همایش ملی الکترونیکی دستاوردهای نوین در علوم مهندسی و پایه
National e-Conference on Advances in Basic Sciences and Engineering
. . . . W W W . A E B S C O N F . I R . . . .

# بررسی داده کاوی توزیع شده با الگوریتم k-means

**نجمه تقی زاده [1]، لاله مداح علی[2]، دکتر محبوبه شمسی[3]، علیرضا آراسته[4]**

1. دانشجوی کارشناسی ارشد فناوری اطلاعات دانشگاه قم
2. دانشجوی کارشناسی ارشدفناوری اطلاعات دانشگاه قم
3. عضوهیئت علمی گروه مهندسی برق و کامپیوتر دانشگاه صنعتی قم
4. فارغ التحصیل مهندسی نرم افزار دانشگاه پیام نور مرکز قم

نویسنده مسئول : نجمه تقی زاده

**چکیده**

اکثر الگوریتم های خوشه بندی نیاز به داده های متمرکز دارند ، اما این الگوریتم ها با توسعه اینترنت و در برخورد با داده های توزیع شده ، با دو چالش روبرو شدند. اول، حجم داده های تولید شده حتی برای ابرکامپیوترها هم خیلی زیاد شده است. دوم، داده ها در چندین مکان ذخیره شده اند و متمرکز کردن آنها در یک جا بسیار پرهزینه خواهد بود، هم چنین محدودیت پهنای باند و حریم شخصی نیز از نگرانی ها و موانع متمرکز سازی داده ها می باشد. به همین دلیل برای حل این مشکلات، داده کاوی توزیع شده یک حوزه تحقیقاتی پرطرفدار شده است. یکی از الگوریتم های خوشه بندی، الگوریتم کامینز است که به عنوان یکی ازبا نفوذترین الگوریتم های داده کاوی مورد استفاده قرار می گیرد و بسیار ساده و مقیاس پذیر است. در سال های اخیر نسخه هایی از این الگوریتم انتشار یافته است که می تواند در برخورد با داده های توزیع شده ، به خوبی عمل کرده و نتایج خوبی را ارائه دهد. در این الگوریتم ها ، نیازی به جمع آوری کردن اطلاعات و داده ها در یک مجموعه متمرکز نیست. در این مقاله قصد داریم که این الگوریتم ها را معرفی و بررسی کنیم.

**واژه‌های کلیدی:** داده کاوی توزیع شده، خوشه بندی،الگوریتم کامنیز،الگوریتم کامینز توزیع شده،نرمال سازی

**مقدمه**

حجم داده های تولید شده در سال های اخیر در حال رشد است. مجموعه اسناد، عکس ها، بیوانفورماتیک و دیگر داده ها با تکنولوژی های جدید، در حال افزایش هستند. در بسیاری از موارد، داده ها در سایت هایمختلف داده ، به طور طبیعی توزیع ، تولید و ذخیره شده اند. در این زمینه، الگوریتم ها باید قادر به استخراج اطلاعات مناسب از داده های توزیع شده با عملکرد خوب محاسباتی باشند.با این حال، اکثر تکنیک های داده کاوی که پیشنهاد می شوند داده های متمرکز را مورد توجه قرار می دهند ، اما متمرکز کردن آنها در یک جا بسیار پرهزینه خواهد بود ، هم چنین محدودیت پهنای باند و حریم شخصی نیز از نگرانی ها و موانع متمرکز سازی داده ها می باشد.

خوشه بندی یکی از تکنیک های پر کاربرد داده کاوی است که در آن ، مجموعه ای از اشیا که معمولا به صورت چند بعدی هستند به گروه ها یا کلاس هایی تقسیم بندی می شوند که در هر گروه، اشیا بر اساس معیارهای از پیش تعیین شده ، به هم شبیه هستند. به طور کلی، فاصله اقلیدسی از مراکز خوشه ها به عنوان معیار شباهت یا تفاوت در نظر گرفته می شود.الگوریتم های خوشه بندی در زمینه های مختلفی مثل داده کاوی، تشخیص الگو ، تئوری یادگیری و دیگر زمینه ها کاربرد دارد.

