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](#_Toc48917141)

[Abstract 2](#_Toc48917142)

[目次 3](#_Toc48917143)

[表目次 5](#_Toc48917144)

[圖目次 6](#_Toc48917145)

[第 1 章 簡介 7](#_Toc48917146)

[第 2 章 相關研究 9](#_Toc48917147)

[2.1資料缺失類型 9](#_Toc48917148)

[2.2缺失值處理方法 9](#_Toc48917149)

[2.3填補法 10](#_Toc48917150)

[2.4 k鄰近點填補法 11](#_Toc48917151)

[第 3 章 問題與方法 13](#_Toc48917152)

[3.1研究動機 13](#_Toc48917153)

[3.2問題定義 13](#_Toc48917154)

[3.3問題分析 14](#_Toc48917155)

[3.4 sk-NN imputation 演算法 15](#_Toc48917156)

[3.5以原天際線評斷填補法的表現優劣 19](#_Toc48917157)

[第 4 章 實驗結果與分析 21](#_Toc48917158)

[4.1實驗環境 21](#_Toc48917159)

[4.1.1實驗平台 21](#_Toc48917160)

[4.1.2實驗資料來源 21](#_Toc48917161)

[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](#_Toc48917162)

[4.2.1實驗目的 22](#_Toc48917163)

[4.2.2實驗方法 22](#_Toc48917164)

[4.2.3實驗結果與分析 22](#_Toc48917165)

[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](#_Toc48917166)

[4.3.1實驗目的 25](#_Toc48917167)

[4.3.2實驗方法 25](#_Toc48917168)

[4.3.3實驗結果與分析 26](#_Toc48917169)

[4.4實驗結論 30](#_Toc48917170)

[第 5 章 結論與未來方向 31](#_Toc48917171)

[5.1結論 31](#_Toc48917172)

[5.2未來研究方向 31](#_Toc48917173)

[參考文獻 32](#_Toc48917174)

# 表目次

[表 3.1 sk-NN imputation演算法符號定義表 16](#_Toc48917175)

[表 4.1 UCI Machine Learning Repository輸入資料集資訊、來源與內容特徵[27] 21](#_Toc48917176)

[表 4.2 k=1各填補法比較表 27](#_Toc48917177)

[表 4.3 k=5各填補法比較表 28](#_Toc48917178)

[表 4.4 k=13各填補法比較表 29](#_Toc48917179)

# 圖目次

[圖 3.1 NaN-Euclidean distance 14](#_Toc48917180)

[圖 3.2 sk-NN imputation演算法 18](#_Toc48917181)

[圖 3.3 Procedure Impute\_Process() 19](#_Toc48917182)

[圖 4.1 k=1時hit ratio versus missing rate圖 23](#_Toc48917183)

[圖 4.2 k=2時hit ratio versus missing rate圖 24](#_Toc48917184)

[圖 4.3 k=3時hit ratio versus missing rate圖 24](#_Toc48917185)

[圖 4.4 k=4時hit ratio versus missing rate圖 25](#_Toc48917186)

[圖 4.5 k=1各填補法比較圖 27](#_Toc48917187)

[圖 4.6 k=5各填補法比較圖 28](#_Toc48917188)

[圖 4.7 k=13各填補法比較圖 29](#_Toc48917189)

# 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], [17]，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], [24]，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[25]，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式距離時，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。

為了驗證方法的有效性，本研究進行兩個模擬實驗，觀察缺失值比例在資料集中越大時，k值對skyline set的影響效果越低，表示參考更多的鄰近點未必具有更好的填補效果。並且發現在缺失值比例高達80%時，本研究的方法填補後，產生的近似天際線與原天際線的相似度仍然可以保持在50%，反觀原始k鄰近填補法填補後的近似天際線的相似度低於20%。顯示出原始k鄰近點填補法對高比例缺失值的不完整資料集填補效果不彰，本研究的方法大幅改善原始k鄰近點填補法的缺點，在面對不同缺失值比例時均具有良好的填補效果。

