回原缺失位置，至此填補其中某一缺失值結束。

Procedure Impute_Process(i, d, k) {
Input : i, d and k, indicating that missing value is found in data sample c_i at dimension d, and k nearest neighbors are used for imputation.
Output : imputed value v'_{id}

Method :
step 1. create a mask array and initialize all values in mask array to False
step 2. // check all elements p_i^h in N_i
2-1. initialize r to 0
2-2. **for** h from 1 to k **do**
2-3. r ← p_i^h
2-4. **if** v_{hd} is missing **then**
2-5. mask[r] ← True
end if
2-6. **else**
2-7. mask[r] ← False
end else
end for
step 3. // retrieve values v_{hd} where mask[h] in mask array assigned to False
3-1. **if** all elements mask[h] in mask array are True **then**
3-2. reset all elements in mask array to False
3-3. sampling the rest of not missing value at column d
end if
3-4. **else**
3-5. retrieve all values v_{hd} in C where mask[h] is False
3-6. compute the mean or weighted mean v'_{id} of those retrieved v_{hd}
end else
step 4. **return** v'_{id} as imputed value
}

圖 3.3 Procedure Impute_Process()

3.5 以原天際線評斷填補法的表現優劣

執行模擬實驗時，為了觀察填補效果[25]對原天際線所造成的影響，本論文採用填補缺失值後的天際線與原天際線兩者之間的漢明距離(hamming distance)作為評斷兩者相似度之標準。漢明距離主要是用於計算兩個字串相對應的位置具有不同字符的個數，換句話說，將一個字串變換成另外一個字串所需要替換多少個字符的總數即為漢明距離。例如：兩等長二進位字串 1011101 與 1001001 由左向右第 3 與第 5 個位元相對位置值不同，故計算此字串的標準漢明距離為 2；同理，toned 與 roses 之間的漢明距離為 3，以此類推。

本論文使用漢明距離定義中須置換字符次數的觀念，因此並沒有要求兩字串必須等長之限制。本論文所採用的是集合上的漢明距離概念，換句話說，任意兩

個集合之間必須插入或刪除多少元素才能使兩集合相同。例如：已知三個集合，$P = \{a, b, c\}, Q = \{e, c, b, d\}, R = \{d, e\}$。P、Q 需要置換三個元素才能使兩集合相同，因此 P、Q 之間的在本論文上所定義的漢明距離為 3，Q、R 的漢明距離為 2，而 P、R 之間的漢明距離則為 5。使用這樣的觀念原因有二：

1. 集合內元素不具有順序性，只能檢查某元素存在與否，此性質在字串問題上即為對應位置是否具有相同值。
2. 兩集合相同的充分且必要條件為兩集合具有相同元素且相異元素個數相同，此性質對應到字串問題上則為兩字串長度必須相同。

判斷原天際線 S 與近似天際線 S'的相似程度之計算方法如下:

1. 天際線 S 與 S'中相異元素總個數為該集合 size。
2. S 與 S'集合具有相同元素個數稱為 hit count。
3. S 之中有的元素但 S' 中沒有的元素之個數以及 S' 之中有的元素但 S 中沒有的元素之個數，將其加總之和，稱為 miss count。miss count 就是本論文所定義的集合上的漢明距離。
4. hit ratio = $hit\ count/(hit\ count + miss\ count)$

今舉例說明:

original skyline set: $\{B, C, E, G, H\}$ size 為 5

estimated-1 skyline set : $\{A, C, D, E, H, F, R\}$，size 為 7

estimated-2 skyline set : $\{B, C, D, E\}$，size 為 4

estimated-1 skyline set 與 original skyline set 具有 2 個元素相同 C、H，hit count = 2，且 A、B、D、G、F、R 並沒有猜中故 miss count = 6，hit ratio 為 $2/2 + 6 = 0.2$。

estimated-2 skyline set 與 original skyline set 具有 3 個相同元素 B、C、E，hit count = 3，且 D、G、H 沒猜中故 miss count = 3，hit ratio 為 $3/3 + 3 = 0.5$。

由上述例子可知，S' size 愈大並不能保證 hit ratio 一定愈好，亦即猜得多不如猜得精準。天際線為一個不被其他點支配的資料點所構成的集合，如果經填補後所找到的天際線集合與原天際線集合之相似度愈高，則可推斷該填補法對天際線所填補效果愈好。本論文用上述相似度指標 hit ratio 來評斷各填補法填補效果之優劣。

第 4 章 實驗結果與分析

本章依序於 4.1 節說明實驗環境、平台與所使用的資料來源。接著 4.2 節觀察 k 值的大小與缺失值比例對原始 k 鄰近點填補法填補缺失值後產生天際線的差異。4.3 節探討不完整資料集在不同 k 值與缺失值比例下，各填補法(包含原始 k 鄰近點填補法、權重型 k 鄰近點填補法與本論文所提出的 sk-NN imputation 填補法)填補缺失資料後所計算出的天際線與原天際線的相似度。

