IDMC2008
11-12 NOV
Amir Kabir University

الگوریتمی جهت یافتن تناظر بین خوشه ها در خوشه بندی های متفاوت به منظور استفاده در خوشه بندی توافقی

حامد زجاجی[۱]، محمد علیشاهی[۲]، بهارک شاکری اسکی[۳]، حسین دلداری[۴]

چکیده

خوشه بندی را می توان یکی از مهمترین مراحل در تحلیل داده ها برشمرد. روش های خوشه بندی بسیاری تاکنون توسعه داده شده‌اند. یکی از این روش ها که در مطالعات اخیر مطرح شده و مورد توجه قرار گرفته است، خوشه بندی توافقی می باشد. این روش بر روی مجموعه ای از خوشه بندی ها عمل می کند. هدف خوشه بندی توافقی، بدست آوردن یک خوشه بندی نهایی است به گونه ای که در آن خوشه ها از کیفیت و پایداری بالاتری نسبت به خوشه بندی های انجام شده اولیه برخوردار باشند. طبق مطالعاتی که انجام داده ایم الگوریتم های خوشه بندی توافقی، خوشه های هم شماره در خوشه بندی های مختلف را متناظر در نظر می گیرند. این در حالی است که در عمل ممکن است خوشه ای با یک شماره مشخص بسیار متفاوت از خوشه ای با همان شماره در خوشه بندی دیگر باشد و این می تواند در پائین آوردن کیفیت خروجی روش تاثیر منفی شدیدی بگذارد. الگوریتمی که در این مقاله پیشنهاد شده است، روشی را جهت تعیین خوشه های متناظر در خوشه بندی های مختلف ارائه می کند. در حقیقت بهتر است الگوریتم ارائه شده را به عنوان یک مرحله پیش پردازش، جهت انجام خوشه بندی توافقی محسوب کنیم. خروجی الگوریتم پیشنهادی در این مقاله، یک ماتریس تناظر بین خوشه‌ها در خوشه بندی های مختلف است که به عنوان ورودی در خوشه بندی توافقی مورد استفاده قرار می گیرد.

کلمات کلیدی

داده کاوی، خوشه بندی توافقی، ماتریس تناظر

An Algorithm to Find Correspondence Clusters in Different Clusterings for Using in Consensus Clustering

Hamed Zojaji[1], Mohammad Alishahi[2], Bahark Shakeri Aski[3], Hosein Deldari[4]

Abstract

Clustering is one of the most important phases of data analysis. Among lots of clustering methods which are developed, consensus clustering is one of the most interested and latest methods. In this method a group of clusterings are processed together in order to obtain a more qualified clustering which has more stability. In consensus clustering algorithms clusters of different clusterings with same labels are considered correspondent. In practice correspondent clusters in different clusterings may differ extremely and this may reduce the quality. In this paper we propose an algorithm which establishes

[1] دانشجوی کارشناسی ارشد گرایش نرم افزار، دانشگاه آزاد اسلامی واحد مشهد hzojaji@gmail.com
[2] دانشجوی کارشناسی ارشد گرایش نرم افزار، دانشگاه آزاد اسلامی واحد مشهد alishahi@ymail.com
[3] دانشجوی کارشناسی ارشد گرایش نرم افزار، دانشگاه آزاد اسلامی واحد مشهد shakeriaski.b@gmail.com
[4] دانشیار، دانشکده فنی و مهندسی دانشگاه فردوسی مشهد hd@ferdowsi.um.ac.ir

correspondence between the clusters of the clusterings. In other word this algorithm is a preprocess for consensus clustering and the output of this algorithm is a correspondence matrix.

Keywords
Data Mining, Consensus Clustering, Correspondence Matrix

1. مقدمه

خوشه بندی اغلب به عنوان اولین و یکی از مهترین گام ها در تحلیل داده می باشد. تاکنون الگوریتم های خوشه بندی بسیاری توسعه یافته است [8،9]، نظیر خوشه بندی سلسله مراتبی[1]، خوشه بندی افرازی[2]، خوشه بندی بر مبنای تراکم[3] و روش هایی که اجتماعی از الگوریتم های خوشه بندی را مورد استفاده قرار می دهند. اغلب روش های خوشه بندی، مطابق با معیار های مشخصی که برای خوشه بندی استفاده می کنند، بر روی یافتن یک پاسخ بهینه و یا نزدیک به پاسخ بهینه متمرکز می شوند. خوشه بندی توافقی[4] مزایایی بیش از آنچه که یک الگوریتم خوشه بندی می تواند به آن دست یابد را فراهم می آورد. در این نوع خوشه بندی، نتایج حاصل از چندین خوشه بندی با هم ترکیب می شوند تا در نهایت به یک خوشه بندی واحد دست یابیم. الگوریتم های خوشه بندی توافقی اغلب: خوشه بندی های بهتری تولید می کنند؛ خوشه بندی ترکیب شده ای را می یابند که به تنهایی توسط هر الگوریتم خوشه بندی دیگری قابل تولید نمی باشد؛ حساسیت کمتری نسبت به نویز دارند؛ و قادر به یکپارچه سازی نتایج از منابع توزیع شده می‌باشند [7]. این روش خوشه بندی کاربرد های گوناگونی در زمینه های مختلف از قبیل یادگیری ماشینی [11،3]، تشخیص الگو [1]، بیوانفورماتیک [10] و داده کاوی [5،4] دارد.