یکی از متداول ترین الگوریتم های خوشه بندی ، الگوریتم کامینز است . این الگوریتم ، بیش از نیم قرن است که مورد بررسی قرار گرفته است.اخیرا الگوریتم کامینز به خاطر سادگی، مقیاس پذیری و کاربرد در نرم افزارهای مختلف، به عنوان یکی از 10 الگوریتم موثر و بانفوذ داده کاوی انتخاب شده است.با این حال،الگوریتم کامینز،به انتخاب نمونه خوشه های اولیه حساس است،پس اگر خوشه های اولیه بدرستی انتخاب نشوند نتایج خوب و بهینه بدست نمی آید. هم چنین در این الگوریتم باید تعداد خوشه های مورد نظر (k) را مشخص کرد و این کار (مشخص کردن تعداد خوشه ها) می تواند کاملا محدود کننده باشد. در سال های اخیر نسخه هایی از این الگوریتم انتشار یافته است که می تواند در برخورد با داده های توزیع شده به خوبی عمل کند.

در قسمت دوم مقاله، به بررسی داده کاوی توزیع شده(DDM) می پردازیم. در قسمت سوم الگوریتم استاندارد کامینز و نسخه بهبود یافته آن را بررسی می کنیم. در قسمت چهارم مقاله،الگوریتم های کامینز توزیع شده را مرور می کنیم.

**داده کاوی توزیع شده**

داده کاوی توزیع شده ، تکنیک کشف الگو ها یا تولید مدل هایی از داده های توزیع شده است که به صورت متمرکز نه امکان پذیر است و نه مطلوب. داده کاوی توزیع شده می تواند در ابر کامپیوترهای موازی، شبکه های نظیر به نظیر و شبکه های حسگر مورد استفاده قرار گیرد. اما، محیط های مختلف مسائل و نگرانی های مختلفی دارند. DDM به دسته های زیر تقسیم می شود.[1]

آکادمی داده
dataacademy.ir

**خوشه های متمرکز در مقابل نظیر به نظیر**

خوشه متمرکز یک هماهنگ کننده دارد. هماهنگ کننده کار را به کامپیوترهای مختلف تقسیم می کند. خوشه متمرکز به کارگیری و هماهنگی ساده تری دارد. و کاررا عادلانه تقسیم می کند و مشکل نقطه شکست(single point of failure)دارد. الگوریتم های نظیر به نظیر به یک سرور مرکزی بستگی ندارند، و هر سایت داده را می گیرد و کار خودش را انجام می دهد.

**مدل تکی در مقابل فرا-یادگیری**

مدل تکی یک الگوریتم داده کاوی داده ها را به بخش های کوچکتر که در هر سایت توزیع شده اند تقسیم می کند. الگوریتم میتواند داده، نتایج میانی، مدل های پیشگو، و نتایج نهایی یک الگوریتم داده کاوی را انتخاب کند و منتقل کند. از طرف دیگر، فرا-یادگیری، به صورت یادگیری از دانش یادگرفته شده تعریف شده است، تکنیک دیگری است که سر و کارش با مشکل محاسبه یک مدل سراسری از پایگاه های داده ایی ذاتا توزیع شده و بزرگ است. هدف فرا-یادگیری محاسبه تعدادی مدل مستقل(classifier) با اعمال برنامه های یادگیری به طور موازی بدون انتقال یا دسترسی مستقیم به سایت است.

**داده همگن در مقابل داده ناهمگن**

در یک سیستم دیتابیس رابطه ای توزیع شده، اطلاعات باید در سایت های مختلف ذخیره شود. هر سایت جداول رابطه ای کامل دارد، داده همگن برای هر سایت ذخیره می شود. که هر سایت اطلاعات جداول مختلف را ذخیره می کند، در این حالت مسئله داده ناهمگن داریم. اکثر داده کاوی موجود، داده همگن را در سایت های مختلف در نظر می گیرد. در حالت داده توزیع شده ناهمگن، ما فقط دانش ناقص را درباره دیتاست کامل مشاهده می کنیم. مدل های محلی مختلف دید محلی از کل مسئله دارند داده کاوی توزیع شده باید یک مدل سراسری از آن مدل های محلی مختلف ایجاد کند. پس، کاوش دادگان ناهمگن، چالش برانگیز است.

