【聚類】五種主要聚類算法

新機器視覺 前天

點擊下方
視覺/圖像重磅乾貨，第一時間送達

新機器視覺
最前沿的機器視覺與計算機視覺技術
206篇原創內容

公眾號

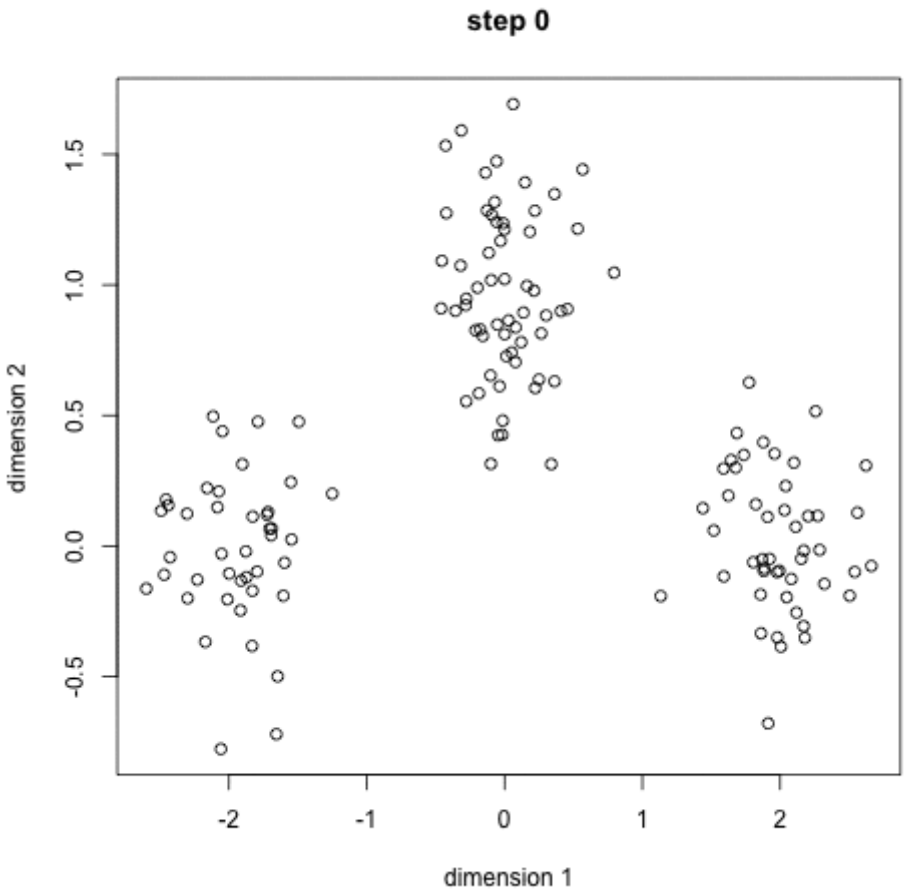
轉自 |

聚類是一種機器學習技術，它涉及到數據點的分組。給定一組數據點，我們可以使用聚類算法將每個數據點劃分為一個特定的組。理論上，同一組中的數據點應該具有相似的屬性和/或特徵，而不同組中的數據點應該具有高度不同的屬性和/或特徵。聚類是一種無監督學習的方法，是許多領域中常用的統計數據分析技術。

在數據科學中，我們可以使用聚類分析從我們的數據中獲得一些有價值的見解。在這篇文章中，我們將研究5種流行的聚類算法以及它們的優缺點。

K-MEANS聚類算法

K-Means聚類算法可能是大家最熟悉的聚類算法。它出現在很多介紹性的數據科學和機器學習課程中。在代碼中很容易理解和實現！請看下面的圖表。



K-Means聚類

1.首先，我們選擇一些類/組來使用並隨機地初始化它們各自的中心點。要想知道要使用的類的數量，最好快速地查看一下數據，並嘗試識別任何不同的分組。中心點是與每個數據點向量相同長度的向量，在上面的圖形中是“X”。