علاوه بر مزایای ذکر شده، استفاده از خوشه بندی توافقی می تواند در کاربرد های دیگری نیز مفید واقع شود. به عنوان مثال می توان به 1)خوشه‌بندی داده‌های اسمی[5] مانند فیلد نام خانوادگی کارمند در بانک اطلاعات پرسنلی، 2) خوشه بندی داده های ناهمگون، 3) کار با داده های ناقص[6]، 4) تعیین تعداد مناسب خوشه ها در انجام خوشه بندی، 5) تشخیص داده های دور افتاده[7] و 6) خوشه بندی با حفظ محرمانگی[8] [2] اشاره‌نمود.

الگوریتم های خوشه بندی توافقی جهت یافتن خوشه بندی نهایی، خوشه های هم شماره در خوشه بندی های مختلف را متناظر در نظر می گیرند، در حالیکه در خوشه بندی های موجود با توجه به نوع الگوریتم، پارامتر های الگوریتم و یا نوع داده ها این مسئله همیشه صادق نمی باشد. به منظور رفع این مسئله و عملیاتی تر نمودن روش های خوشه بندی توافقی باید تناظر بین خوشه ها در خوشه بندی های مختلف را تشخیص دهیم.

روش های موجود جهت اندازه گیری تشابه بین دو خوشه، در روش خوشه بندی سلسله مراتبی به منظور ترکیب خوشه ها، مورد استفاده قرار می‌گیرند. اما این روش ها تنها در شرایطی میزان فاصله و یا میزان شباهت دو خوشه را اندازه گیری می کنند که هر دو خوشه در یک خوشه بندی قرار داشته باشند.

هدف ما در این مقاله ارائه روشی جهت یافتن خوشه های مشابه و یا متناظر در خوشه بندی های مختلف می باشد. لازم به ذکر است که منظور از خوشه بندی ها مختلف، نتایج خوشه بندی های انجام شده بر روی یک مجموعه داده ای می باشد. این مجموعه داده می تواندبه صورت عمودی یا افقی توزیع شده و یا اینکه متمرکز باشد. ساختار ادامه مقاله به این صورت می باشد: در بخش 2 به بررسی فعالیت ها مشابه در اندازه گیری فاصله دو خوشه و تشریح روش خوشه بندی توافقی می پردازیم؛ در بخش 3 الگوریتم های پیشنهادی را مورد بررسی قرار خواهیم داد؛ در بخش 4 پیچیدگی الگوریتم تحلیل می شود؛ در بخش 5 به بررسی نتایج اجرا الگوریتم بر روی یک مجموعه داده ای خواهیم پرداخت؛ بخش 6 نیز به نتیجه گیری اختصاص یافته است.

2. پیشینه کاوی پژوهش

در این بخش ابتدا در زمینه روش های اندازه گیری فاصله، به معرفی روش هایی جهت تشخیص فاصله دو خوشه می پردازیم و سپس روال کلی خوشه بندی توافقی را جهت درک مناسب از الگوریتم پیشنهادی مورد بررسی قرار خواهیم داد.

1.2. روش های اندازه گیری فاصله

روش های موجود جهت اندازه گیری میزان تشابه بین دو خوشه در الگوریتم های خوشه بندی سلسله مراتبی مورد استفاده قرار می گیرند. خوشه‌بندی سلسله مراتبی به دو صورت تقسیم کننده[9] یا تجمیع کننده[10] انجام می شود [6]. در این روش خوشه بندی، به اندازه گیری میزان فاصله و یا میزان شباهت دو خوشه نیاز است تا بتوان یک خوشه را به دو خوشه مستقل تقسیم نمود (در حالت تقسیم کننده) و یا اینکه دو خوشه مشابه را به یک خوشه واحد تبدیل کرد (در حالت تجمیع کننده).

چهار روش عمده جهت اندازه گیری میزان فاصله بین دو خوشه وجود دارد، این روش ها در روابط (1) تا (4) آمده اند [6]. $|p-p'|$ فاصله بین دو شئ p و p'، m_i مرکز خوشه‌ی C_i، n_i تعداد اشیاء در خوشه‌ی C_i می باشد.