4.1 實驗環境

4.1.1 實驗平台

本實驗的硬體設備包括處理器為 Intel® Core™ i7-6700 CPU @ 3.40GHz，記憶體為 16.0GB，作業系統為 Microsoft Windows 10 Profession version 2004 64bits。開發環境主要使用的程式語言為 Python 3.8.2 版本，並以 Anaconda 整合開發環境(IDE)。實驗程式架設內建於 Anaconda 的 Jupyter Lab 與 Notebook 虛擬環境中，並引用包含處理資料流的 pandas 套件、數學與矩陣函式相關的 numpy 套件、機器學習與資料挖掘所需要的 sklearn 套件與數據視覺化的 matplotlib 套件。本研究利用 Office Professional Plus 2019 Excel 來輔助實驗結果分析。

4.1.2 實驗資料來源

本研究使用的資料來源為 UCI Machine Learning Repository[29]中純數值資料類型的資料集，輸入資料集名稱分別為 Bike Sharing dataset、Real Estate Valuation dataset、Real-time Election Results Portugal 2019 dataset 三個資料集。資料集資訊、來源與內容特徵呈現於表 4.1[29]。

Bike Sharing dataset 之資料性質屬於單變量(univariate)，共有 17389 筆資料，屬性特徵(attribute characteristics)均為整數(integer)與實數(real)，特徵欄位(attributes)總共有 16 個特徵。Real Estate Valuation dataset 之資料性質屬於多變量(multivariate)，共有 414 筆資料，屬性特徵均為整數與實數，特徵總共有 7 個。Real-time Election Results Portugal 2019 dataset 之資料性質屬於多變量，共有 21643 筆，屬性特徵均為整數與實數，總共有 29 個特徵。

表 4.1 UCI Machine Learning Repository 輸入資料集資訊

	Bike Sharing dataset	**Real Estate Valuation dataset**	**Real-time Election Results Portugal 2019 dataset**
Data Set Characteristics	univariate	multivariate	multivariate, time-series, text
Number of Instances	17389	414	21643
Attribute Characteristics	integer, real	integer, real	integer, real
Number of Attributes	16	7	29

4.2 實驗一: 原始 k 鄰近點填補法對天際線的相似度

4.2.1 實驗目的

本實驗目的是針對不同鄰近點 k 值與缺失值比例(missing rate)，原始 k 鄰近點填補法對填補缺失資料後的天際線與原始天際線之相似度差異。

4.2.2 實驗方法

本實驗使用 Real Estate Valuation dataset 作為輸入資料集，由於特徵數量只有 7 個，因此取 k 值時只採 1 到 4 作為觀察對象。本實驗將同一缺失資料集於不同缺失值比例下，隨著 k 值增加，觀察原始 k 鄰近點填補法是否可以得到更高的天際線相似度。本實驗採用原始完整資料集中所得出天際線作為比較基準，hit ratio (計算方式詳見 3.5 節)愈高則相似度愈高，表示填補結果愈準確。

4.2.3 實驗結果與分析

實驗結果如圖 4.1、4.2、4.3、4.4 所示。圖 4.1 顯示，當 k=1 且缺失值比例尚未達到 20%時，原始 k 鄰近點填補法之 hit ratio 已經降至約 50%左右，且缺失值比例提高至 40%時，hit ratio 只剩下 40%左右。圖 4.2 顯示，k=2 且缺失值比例為 20%時，原始 k 鄰近點填補法之 hit ratio 比 k=1 時稍微上升約 10%；不過缺失值

比例為 40%時，hit ratio 與 k=1 的時候一致，推測可能是因為鄰近點變多，使得 hit ratio 稍微提升。

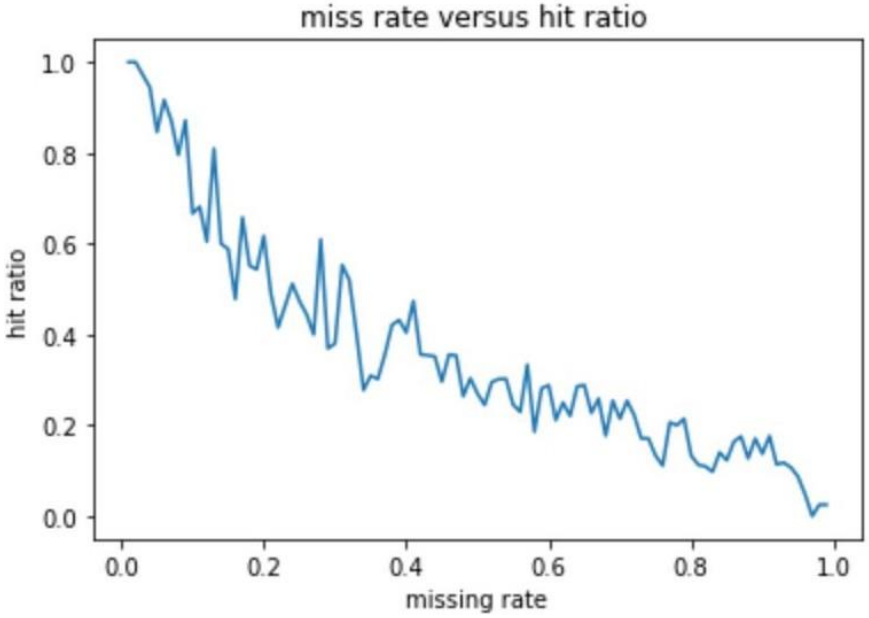

圖 4.1 不同缺失值比例下原始 k 鄰近點填補法之 hit ratio (k=1)

