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

(初稿)

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

**誌謝辭**

說明：表達對師長、受訪者、同學、家人等感謝之意，以一頁為原則，最多不超過兩頁。(非必備，由各系所自行決定)

# ­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 relys 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 avaliable 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

[摘要 1](#_Toc49205764)

[Abstract 2](#_Toc49205765)

[目次 3](#_Toc49205766)

[表目次 5](#_Toc49205767)

[圖目次 6](#_Toc49205768)

[第 1 章 簡介 7](#_Toc49205769)

[第 2 章 相關研究 9](#_Toc49205770)

[2.1資料缺失類型 9](#_Toc49205771)

[2.2缺失值處理方法 9](#_Toc49205772)

[2.2.1丟棄法 10](#_Toc49205773)

[2.2.2填補法 10](#_Toc49205774)

[2.3 k鄰近點填補法 12](#_Toc49205775)

[第 3 章 問題與方法 13](#_Toc49205776)

[3.1研究動機 13](#_Toc49205777)

[3.2問題定義 14](#_Toc49205778)

[3.3問題分析 14](#_Toc49205779)

[3.4 sk-NN imputation 演算法 15](#_Toc49205780)

[3.5以原天際線評斷填補法的表現優劣 19](#_Toc49205781)

[第 4 章 實驗結果與分析 21](#_Toc49205782)

[4.1實驗環境 21](#_Toc49205783)

[4.1.1實驗平台 21](#_Toc49205784)

[4.1.2實驗資料來源 21](#_Toc49205785)

[4.2實驗一: k值大小與缺失值比例對天際線結果的影響 22](#_Toc49205786)

[4.2.1實驗目的 22](#_Toc49205787)

[4.2.2實驗方法 22](#_Toc49205788)

[4.2.3實驗結果與分析 22](#_Toc49205789)

[4.3實驗二: 各填補法產生的天際線與原天際線之相似度 25](#_Toc49205790)

[4.3.1實驗目的 25](#_Toc49205791)

[4.3.2實驗方法 25](#_Toc49205792)

[4.3.3實驗結果與分析 26](#_Toc49205793)

[4.4實驗結論 29](#_Toc49205794)

[第 5 章 結論與未來方向 31](#_Toc49205795)

[5.1結論 31](#_Toc49205796)

[5.2未來研究方向 31](#_Toc49205797)

[參考文獻 32](#_Toc49205798)

# 表目次

[表 2.1 含缺失值資料集M 10](#_Toc49253711)

[表 2.2 刪除資料列 11](#_Toc49253712)

[表 2.3 刪除維度 11](#_Toc49253713)

[表 2.4 過度刪除資料列 11](#_Toc49253714)

[表 2.5 過度刪除維度 12](#_Toc49253715)

[表 3.1 sk-NN imputation演算法符號定義表 18](#_Toc49253716)

[表 4.1 UCI Machine Learning Repository輸入資料集資訊、來源與內容特徵 24](#_Toc49253717)

[表 4.2 k=1各填補法比較表 28](#_Toc49253718)

[表 4.3 k=5各填補法比較表 29](#_Toc49253719)

[表 4.4 k=13各填補法比較表 30](#_Toc49253720)

# 圖目次

[圖 3.1 NaN-Euclidean distance 15](#_Toc49205804)

[圖 3.2 sk-NN imputation演算法 18](#_Toc49205805)

[圖 3.3 Procedure Impute\_Process() 19](#_Toc49205806)

[圖 4.1 k=1時hit ratio versus missing rate圖 23](#_Toc49205807)

[圖 4.2 k=2時hit ratio versus missing rate圖 23](#_Toc49205808)

[圖 4.3 k=3時hit ratio versus missing rate圖 24](#_Toc49205809)

[圖 4.4 k=4時hit ratio versus missing rate圖 25](#_Toc49205810)

[圖 4.5 k=1各填補法比較圖 27](#_Toc49205811)

[圖 4.6 k=5各填補法比較圖 28](#_Toc49205812)

[圖 4.7 k=13各填補法比較圖 29](#_Toc49205813)