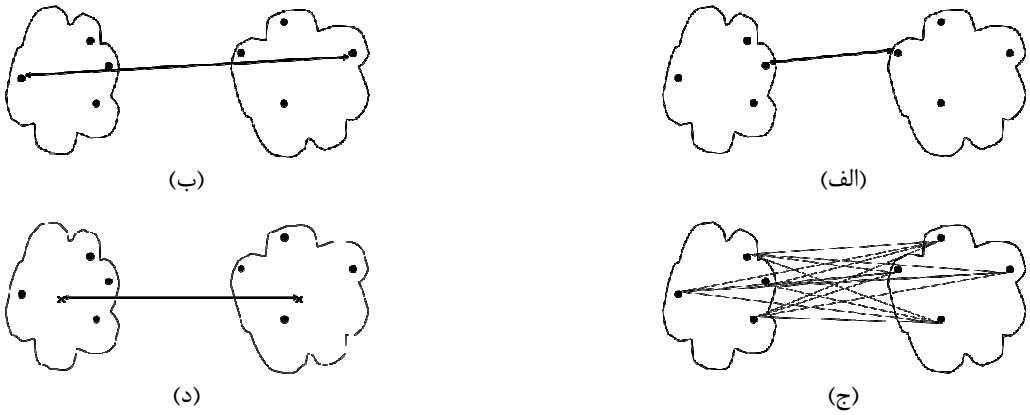
(1) فاصلهی کمینه : $d_{\min}(C_i, C_j) = min_{p\in c_i, p'\in c_j}|p-p'|$

(2) فاصلهی بیشینه : $d_{max}(C_i, C_j) = max_{p\in c_i, p'\in c_j}|p-p'|$

(3) فاصله میانگین : $d_{avg}(C_i, C_j) = \frac{1}{n_i n_j}\sum_{p\in c_i}\sum_{p'\in c_j}|p-p'|$

(4) فاصله مراکز : $d_{mean}(C_i, C_j) = |m_i - m_j|$

در شکل (1) هر کدام از چهار روش اندازه گیری فاصله بین دو خوشه را نشان داده شده است.

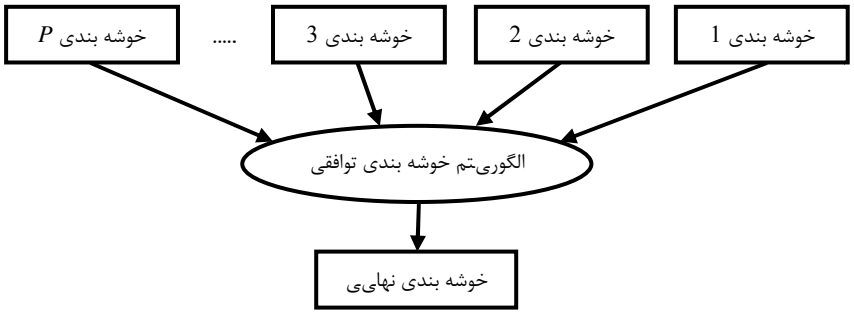


شکل (1) - چهار روش اندازه گیری فاصله بین دو خوشه : (الف) اندازه گیری فاصله با استفاده از فاصله نزدیکترین اشیاء در دو خوشه. (ب) اندازه گیری فاصله با استفاده از فاصله دورترین اشیاء در دو خوشه. (ج) اندازه گیری فاصله با استفاده از میانگین مجموع فاصله بین هر دو شئ. (د) اندازه گیری فاصله با استفاده از فاصله مراکز دو خوشه.

روش های مذکور در شرایطی قابل استفاده می باشند که خوشه هایی که نیاز به اندازه گیری فاصله بین آنها وجود دارد در یک خوشه بندی قرار داشته باشند یا به عبارتی دیگر اشتراک داده های بین هر دو خوشه تهی باشد. این روش ها در تعیین تناظر بین خوشه ها در خوشه بندی های مختلف قابل استفاده نمی باشند، چرا که خوشه های موجود در دو یا چند خوشه بندی، اشتراک داده ای دارند و این مسئله موجب می شود تا روش های موجود نتوانند میزان فاصله بین خوشه ها را در چنین حالتی بدست آورند.

2.2. خوشه بندی توافقی

مجموعه $X = \{x_1, x_2, \ldots, x_N\}$ شامل N شئ و $\Pi = \{\pi_1, \pi_2, \ldots, \pi_P\}$ شامل نتایج P خوشه بندی می باشد. هر خوشه بندی نیز شامل S خوشه است که در هر خوشه تعدادی از عناصر مجموعه X قرار گرفته است. هدف خوشه بندی توافقی، یافتن خوشه بندی جدید π^* از اشیاء مجموعه X است به گونه ای که بهترین نتیجه از اجتماع خوشه بندی های Π بدست آید [7]. شکل (2) مدل خوشه بندی توافقی را نشان می‌دهد.



شکل (2) – مدل خوشه بندی توافقی

جهت نشان دادن یکی از روش های خوشه بندی توافقی که در [2] آمده است از یک مثال استفاده می کنیم. فرض کنید مجموعه داده ای $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ که شامل شش شئ می باشد را در اختیار داریم. سه خوشه بندی مختلف بر روی این داده ها بدست آورده ایم، $\pi_1 = \{\{x_1, x_2\}, \{x_3, x_4\}, \{x_5, x_6\}\}$ ، $\pi_2 = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5\}, \{x_6\}\}$ و $\pi_3 = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}\}$ شامل جدول (1) نشان می دهد که هر داده در هر یک از سه خوشه بندی به کدام خوشه تعلق دارد. جهت یافتن خوشه بندی نهایی π^* در این مثال از رأی‌گیری استفاده شده است. به عنوان مثال داده x_2 در خوشه بندی های π_2 و π_3 در خوشه شماره دو قرار گرفته است، بنابراین چون این شماره خوشه بیشینه است به عنوان شماره خوشه برنده در رای گیری مشخص می شود. با توجه به روشی که توضیح داده شد، خوشه بندی نهایی بدست آمده به صورت $\pi^* = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5, x_6\}\}$ خواهد بود.