本論文後續的章節結構如下: 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

本論文相關研究有四個面向：資料缺失類型(types of missing value)、缺失值處理方法(missing value handling)、填補法(imputation)以及k鄰近點填補法(k-nearest neighbor imputation)。

## 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], [25]。

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，故此一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

缺失值處理可分為兩種策略，丟棄法(dropout)與填補法(imputation)[3], [6], [14]。

[刪除資料列怎麼做，刪除特徵欄怎麼做]

丟棄法可分為刪除資料列以及刪除特徵欄兩種方式。當某一筆資料中含有缺失值時，則將該筆資料列刪除。同理，當某一

[會考慮刪除資料列的原因，以及其缺點]

[會考慮刪除特徵欄位的原因，以及其缺點]

如果一個資料列裡面的欄位當中有缺失值，則將整筆資料列刪除，此種刪除方式可以在整個資料集中只含有少量缺失值時採用，使剩餘的資料成為無缺失值的完整資料集。丟棄法的缺點是，若缺失值佔資料集整體比例太高且是完全隨機缺失類型模型，則刪除資料列的方法會讓原資料集當中剩下堪用的資料筆數變得非常少，可參考之資料點也很少，致使最終填補法效果不彰。若整體資料集的某一維度或某一特徵欄位缺失筆數的數量太多，甚至遠多於其他特徵欄位缺失狀況時，則可選擇直接放棄該特徵欄位。遇缺失值時無論就上述採用刪除資料列或者是刪除特徵欄位，雖然可能不需要面對因具有缺失值所造成資料的不完整性，然而在另一方面，無論是丟棄資料列或是丟棄整個特徵欄位，仍然需要面臨因丟棄法所產生的額外問題 : 喪失資料表現特徵性質以及資料筆數來源不足。

填補法[3]是運用統計方法來計算缺失值所需要被填入的填補值，[介紹填補法的種類]~~最簡易補值的方法是對所有缺失值特徵欄位只賦予固定單一數值~~，例如可以取該特徵欄位的眾數、平均值、中位數、極大值、極小值等等[14]。若考量不影響整體輸入資料集分布、統計模型以及避免遺失了資料點可能在輸入資料集所表現的特性，參考資料集中非缺失的數值作為缺失值填補依據比較具有可行性且預期會有比較好的效果(即接近完整資料集的特徵)。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上與天際線查詢演算法的適用性上，填補法會比丟棄法來得更適合。

(加薪)

當資料缺失的表現是什麼樣的時候採用丟棄法，而丟棄法分為刪除資料列及刪除特徵欄兩種方式，當資料列的缺時欄位較多時採用刪除資料列，當資料欄缺失值較多採用刪除特徵欄。說明優缺點。

而填補法的使用時機是在資料如何表現時…

## 2.3填補法

眾多缺失值填補法當中又可以[XXX]區分為單一填補法(single imputation)與多重填補法(multiple imputations，MI)兩種[1], [6], [21]。 [要區分好單一填補法與多重填補法定義的差異]

[說明單一填補法定義，以及說明有甚麼填補法屬於單一填補法]

單一填補法[3], [14]主要目的是填補某一特定欄位中遺失的值。亦即使計算出可能具有不只一種可參考解，也會依照不同演算法的機制來挑選其中一個值作為最後填補該缺失欄位。其中屬於單一填補法的有熱卡填補法(hot deck imputation)[9]、冷卡填補法(cold deck imputation)[20]、平均值填補法(mean imputation or mean substitution)、迴歸填補法(regression imputation)[10]

[屬於單一填補法的種類]

熱卡填補法會，另一種形式形容熱卡填補法為last observation carried forward (LOCF)，其中根據任何資料集中的變數做排序，

冷卡填補法

平均值填補法，

迴歸填補法

當缺失值佔資料集比例很小時，採取單一填補法的確是一種簡易卻又實用的方法。因為單一填補法計算容易，所以也最常被用來當作填補缺失值的方法之一。必須注意的一點是，單一填補法在面臨一定程度的缺失情形時，可能會面臨到一個嚴重的問題。一旦被填補進缺失欄位後，便會被當作真正的資料值而無法再分辨出原始資料集的值與被填補值之間的真偽。此一疑慮甚至可能造成日後分析結果時被誤導。