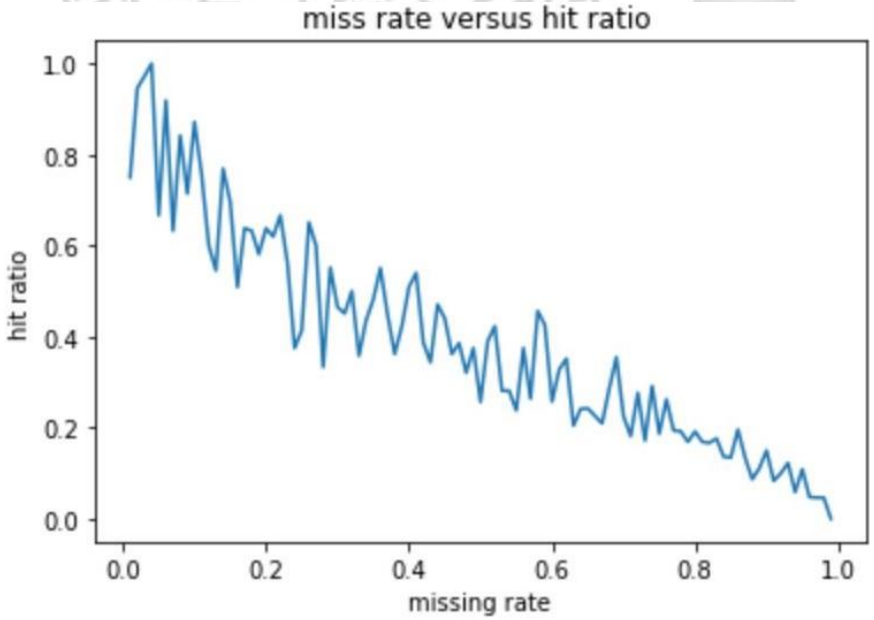

圖 4.2 不同缺失值比例下原始 k 鄰近點填補法之 hit ratio (k=2)

圖 4.3 顯示，k=3 且缺失值比例為 20%時，此時即使可參考的鄰近點數增加，原始 k 鄰近點填補法之 hit ratio 仍然未超過 60%，且缺失值比例為 40%時 hit ratio 降至約 30%。圖 4.4 顯示，k=4 且缺失值比例為 20%時，可參考的鄰近點為圖 4.1 的四倍，但是此時原始 k 鄰近點填補法之 hit ratio 大約為 40%，比 k=1 時

的 hit ratio 為 50%還低；且缺失值比例為 40%時，hit ratio 也沒超過 k=1 時的 40%。由此可知 k=4 並沒有比 k=1 時相似度更高。

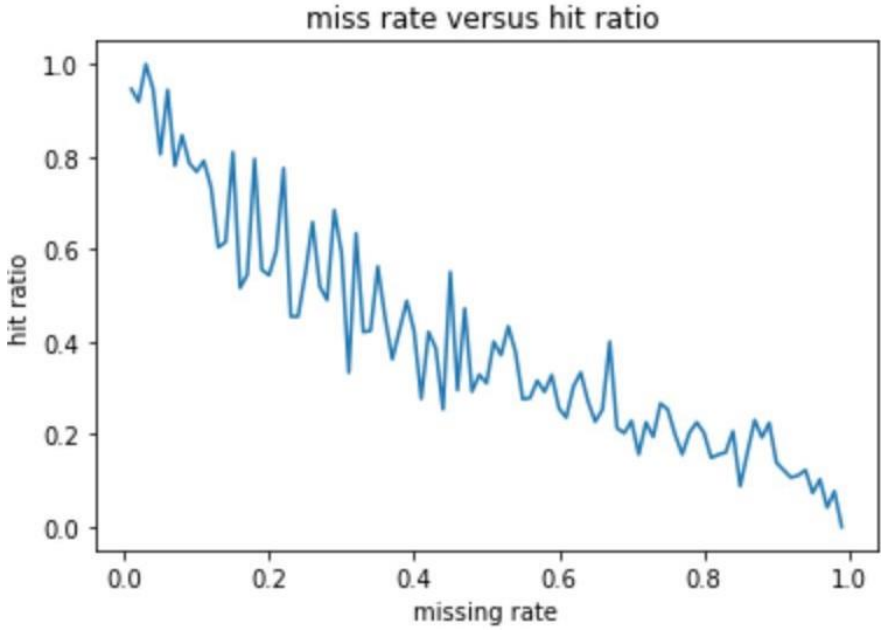

圖 4.3 不同缺失值比例下原始 k 鄰近點填補法之 hit ratio (k=3)