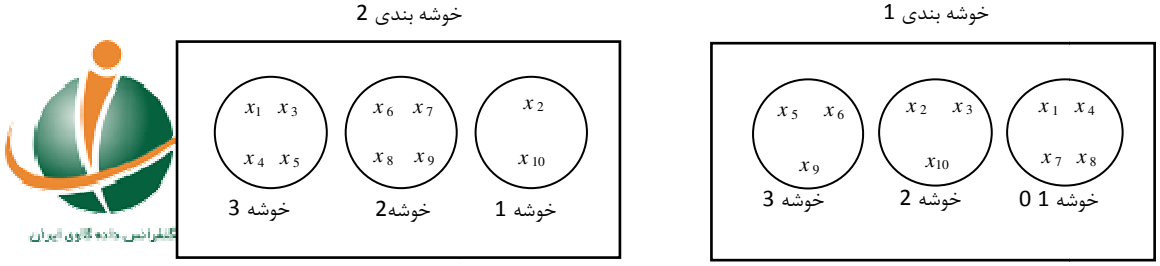
جدول (1) – نتایج خوشه بندی ها

	π_1	π_2	π_3	π^*
x_1	1	1	1	1
x_2	1	2	2	2
x_3	2	1	1	1
x_4	2	2	2	2
x_5	3	3	3	3
x_6	3	4	3	3

3. الگوریتم های تشخیص تناظر بین خوشه ها

در این بخش به تشریح الگوریتم های پیشنهادی به منظور تشخیص تناظر بین خوشه ها در دو حالت تعداد خوشه های برابر و تعداد خوشه های متفاوت در خوشه‌بندی ها می پردازیم. قبل از تشریح الگوریتم مسئله ای که نیاز به حل آن داریم را با یک مثال مورد بررسی قرار خواهیم داد.

همانطور که در شکل (3) مشاهده می‌شود مجموعه داده ای به صورت $X = \{x_1, x_2, \ldots, x_{10}\}$ می باشد. همچنین دو خوشه بندی داریم که هر یک دارای سه خوشه می باشند. با یک نگاه اجمالی به وضعیت خوشه ها در دو خوشه بندی و در نظر گرفتن داده ها می‌توانیم به این نتیجه برسیم که خوشه های 1 و 2 و 3 از خوشه بندی یک به ترتیب با خوشه‌های 3 و 1 و 2 از خوشه بندی دو متناظر و یا در واقع تشابه بیشتری نسبت به هم دارند.



شکل (3) - دو نمونه خوشه بندی بر روی مجموعه‌ی داده ای X

به طور کلی تشخیص تناظر به کمک الگوریتم پیشنهادی، به شرحی که در ادامه می آید، صورت می‌پذیرد. در این الگوریتم هر یک از خوشه ها به صورت یک الگو بیتی ها نشان داده می شود. تعداد بیت های الگو، برابر تعداد اشیاء در مجموعه داده ای می باشد. هر بیت در این الگو متناظر یکی از داده ها است، در صورت وجود آن داده در خوشه مورد نظر بیت متناظر آن یک و غیر این صورت مقدار آن صفر در نظر گرفته می شود. به عنوان مثال الگو بیتی برای خوشه شماره یک از خوشه بندی یک و خوشه شماره سه از خوشه بندی دو، به ترتیب به صورت (1001001100) و (1011100000) می‌باشد. بنابراین جهت تشکیل الگو بیتی به ازاء هر خوشه از روابط (5) و (6) استفاده می کنیم.

$$B_{im} = b_1 b_2 b_3 \dots b_N \quad i \in [1..P],\ m \in [1..S] \qquad (5)$$

$$b_k = f(\pi_i(x_k) = m) \quad k \in [1..N] \qquad (6)$$

که i شماره خوشه بندی، m شماره خوشه، N تعداد اشیاء در مجموعه داده ای، P تعداد خوشه بندی ها، S تعداد خوشه ها، π_i مجموعه خوشه ها به صورت $\{C_1, C_2, \dots, C_S\}$ و $\pi_i(x_k)$ شماره خوشه ای در π_i است که داده x_k در آن قرار دارد، می‌باشد. تابع f نیز در صورت برابری شرط $\pi_i(x_k) = m$ مقدار یک و در غیر این صورت مقدار صفر را بر می گرداند.