**داده کاوی توزیع شده و داده کاوی موازی**

هم داده کاوی توزیع شده و هم داده کاوی موازی فرآیند داده کاوی را سرعت می بخشند اما از جهاتی متفاوت هستند. DDM اغلب الگوریتم های کاوش متفاوت یا یکسانی را اعمال می کنند برای داده محلی و بین واحدهای پردازش ارتباط برقرار می کند و کشف الگوی محلی را ترکیب می کند با الگوریتم های داده کاوی محلی از پایگاه های داده ایی محلی به یک راه حل دانش جهانی. در این حالت، دانش کشف شده اغلب با دانش کشف شده توسط اعمال الگوریتم های داده کاوی به کل دیتاست فرق دارد. دقت یا کارایی DDM تقریبا پیشگویی اش سخت است چون بستگی دارد به افراز داده، زمان بندی کار، و سنتز جهانی. در مقابل با DDM ، یک الگوریتم داده کاوی موازی همان دانشی را پیدا می کند که با الگوریتم ترتیبی اش یافت شده به علت اعمال الگوریتم موازی جهانی روی کل دیتاست. دقت آن نسبت به DDM بیشتر ضمانت شده است.

**الگوریتم کامینز**

**الگوریتم کامینز استاندارد**

در این الگوریتم، داده ها را به k خوشه مجزا تقسیم کنیم.این الگوریتم به دو فاز مجزا تقسیم می شود: در فاز اول برای هر خوشه یک نقطه را به عنوان نقطه ثقل خوشه(centroid) یا نقطه مرکزی خوشه بدست می آوریم.و درفاز بعدی بدست می آوریم که هر نقطه از مجموعه داده به کدام مرکز خوشه(centriod)نزدیک تر است و آن نقطه را به خوشه مربوطه نسبت می دهیم.در حالت کلی برای بدست آوردن فاصله بین نقاط داده و مراکز خوشه ها، از فاصله اقلیدسی استفاده می شود. زمانی که تمام نقاط در خوشه ها قرار گرفتند مرحله اول به اتمامرسیده و خوشه بندی اولیه انجام شده است.سپس دوباره برای خوشه ها مراکز جدیدی را بدست می آوریم و فاصله هر نقطه را نسبت به این نقاط مرکزی اندازه می گیریم تا خوشه ها به روز شوند و این کار تا زمانی ادامه پیدا می کند که خوشه ها همگرا شوند.[2]

**Algorithm 1**: The k-means clustering algorithm

Input:

D = {d1, d2,.....,dn} //set of *n* data items.*k*// Number of desired clusters

Output:

A set of *k* clusters.

Steps:

1. Arbitrarily choose *k* data-items from D as initialcentroids;
2. Repeat
   - Assign each item *d*i to the cluster whichhas the closest centroid;
   - Calculate new mean for each cluster;
   - Until convergence criteria is met.

اما اشکال عمده این الگوریتم این است که با توجه به مقدار مراکز اولیه، خوشه های متفاوتی تولید می شودو در نتیجه کیفیت خوشه های نهایی شدیدا به انتخاب مراکز اولیه خوشه ها وابسته است. این الگوریتم از لحاظ محاسباتی گران و متناسب با تعداد نقاط، تعداد خوشه ها و تعداد تکرارها نیاز به زمان دارد.در قسمت بعد الگوریتم اصلاح شده کامینز را بررسی می

آکادمی داده
dataacademy.ir

**الگوریتم کامینز بهبود یافته**

همانطور که بیان شد، الگوریتم خوشه بندی کامینز استاندارد از لحاظ محاسباتی سنگین و کیفیت نتایج خوشه های آن، شدیدا به انتخاب مراکز اولیه خوشه ها وابسته است. یه همین دلیل محققان سعی کرده اند تا با ارائه روش هایی این نقایص را برطرف کنند و الگوریتم کامینز را بهبود ببخشند.

در روش کامینز بهبود یافته، هر دو فاز الگوریتم کامینز برای بهبود دقت و کارایی آن اصلاح شده است. در مرحله اول، مراکز اولیه خوشه ها برای تولید خوشه ها با دقت بالاتر از یک روش سیستماتیک به جای انتخاب تصادفی استفاده میکنند.

مرحله دوم با تشکیل خوشه های اولیه بر اساس فاصله نسبی هر نقطه (داده) از مراکز اولیه خوشه ها شروع می شود. این خوشه ها متعاقبا بوسیله یک روش اکتشافی تنظیم می شوند، بنابراین کارایی بهبود می یابد.[4[,]3]

هر دو مرحله الگوریتم، در دو الگوریتم 2 و 3 توضیح داده شده که مطابق زیر است :

Algorithm 2: Finding the initial centroids
Input:
D = {d1, d2,.....,dn} // set of *n* data items*k* // Number of desired clusters
Output:
Aset of *k* initial centroids .
Steps:
1. Set m = 1;
2. Compute the distance between each data point and allother data- points in the set D;
3. Find the closest pair of data points from the set D andform a data-point set
Am (1<=m <= k) which containsthese two data- points, Delete these two data points from
the set D;
4. Find the data point in D that is closest to the datapoint setAm, Add it to Am and deleteit from D;
5. Repeat step 4 until the number of data points in Amreaches 0.75*(n/k);
6. If m<k, then m = m+1, find another pair of datapointsfrom D between which the distance is the shortest, formanother data-point set Am and delete them from D, Go tostep 4;
7. For each data-point set Am (1<=m<=k) find thearithmetic mean of the vectors of datapoints in Am,these means will be the initial centroids.