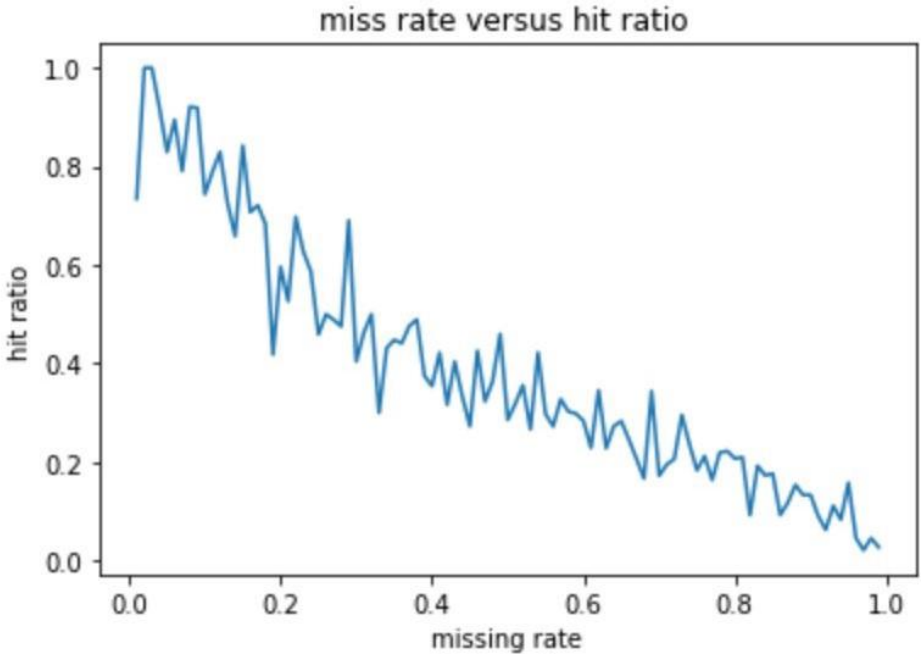

圖 4.4 不同缺失值比例下原始 k 鄰近點填補法之 hit ratio (k=4)

觀察圖 4.1 到圖 4.4 可知，隨著缺失值比例在資料集當中逐漸增加，原始 k 鄰近點填補法的準確率(即相似度)並沒有因為參考更多的鄰近點而有明顯地改善

填補效果。原始 k 鄰近點填補法期待透過增加鄰近點的數量，以提升填補值的品質，但是這將使得計算錯誤的填補值因為參考更多無效的鄰近點反而更嚴重。隨著缺失值比例逐漸上升，加上原始 k 鄰近點填補法中對鄰近點不足 k 個卻從缺不補的問題，致使即使計算鄰近點值的平均值也會逐漸失效，這同時也代表參考鄰近點值之可靠度會因高缺失值比例而降低。

4.3 實驗二: 各填補法產生的天際線與原天際線之相似度

4.3.1 實驗目的

本實驗的目的是針對固定鄰近點 k 值，隨著缺失值比例上升，觀察各填補法(包含原始 k 鄰近點填補法[14]、權重型 k 鄰近點填補法[7]與本論文所提出的 sk-NN imputation 填補法)於填補缺失資料後所產生的天際線與原天際線的相似度差異。

4.3.2 實驗方法

本實驗使用的資料集為 Bike Sharing dataset，特徵欄位共有 16 個特徵，因此 k 值最大範圍可以到 15，分別取三種不同 k 值，執行原始 k 鄰近點填補法、權重型 k 鄰近點法以及本研究提出的 sk-NN imputation 填補法，產生經過填補後的完整資料集。然後再分別執行天際線查詢演算法的程式 BNL[2]獲得近似天際線，觀察比較與原天際線的相似度差異。我們以 3.5 節的評測方法計算近似天際線與完整資料之原天際線的相似度。相似度愈高，則表示該填補法的效果愈好。

4.3.3 實驗結果與分析

當 k=1 時，實驗結果如表 4.2 與圖 4.5 所示。我們可以觀察到在缺失值比例(missing rate)由 20%提高至 30%間，原始 k 鄰近點填補法與權重型 k 鄰近點填補法所產生的天際線與原天際線的相似程度(後簡稱相似度)都從原本的 70%驟降至 53.8%，而本研究所提出的 sk-NN imputation 填補法仍然可以維持在 81.8%，直到缺失值比例為 50%時相似度才降到 63.6%。甚至當缺失值比例達到 70%時，原始 k 鄰近點填補法與權重型 k 鄰近點填補法產生的天際線的相似度分別只剩下 38.4%與 28.5%，但是 sk-NN imputation 填補法還可以維持 63.6%。

圖 4.5 更可以看出，當缺失值比例大於 75%以上時，sk-NN imputation 填補法

的結果已經與其他兩者有明顯的差距。由此可知，當鄰近點參考數量稀少且較高缺失值比例時，本論文提出的方法所產生的天際線比較接近原天際線。

表 4.2 不同缺失值比例下各填補法相似度比較表 (k=1)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
k-NN	0.8	0.7	0.538	0.6	0.421	0.5	0.384	0.25	0.1
weighted k-NN	0.8	0.7	0.538	0.6	0.421	0.5	0.285	0.25	0.1
sk-NN	0.8	0.8	0.818	0.909	0.636	0.75	0.7	0.636	0.545