2.每個數據點通過計算點和每個組中心之間的距離進行分類，然後將這個點分類為最接近它的組。

3.基於這些分類點，我們通過取組中所有向量的均值來重新計算組中心。

4.對一組迭代重複這些步驟。你還可以選擇隨機初始化組中心幾次，然後選擇那些看起來對它提供了最好結果的來運行。

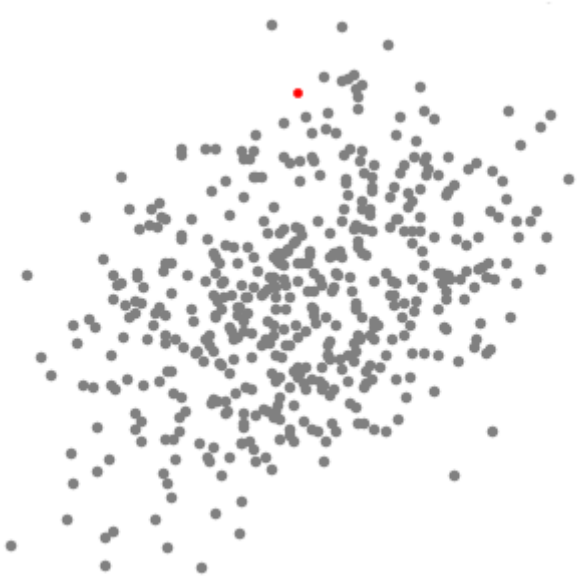
K-Means聚類算法的優勢在於它的速度非常快，因為我們所做的只是計算點和群中心之間的距離;它有一個線性複雜度O(n)。

另一方面，K-Means也有幾個缺點。首先，你必須選擇有多少組/類。這並不是不重要的事，理想情況下，我們希望它能幫我們解決這些問題，因為它的關鍵在於從數據中獲得一些啟示。K-Means也從隨機選擇的聚類中心開始，因此在不同的算法運行中可能產生不同的聚類結果。因此，結果可能是不可重複的，並且缺乏一致性。其他聚類方法更加一致。

K-Medians是另一種與K-Means有關的聚類算法，除了使用均值的中間值來重新計算組中心點以外，這種方法對離群值的敏感度較低（因為使用中值），但對於較大的數據集來說，它要慢得多，因為在計算中值向量時，每次迭代都需要進行排序。

均值偏移聚類算法

均值偏移（Mean shift）聚類算法是一種基於滑動窗口（sliding-window）的算法，它試圖找到密集的數據點。而且，它還是一種基於中心的算法，它的目標是定位每一組群/類的中心點，通過更新中心點的候選點來實現滑動窗口中的點的平均值。這些候選窗口在後期處理階段被過濾，以消除幾乎重複的部分，形成最後一組中心點及其對應的組。請看下面的圖表。