Algorithm 3: Assigning data-points to clusters
Input:
D = {d1, d2,.....,dn} // set of n data-points.
C = {c1, c2,.....,ck} // set of k centroids
Output:
A set of k clusters
Steps:
1. Compute the distance of each data-point di (1<=i<=n) toall the centroids cj (1<=j<=k) as d(di, cj);
2. For each data-point di, find the closest centroid cj andassign di to cluster j.
3. Set ClusterId[i]=j; // j:Id of the closest cluster
4. Set Nearest_Dist[i]= d(di, cj);
5. For each cluster j (1<=j<=k), recalculate the centroids;
6. Repeat
7. For each data-point di,
7.1 Compute its distance from the centroid of thepresent nearest cluster;
7.2 If this distance is less than or equal to the presentnearest distance, the data-point stays in the cluster;
Else
7.2.1 For every centroid cj (1<=j<=k)
Compute the distance d(di, cj);
Endfor;
7.2.2 Assign the data-point di to the cluster withthe nearest centroid cj
7.2.3 Set ClusterId[i]=j;
7.2.4 Set Nearest_Dist[i]= d(di, cj);
Endfor;
8. For each cluster j (1<=j<=k), recalculate the centroids;
Until the convergence criteria is met.

آکادمی داده
dataacademy.ir

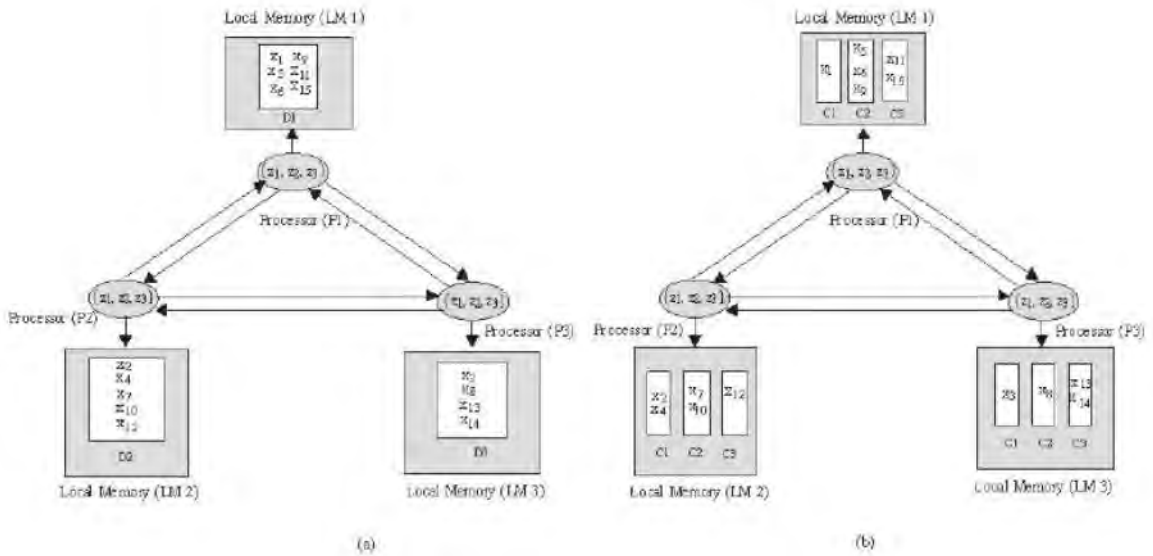
**الگوریتم های کامینز توزیع شده**

همانطور اشاره شد، با توجه به توزیع شدگی داده ها در مجموعه سایت های مختلف و عدم متمرکز کردن آنها در یک مجموعه، نیاز به الگوریتم هایی می باشد که در محیط های توزیع شده به خوبی عمل کنند. الگوریتم کامینز یکی از الگوریتم هایی است که می تواند در این موارد نتایج مورد قبولی را ارائه دهد، این الگوریتم به صورت استاندارد قادر خواهد بود در یک مجموعه متمرکز کار کند، لکن با کمی تغییر در ساختار این الگوریتم و ترکیب آن با دیگر الگوریتم ها می توان آن را به صورت توزیع شده استفاده کرد.در زیر به ذکر نمونه هایی از این الگوریتم که در محیط های توزیع شده کار می کنند، می پردازیم.