# 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，本研究提出了一個賦予不同鄰近點合理權重值的方法以及挑選鄰近點機制。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挑選鄰近點機制，儘可能地尋找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。

為了驗證本方法的有效性，本研究進行兩個模擬實驗。分別觀察k值與缺失值比例對天際線結果的影響，以及比較各填補法填補後所產生的天際線與原天際線的相似程度。本研究方法在缺失值比例較低時，透過更合理的差別權重分配，使得較接近的鄰近點對缺失值有更大的影響力，計算出比原始k鄰近填補法更適合的填補值。而在缺失值比例較高的情況下，比原始k鄰近填補法更積極地找其他可參考的點作為鄰近點後計算填補值。實驗結果顯示，尤其在k值足夠大時(k值越接近維度個數為越大)，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。在缺失值比例高(大於50%)時，與原天際線相似度評斷的填補效果也都超過六成。

本研究的主要貢獻在於改善原始k鄰近點填補法不合理的權重分配，以提升鄰近點的影響力。以及在當鄰近點不足k個時，提供鄰近點的選擇方式，以找到足夠的可參考的鄰近點進行填補，避免退化為填補固定數值。這使得在高缺失值比例情況下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。

# 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)、完全非隨機缺失類型(missing not at random，MNAR)以及完全隨機缺失類型(missing completely at random，MCAR)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。例如有關心情沮喪議題的問卷較常見到缺少男性族群的資料，使得男性族群與該問卷議題看似有相關性，但這只是男性較不願意填寫此類型問卷，事實上資料缺失因素與男性族群毫無關係，此一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。此一類型的缺失值與某一維度具有一定程度的相關性，屬於此一類型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，難免會遇到手上的資料集含有若干個資料欄位是不存在的，這樣的不存在任何資料的情況被稱為資料缺失，針對這些沒有資料的欄位，我們統一稱之為缺失值。

面臨資料集內含有缺失值時，為了讓整體資料集可以做後續的統計、分析、計算等工作。我們必須先對資料集內的缺失值做一些處理，無論是忽視不管、把含有缺失值的資料移除，或者是賦予這些缺失值一些替代的值，都可以是處理缺失值的其中一種方法。這些方法稱為缺失值處理方法，而缺失值處理方法可分為三種策略，丟棄法(dropout)與填補法(imputation)[3], [6], [14]，而k鄰近點填補法是效果較好的填補法，以下分別說明之。

### 2.2.1丟棄法

[介紹丟棄法]

首先介紹丟棄法，顧名思義，此類型的方法面對缺失值所採取的行為，是對缺失值所在的位置，根據目的不同以及適合的資料缺失類型。將缺失值從原資料集當中移除，以達到移除後剩下資料全部為完整資料(每個維度皆有值)且資料集為完整資料集(不含缺失值的資料集)。

[說明丟棄法種類]

即使缺失值的位置相同，我們也能以移除的方式不同，使得移除後有不同的結果。若存在有一筆資料點含有缺失值，將該資料點從原資料集中整筆刪除，這種移除缺失值的方式稱為刪除資料列。同理，若存在有一個維度含有缺失值，將整個維度從資料集內刪除，這種移除缺失值的方式稱為刪除維度。以下分別介紹刪除資料列與刪除維度兩種。表2.1為一個含有若干缺失值的資料集，表中以橫線顯示缺失值欄位。

[分別說明刪除資料列]

刪除資料列，也稱為名單去除法(listwise deletion)。根據刪除資料列的定義，將具有缺失值的資料點刪除，經刪除後會剩下id為2與id為4的資料點，如表2.2所示。假設資料集內的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，此時比較適合採用刪除資料列來移除資料集的缺失值。理由是在上述假設中，將含有缺失值的資料點整筆刪除。而刪除後剩下的完整資料集相較於刪除前的資料集，等同於從一群資料母群體內取其中一部分的子集合 ，而 就是蒐集那些在所有維度中都沒有缺失值資料點的集合。且子集合 的大小會受到缺失值比例增加而減少。理論上 的分佈會與母群體(原資料集)的一致，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。當缺失值比例較高的時候，同時也意味著資料集內的完整資料較少，凸顯出資料集內可用的完整資料點個數不足的問題。