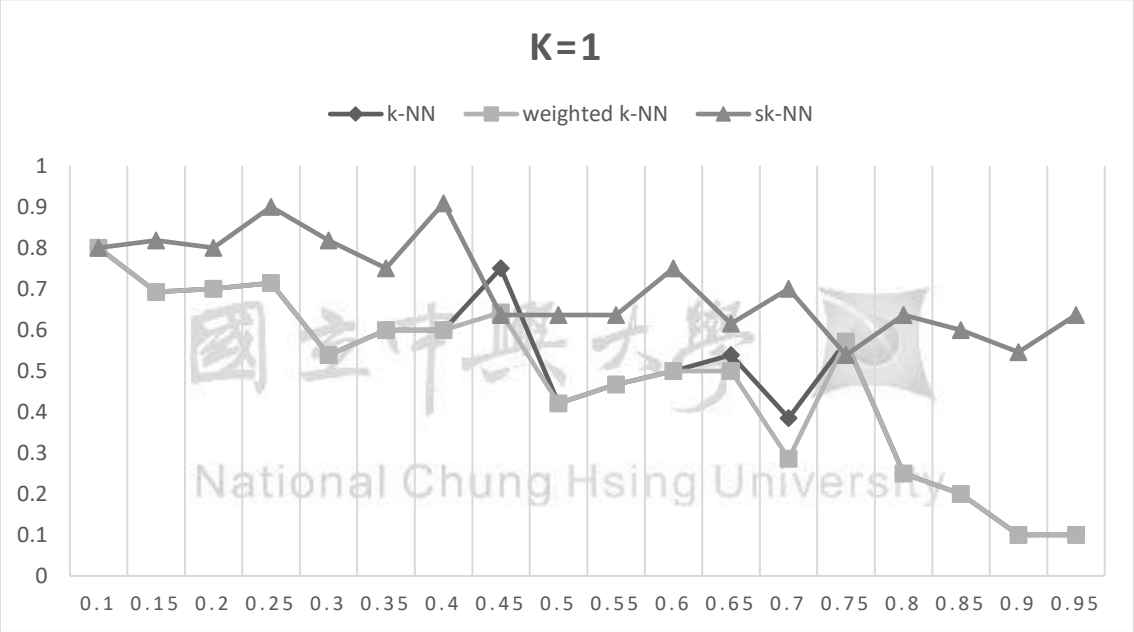

圖 4.5 不同缺失值比例下各填補法相似度比較圖 (k=1)

當 k=5 時，實驗結果如表 4.3 與圖 4.6 所示。我們可以觀察到當鄰近點比較多，原始 k 鄰近點填補法與權重型 k 鄰近點填補法比 k=1 時的表現更好一些。也可以觀察出在缺失比例介於 40%至 50%間，原始 k 鄰近點填補法有機會擁有較好的填補效果，這是因為缺失程度不高，原始 k 鄰近點填補法還能夠以足夠的 k 與鄰近點計算平均值後填回。即使缺失值比例已達 70%，原始 k 鄰近點填補法與權重型 k 鄰近點填補法都還有 60%與 66.6%的相似度。

圖 4.6 顯示，當 k=5 時，sk-NN imputation 填補法除了在缺失值比例為 40%與 50%時，相似度略低於其他兩者。在缺失值比例超過 80%的情形下，sk-NN imputation 填補法所產生天際線的相似度至少與其他兩種填補法相同或高於其他兩種填補法。甚至在缺失值比例高達 90%時候，sk-NN imputation 填補法所產生天際線的相似度為 k 鄰近點填補法的 2 倍。

表 4.3 不同缺失值比例下各填補法相似度比較表 (k=5)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
k-NN	0.9	0.666	0.666	1	0.625	0.533	0.6	0.454	0.181
weighted k-NN	0.9	0.625	0.6	1	0.529	0.692	0.666	0.1	0.25
sk-NN	0.9	0.833	0.909	0.818	0.5	0.8	0.6	0.5	0.363

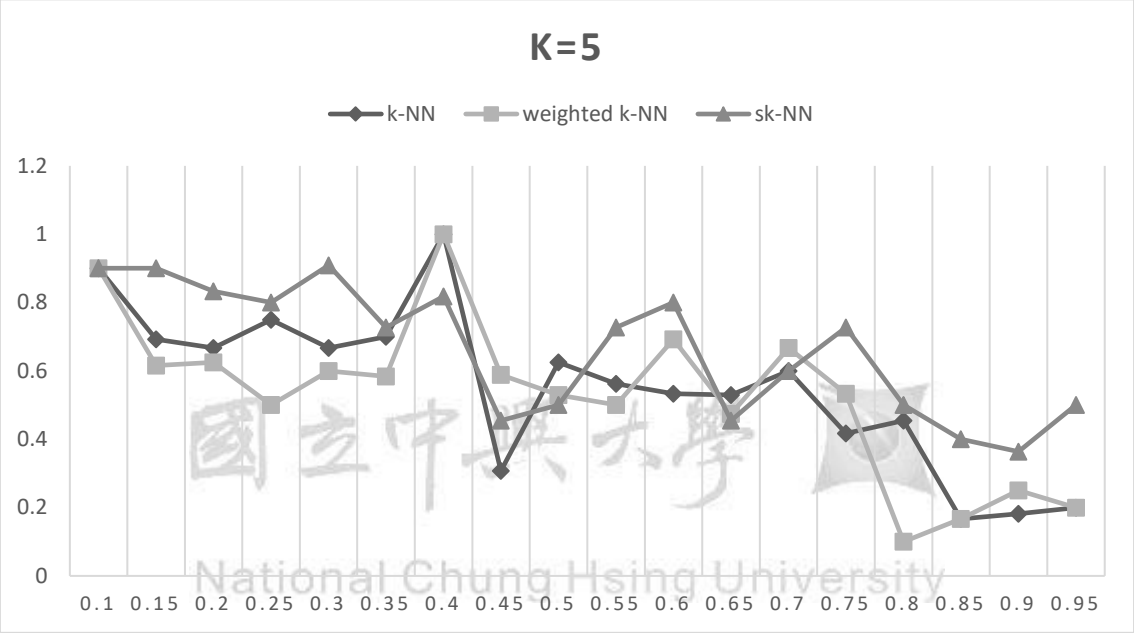

圖 4.6 不同缺失值比例下各填補法相似度比較圖 (k=5)