單滑動窗口的均值偏移聚類

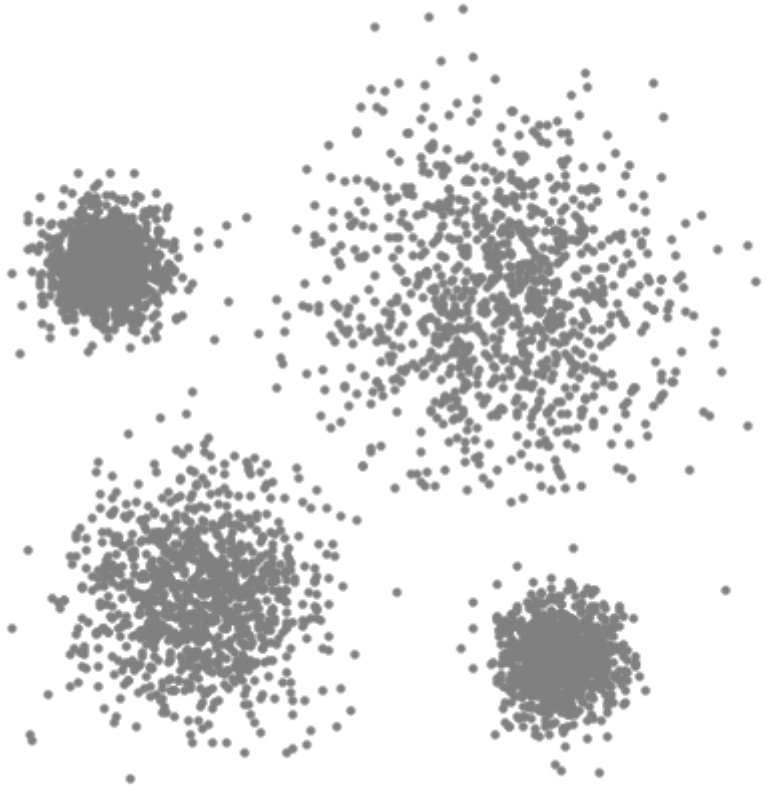
1.為了解釋這一變化，我們將考慮二維空間中的一組點（就像上面的例子）。我們從一個以點C（隨機選擇）為中心的圓形滑窗開始，以半徑r為內核。均值偏移是一種爬山算法（hill climbing algorithm），它需要在每個步驟中反复地將這個內核移動到一個更高的密度區域，直到收斂。

2.在每一次迭代中，滑動窗口會移向密度較高的區域，將中心點移動到窗口內的點的平均值（因此得名）。滑動窗口中的密度與它內部的點的數量成比例。自然地，通過移向窗口中點的平均值，它將逐漸向更高的點密度方向移動。

3.我們繼續根據均值移動滑動窗口，直到沒有方向移動可以容納內核中的更多點。看看上面的圖表;我們一直在移動這個圓，直到我們不再增加密度（也就是窗口中的點數）。

4.步驟1到3的過程是用許多滑動窗口完成的，直到所有的點都位於一個窗口內。當多個滑動窗口重疊的時候，包含最多點的窗口會被保留。然後，數據點根據它們所在的滑動窗口聚類。

下面展示了從端到端所有滑動窗口的整個過程的演示。每個黑點代表一個滑動窗口的質心，每個灰色點都是一個數據點。



均值偏移聚類的整個過程

與K-Means聚類相比，均值偏移不需要選擇聚類的數量，因為它會自動地發現這一點。這是一個巨大的優勢。聚類中心收斂於最大密度點的事實也是非常可取的，因為它非常直觀地理解並適合於一種自然數據驅動。缺點是選擇窗口大小/半徑r是非常關鍵的，所以不能疏忽。

DBSCAN聚類算法

DBSCAN(Density-Based Spatial Clustering of Applications with Noise)是一個比較有代表性的基於密度的聚類算法，類似於均值轉移聚類算法，但它有幾個顯著的優點。

DBSCAN笑臉聚類

1.DBSCAN以一個從未訪問過的任意起始數據點開始。這個點的鄰域是用距離ε（所有在ε距離的點都是鄰點）來提取的。

2.如果在這個鄰域中有足夠數量的點（根據minPoints），那麼聚類過程就開始了，並且當前的數據點成為新聚類中的第一個點。否則，該點將被標記為噪聲（稍後這個噪聲點可能會成為聚類的一部分）。在這兩種情況下，這一點都被標記為“訪問（visited）”。