表 2.1 含缺失值資料集M

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

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 |

[分別說明刪除維度]

假設資料集內缺失值的缺失類型為不完全隨機缺失類型(MNAR)，且在某個維度d缺失的情況非常嚴重時，此時可以考慮將維度d整個刪除。也就是丟棄法中另一種移除缺失值的方法，刪除維度。如表2.3所示，經刪除後會剩下weight與id兩個維度。刪除維度在資料處理的角度就相當於將原本的資料集降維，好處是在對很多演算法而言可以減少執行時間。但是過度刪除維度也會有其衍生的問題，由於刪除後所剩的維度不足或是被刪除的維度對於資料集本身有很大的代表性，讓剩下的維度無法完整地表現資料集的特徵。例如：資料集內的分類標籤(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，導致刪除後的分類不準確的問題。

表 2.3 刪除維度

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

[說明丟棄法優缺點]

採用丟棄法的優點是可以透過簡單移除缺失值的方式，同時達到保有完整資料集且不會增加原資料集的偏差。然而，丟棄法的缺點是，在丟棄法的過程中可能會因為過度刪除產生額外的問題。

[過度刪除資料列缺點]

以刪除資料列來說，過度刪除資料列會導致資料點的數量不足。如表2.4表示由表2.2過度刪除資料列的結果，若是id為2的資料點任何維度也存在著缺失值，則原資料集剩下一筆id為4的資料點。

表 2.4 過度刪除資料列

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | age | gender | height(cm) | weight(kg) |
| 4 | 21 | Female | 158 | 47 |

[過度刪除維度缺點]

以刪除維度來說，過度刪除維度會喪失資料集表現的特徵。如表2.5為由表2.3因過度刪除維度後，失去維度id後喪失原資料集表現特徵。失去兩者都會損失因為移除缺失值而喪失資料集內的資訊。

表 2.5 過度刪除維度

|  |
| --- |
| weight(kg) |
| 53 |
| 75 |
| 67 |
| 47 |

### 2.2.2填補法

[說明填補法]

接著介紹填補法，不同於移除缺失值，另一種面對缺失值的處理方法，透過觀察資料集內的資料點，參考那些不含缺失值的資料點後，賦予缺失值一個合理的參考值。這樣計算出一個合理的參考值給予缺失值的過程，稱之為填補法。填補法與丟棄法相同的地方是，填補後仍然可以保持資料集的完整性；不同的是，經填補法填補後的資料個數與填補前一致。

[填補法種類，如何區別]

根據填補後可以產生一組或是多組的資料集數量來看，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，單一填補法經單次填補完後只會產生一組經填補過後的完整資料集，之後便會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過後，產生多組不同的完整資料集，再將這些多組填補結果加以合併後才做後續的統計與分析。

[說明那些屬於單一填補法]

單一填補法[3], [14]主要原理是填補某筆資料的缺失值。如果計算過程中產生多個可能的值，也會根據不同演算法的機制來挑選其中一個值來填補該缺失值。

根據單一填補法的定義，屬於單一填補法的有：固定數值填補法、熱卡填補法(hot deck imputation)[9]、冷卡填補法(cold deck imputation)[21]、平均值填補法(mean imputation or mean substitution)、迴歸填補法(regression imputation)與最鄰近填補法(nearest neighbor method)[10]，於下面依序介紹。

[固定數值填補法]

固定數值填補方法是對相同維度的缺失值賦予同一固定數值，例如以缺失值相同維度下的眾數、平均值、中位數、極大值、極小值等等[14]。此方法最明顯的優點就是計算量即少，填補過程最簡單且不受任何缺失種類與資料集分布。但填補固定數值的問題在於，當缺失值數量更多的時候，由於在相同維度下的缺失值也都填補了相同的值，使得降低了相異資料點之間的差異性，造成填補的效果很差。除非缺失值數量極少，注重差異性的資料集不建議以固定數值填補缺失值。

[熱卡、冷卡填補法]