با در اختیار داشتن الگو های بیتی تلاش بر آن است تا با کمک آنها، یک ماتریس فاصله ایجاد کنیم. در واقع عناصر این ماتریس اختلاف بین هر دو خوشه از دو خوشه بندی مختلف را مشخص می‌کنند. به منظور محاسبه اختلاف یا میزان فاصله بین دو خوشه از فاصله همینگ استفاده خواهیم کرد. میزان فاصله همینگ بین دو خوشه با استفاده از الگو های بیتی محاسبه می شود، به این صورت که تعداد تفاوت ها بین بیت های متناظر شمارش می‌شود. به عنوان مثال فاصله بین خوشه شماره یک از خوشه بندی یک و خوشه شماره سه از خوشه بندی دو برابر 4 می شود. بنابراین رابطه (7) را خواهیم داشت.

$$d_{mn} = HammingDistance(B_{im}, B_{jn}) \quad i,j \in [1..P],\ m,n \in [1..S] \qquad (7)$$

d_{mn} عنصر سطر mام و ستون nام ماتریس فاصله است که فاصله بین خوشه شماره m از خوشه بندی π_i و خوشه شماره n از خوشه بندی π_j در آن قرار داده می شود.

پس از محاسبه ماتریس فاصله، ماتریس تناظر را با استفاده از آن بدست می آوریم. در واقع ماتریس تناظر پاسخ مسئله است، که سطر های این ماتریس را خوشه ها و ستون های آن را خوشه بندی ها تشکیل می‌دهند. نحوه ایجاد ماتریس تناظر با توجه به دو پارامتر تعداد داده‌ها و تعداد خوشه‌ها در هر خوشه بندی متفاوت است که در ادامه هر یک از این حالات را بررسی کرده و الگوریتم های پیشنهادی را مطرح می کنیم.

1.3. الگوریتم تشخیص تناظر درحالت تعداد خوشه های برابر

در شرایطی که تعداد داده ها و خوشه ها در خوشه بندی های مختلف یکسان باشد، روال ایجاد ماتریس تناظر به این صورت است که ابتدا یکی از خوشه بندی ها به عنوان خوشه بندی مرجع $\pi_{BaseClustering}$ در نظر گرفته می شود و سعی کرده تناظر بین خوشه های دیگر خوشه بندی ها را با آن بیابیم.

در الگوریتم پیشنهادی هر یک از خوشه بندی ها را یک به یک انتخاب کرده، و سپس با استفاده از الگو های بیتی و رابطه (7) فاصله هر یک از خوشه های موجود در آنها را با خوشه های $\pi_{BaseClustering}$ بدست می آوریم. هر یک از مقادیر بدست آمده در مرحله قبل را در یک بردار مرتب صعودی (بر حسب فاصله دو خوشه) با نام *SortedDist* قرار می دهیم. ترتیب صعودی فاصله ها در بردار مرتب *SortedDist* نشان‌دهنده اولویت میزان تشابه خوشه ها به یکدیگر است.

به عنوان مثال اولین عنصر در بردار، دو خوشه ای را معرفی می کند که بیشترین تشابه را به هم دارند. بنابراین انتخاب S (تعداد خوشه ها) عنصر ابتدایی این بردار به معنی تشخیص تناظر بین شبیه ترین خوشه ها در دو خوشه بندی می باشد. شرط انتخاب دو خوشه به عنوان خوشه های متناظر این است که هیچکدام از دو خوشه قبلا به عنوان خوشه متناظر انتخاب نشده باشند، در صورت عدم برقراری این شرط عنصر بعدی در بردار مرتب *SortedDist* بررسی می گردد. شماره خوشه هایی که از $\pi_{BaseClustering}$ انتخاب می شوند در مجموعه $Selected_{BaseClustering}$ و شماره خوشه هایی که از π_y انتخاب می شوند در $Selected_y$ قرار داده می شوند تا بتوان برقراری یا عدم برقراری شرط مذکور را بررسی نمود. انتخاب عنصر تا زمانی صورت می گیرد که هر یک از خوشه های $\pi_{BaseClustering}$ متناظر با یکی از خوشه های π_y شود. روال تشریح شده در الگوریتم (1) آورده شده است.

در شرایطی که تعداد خوشه ها در خوشه بندی های مختلف برابر است، ممکن است که تعداد داده ها در خوشه بندی های مختلف، متفاوت باشد. این حالت در شرایطی می تواند رخ دهد که برخی از خوشه بندی ها بر روی زیر مجموعه ای از مجموعه داده ای انجام شوند. در چنین حالتی باید شرایط زیر را در انتخاب خوشه بندی مرجع لحاظ کنیم :

1) خوشه بندی ای به عنوان خوشه بندی مرجع انتخاب می گردد که شامل تمامی داده های مجموعه داده ای باشد.
2) الگوی بیتی که برای خوشه ها استفاده می‌کنیم می‌بایست برابر با تعداد کل داده ها باشد و در خوشه بندی هایی که تعدادی از داده ها را نداریم در الگوی بیتی به ازای آن داده ها مقدار صفر قرار می گیرد.