3.對於新聚類中的第一個點，其ε距離附近的點也會成為同一聚類的一部分。這一過程使在ε鄰近的所有點都屬於同一個聚類，然後重複所有剛剛添加到聚類組的新點。

4.步驟2和步驟3的過程將重複，直到聚類中的所有點都被確定，就是說在聚類附近的所有點都已被訪問和標記。

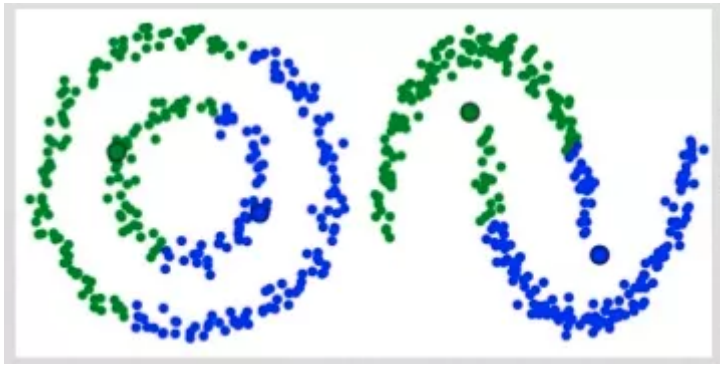
5.一旦我們完成了當前的聚類，就會檢索並處理一個新的未訪問點，這將導致進一步的聚類或噪聲的發現。這個過程不斷地重複，直到所有的點被標記為訪問。因為在所有的點都被訪問過之後，每一個點都被標記為屬於一個聚類或者是噪音。

DBSCAN比其他聚類算法有一些優勢。首先，它不需要一個預設定的聚類數量。它還將異常值識別為噪聲，而不像均值偏移聚類算法，即使數據點非常不同，它也會將它們放入一個聚類中。此外，它還能很好地找到任意大小和任意形狀的聚類。

DBSCAN的主要缺點是，當聚類具有不同的密度時，它的性能不像其他聚類算法那樣好。這是因為當密度變化時，距離閾值ε和識別鄰近點的minPoints的設置會隨著聚類的不同而變化。這種缺點也會出現在非常高維的數據中，因為距離閾值ε變得難以估計。

使用高斯混合模型（GMM）的期望最大化（EM）聚類

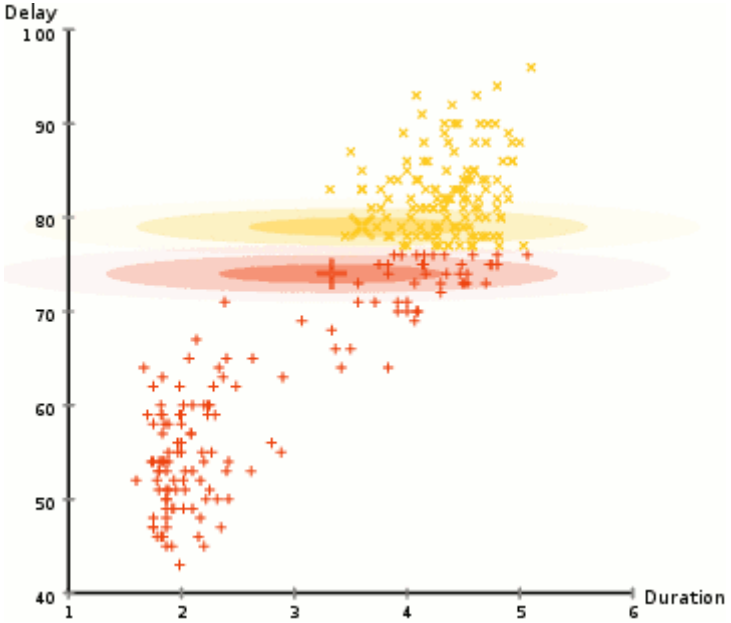
K-Means的一個主要缺點是它對聚類中心的平均值的使用很簡單幼稚。我們可以通過看下面的圖片來了解為什麼這不是最好的方法。在左邊看起來很明顯的是，有兩個圓形的聚類，不同的半徑以相同的平均值為中心。K-Means無法處理，因為聚類的均值非常接近。在聚類不是循環的情況下，K-Means也會失敗，這也是使用均值作為聚類中心的結果。



K-Means的兩個失敗案例

高斯混合模型（GMMs）比K-Means更具靈活性。使用高斯混合模型，我們可以假設數據點是高斯分佈的;比起說它們是循環的，這是一個不那麼嚴格的假設。這樣，我們就有兩個參數來描述聚類的形狀：平均值和標準差！以二維的例子為例，這意味著聚類可以採用任何形式的橢圓形狀（因為在x和y方向上都有標準差）。因此，每個高斯分佈可歸屬於一個單獨的聚類。