[說明多重填補法]

多重填補法首次在1978年被Rubin提出來[19]，更著重於分析與解決問題上。多重填補法仰賴於資料集上模擬分布模型[18]，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有大量的計算需求。

## 2.4 k鄰近點填補法

k鄰近點填補法(k-Nearest Neighbor imputation，k-NN imputation)[8], [16]是一個很實用的主流填補缺失值方法。k鄰近點填補法的核心概念是在多維度空間資料集中對某個具有缺失值的點p，找尋k個與p鄰近的點作為填補點p的參考值。k鄰近點填補法可以被應用於連續型資料(continuous data)上，也可以被應用於離散型資料(discrete data)、有序型資料(ordinal)甚至是分類型資料(categorical data)[26]，幾乎均適用在各種的資料種類[25]。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。

k鄰近點填補法的優點是比單一填補值，例如平均數、中位數、極值、或是眾數等填補法準確許多，原因是該方法會同時參照其他與該缺失值相鄰點去預測其應該原有的合理值；尤其是在尋找天際線時，單一值填補法對於尋找天際線不會有更好的幫助，反而會因此增加許多不必要的計算在比較相同維度上具有相同值(因缺失而被填補回去的相同值)。類似這樣的問題在無論考慮單一維度或整體資料集缺失值密度，當missing rate愈趨增加時，其餘具有完整資料值的點也會越來越少。

[kNN imputation 缺點]

縱然如此，k鄰近點填補法也存在著某些潛在的缺點。身為非監督式填補法，它會參考其他非缺失值，換句話說，此類型填補法並不會對資料內容做任何過濾或預處理，對於已存在的資料極為敏感，很容易受輸入資料集之中非缺失值資料的影響，導致可能無法精準地將資料集內的原缺失部分正確地填補回去。

k鄰近點填補法另外一個缺點是，針對資料量大而且高維度的巨量資料，k鄰近點填補法必須先行儲存整個欲參考的鄰近點於記憶體中，並且逐一計算求出各點兩兩之距離，使得距離計算量隨著資料點的增加成指數成長。對某一個缺失值資料點p而言，尋找可參考的鄰近點過程中，k鄰近點填補法考量了所有點q與點p的距離。依距離值由小到大排序，與點p所有鄰近點之中從第一個至第k個相鄰近點都具有相同的權重值[15]，但相同權重值這點卻與k鄰近點填補法中，希望找出最鄰近點作為最有參考價值填補該缺失值的概念相違背[25]。

# 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

尋找天際線時需要去比較所有特徵欄位的值，也就是說每一筆資料的每一個特徵欄位都必須存在有值，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，期待蒐集到的每一個特徵欄位都不能有缺失值似乎是不切實際的。

為了保證資料集的完整性，遇到缺失值的時候最直覺採用的策略為丟棄法。丟棄法又有兩種刪除資料的方式，分別為刪除缺失值所在的資料列以及刪除缺失值所在的特徵欄位。隨著不完整資料集中缺失值的比例越來越高，刪除缺失值所在的資料列會讓所剩資料數量不足，而刪除缺失值所在的特徵欄位則會喪失原資料集所表現的特徵。因此本研究認為不應該在具有大量缺失值的資料集當中，直接將資料列或特徵欄位刪除。反之運用填補法來填補適合的值，可以同時解決上述兩個問題，也是為何填補法比丟棄法更被多數研究採納的原因。

針對在不完整資料集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。從填補後的結果來看，k鄰近點填補法是眾多填補法當中表現比較好的[22]。k鄰近點填補法會先蒐集缺失值鄰近點的特徵值，再計算這些特徵值之算術平均作為填補值，這使得被填補值相較於以單一數值的填補法更具有參考性。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資料點的歐式距離會導致找錯鄰近點，其二是k鄰近點填補法遇到可參考的鄰近點不足時，會簡化為填補單一值[7]。因此本研究針對這些問題分別提出差別權重分配與新的選擇鄰近點機制以改善k鄰近填補法的缺點。