**الگوریتم کامینز توزیع شده در محیط های موازی**

اگر به الگوریتم کامینز استاندارد نگاه کنیم میبینیم که توزیع شدگی به طور ذاتی درون آن نهفته است.فرایند با یک مجموعه داده آغاز می شود و سپس مراکز اولیه خوشه ها انتخاب می گردند.برای اجرای الگوریتم کامینز استاندارد به یک سیستم پردازنده واحد احتیاج است.برای هر داده در هر تکرار عضویت آن در خوشه ها محاسبه می شود. در این وضعیت، پردازنده همه ویژگیهای ساختاری الگوریتم کامینز را در حافظه محلی نگهداری می کند و تکرار گام های الگوریتم کامینز نقاط مرکزی نهاییZ را محاسبه می کند.

در محیط های توزیع شده ما از پردازنده های شبکه ای برای الگوریتم کامینز استفاده میکنیم. فرض میکنیم که مجموعه داده ها در پردازنده های شبکه توزیع شده اند. این پردازنده ها، فرایند را در یک روش تعاملی اجرا می کنند.شکل 1(a) و 1(b) توزیع داده را در یک شمای توزیع شده در قبل و بعد از اجرای الگوریتم نوزیع شده نشان می دهد.[5]




شکل (1) شماء از اجرای الگوریتم k-means در محیطهای توزیع شده (a) قبل از خوشه بندی (b) بعد از خوشه بندی

فرض میکنیم که داده های اولیه توزیع شده، مطلقا دلخواه و مستقل هستند.هر پردازنده$P_i$به طور تصادفی، یک مجموعه برداری از مراکز ($Z_i=\{z_k$ for $k=1$ to $K\}$) را مقدار دهی اولیه می کند. این مراکز، مراکز قدیمی نام گذاری می شوند$\{Z_k^{(old)}\}$ . بعد از این مقدار دهی اولیه، مراکز خوشه های محلی از هر پردازنده $P_i$ به دیگر پردازنده ها پخش می شود.

در داخل حلقه تکرار اصلی الگوریتم کامینز توزیع شده، داده ها در هر پردازنده برای محاسبه فاصله از مراکز خوشه های مربوطه به صورت موازی استفاده می شود. بعد از توزیع محلی، بردار مراکز جدید$\{Z_k^{(new)}\}$ با استفاده از داده ها محلی طبقه بندی شده و مراکز قدیمی،محاسبه می شود.این کار برای همه k ها انجام میگیرد.هم چنین دقت داشته باشید که از مراکز قدیمی خوشه ها نیز برای محاسبه مراکز جدید استفاده می شود بخاطر اینکه از ایجاد خوشه های خالی در هر پردازنده جلوگیری شود.

گام محاسبه مرکز، یکی از مهمترین بخش های این الگوریتم است. از نظر ریاضیاتی، تفاوت بین محاسبه مراکز در الگوریتم کامینز معمولی و کامینز توزیع شده به صورت زیر است :

محاسبه مرکز در الگوریتم کامینز استاندارد :

$$Z_k^{(new)} = \frac{1}{n_k}\{\sum_{x_j \in c_j}(X_j)\} \qquad (1)$$

محاسبه مرکز در الگوریتم کامینز توزیعی :

$$Z_k^{(new)} = \frac{1}{n_k+1}\{\sum_{x_j \in c_j}(X_j) + Z_k^{(old)}\} \qquad (2)$$

بردار مراکز جدید در بین همه پردازنده ها، توزیع شده است.برای هر پردازنده $P_i$، بردار مرکز جدید بصورت میانگین نقاط و مراکز قدیمی برای همه مقادیر k=1 تا k بدست می آید. ، می کنند.

آکادمی داده
dataacademy.ir

فرایند بالا ادامه می یابد تا زمانی که دیگر بردار مرکز، تغییری نکند وثابت بماند.استراتژی محاسبه مراکز جدید باعث می شود این الگوریتم از دیگر الگوریتم های کامینز متفاوت باشد.در غیر اینصورت همان الگورتم کامینز اصلی می باشد.