熱卡填補法也稱為last observation carried forward (LOCF)，會依照每個維度給與不同條件下，資料集中的變數根據所設定的條件做排序，以最常或者最近被觀測到的數值填補該維度值下的缺失值。而冷卡填補法與熱卡填補法最大的不同，就是以最不常或者最少被觀測到的值來填補該維度下的缺失值。

[平均值值填補法]

平均值填補法也就是上述提到填補單一固定數值中的填補平均數一例，是以各別維度中所有已知的數值計算其平均值，作為缺失值的填補值。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，也間接影響資料的分布方式。

[迴歸填補法]

迴歸填補法的假設為含有缺失值的連續型資料與資料集當中的其他維度具有一定的相關性。即缺失值類型為MNAR，缺失值可以經迴歸計算後所得出的預測結果並取代該缺失值。此填補方法會使得填補值的變異數降低，故需要再加上隨機誤差作為最終缺失值的填補值。與平均填補法相同是，當缺失值比例很高時榮以影響資料的分布情形，因此兩者都只適合在缺失值比例較小時使用。

[最鄰近值填補法]

最鄰近填補法由Fix與Hodges在1951提出[22]，其原理為一資料集已知n樣本資料，其中某一個 為資料集中第 筆資料，若存在一筆資料為 ，則可以運用那n筆資料中的已知訊息將該 歸類。在資料歸類的過程中，會牽涉到任意兩筆資料距離(distance)的遠近或是相異程度的大小。用適合計算該資料集的距離方式測量出任兩筆資料之間的距離，若距離越小則可以推測兩者的值越相近。

[介紹多重填補法，步驟]

以上填補法屬於單一填補法，

多重填補法首次在1978年被Rubin提出來[20]，著重於分析與解決問題上。多重填補法利用蒙地卡羅法(Monte Carlo method)的概念，將缺失資料填補m個可能的數值，產生m個可能的資料集並進行分析。多重填補法主要有三個過程：

第一，對缺失值被重複填補m次以產生m個完整資料集。第二，將m個完整資料集使用正規程序進行分析。第三，將m個結果合併以進行後續的統計推論。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。多重填補法仰賴於資料集上模擬分布模型[19]，在遇到缺失資料時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需求且在實務上不容易進行模擬。

[填補法優、缺點]

填補法的優點

在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性，參考無缺失值的資料點以填補缺失值較有可參考性且預期會有比較好的效果，也比較接近完整資料集的特徵。

### 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。k鄰近點填補法的填補過程敘述如下，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是，無論維度多寡，填補值的計算方式均相同。計算過程也與任何維度無關性，這使得k鄰近點填補法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。且比起填補固定數值，例如平均數、中位數、極值、眾數等，k鄰近點填補法來的更準確許多。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值。

然而，k鄰近點填補法也存在某些缺點。第一，由於k鄰近點填補法參考鄰近點的值，在無資料預處理的前提下，若挑選到不好的鄰近點或者鄰近點含有偏差值，可能無法填補更具參考性的值於缺失值上。第二，k鄰近點填補法必須計算所有資料點之間的距離，此過程很容易增加大量計算時間。以n筆資料為例，新增一筆資料點時就必須多n次的距離計算。此時若再新增一筆資料點時，必須再增加n+1次的距離計算(目前總共有n+1筆資料)，以此類推。使得資料點間的距離計算量會隨著資料點的增加而呈現指數成長。

# 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 應該為 且 應該為 ，故 。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 為 ，而新的AC距離 則為，使得 。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，尋找其他在與缺失值相同維度但不具缺失值的點，接著從這些點採樣其中k個點，並且取採樣後的k個點在相同維度上的值取其平均值，最後以該平均值填補原缺失值。目的是不讓距離值因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 含有若干個缺失值，常數 為可參考鄰近點的上限個數，n與m分別為 的資料點個數與 的維度個數。 表示 中第 筆資料點，且 為該資料點 於維度 的值。