## 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2.2.1節中提到的完全隨機缺失類型(MCAR)，表示缺失值與各欄位毫無相關性。我們將填補後的完整資料集，再計算求出近似天際線，並與原先無缺失資料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。若相似程度越高，則該填補法的填補效果越好。

## 3.3問題分析

所有計算距離公式中[8], [16]，歐式距離的計算方法[13]是採資料集中兩兩資料點相對應維度的差平方再取平方根。若是至少一個維度具有缺失值，則在計算歐式距離時並不會採計具有缺失值的數值差的平方和，此計算方式是最廣為主流的算法[22]。本論文不打算更改這種計算方式，但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。



圖 3.1 NaN-Euclidean distance

今舉例說明。如圖3.1所示A、B、C三點座標分別為(1, 1)、(2, 8)、(3, 3)，在沒有任何缺失值下，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座標設為缺失表示缺失值時，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。

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(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，更容易形成有如該欄位直接被刪除一樣無意義地比較的結果。

鑒於以上分析，本論文提出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，並且在缺失值比例升高時更能夠一定程度地解決原始k鄰近點填補法的缺點。我們對有缺失值的維度其餘非缺失值做採樣後取平均值，目的在於不讓缺失值的距離計算導致填補後找尋到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 含有若干個缺失值與屬於自然數的常數 作為限制可參考鄰近點的上限個數，兩者皆為演算法的輸入參數。其中n與m分別為 的資料列個數與 的維度個數。 表示 中第 筆資料列，且 為該資料列 於維度 的值。

為不完整資料集 中記錄任兩資料列 與 之間距離的距離矩陣，且其中 為任兩資料列 與 之歐氏距離。 是一個記錄資料點 在每一個維度 的值 相對其他任意相異資料點 的權重矩陣， 則表示兩資料點 與 之間的權重值。 為一個包含 與 兩種類別的類別型集合，其目的是用來決定計算某一缺失值的被參考維度值 的權重值該如何分配。若 設為 ，則所有權重值 均被設為1 ; 若 設為 ，則 會依照 中任兩資料列 與 之間距離 的倒數作為該權重值。

NN list蒐集已排序串列 。每一個 內記錄所有資料點 按照與其相距的歐氏距離 由小至大排序好後，記錄該順序所對應鄰近點的index。 代表 所有相鄰近點中，經計算歐氏距離後 不為0的且與 第j個最接近資料點的index。

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，演算法輸出一個已被填補所有缺失值後的資料集 。

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 j 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。step5列出每一筆資料點其所有鄰近點，並且依照所有鄰近點與該資料點的歐式距離值，由小到大排序。step6遍歷輸入資料集中所有點並且針對找到的每一個缺失值各別依照其鄰近點填補新值，一旦找到某一資料點的某一個維度含有缺失值，便會執行副程式Impute\_Process()如圖3.3所示。step7則回傳填補後的新完整資料集。

|  |
| --- |
| 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, j, k)  **end if**  **end for**  step 7. **return** an imputed data set  } |

圖 3.2 sk-NN imputation演算法

|  |
| --- |
| Procedure Impute\_Process(i, j, k) {  Input : row index i, column index j, nearest neighbor k  Output : imputed value at index (i, j)  Method:  step 1. 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 j  **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[23]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[27]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。

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，呈現於表4.1。

表 4.1 UCI Machine Learning Repository輸入資料集資訊、來源與內容特徵[27]

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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，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

本實驗方法為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，本論文採用原完整資料集中所得出天際線作為最終填補效果的依據。

### 4.2.3實驗結果與分析

根據圖4.1實驗一顯示結果，隨著缺失比率在資料集當中增加，原始k鄰近點填補法的準確率並沒有因為找尋更多的鄰近點數量改善填補效果。

由圖4.2觀察可知，原始k鄰近點填補法只著重在將鄰近參考點的數量逐漸地增加，但因為缺失比例也逐漸增加，再配合原始k鄰近點填補法中對可供參考點不足從缺的機制，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嚴重不足。