این الگوریتم بصورت زیر است :

Algorithm 4: The Distributed k -means Algorithm
Input: a set S of d-dimensional data and an integer K.
Output :K clusters
Begin
for each processor $P_i$ do in parallelinitialize a set of center vectors $z_k$ fork=1 to K ,
(these centers are called old centers $\{Z_k^{(old)})\}$)

Repeat
for each processor $P_i$do in parallelbegindistribute local data in $P_i$into k classesaccording to
minimum distance from $z_k$for k =1 to Kcompute new center vectors $\{Z_k^{(new)}\}$ for k =1,... .K
considering old centers $\{Z_k^{(old)}\}$ as data items
end
for each processor $P_i$broadcast$\{Z_k^{(new)}\}$, for k=1,…, K, to all other processors
for each processor $P_i$do in parallelbegin
for k=l,... ,K do
compute average of$\{Z_k^{(new)}\}$ from self and those received from other processors and replace $\{Z_k^{(old)}\}$with this average
end
until center vectors are stable
end

**الگوریتم خوشه بندی نرم توزیع شده گروه محور(DSCA)**

الگوریتم کامینز توزیع شده(DKM) گروه محور، یکی دیگراز الگوریتم های خوشه بندی توزیع شده است.از این الگوریتم به منظور خوشه بندی کردن داده های محلی استفاده می شود.نقاط مرکزی خوشه های محلی، در یک مکان مرکزی ترکیب می شوند و با استفاده از الگوریتم کامینز خوشه بندی می شوند که نقاط مرکزی سراسری را تولید می کنند.از این نقاط سراسری برای به روز کردن خوشه های محلی استفاده می شود.به منظور تولید خوشه های نرم سراسری، الگوریتم کامینز برای پشتیبانی از الگوریتم کامینز اصلاح شده است.[6]

**خوشه بندی نرم و خوشه بندی سخت**

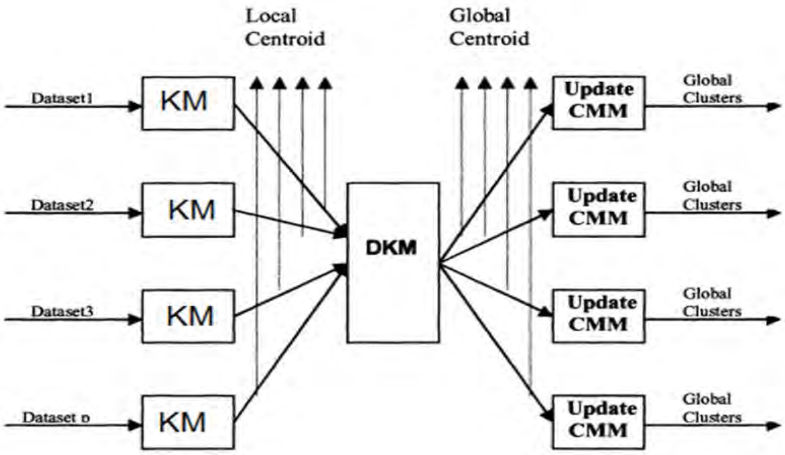
ب ه طور طبیعی خوشه ها می توانند نرم یا سخت باشند. در خوشه بندی های معمول، اشیاء مشابه در یک خوشه جمع آوری می شوند در حالی که اشیاء با ویژگیهای متفاوت در خوشه های دیگر هستند.به این خوشه ها که نقطه مشترک ندارند خوشه های سخت می گویند.اما در خوشه بندی نرم اشیاء می توانند در بیش از یک خوشه عضو باشند. خوشه بندی نرم ممکن است بصورت فاری باشد و یا مرزهای ناهموار داشته باشد.

روش معمول کامینز، کهدر اکثر برنامه های داده کاوی پیدا میشود، همیشه منجر به خوشه های سخت میشود.اما در عمل، ویژگی های یک شی ممکن است در دسته های مختلف به نمایش در بیاید. در چنین مواردی شی باید به چندیدن خوشه متعلق باشدو در نتیجه لزوما مرزهای خوشه ها با هم تداخل دارند. الگوریتم های خوشه بندی فازی، مانند الگوریتم C-Means این احتمال را ایجاد میکنند که یک شی به چندین خوشه تعلق داشته باشد.[7]

