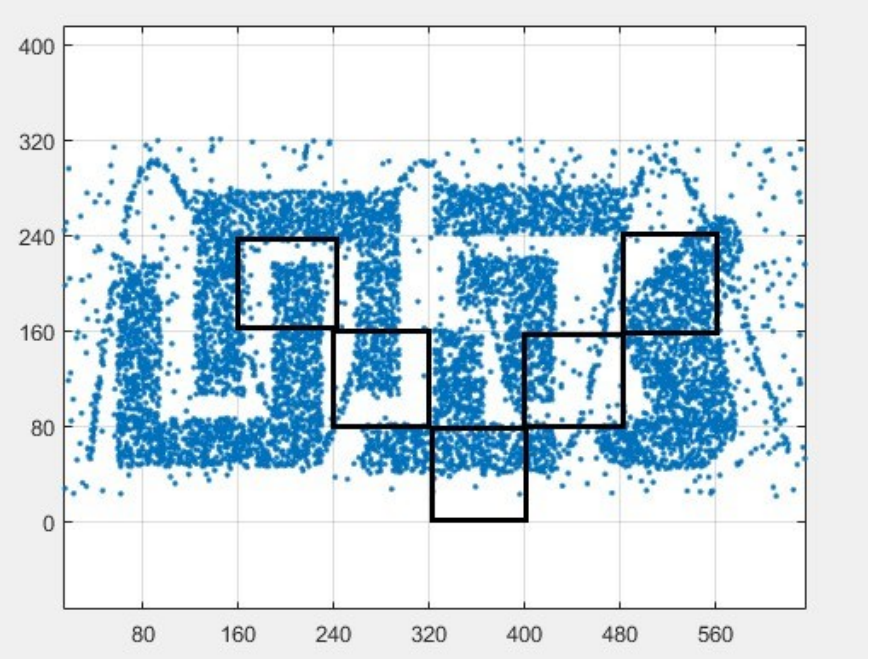
۱-۲-۳- علت انتخاب روش

ما در این روش به منظور کاهش زمان اجرا از روش گریدبندی نقاط استفاده کرده ایم و ابعاد اندازه ی گرید را نیز با توجه به تعداد نقاط انتخاب میکنیم که این موج شده است مرتبه زمانی به O(n) برسد. یکی از معایب الگوریتم DBSCAN، عدم وجود ترتیب مشخص در انتخاب نقاط است که این موج میشود در مواردی که خوشه ها، از لحاظ چگالی با یکدیگر تفاوت چشمگیری دارند ،نتواند نتیجه مطلوبی بدهد. بدین منظور ما برای خوشه بندی نقاط موجود در گرید از خانه هایی با چگالی بیشتر شروع می کنیم و ابتدا خانه های چگالتر را خوشه بندی می کنیم و هر چه چگالی خانه های گرید کمتر می شوند، ما پارامتر Eps را بیشتر می کنیم تا بتوانیم خوشه هایی که تعداد نقاط کمتری دارند را بخوبی بیابیم.

۲-۲-۳- تشریح کامل روش تحقیق

ابتدا با توجه به تعداد نقاط موجود در مجموعه داده ی مورد نظرمان، ابعاد گریدمان را $\sqrt[]{n} * \sqrt[]{n}$ در نظر می گیریم، که n معرف تعداد نقاط موجود در مجموعه داده است.
سپس هر یک از نقاط را در خانه های گریدمان قرار می دهیم و نقاط را گریدبندی می کنیم و گریدمان را تشکیل می دهیم. در شکل زیر میبینیم نقاط چگونه در هر یک از خانه های گریدمان قرار گرفته اند.

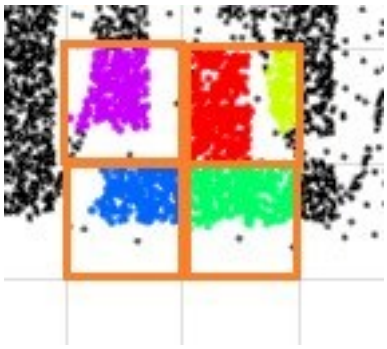


شکل ۳-۳: نحوه ی قرار گرفتن هر یک از نقاط در گرید

حال هر یک از خانه های گرید را با الگوریتم DBSCAN شروع به خوشه بندی می کنیم به این صورت که، در ابتدا خانه با بیشترین تعداد نقاط (بیشترین چگالی) را پیدا کرده و با Minpts و Eps مناسب خوشه

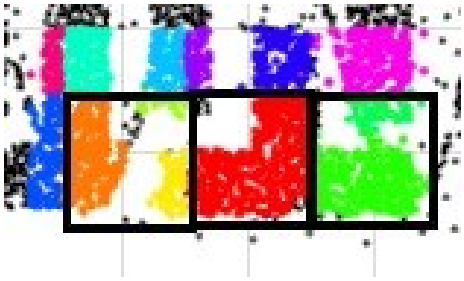
بندی می کنیم. برای خوشه بندی سایر خانه ها، ۵ مرحله در نظر می گیریم و در هر مرحله خانه هایی با چگالی مشخص را خوشه بندی می کنیم و در هر مرحله Eps را ۱۰ درصد افزایش می دهیم.