為不完整資料集 中任兩資料點 與 之間距離的距離矩陣，且其中 為任兩資料點 與 之歐氏距離。 是一個資料點 在每一個維度 的值 相對其他任意相異資料點 的權重矩陣，代表兩資料點 與 之間的權重值。 是一個決定權種值的函數，用來決定某一缺失值的被參考維度值 的權重值。若 值為 ，則所有權重值 均被設為1 ; 若 值為 ，則 會以 中任兩資料點 與 之間距離 的倒數作為該權重值。

NN list是一個由元素 所構成的有序串列。每一個 內記錄k個鄰近 的資料點 的索引值(即 中的第j筆資料點)。而索引值j的順序則是依照資料點 與 相距的距離值 ，由小至大排序。 記錄 所有相鄰近點中， 不為0的且與 第h個最接近資料點 的索引值j。

最後說明填補過程中會使用到的符號，若輸入資料集 當中某 為缺失值，則經過本論文填補法計算過後會將會賦予該缺失值一個新值 填入原缺失欄位。mask為一個長度同為輸入參數k值的陣列，去記錄某 最鄰近k個參考點在發現該缺失值所在的維度 是否為缺失值。若被參考的鄰近點在與該缺失值相同維度值也為缺失值，則標註為True，否則為False。最終，演算法輸出一個原始缺失值位置均已被填補的資料集 。

表 3.1 sk-NN imputation演算法符號定義表

|  |  |
| --- | --- |
| notation | description |
| n | numbers of data instances |
| m | dimensionality of input data set |
| incomplete data set ,  = | an incomplete data set of size |
| , k | specified constant k to determine number of neighbors |
|  | data value of the data instance,  at index row and column |
|  | data instance at row |
|  | a symmetric distance matrix, records the distance between two pair wised data instances |
|  | distance between and , denoted as , |
|  | a weight matrix record to pairwise any and , i |
| = | weighting value with respect to between and |
|  | type of weighting |
| NN list | record a sorted nearest neighbor list between all pairs of and |
|  | a sorted list at index i in NN list to keep all neighbors of , where |
|  | nearest neighbor of element at index h in sorted list |
|  | imputed value |
| mask | the neighbor of is missing or not |
| imputed data set | an imputed data set of size |

圖3.2為本論文所提出的sk-NN imputation演算法，輸入參數為一不完整資料集、一個自設常數k、以及決定權重值給與的方式，執行過程依序為：step1到step2載入輸入不完整資料集並初始化imputed data set。step3先初始化距離矩陣，step3-1到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 來決定計算任兩資料點之間的權重值，若 為uniform如同k鄰近點填補法給予相同權重，若 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越大對彼此的影響力越小。step5列出每一筆資料點其所有鄰近點並儲存於 中表示的所有鄰近點，並且依照與該點 的歐式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，執行副程式Impute\_Process()如圖3.3所示。step7則回傳填補後的新完整資料集，結束sk-NN imputation演算法。

|  |
| --- |
| Algorithm sk-NN imputation () {  Input : incomplete data set , constant k, weight type  Output : imputed data set  Method:  step 1. load incomplete data set  step 2. initialize imputed data set , as a copy of  step 3. initialize all values of distance matrix into zero  3-1. **for** each in distance matrix **do**  3-2. Euclidean distance of pairwise data samples  **end for**  step 4. initialize all values of weight matrix into zero  4-1. **for** each in weight matrix W **do**  4-2. =  **end for**  step 5. establish a nearest neighbor list to store all nearest neighbors with respect to certain  5-1. initialize an empty NN list, with size n  5-2. **for** each in **do**  5-3. retrieve all between any pair of the and from  5-4. sorting in ascending order, keep track of corresponding  5-5. **repeat** from step 5-6 to step 5-9  5-6. if 0  5-7. append into  5-8. **until** size of or all inserted into  5-9. update  **end for**  step 6. search all missing values among data set, then impute new value back into the missing position  6-1. **for** each in **do**  6-2. **if** is missing **then**  6-3. call procedure Impute\_Process(i, d, k)  **end if**  **end for**  step 7. **return** an 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為初始化表示鄰近點缺失情形的mask array。step2找尋對於每一個缺失值在該維度下的鄰近點的值是否也為缺失值。若為缺失值則在mask array中標記True，否則為False。step3則是檢視所有該缺失值的鄰近點在相同維度下的值是否全部都缺失，若mask的值皆為True，則判定可參考的鄰近點在相同維度下的值也都是缺失值。此時就會觸發採樣機制去參考相同維度下不是缺失值的點(不一定為鄰近點，因為鄰近點已經不具參考性)。step4將計算過後的填補值回傳至原缺失位置，至此填補其中某一缺失值結束。