الگوریتم (1) - تشخیص تناظر درحالت تعداد خوشه های برابر

Input: *BitPattern*[1..*S* , 1..*P*] as **bit_array**
BaseClustering as **number**
Output: *Corresponding*[1..*S* , 1..*P*] as **number**

declare *SortedDist* **as sorted_list each element**=*(Distance, CIndex*$_{BaseClustering}$*, CIndex*$_y$*)*
declare *Selected*$_{BaseClusterig}$ **as int_list**
declare *Selected*$_y$ **as int_list**

foreach $\pi_y \in \{\pi_1, \pi_2, \pi_3, \dots, \pi_P\}$
  for $i \in [1..S]$
    for $j \in [1..S]$
      SortedDist.**Add**(*HammingDistance*(*BitPattern*[*i*, *BaseClustering*], *BitPattern*[*j*, *y*]), *i*, j)
  repeat
    e = **SelectNext**(*SortedDist*)
    if(*e.CIndex*$_{BaseClustering} \in$ *Selected*$_{BaseClustering}$ **or** *e.CIndex*$_y \in$ *Selected*$_y$)
      continue
    else
      Selected$_{BaseClustering}$.**Add**(*e.CIndex*$_{BaseClustering}$)
      Selected$_y$.**Add**(*e.CIndex*$_y$)
      Corresponding[*e.CIndex*$_y$, *y*] = *e.CIndex*$_{BaseClustering}$
    end if
  until |*Selected*$_{BaseCluster}$| = S
end for π_y

2.3. الگوریتم تشخیص تناظر درحالت تعداد خوشه های متفاوت

در شرایطی که تعداد داده ها در خوشه بندی های مختلف یکسان و تعداد خوشه ها در خوشه بندی ها متفاوت باشد با رعایت دو شرط زیر و استفاده از الگوریتم (2) ماتریس تناظر را ایجاد می‌کنیم.

1) آن خوشه بندی ای که تعداد خوشه های آن بیشینه است به عنوان خوشه بندی مرجع در نظر گرفته شود.
2) در شرایطی که تعداد خوشه ها در خوشه بندی ها مختلف باشد، ممکن است یک یا چند خوشه از خوشه بندی با تعداد خوشه های بیشتر، با یک خوشه از خوشه بندی با تعداد کمتری خوشه متناظر شود. بنابراین الگوریتم (2) را به صورتی که در ادامه می آید ارائه می‌دهیم تا امکان ایجاد تناظر بین چند خوشه با یک خوشه فراهم شود. حداکثر تعداد خوشه هایی که می تواند با یک خوشه متناظر گردد به اندازه تفاضل تعداد خوشه‌های $\pi_{BaseClustering}$ با خوشه بندی دیگر می باشد.

در شرایطی که تعداد داده ها و خوشه ها در خوشه بندی ها متفاوت باشد، آن خوشه بندی ای را که تمامی داده ها را در بر دارد به عنوان خوشه بندی مرجع در نظر می‌گیریم. البته در این حالت لزوما خوشه بندی مرجع دارای تعداد خوشه بیشتری نمی باشد. بنابراین تغییراتی باید در الگوریتم (2) اعمال گردد. در حالتی که شرط $|\pi_{BaseClustering}| - |\pi_y| \geq 0$ برقرار باشد (به عبارتی دیگر تعداد خوشه های خوشه بندی مرجع بیش از خوشه بندی دیگر است) دقیقا مانند الگوریتم (2) عمل می کنیم در غیر این صورت کافی است دو شرط (*i*) و (*ii*) جا به جا شوند.

4. تحلیل پیچیدگی

در این قسمت الگوریتم های پیشنهادی در این مقاله را از لحاظ پیچیدگی زمانی مورد تحلیل و بررسی قرار می دهیم. الگوریتم (1) از یک حلقه اصلی تشکیل شده است که به تعداد خوشه بندی ها (P) به ازای هر یک از π_i ها تکرار می شود؛ در داخل حلقه، فاصله های بدست آمده به ازای هر دو خوشه در یک بردار مرتب قرار می گیرند که اگر تعداد عناصر بردار را M (مجذور تعداد خوشه ها) در نظر بگیریم، این عمل از مرتبه $O(M^2)$ خواهد بود؛ در داخل حلقه اصلی حلقه دیگری وجود دارد که به تعداد خوشه های خوشه بندی مرجع (S) تکرار می شود. بنابراین پیچیدگی زمانی این الگوریتم در بدترین حالت از مرتبه $O(P.(M^2 + S))$ خواهد بود. پیچیدگی زمانی برای الگوریتم (2) نیز مشابه الگوریتم (1) می باشد.

الگوریتم (2) – تشخیص تناظر درحالت تعداد خوشه های متفاوت

Input: *BitPattern*[1..*S* , 1..*P*] as **bit_array**
BaseClustering as **number**
Output: *Corresponding*[1..*S* , 1..*P*] as **number**

declare *SortedDist* **as sorted_list each element**=*(Distance, CIndex*$_{BaseClustering}$*, CIndex*$_y$*)*
declare *Selected*$_{BaseClusterig}$ **as int_list**
declare *Selected*$_y$ **as int_list**
declare *counter* **as number initialize with** 0