為了找到每個聚類的高斯分佈的參數（例如平均值和標準差）我們將使用一種叫做期望最大化（EM）的優化算法。看看下面的圖表，就可以看到高斯混合模型是被擬合到聚類上的。然後，我們可以繼續進行期望的過程——使用高斯混合模型實現最大化聚類。



使用高斯混合模型來期望最大化聚類

1.我們首先選擇聚類的數量（如K-Means所做的那樣），然後隨機初始化每個聚類的高斯分佈參數。通過快速查看數據，可以嘗試為初始參數提供良好的猜測。注意，在上面的圖表中可以看到，這並不是100%的必要，因為高斯開始時的表現非常不好，但是很快就被優化了。

2.給定每個聚類的高斯分佈，計算每個數據點屬於特定聚類的概率。一個點離高斯中心越近，它就越有可能屬於那個聚類。這應該是很直觀的，因為有一個高斯分佈，我們假設大部分的數據都離聚類中心很近。

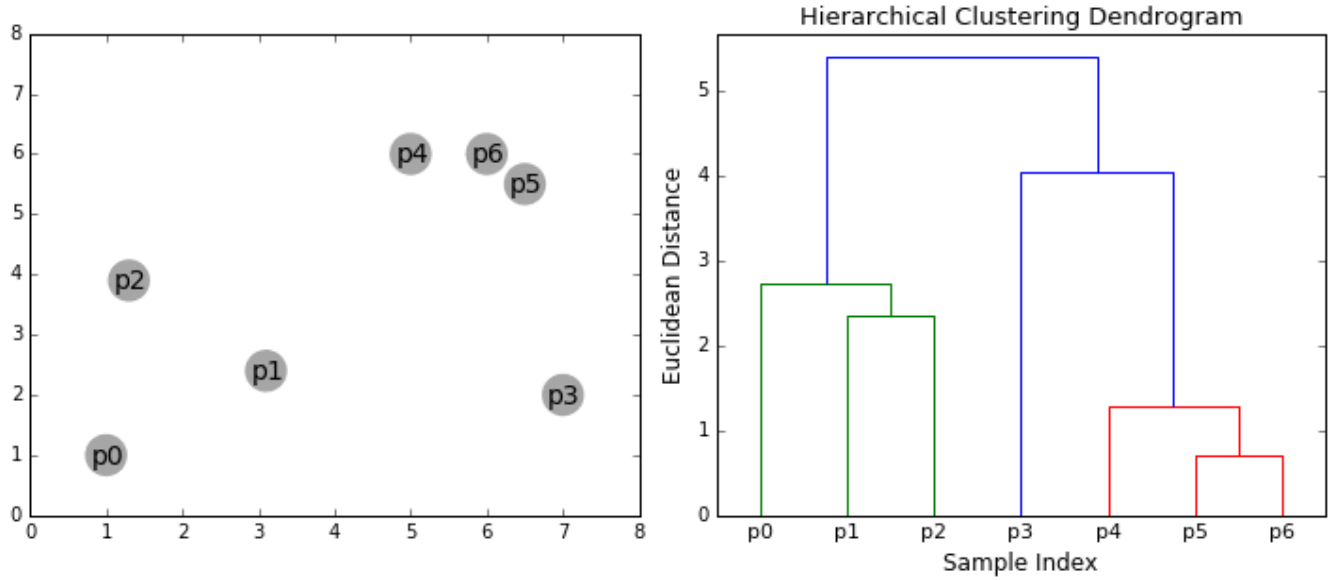
3.基於這些概率，我們為高斯分佈計算一組新的參數，這樣我們就能最大程度地利用聚類中的數據點的概率。我們使用數據點位置的加權和來計算這些新參數，權重是屬於該特定聚類的數據點的概率。為了解釋這一點，我們可以看一下上面的圖，特別是黃色的聚類作為例子。分佈在第一次迭代中是隨機的，但是我們可以看到大多數的黃色點都在這個分佈的右邊。當我們計算一個由概率加權的和，即使在中心附近有一些點，它們中的大部分都在右邊。因此，自然分佈的均值更接近於這些點。我們還可以看到，大多數點都是“從右上角到左下角”。因此，標準差的變化是為了創造一個更符合這些點的橢圓，從而使概率的總和最大化。