當 k=13 時，實驗結果如表 4.4 與圖 4.7 所示。我們可以觀察到，原始 k 鄰近點填補法與權重型 k 鄰近點填補法在缺失值比例從 30%到 50%的區間以及從 50%到 90%的區間，兩者的相似度分別從 75%與 81.8%下降至 18.1%。反觀本研究提出的 sk-NN imputation 填補法，除了在 40%的缺失值比例時其相似度為 45.4%以外，其他缺失值比例下幾乎都維持在 70%以上的相似度；意味高缺失值比例時，sk-NN imputation 填補法所觸發的採樣機制有其效果。

表 4.4 不同缺失值比例下各填補法相似度比較表 (k=13)

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
k-NN	0.9	1	0.75	0.333	0.727	0.583	0.615	0.384	0.181
weighted k-NN	0.9	0.833	0.818	0.4	0.277	0.529	0.428	0.352	0.181
sk-NN	0.9	1	0.833	0.454	0.818	0.7	0.727	0.75	0.7

圖 4.7 顯示，在缺失值比例 30%以下時，sk-NN imputation 填補法幾乎與原始 k 鄰近填補法無異。原因是此時可以參考的鄰近點足夠多且缺失值比例不高，加上 k 值夠大因此不易觸發採樣機制，使得 sk-NN imputation 填補法所計算的填補值與原始 k 鄰近填補法大部分都一樣。並且由圖 4.7 可以發現，缺失值比例大於 55%之後 sk-NN imputation 填補法與其他兩者開始有顯著的差異，原始 k 鄰近填補法與權重型 k 鄰近點填補法的相似度幾乎呈現嚴格下降趨勢，但 sk-NN imputation 填補法仍然維持在 60%至 80%之間。由此可知即使在對原始 k 鄰近填補法最有優勢的情形下(k=13)，在高缺失值比例下本研究所提出 sk-NN imputation 填補法對天際線的填補效果明顯具有優勢。

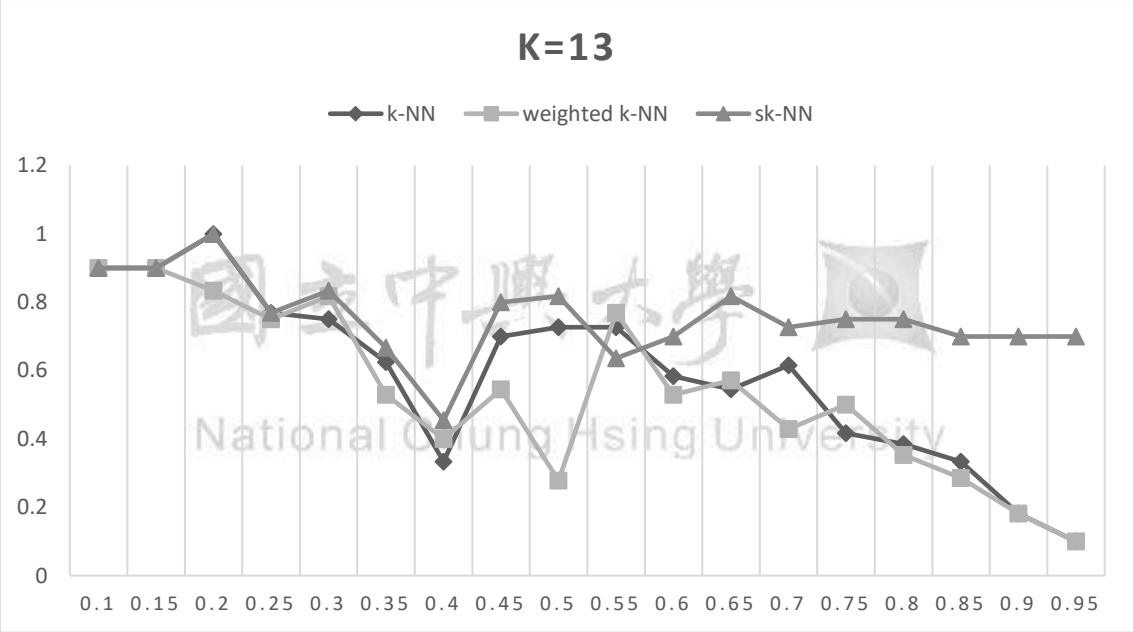

圖 4.7 不同缺失值比例下各填補法相似度比較圖 (k=13)

4.4 實驗結論

由實驗二結果(如圖 4.5、4.6、4.7 所示)可知，在 k 值無論為 1、5 或是 13 時，當缺失值比例上升，原始 k 鄰近點填補法與權重型 k 鄰近點填補法的填補效果都不好，分析其原因有二。