**تشریح الگوریتم**

الگوریتم خوشه بندی نرم توزیع شده گروه محور(DSCA) در شکل 2 ، نمایش داده شده است.در ابتدا مجموعه داده های محلی، با استفاده از الگوریتم k-means تا ماتریسی از مقادیر عضویت خوشه ها (CMM) و نقاط مرکزی محلی بدست آید.در این مرحله، فقط نقاط مرکزی محلی به یک مکان مرکزی منتقل می شوند و در آنجا با هم ترکیب شده و به اصطلاح CentroidDataset را نامیده می شوند.در این زمان، الگوریتم کامینز(DKM) به کار می رود تا نقاط مرکزی محلی ( local centroid) ها را که مشابه هستند گروه بندی کرده وسپس میانگین آنها، نتایج مورد نیاز برای فرایند خوشه بندی توزیعی نقاط سراسری (global Centroid) است. در گام آخر نیز، ماتریس CMM که در اولین مرحله تولید شد را توسط این نقاط سراسری(Global Centroid) به روز میکنیم تا در نهایت خوشه های نهایی بدست آید. بدون در نظر گرفتن تعداد خوشه های تولید شده در مجموعه داده های محلی، تعداد k مرکز سراسری بدست می آید.

آکادمی داده
dataacademy.ir




شکل 2 چارچوب برای خوشه بندی نرم توزیع شده

**الگوریتم k-means توزیع شده نرمال**

**نرمال سازی**

پیش پردازش اغلب قبل از استفاده از هر الگوریتم داده کاوی برای بهبود کارایی نتیجه نیاز است. نرمال سازی داده یکی از رویه های پیش پردازشی در داده کاوی است، که صفت درون یک محدوده خاص قرار می گیرد مثل 1.0- تا 1.0 یا 0.0 تا 1.0 . نرمال سازی قبل از خوشه بندی برای معیار فاصله استفاده می شود، مثل فاصله اقلیدسی، که به اختلاف در اندازه یا مقیاس صفات حساس هستند. در اپلیکیشن های واقعی، به دلیل تفاوت در محدوده مقدار صفات، یک صفت باید به دیگری غلبه کند. نرمال سازی مانع از زیادی وزن دادن به صفات با محدوده بزرگ مثل حقوق نسبت به صفات با محدوده کوچکتر می شود مثل سن. هدف، مساوی سازی اندازه یا بزرگی و تنوع این صفات است.

روش های مختلفی برای نرمال سازی داده وجود دارد، نرمال سازی Min-max، Z-score، و مقیاس ده دهی (decimal scaling) . نرمال سازی Min-max یک تغییر خطی روی داده اصلی ایجاد می کند. فرض کنید $min_a$ و $max_a$ مقادیر کمینه و بیشینه برای صفت Aهستند. نرمال سازی Min-max یک مقدار vA-v را در محدوده (0و1) اینگونه نگاشت می کند:

$$v' = \frac{v - min_a}{(max_a - min_a)} \quad (3)$$

در نرمال سازی Z-score، مقادیر برای یک صفت A بر اساس میانگین و انحراف معیار A نرمال می شود. A مقدار vA به v نرمال می شود اینگونه:

$$v' = \frac{v - \bar{A}}{\sigma_A} \quad (4)$$

که $\bar{A}$ و $\sigma_A$ میانگین و انحراف معیار صفت A هستند. این روش نرمال سازی مفید است وقتی که مقدار واقعی کمینه و بیشینه صفت A مجهول هستند، یا وقتی که پرت هایی هستند که بر نرمال سازی min-max غلبه می کند. نرمال سازی مقیاس ده دهی نقطه ده دهی مقادیر صفت A را منتقل می کند. تعداد نقاط ده دهی منتقل شده به مقدار محض بیشینهA بستگی دارد. مقدار vA به v نرمال می شود اینگونه:

$$v' = \frac{v}{10^j} \quad (5)$$

که j کوچکترین عدد صحیحی است که $Max(|v'|) < 1$ .