|  |
| --- |
| Procedure Impute\_Process(i, d, k) {  Input : row index i, column index j, nearest neighbor k  Output : imputed value at index (i, d)  Method:  step 1. create a mask array and initialize all elements in mask array to False  step 2. find all elements in with respect to is not missing in  2-1. **for** each index from 0 to k **do**  2-2. r  2-3. **if** is missing **then**  2-4. mask[r] True  **end if**  2-5. **else**  2-6. mask[r] False  **end else**  **end for**  step 3. retrieve values in which index in mask array assigned to False  3-1. **for** each index in mask **do**  3-2. **if** all elements in mask array are not all True **then**  3-3. retrieve all value in  **end if**  3-4. **else**  3-5. reset all elements in mask array to False  3-6. sampling the rest of not missing value at column d  **end else**  3-7. evaluate mean or weighted mean for imputed value  3-8. assign imputed value to  step 4. **return** 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

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。漢明距離主要是用在計算兩個字串相對應的位置不同字符的個數，換句話說，將一個字串變換成另外一個字串所需要替換字符的總個數即為漢明距離。本論文使用漢明距離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。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。

本論文所採用的是對集合上的漢明距離概念，換句話說，集合A必須插入或刪除多少元素才能使兩集合相同，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 與近似天際線 的相似程度計算方法如下:

* 1. 天際線 與 中相異元素總個數為該集合size。
  2. 與 集合具有相同元素個數稱為hit count。
  3. 之中有的元素但 中沒有的元素個數以及 之中有的元素但 中沒有的元素的個數總和，稱為miss count。miss count就是本論文所定義的set hamming distance。

4. hit ratio =

舉例說明:

original skyline set: size為5

estimated-1 skyline set : ，size為7

estimated-2 skyline set : ，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 = 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 = 0.5。

由上述例子可知， 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來評斷各填補法填補效果優劣之依據。

# 實驗結果與分析

本章節依序於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天際線結果的影響。4.3節不完整資料集分別在不同k值與缺失值比例下，比較原始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填補法各填補法所產生的天際線與原天際線的相似度衡量各填補法對資料的效果。

## 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)，實驗所執行的程式架設內建於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。並利用Office Professional Plus 2019 Excel來輔助實驗結果分析。

### 4.1.2實驗資料來源

本研究所使用的資料集來源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資料集性質屬於單變量(univariate)，共有17389筆資料個數，資料特徵(attribute characteristics)均為整數(integer)與實數(real)，特徵欄位(attributes)總共有16個特徵欄位。

Real estate valuation dataset資料集性質屬於多變量(multivariate)，共有414筆資料個數，資料特徵均為整數與實數，特徵欄位總共有7個。

Real-time Election Results Portugal 2019 dataset資料集性質屬於多變量，共有21643筆，資料特徵均為整數與實數，特徵欄位總共有29個。

表 4.1 UCI Machine Learning Repository輸入資料集資訊、來源與內容特徵

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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: k值大小與缺失值比例對天際線結果的影響

### 4.2.1實驗目的

本實驗目的是在探討參考鄰近點k值越大與缺失值比例(missing rate)，對填補後的資料集找天際線是否會有更好的填補效果。

### 4.2.2實驗方法

本實驗方法使用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。本實驗方法將同一缺失資料集分別從k=1測試，觀察相同k值分別於不同缺失值比例下情況之下，隨著k值增加是否可以得到更佳的準確度。衡量此實驗效果，本實驗採用原完整資料集中所得出天際線作為基準，若hit ratio越高則表示填補結果越準確。