原因之一是以目前主流計算含有缺失值的距離公式，明確地定義出對應相同的維度上兩資料點都必須有值才能做計算。若兩資料點在相對應維度(假設維度 d)上至少有一為缺失值，則維度 d 那一項就不會被納入歐氏距離公式計算。這樣的結果使得維度 d 在兩資料點距離計算上的影響力會直接被忽視，這也是原始 k 鄰近點填補法在找尋最鄰近點時會誤判鄰近關係的主要原因之一。

另一個填補效果都不好的原因是在尋找 k 個鄰近點的過程中，可能無法滿足 k 個鄰近點在維度 d 上都沒有缺失值。原始 k 鄰近點填補法在遇到此種情況時，會選擇從缺不補。如此看似正常的步驟，探討在計算填補值的過程就會發現到，剩下不足 k 個的鄰近點在維度 d 值之權重其實正在無形地上升。說明如下:原始 k 鄰近點填補法如果找得到 k 個鄰近點，在維度 d 上都有值的話，平均每個鄰近點所分配到的權重值為 $1/k$ ；若找不到 k 個鄰近點，例如只找到 ε 個，其中 $1 \leq \varepsilon < k$ ，此時每個點的權重值將為 $1 \geq 1/\varepsilon > 1/k$。而 ε 值會隨著缺失值比例上升而降低，ε 與缺失值比例呈現負相關。換句話說，缺失值比例增加，可以找得到滿足 k 個鄰近點在維度 d 上不含有缺失值的機會就會降低。因此當缺失值比例愈高的時候，權重值被迫上升的狀況也就愈嚴重，導致原始 k 鄰近點填補法所填補的新值雖然表面上是看似公平的平均值，但事實上填補的新值幾乎退化為固定數值填補法(因為此時$\varepsilon = 1$)，如 2.2.2 節中所提及。因此填補效果不好也是可預期的。

即使試圖只依靠權重法計算加權平均值來試圖彌補填補值計算錯誤，也無法解決鄰近點不足的問題。這也是為何無法單純以權重型 k 鄰近點填補法就可以解決的。由實驗結果可知，在高缺失值比例下，挑選出更具參考性鄰近點的影響力比挑選出更多鄰近點的影響力更大

第 5 章 結論與未來方向

本本章分為兩節，5.1 節總結本研究，5.2 節探討未來可能的研究方向與工作。

5.1 結論

天際線查詢演算法無法適用於具有缺失資料的不完整資料集。針對完全隨機缺失類型(MCAR)的不完整資料集，本研究提出一個概念源自於 k 鄰近點填補法的 sk-NN imputation 演算法，能夠填補缺失資料，使得天際線查詢演算法可以運作於填補缺失資料後的完整資料集。sk-NN imputation 演算法在原始 k 鄰近點填補法中，加入鄰近差別權重分配方法以及對鄰近點採樣的機制，因此改善了原始 k 鄰近點填補法在高缺失值比例時，會退化為填補固定數值以及無法找到有效鄰近點，導致相似度很低的缺點。根據 4.3.3 節的實驗結果分析，在大部分的資料缺失情形下，sk-NN imputation 演算法填補缺失資料後所產生的天際線與原始天際線之相似度，比 k 鄰近點填補法所產生的天際線與原始天際線之相似度平均高出 10 到 20%。因此本研究提出的差別權重分配與新的鄰近點選擇機制，能夠改善天際線查詢演算法在不完整資料集中找到近似原始的天際線。

5.2 未來研究方向

在實驗的過程中，我們注意到當 k 值很小(例如 k=1)且缺失比例低時，填補缺失值容易觸發對鄰近點採樣的機制。因此我們希望未來能夠更進一步的去平衡 k 值與缺失值比例之間的關係，降低低缺失值比例時啟動採樣機制的頻率，並多利用鄰近點填補缺失值。其可能做法是針對觸發條件設一個門檻值，例如利用 4.4 節提到的 ε 值，限制啟動採樣機制的頻率。此外，在未來的研究方向上，我們認為也可以根據不同資料缺失類型，分別設計適合隨機缺失(MAR)以及完全非隨機缺失類型(MNAR)的填補法。

參考文獻

[1] A. A. Alwan, H. Ibrahim, N. Udzir, and F. Sidi, “Missing Values Estimation for Skylines in Incomplete Database,” *The International Arab Journal of Information Technology*, vol. 15, no. 1, pp. 66–75, 2018.

[2] S. Borzsony, D. Kossmann, and K. Stocker, “The Skyline Operator,” *Proceedings of the 17th International Conference on Data Engineering*, pp. 421–430, 2001.

[3] S. Deepa Kanmani, E. Kirubakaran, R. E. Blessing Vinoth, and A. S. Ebenezer, “An Effective Imputation Technique for Improving The Performance of Skyline Queries for Incomplete Database,” *Proceedings of the International Conference on Data Science and Communication (IconDSC)*, pp. 1–5, 2019.

[4] G. B. Dehaki, H. Ibrahim, N. I. Udzir, F. Sidi, and A. A. Alwan, “Efficient Skyline Processing Algorithm over Dynamic and Incomplete Database,” *Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services*, pp. 190–199, 2018.