步驟2和3被迭代地重複，直到收斂，在那裡，分佈不會從迭代到迭代這個過程中變化很多。

使用高斯混合模型有兩個關鍵的優勢。首先，高斯混合模型在聚類協方差方面比K-Means要靈活得多;根據標準差參數，聚類可以採用任何橢圓形狀，而不是局限於圓形。K-Means實際上是高斯混合模型的一個特例，每個聚類在所有維度上的協方差都接近0。其次，根據高斯混合模型的使用概率，每個數據點可以有多個聚類。因此，如果一個數據點位於兩個重疊的聚類的中間，通過說X%屬於1類，而y%屬於2類，我們可以簡單地定義它的類。

層次聚類算法

層次聚類算法實際上分為兩類：自上而下或自下而上。自下而上的算法在一開始就將每個數據點視為一個單一的聚類，然後依次合併（或聚集）類，直到所有類合併成一個包含所有數據點的單一聚類。因此，自下而上的層次聚類稱為合成聚類或HAC。聚類的層次結構用一棵樹（或樹狀圖）表示。樹的根是收集所有樣本的唯一聚類，而葉子是只有一個樣本的聚類。在繼續學習算法步驟之前，先查看下面的圖表。



合成聚類

1.我們首先將每個數據點作為一個單獨的聚類進行處理。如果我們的數據集有X個數據點，那麼我們就有了X個聚類。然後我們選擇一個度量兩個聚類之間距離的距離度量。作為一個示例，我們將使用平均連接（average linkage）聚類，它定義了兩個聚類之間的距離，即第一個聚類中的數據點和第二個聚類中的數據點之間的平均距離。

2.在每次迭代中，我們將兩個聚類合併為一個。將兩個聚類合併為具有最小平均連接的組。比如說根據我們選擇的距離度量，這兩個聚類之間的距離最小，因此是最相似的，應該組合在一起。

3.重複步驟2直到我們到達樹的根。我們只有一個包含所有數據點的聚類。通過這種方式，我們可以選擇最終需要多少個聚類，只需選擇何時停止合併聚類，也就是我們停止建造這棵樹的時候！

層次聚類算法不要求我們指定聚類的數量，我們甚至可以選擇哪個聚類看起來最好。此外，該算法對距離度量的選擇不敏感;它們的工作方式都很好，而對於其他聚類算法，距離度量的選擇是至關重要的。層次聚類方法的一個特別好的用例是，當底層數據具有層次結構時，你可以恢復層次結構;而其他的聚類算法無法做到這一點。層次聚類的優點是以低效率為代價的，因為它具有$O(n^3)$的時間複雜度，與K-Means和高斯混合模型的線性複雜度不同。

—版權聲明—

僅用於學術分享，版權屬於原作者。

若有侵權，請聯繫微信號:yiyang-sy 刪除或修改！

—THE END—

走进新机器视觉 · 拥抱机器视觉新时代

新机器视觉 —— 机器视觉领域服务平台

媒体论坛/智库咨询/投资孵化/技术服务

商务合作 ：

投稿咨询 ：

产品采购 ：

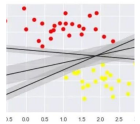
（微信号）

长按扫描右侧二维码关注“新机器视觉”公众号

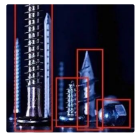
新机器视觉
New machine vision

喜歡此內容的人還喜歡

優雅的讀懂支持向量機SVM 算法
新機器視覺



深入了解目標檢測深度學習算法的技術細節
新機器視覺



收藏| 各種Optimizer 梯度下降優化算法回顧和總結
新機器視覺