### 4.2.3實驗結果與分析

觀察實驗一結果，根據圖4.1顯示當k=1且缺失值比例尚未達到20%時，hit ratio已經降至約50%左右，且缺失值比例為40%時，hit ratio剩下40%左右。



圖 4.1 k=1時hit ratio versus missing rate圖

根據圖4.2顯示，k=2且缺失值比例為20%時，雖然可能因為可參考的鄰近點增加而使得hit ratio稍微上升約10%，且缺失值比例為40%時，hit ratio依然保持40%上下。



圖 4.2 k=2時hit ratio versus missing rate圖

根據圖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增加，hit ratio仍然未超過60%，且缺失值比例為40%時，hit ratio降至約30%。



圖 4.3 k=3時hit ratio versus missing rate圖

根據圖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40%的hit ratio。由此可知k=4並沒有比k=1時效果更好。



圖 4.4 k=4時hit ratio versus missing rate圖

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並沒有因為參考更多的鄰近點而明顯改善其填補效果。原始k鄰近點填補法只著重在增加鄰近參考點的數量，若是可供參考的鄰近點已經為數不多，取再多的鄰近點只會使得填補值的計算錯誤更加劇。缺失值比例逐漸上升，加上原始k鄰近點填補法中對可參考點不足k個卻從缺不補的問題，致使即使計算鄰近點值的平均也會逐漸失效，這同時也意味著可供參考點數量以及參考值之可靠性會因為高缺失值比例而嚴重不足。

## 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

在同一k值下，在不同缺失值比例下，本論文方法與原始k鄰近點填補法即所能夠找回近似天際線的程度。

### 4.3.2實驗方法

本實驗所使用的資料集為Bike Sharing dataset，因此k值最大範圍可以到16，分別取三種不同k值分別做三次比較，並觀察原始k鄰近點填補法、權重型k鄰近點法以及本論文方法填補後的值所能夠找回原天際線的程度。x軸為缺失值佔整體資料集當中的比例，y軸為填補所有缺失值之後，再分別跑同一支天際線查詢演算法的程式(以BNL為例)，經各填補法填補後所產生出的天際線與無缺失值的原天際線比較後。以3.5節的評測方法計算出前後兩個天際線的相似程度，若越接近原天際線則y軸的值越接近1.0，表示該填補效果越好。

### 4.3.3實驗結果與分析

實驗二結果顯示，如表4.2所示在k=1下，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(missing rate)由20%上升到30%間，其各別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填補法仍然維持在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相似度分別只剩下38.4%與28.5%。sk-NN填補法還可以維持63.6%。

表 4.2 k=1各填補法比較表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

由圖4.5更可以看出來缺失值比例大於75%以上時，sk-NN的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。

圖 4.5 k=1各填補法比較圖

表4.3顯示在k=5時，各個填補法所產生的天際線與原天際線的相似度。可以看出來當鄰近點比較多，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。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。

表 4.3 k=5各填補法比較表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

而sk-NN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，但大部分的缺失情形都是至少甚至優於其他兩種填補法。甚至在缺失值比例90%時候，sk-NN填補法所產生天際線的相似度為k鄰近點填補法的2倍。

圖 4.6 k=5各填補法比較圖

當k=13時，各填補法所產生的天際線與原天際線的相似度。從表4.4中可以看出來，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填補法，除了40%缺失值比例相似度45.4%以外，其他情形下幾乎都維持在70%以上的相似度。表示高缺失值比例時所觸發的採樣機制有其效果。

表 4.4 k=13各填補法比較表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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 |

由圖4.7觀察到，在缺失值比例30%以下時，sk-NN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填補法所計算值與原始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填補法仍然維持在60%至80%之間。由此可知即使在對於原始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填補法對天際線的填補效果明顯具有優勢。

圖 4.7 k=13各填補法比較圖

本論文的方法之中有採取採樣的機制，此機制在k值不大的時候且缺失值比例不高時會比較容易啟動，但隨著缺失比率增加下，增加k值所帶來的益處會越來越不明顯。而本論文中適時觸發採樣的方式，比增加k值有更大的機會找到有效的鄰近點。從表4.2與4.3中可以看出原始k鄰近點填補法從缺失比例約為30%時，相似度度就開始急遽下降，雖然中間偶有小幅度的上升，但可看出下降程度在缺失值比例大於75%以後下降幅度加劇，表4.4更是提前在缺失值達55%時就開始大幅下滑，這都顯示出一現象，原始k鄰近點填補法參考鄰近點的機制，在缺失率高下無法具有穩定的填補效果。