[5] Y. Gulzar, A. A. Alwan, N. Salleh, I. F. A. Shaikhli, and S. I. M. Alvi, “A Framework for Evaluating Skyline Queries over Incomplete Data,” *Procedia Computer Science*, vol. 94, pp. 191–198, 2016.

[6] Y. Gulzar, A. A. Alwan, and S. Turaev, “Optimizing Skyline Query Processing in Incomplete Data,” *IEEE Access*, vol. 7, pp. 178121–178138, 2019.

[7] C. Hasler and Y. Tille, “Balanced k-Nearest Neighbor Imputation,” *Statistics*, vol. 50, no. 6, pp. 1310–1331, 2016.

[8] J. Huang, J. W. Keung, F. Sarro, Y.-F. Li, Y. T. Yu, W. K. Chan, and H. Sun, “Cross-Validation Based k Nearest Neighbor Imputation for Software Quality Datasets: An Empirical Study,” *Journal of Systems and Software*, vol. 132, pp. 226–252, 2017.

[9] D. W. Joenssen, “Hot Deck Methods for Imputing Missing Data,” *Machine Learning and Data Mining in Pattern Recognition*, vol. 7376, pp. 63–75, 2012.

[10] H. Kang, “The Prevention and Handling of The Missing Data,” *Korean Journal of Anesthesiology*, vol. 64, no. 5, p. 402, 2013.

[11] M. E. Khalefa, M. F. Mokbel, and J. J. Levandoski, “Skyline Query Processing for Incomplete Data,” *Proceedings of the IEEE 24th International Conference on Data Engineering*, pp. 556–565, 2008.

[12] J. Lee, H. Im, and G. You, “Optimizing Skyline Queries over Incomplete Data,” *Information Sciences*, vol. 361, pp. 14–28, 2016.

[13] J. Lee, G. You, S. Hwang, J. Selke, and W.-T. Balke, “Interactive Skyline Queries,” *Information Sciences*, vol. 211, pp. 18–35, 2012.

[14] R. Malarvizhi and D. A. S. Thanamani, “K-Nearest Neighbor in Missing Data

Imputation," *International Journal of Engineering Research and Development*, vol. 5, no. 1, pp. 5–7, 2012.
[15] X. Miao, Y. Gao, G. Chen, and T. Zhang, "k -Dominant Skyline Queries on Incomplete Data," *Information Sciences*, vol. 367–368, pp. 990–1011, 2016.
[16] X. Miao, Y. Gao, G. Chen, B. Zheng, and H. Cui, "Processing Incomplete k Nearest Neighbor Search," *IEEE Transactions on Fuzzy Systems*, vol. 24, no. 6, pp. 1349–1363, 2016.
[17] T. A. Myers, "Goodbye, Listwise Deletion: Presenting Hot Deck Imputation as an Easy and Effective Tool for Handling Missing Data," *Communication Methods and Measures*, vol. 5, no. 4, pp. 297–310, 2011.
[18] W. Ren, X. Lian, and K. Ghazinour, "Skyline Queries over Incomplete Data Streams," *The VLDB Journal*, vol. 28, no. 6, pp. 961–985, 2019.
[19] P. Royston, "Multiple Imputation of Missing Values," *The Stata Journal*, vol. 4, no. 3, pp. 227–241, 2004.
[20] D. B. Rubin, "Multiple Imputations in Sample Surveys-A Phenomenological Bayesian Approach to Nonresponse," *Proceedings of the Survey Research Methods Section of the American Statistical Association*, vol. 1, pp. 20–34, 1978.
[21] J. Shao, "Cold Deck and Ratio Imputation," *Survey Methodology*, vol. 26, no. 1, pp. 79–86, 2000.
[22] B. W. Silverman and M. C. Jones, "An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)," *International Statistical Review*, vol. 57, no. 3, p. 233, 1989.
[23] G. Tonini, M. Ricerche, S. Scartoni, M. Ricerche, C. Paoli, and M. Ricerche, "Missing Data For Repeated Measures: Single Imputation VS Multiple Imputation," *Proceedings of PharmaSUG Conference*, p. 10, 2015.
[24] G. Tutz and S. Ramzan, "Improved Methods for The Imputation of Missing Data by Nearest Neighbor Methods," *Computational Statistics & Data Analysis*, vol. 90, pp. 84–99, 2015.
[25] J. Van Hulse and T. M. Khoshgoftaar, "Incomplete-Case Nearest Neighbor Imputation in Software Measurement Data," *Information Sciences*, vol. 259, pp. 596–610, 2014.
[26] Y. Wang, Z. Shi, J. Wang, L. Sun, and B. Song, "Skyline Preference Query Based on Massive and Incomplete Dataset," *IEEE Access*, vol. 5, pp. 3183–3192, 2017.
[27] K. Zhang, H. Gao, X. Han, Z. Cai, and J. Li, "Modeling and Computing Probabilistic Skyline on Incomplete Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 7, pp. 1405–1418, 2019.
[28] S. Zhang, "Nearest Neighbor Selection for Iteratively kNN Imputation," *Journal of*

Systems and Software, vol. 85, no. 11, pp. 2541–2552, 2012.
[29] “UCI Machine Learning Repository,” 2013. [Online]. Available: https://archive.ics.uci.edu/ml/index.php.