foreach $\pi_y \in \{\pi_1, \pi_2, \pi_3, \dots, \pi_P\}$
  for $i \in [1..S]$
    for $j \in [1..S]$
      SortedDist.**Add**(*HammingDistance*(*BitPattern*[*i*, *BaseClustering*], *BitPattern*[*j*, *y*]), *i*, j)
  repeat
    e = **SelectNext**(*SortedDist*)
    if(*e.CIndex*$_{BaseClustering} \in$ *Selected*$_{BaseClustering}$) (*i*)
      continue
    else
      if(*e.CIndex*$_y \in$ *Selected*$_y$)
        if(*counter* == $|\pi_{BaseClustering}|$-$|\pi_y|$) (*ii*)
          continue
        else
          counter++
        end if
      end if
      Selected$_{BaseClustering}$.**Add**(*e.CIndex*$_{BaseClustering}$)
      Selected$_y$.**Add**(*e.CIndex*$_y$)
      Corresponding[*e.CIndex*$_y$, *y*] = *e.CIndex*$_{BaseClustering}$
    end if
  until |*Selected*$_{BaseCluster}$| = S
end for π_y

5. پیاده سازی و آزمون

در این بخش ابتدا مجموعه داده ای که در اجرای الگوریتم مورد استفاده قرار گرفته است، تشریح می گردد و سپس نتایج حاصل از پیاده سازی و اجرای الگوریتم پیشنهادی جهت نشان دادن عملکرد آن ارائه می شود.

1.5. مجموعه داده ای

برای انجام آزمون در این مورد از مجموعه‌ی داده ای glass که شامل 214 شئ داده ای می باشد، استفاده شده است. این مجموعه داده شامل اطلاعاتی در مورد شش نوع شیشه[11] می باشد. مشخصات کامل این مجموعه‌ی داده ای در جدول (2) آمده است.

جدول (2) – مشخصات مجموعه داده ای glass

نام جدول	تعداد داده ها	تعداد فیلد ها	تعداد کلاس ها	نوع فیلد ها
glass	214	10	6	عددی (اعشاری)

این مجموعه داده ای یکی از مجموعه های UCI Machine Learning Repository می باشد، که در پروژه ها و تحقیقات مرتبط با یادگیری ماشینی و تحلیل داده ها استفاده می شود.

2.5. نتایج حاصل

ابتدا بر روی مجموعه‌ی داده ای glass سه بار عمل خوشه بندی را به ترتیب با روش های Expectation Maximization (EM)، K-Means و FarthestFirst انجام می دهیم. تعداد خوشه ها به منظور انجام خوشه بندی در هر یک از روش ها شش در نظر گرفته شده است. انجام خوشه بندی نیز با استفاده از نرم افزار Weka نسخه‌ی 3.5.8 بوده است. نتایج این خوشه بندی در جدول (3) آمده است؛ سطر ها در جدول (3) برچسب کلاس ها را نشان می دهند و ستون ها نیز نشان دهنده یک خوشه می باشند. هر مقدار در جدول (3) بیانگر تعداد داده هایی از هر کلاس است که در یک خوشه خاص قرار گرفته است. لازم به ذکر است که الگوریتم نیز با استفاده از C#.NET 2008 پیاده سازی شده است.

جدول (3) – نتایج انجام سه خوشه بندی بر روی مجموعه داده ای glass

خوش بندی 3 FarthestFirst						خوش بندی 2 K-Means						خوش بندی 1 Expectation Maximization						
C_6	C_5	C_4	C_3	C_2	C_1	C_6	C_5	C_4	C_3	C_2	C_1	C_6	C_5	C_4	C_3	C_2	C_1	برچسب کلاس / خوشه
0	0	0	0	0	70	0	38	17	0	0	15	37	19	1	0	13	0	build wind float
10	0	0	0	1	65	11	42	2	0	0	21	42	1	4	12	17	0	build wind non-float
0	0	0	0	0	17	0	12	2	0	0	3	11	5	0	0	1	0	vehic wind float
5	0	1	2	0	5	7	1	0	2	2	1	0	0	1	9	0	3	containers
3	2	0	0	0	4	1	5	0	0	3	0	0	3	1	4	0	1	tableware
1	0	4	0	0	4	1	2	3	0	23	0	0	0	6	0	1	22	headlamps

همانطور که در جدول (3) مشخص است، به عنوان مثال خوشه‌ی C_1 در خوشه بندی 1 بسیار متفاوت از خوشه هم شماره آن یعنی C_1 در خوشه‌بندی 2 و 3 می‌باشد. جهت اجرای روش پیشنهادی و یافتن خوشه های متناظر، خوشه بندی 1 را به عنوان خوشه بندی مرجع انتخاب می کنیم. در روش پیشنهادی، قبل از تعیین خوشه های متناظر، ماتریس فاصله خوشه بندی مرجع با یک خوشه بندی دیگر محاسبه می شود و سپس مقادیر بدست آمده مرتب شده، تا در نهایت خوشه هایی که بیشترین تشابه را به هم دارند، به عنوان خوشه های متناظر انتخاب گردند. دو ماتریس نشان داده شده در شکل (4) ماتریس فاصله خوشه بندی مرجع تا هر کدام از خوشه بندی های 2 و 3 می باشد.