برای انتخاب بازهی مناسب در هر مرحله، از تعداد نقاط موجود در خانه ی گرید با بیشترین چگالی استفاده می کنیم. به این صورت که تعداد نقاط را بر تعداد مراحل تقسیم می کنیم و بازهی انتخابی در هر مرحله را بدست میاوریم. فرض کنید اگر تعداد نقاط خانه ی گرید ما که چگالتر از بقیه است، برابر ۱۰۰ باشد، اگر ۱۰۰ را بر ۵ تقسیم کنیم، بازه انتخابی ما ۲۰ خواهد بود، در نتیجه در مرحله اول خانه هایی از گرید انتخاب می شوند که، تعداد نقاطشان بین ۱۰۰ تا ۸۰ بوده، در مرحله دوم بایستی تعداد نقاطشان بین ۸۰ تا ۶۰ باشد و به همین ترتیب . در شکل زیر مرحله اول خوشه بندی را می بینیم.



شکل ۴-۳: شکل خوشه ها در مرحله ی اول

در مرحله ی بعد هریک از خانه های گرید را به اندازه نصف سایز گریدمان، به سمت راست و بالا انتقال می دهیم و مراحل بالا را تکرار می کنیم و دوباره عمل خوشه بندی را با استفاده از الگوریتم DBSCAN انجام میدهیم.



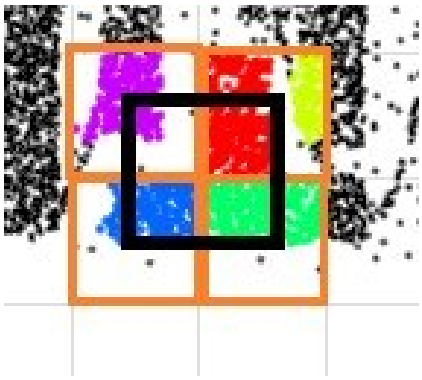
شکل ۵-۳: شکل خوشه ها در مرحله ی دوم

حال ما ۲ نتیجه داریم، بر اساس اشتراک خوشه ها، خوشه های نتیجه اولیه را ترکیب می کنیم به این صورت که اگر نتیجه اول را c1 و نتیجه دوم را c2 بنامیم

نقاطی که در c2 دارای یک شماره خوشه هستند ولی در c1 شماره خوشه متفاوتی دارند با در نظر گرفتن حداقل تعداد نقاط مشترک، خوشه های موجود در c1 را ترکیب می کنیم.

به طور مثال نقاط ۱و۲و۳و۴ در c1 دارای شماره خوشه ۲ هستند و همین نقاط در c2 در خوشه شماره ۵ قرار گرفته اند. پس در اینجا ما ۴ نقطه داریم که در مرحله اول در شماره ۲ و در مرحله دوم در شماره ۵ قرار گرفته اند، حال فرض کنید نقاط ۵و ۶و۷و ۸و ۹ در c۱ شماره خوشه ۳ را دارند و در c۲ شماره خوشه ۵، پس در اینجا۵ نقطه داریم که در مرحله اول در شماره ۳ و در مرحله دوم شماره۵ ، اینجا می توانیم ۲ خوشه ۲و ۳ در مرحله اول را ترکیب کنیم، ما حداقل تعداد اشتراک نقاط را برابر با Minpts در نظر می گیریم که در اینجا به طور مثال ۴ در نظر گرفته شده است.

در شکل زیر ۲ شکل ۴-۳و ۵-۳ را مقایسه کرده ایم .



شکل ۶-۳: مقایسه۲ مرحله

در شکل بالا مربع مشکی رنگ یک خانه از گریدی است که در مرحله ی دوم تشکیل شده است و در شکل (۵-۳) دیدیم نقاط موجود در آن چگونه خوشه بندی شد، پس ما می توانیم خوشه های سبز ،آبی و قرمز، بوجود آمده در مرحله اول را ترکیب کنیم.

در شکل زیر شبه کد روش پیشنهادی را می بینیم.

```
Algorithm GM2- DBSCAN
Input: D, Minpts, Eps
Output: Set of clusters
  begin
  create grid ∛n * ∛n
  for (each) unvisited point p in the dataset D do
    Put point p in grid
  c1 = grid_clustring (grid)
  grid = we shift the grid
  c۲ = grid_clustring (grid)
  cluster_num_c1, cluster_num_c2 = cluster num in c1, cluster num in c2
  create compare_result = matrix (cluster_num_c1, cluster_num_c2)
   for each point in D
    compare_result(c1(point), c2(point)) += 1
   end for
   for each column in compare_result
     rows = find which rows that have at least Minpts in compare_result (row, column)
     combine this rows together in c1
   end for
   return c1
```

شکل ۷-۳: شبه کد روش پیشنهادی

```
Function grid_clustring (grid)
  for each cell in grid with maximum point
    result = do DBSCAN (points, Eps, Minpts) clustering for cells point
  end for
  max_point = point number in cell with maximum points
  point_rang = max_point/5
  while i<=5
    Eps = Eps + 0.1*Eps
    for each unvisited cell with sum points in rang (max_point - point_rang, max_point + point_rang)
      result += do DBSCAN (points, Eps, Minpts) clustering for cells point
    end for
    i +=1
  end while
  return result
```

شکل ۸-۳: شبه کد روش پیشنهادی

در فصل بعدی به ارزیابی الگوریتم پیشنهادی می پردازیم و بر روی ۷ مجموعه داده ی مختلف ۲ الگوریتم، روش پیشنهادی و DBSCAN را از لحاظ نتیجه ی خوشه بندی و مدت زمان اجرا با یکدیگر مقایسه می کنیم