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



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



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



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

## 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missing rate程度下，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k值最大範圍可以到17，故分別取三種不同k值分別做三次比較，以觀察原始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skyline set的程度。x軸為缺失值佔整體資料集當中的比例，y軸為最填補所有缺失值之後，再分別跑同一支尋找skyline query algorithm的程式，並與缺失前的原skyline set做比較計算出相似程度。若越接近原skyline set則y軸的值越接近1.0。

### 4.3.3實驗結果與分析

實驗二結果顯示出，本實驗方法雖然在某些k值不大情形下準確度會略差，如表4.2與圖4.4中，當k=5且缺失比率0.3到0.55間，本論文方法準確度有下降，此時原始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下，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。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比例不高時會比較容易啟動，但隨著缺失比率增加下，k值的增加所帶來的益處會越不明顯，而此時本論文中採樣的方式反而可以起到更大機會能有效的找到鄰近參考點。

從表4.1與4.2中可以看出原始k鄰近點填補法大約從缺失比率30%時準確度就開始急遽下降，雖然中間可能有小幅度的上升，但可看出下降程度在75%以後下降幅度又更加嚴重，表4.3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。

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 |
| kNN | 0.8 | 0.7 | 0.538 | 0.6 | 0.421 | 0.5 | 0.384 | 0.25 | 0.1 |
| weighted-kNN | 0.8 | 0.7 | 0.538 | 0.6 | 0.421 | 0.5 | 0.285 | 0.25 | 0.1 |
| skNN | 0.8 | 0.8 | 0.818 | 0.909 | 0.636 | 0.75 | 0.7 | 0.636 | 0.545 |

圖 4.5 k=1各填補法比較圖

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 |
| kNN | 0.9 | 0.666 | 0.666 | 1 | 0.625 | 0.533 | 0.6 | 0.454 | 0.181 |
| weighted-kNN | 0.9 | 0.625 | 0.6 | 1 | 0.529 | 0.692 | 0.666 | 0.1 | 0.25 |
| skNN | 0.9 | 0.833 | 0.909 | 0.818 | 0.5 | 0.8 | 0.6 | 0.5 | 0.363 |

圖 4.6 k=5各填補法比較圖

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 |
| kNN | 0.9 | 1 | 0.75 | 0.333 | 0.727 | 0.583 | 0.615 | 0.384 | 0.181 |
| weighted-kNN | 0.9 | 0.833 | 0.818 | 0.4 | 0.277 | 0.529 | 0.428 | 0.352 | 0.181 |
| skNN | 0.9 | 1 | 0.833 | 0.454 | 0.818 | 0.7 | 0.727 | 0.75 | 0.7 |

圖 4.7 k=13各填補法比較圖

## 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。

1. 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。

1. 遇到缺失值計算距離的機制，當兩兩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] W. Ren, X. Lian, and K. Ghazinour, “Skyline Queries over Incomplete Data Streams,” *The VLDB Journal*, vol. 28, no. 6, pp. 961–985, 2019.

[18] P. Royston, “Multiple Imputation of Missing Values,” *The Stata Journal*, vol. 4, no. 3, pp. 227–241, 2004.

[19] 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.

[20] J. Shao, “Cold deck and ratio imputation,” *Survey Methodology*, vol. 26, no. 1, pp. 79–86, 2000.

[21] 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.

[22] 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.

[23] J. Van Hulse and T. M. Khoshgoftaar, “Incomplete-Case Nearest Neighbor Imputation in Software Measurement Data,” *Information Sciences*, vol. 259, pp. 596–610, 2014.

[24] 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.

[25] 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.

[26] S. Zhang, “Nearest Neighbor Selection for Iteratively kNN Imputation,” *Journal of Systems and Software*, vol. 85, no. 11, pp. 2541–2552, 2012.

[27] “UCI Machine Learning Repository,” 2013. [Online]. Available: https://archive.ics.uci.edu/ml/index.php.