	C_1	C_2	C_3	C_4	C_5	C_6
C_1	66	10	24	50	124	44
C_2	34	60	34	52	119	52
C_3	61	49	27	47	121	9
C_4	51	33	15	31	105	31
C_5	60	56	30	16	116	48
C_6	102	118	92	114	38	110

(الف)

	C_1	C_2	C_3	C_4	C_5	C_6
C_1	187	27	24	11	26	43
C_2	133	33	34	57	34	51
C_3	174	26	27	50	27	10
C_4	168	12	15	28	13	30
C_5	137	29	30	53	30	47
C_6	75	91	92	115	92	109

(ب)

شکل (4) – ماتریس های فاصله: (الف) ماتریس فاصله بین خوشه های خوشه بندی مرجع و خوشه بندی 2. (ب) ماتریس فاصله بین خوشه های خوشه بندی مرجع و خوشه بندی 3

اگر مقادیر ماتریس فاصله که در شکل (4) آمده است را مرتب شده در نظر بگیریم، می توان ماتریس تناظر بین خوشه بندی مرجع و خوشه‌بندی‌های 2 و 3 را محاسبه نمود. شکل (5) نشان دهنده ماتریس تناظر است. تلاقی هر سطر و ستون نشان دهنده این است که هر خوشه در هر یک از خوشه‌بندی ها، متناظر کدام خوشه در خوشه بندی مرجع می باشد. به عنوان مثال تقاطع سطر C_6 و ستون π_2 بدین معنی است که خوشه شماره 6 در خوشه بندی 2، متناظر با خوشه شماره 2 در خوشه بندی مرجع می باشد.

	π_1	π_2	π_3
C_1	0	1	5
C_2	1	0	3

C_3	2	3	4
C_4	3	4	0
C_5	4	5	1
C_6	5	2	2

شکل (5) – ماتریس تناظر

6. نتیجه گیری

با توجه به آنچه که در این مقاله مطرح شد، خوشه بندی توافقی به این صورت عمل می کند که خوشه ها با شماره یکسان در خوشه بندی های مختلف را به عنوان خوشه های متناظر در نظر می گیرد. همانطور که در نتایج پیاده سازی هم مشاهده شد در عمل ممکن است خوشه های هم شماره در خوشه بندی های مختلف بسیار متفاوت از یکدیگر باشند؛ از اینرو بدیهی است که در این حالت، با کاهش کیفیت خوشه های به دست آمده در خوشه بندی توافقی مواجه خواهیم شد. بنابراین در این مقاله الگوریتمی جهت تشخیص دقیق تناظر بین خوشه ها در خوشه بندی های مختلف ارائه کردیم. استفاده از الگوریتم پیشنهادی به عنوان مرحله ای قبل از انجام خوشه بندی توافقی منجر به بهبود نتایج، نسبت به حالتی می شود که دقتی در متناظر گرفتن خوشه وجود ندارد.

تقدیر و تشکر

از جناب آقای صفاری، مدیر عامل شرکت فن آوری اطلاعات و پردازش و جناب آقای مهندس احمد خادم زاده عضو هیئت علمی دانشگاه آزاد اسلامی واحد مشهد که ما را در تهیه این مقاله یاری نموده اند، کمال تشکر را داریم.

مراجع

[1] A. Fred, A. K. Jain; "*Data Clustering using evidence accumulation"*, ICPR, 2002.
[2] A. Gionis, H. Mannila, P. Tsaparas; "*Clustering Aggregation*", ACM Transactions on Knowledge Discovery from Data (TKDD), 2007.
[3] A. Strehl, J. Ghosh; "*Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions*", Journal of Machine Learning Research, 2002.
[4] A. Topchy, A. K. Jain, W.Punch; "*A Mixture Model of Clustering Ensemble*", SDM, 2004.
[5] C. Boulis, M. Ostendorf; "*Combining Multiple Clustering Systems*", PKDD, 2004.
[6] J. Han, M. Kamber; *Data Mining Concepts and Techniques*, Second Edition, Morgan Kaufmann, 2006.
[7] N. Nguyen, R. Caruana; "*Consensus Clustering*", Proceedings of the Sixth International Conference on Data Mining (ICDM), 2007.
[8] P. Berkhin; "*Survey of Clustering Data Mining Techniques*", Accrue Software Institude, 2002.
[9] S. B Kotsiantis, P.E Pintelas; "*Recent Advanced in Clustering: A Brief Survey*", WSEAS Transactions on Information Science and Applications 1, 73-81, 2004.
[10] V. Filkov, S.Skiena; "*Integration Microarray Data by Consensus Clustering*", International Conference on Tools with Artificial Intelligence , 2003.
[11] X. Z. Fern, C.E.Brodley; "*Random Projection for High Dimensional Data Clustering : A Cluster Ensemble Approach*", ICML, 2003.

[1] Hierarchical clustering
[2] Partitioning clustering
[3] Density-based clustering
[4] Consensus clustering
[5] Nominal data
[6] Missing data
[7] Outliers data
[8] Privacy-preserving

[9] Divisive
[10] Agglomerative
[11] http://archive.ics.uci.edu/ml/datasets/Glass+Identification