## 4.4實驗結論

由實驗二中的圖可以知道無論是原始k鄰近點填補法或是權重型k鄰近點填補法，在k值分別為1、5與13時，隨著missing rate在整體資料集中升高，亦無法有很好的填補效果，經分析其原因有二。

其一是在考慮最鄰近的k值作為參考該維度的值時，可能會遇到不足k個不為缺失值，而原始k鄰近點填補法在遇到此種情況時，會選擇從缺不補，使得剩下的不足k個被參考鄰近點該維度值之權重無形中上升。當缺失值比例越高的時候，此狀況也就越趨明顯，最後原始k鄰近點填補法所填補的新值雖然為平均，但也幾乎被簡化為單一值填補法的效果，最終效果如同只填補均值、眾數、或最大最小數結果一樣。

其二是當遇到缺失值計算距離的機制，當兩資料點計算出距離時，若兩個資料點在相對應維度上其中一點至少有一數值為缺失值，則在計算歐氏距離時該維度值之間的差平方並不會被納入歐氏距離的計算式中，使該維度對距離上的影響力被無視，也是原始k鄰近點填補法在找尋最接近鄰近點時會被誤判鄰近關係的主要原因之一。

上述兩個原因仍無法藉由權重法計算加權平均數來彌補此一現象的缺陷，因此亦可看出即使採用權重型k鄰近點填補法也不會有太好的填補效果，從此可以得知，挑選可參考性鄰近點的值在高缺失值比例下，其影響力遠比參考更多鄰近點來的大。

# 結論與未來方向

本章分為兩節，5.1節總結本研究，5.2節探討未來可研究的方向與工作。

## 5.1結論

本研究提出sk-NN Imputation演算法，利用給予不同權重值以及新的採樣機制。同時改善k鄰近點填補法中，因為含有缺失值導致找錯鄰近點與鄰近點不足的困境。本研究發現，資料集的missing rate超過20%下且當k為1時， k鄰近點填補法所產生的近似天際線與原天際線的相似度已驟降至50%。而sk-NN Imputation演算法所產生的近似天際線與原天際線的相似度至少有80%。當k為5時，k鄰近點填補法的相似度剩下66.6%而sk-NN Imputation演算法卻可以保有83.3%的相似度。即使在對k鄰近點填補法最有利的情形，k取13下，資料集的missing rate為80%時，k鄰近點填補法的相似度也僅剩下38.4%但sk-NN Imputation演算法的相似度卻仍然可以維持在75%。根據實驗證實，本論文所提出的sk-NN Imputation演算法在解決改善不完整資料集時執行天際線查詢演算法具有良好的填補效果。

## 5.2未來研究方向

在未來的研究方向上，我們認為可以根據不同缺失類型，找出分別適合隨機缺失以及完全非隨機缺失類型的填補法。甚至觀察不同維度之間，其中是否一些維度對於資料集本身更具有影響力，給予維度間不同權重值。若是輸入資料集具有部分天際線相關的資訊，則也可以透過目前已知的部分天際線去填補可能的缺失值，以增加近似天際線的相似度。

# 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,” in *Proceedings 17th International Conference on Data Engineering*, Heidelberg, Germany, 2001, pp. 421–430.

[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 and U. Bankhofer, “Hot Deck Methods for Imputing Missing Data,” in *Machine Learning and Data Mining in Pattern Recognition*, vol. 7376, P. Perner, Ed. Springer Berlin Heidelberg, 2012, pp. 63–75.

[10] H. Kang, “The prevention and handling of the missing data,” *Korean J. Anesthesiol.*, 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,” *Commun. Methods Meas.*, vol. 5, no. 4, pp. 297–310, Oct. 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, “E. Fix and J.L. Hodges (1951): 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.