**الگوریتم توزیع شده نرمال کامینز(NDKM)**

DKM ابتدا در سایت های محلی با الگوریتمk-means خوشه بندی را انجام می دهد، سپس تمام مقادیر مراکز را به سایت مرکزی می فرستد، در آخر مقادیر مرکزی سراسری ، مجددا از k-means بدست می آید. NDKM نسخه توسعه یافته DKM هست، که برای استاندارد سازی اشیا داده ای به یک محدوده انجام می شود. رویه گام به گام الگوریتم پیشنهادی در زیر آمده است. ابتدا مقادیر بیشینه و کمینه بردارهای ویژگی از تمام دیتاست های محلی استخراج می شود و به یک مکان مرکزی ارسال می شود، که مقادیر بیشینه و کمینه مشخص شده اند. این دو مقدار برای انجام نرمال سازی سراسری با رویه نرمال سازیmin-maxبه سایت های محلی منتقل می شوند. سپس، اشیا نرمال شده برای ایجاد ماتریس مراکز و اندیس خوشه برای هر دیتاست،با k-means دسته بندی می شوند.. تمام مرکزهای محلی با الگوریتم k-means، برای گروه کردن مرکزهای مشابه و کسب مرکزهای سراسری دسته بندی و ادغام می شوند. مرکزهای سراسری به سایت محلی منتقل می شود، که فاصله اقلیدسی هر شی از مجموعه سراسری مرکزها محاسبه می شوند و به مرکز نزدیکترین خوشه منتسب می شود.الگوریتم زیر با تغییر دو گام اول، می تواند برای انواع دیگر رویه نرمال سازی پیاده سازی شود.

**Algorithm 5:** NDKM
**Input**: Homogeneous p datasets, each with d dimensions
**Output** :Global partitions of p datasets

آکادمی داده
dataacademy.ir

**Procedure**

Step 1: Find Maximum And Minimum Values of each feature froneach local dataset and Transmit them into Central Place
Step 2: Compute Maximum And Minimum Values at Central Place
Step 3: Normalize real Scalar Values of Local datasets with Global Maximum And Minimum Values
Step 4: Cluster each Local dataset By K-Means Algorithm and obtain Centroid Matrix Along with cluster index for each dataset
Step 5: Merge cluster centroids of local datasets into a single dataset named as centroids Dataset at Central Place
Step 6: Cluster centroids dataset using K-Means Algorithm to obtain Global Centroids
Step 7: Update local Cluster indices by assigning each object to nearest cluster centroid, After Computing Euclidean distance between the object and global centroid

**نتیجه گیری**

با توجه به رشد داده های توزیع شده و عدم جابه جایی این داده ها به یک مجموعه متمرکز ، داده کاوی توزیع شده می تواند بسیار مفید باشد.در این میان الگوریتم های متفاوتی برای این زمینه وجود دارد. در این مقاله روش های داده کاوی توزیع شده با الگوریتم کامینز را بررسی کردیم. برخی از این الگوریتم ها به دلیل اینکه به صورت موازی اجرا می گردند می توانند سرعت داده کاوی را به طور قابل ملاحظه ای افزایش دهند.هم چنین بنابر اهمیت امنیت داده در محیط های توزیع شده، الگوریتم های گروه محور می توانند بدون نیاز به جا به جا کردن اطلاعات در این محیط ها، داده کاوی را انجام دهند.

**منابع**

[1] Rekha Sunny T, Survey on Distributed Data Mining in P2P Networks
[2]K. A. Abdul Nazeer, Improving the Accuracy and Efficiency of the k-means Clustering Algorithm, 2009 Vol I
[3] Fahim A.M, Salem A. M, Torkey A and Ramadan M. A, “An Efficient enhanced k-means clustering algorithm,” Journal of Zhejiang University, 10(7):1626–1633, 2006
[4] Yuan F, Meng Z. H, Zhang H. X and Dong C. R, “A New Algorithm to Get the Initial Centroids,” Proc. of the 3rd International Conference on Machine Learning and Cybernetics, pages 26–29, August 2004.
[5] Malay K. Pakhira, Clustering Large Databases in DistributedEnvironment,2009
[6] N. KarthikeyaniVisalakshi, ENSEMBLE BASED DISTRIBUTED SOFTCLUSTERING,IEEE,2008
[7] Zhou A, Cao F, Van Y, Sha C, He X, "Distributed Data StreamClustering: A Fast EM-based Approach", ICDE 2007, IEEE 23rdInternational Conference on Data Engineering, pp. 736-745 ,April 2007
[8]SouptikDatta, Chris R. Approximate Distributed K-Means Clustering over a Peer-to-Peer Network,2009
[9] M.C. Naldi, Evolutionary k-means fordistributeddatasets,30-42,2014
[10] JiGenlin, Ling Xiaohan, Ensemble learning based distributed clustering, Lecture Notes In Computer Science, Springer-Verlag, Vol. 4819, pp. 312-321, November 2007.
[11] Amir Ben-Dor, Ron Shamir and Zohar Yakini, Clustering Gene Expression Patterns, Journal of Computational Biology, 6(3/4): 281-297, 1999
[12] Margaret H. Dunham, Data Mining- Introductory and Advanced Concepts, Pearson Education, 2006.

آکادمی داده
dataacademy.ir