سید حامد سادات 400521405

بخش اول:

الگوریتم ژنتیک داخل یک کلاس به همین اسم پیاده سازی شده است

نکات مهم درباره ساخت جمعیت اولیه:

جمعیت اولیه به تعداد گفته شده در آرگومان تابع ژنوم تعریف کرده

و در هر جایگاه هر ژنوم آن یک عدد رندوم بین 1 و تعداد خوشه تعریف شده قرار میدهد

یک قابلیت به نام dynamic clustering به این الگوریتم اضافه شده که اجازه میدهد که بجای تعیین ثابت تعداد خوشه ها،یک بازه برای آن تعریف کرد که هنگام ساخت اولیه ژنوم،یک عدد بین این بازه برای تعداد خوشه های ژنوم مورد نظر انتخاب شود

البته که میتوان بین حالت عادی این حالت،با دادن مقدار true یا false حالت مورد نظر را انتخاب کرد

تابع fitness:

این تابع مقدار fitness برای هر ژنوم را بعد صدا زدن برمیگرداند

برای محاسبه میزان مفید بودن هر ژنوم از تابع Silhouette Coefficient استفاده شده که با کمک فاصله از میانگین هر خوشه و فاصله از میانگین نزدیک ترین خوشه محاسبه میشود

و در نهایت در ازای هر ژنوم،یک عدد بین -1 و 1 میدهد که هرچه این عدد به 1 نزدیک تر باشد یعنی خوشه بندی دقیقتر است

و برای راحتتر کار کردن با نتیجه تابع fitness تمام اعداد بدست آمده بین 1 تا 10 scale میشوند(بعدا برای crossover استفاده میشود)

تابع crossover:

این تابع مسئولیت ساخت نسل بعدی را به عهده دارد

اول مقدار fitness برای همه ژنوم ها را گرفته و تعداد اعضای نسل بعد را میگیرد(بازه حداقل و حداکثر به عنوان آرگومان داده شده و باید متناسب با داده اولیه باشد،سپس از بین این بازه یک عدد به را عنوان تعداد اعضای نسل بعد انتخاب میکند)

سپس دوبرابر این تعداد،والدین را انتخاب میکنیم که دو به دو با هم ترکیب شوند
نحوه انتخاب والدین به این گونه هست که به ژنوم هایی که تیجه تابع fitness آنها بزرگتر است،به میزان مقدار تابع fitness احتمال انتخاب شدن بالاتری دارند

نحوه crossover تا حد امکان شبیه به چینش کروموزوم ها در انسان در آمده که به شکل زیر است
تعداد عناصر گرفته شده از والدین یکسان بوده اما ترتیب گرفتن آنها معلوم نیست

یعنی ممکن است 2 تای اول از والد1، 3 تای بعدی از والد2، 1دونه بعدی از والد6،1 تای بعدی از والد2 و... باشد

اما مطمئن هستیم که تعداد کروموزوم های گرفته شده از هر والد برابر است

اگر از dynamic clustering استفاده شده باشه،در آخر،خوشه تعداد خوشه بزرگتر به فرزند داده میشه

برای جهش هم دو خوشه های دو کروموزوم به صورت رندوم جابجا میشود

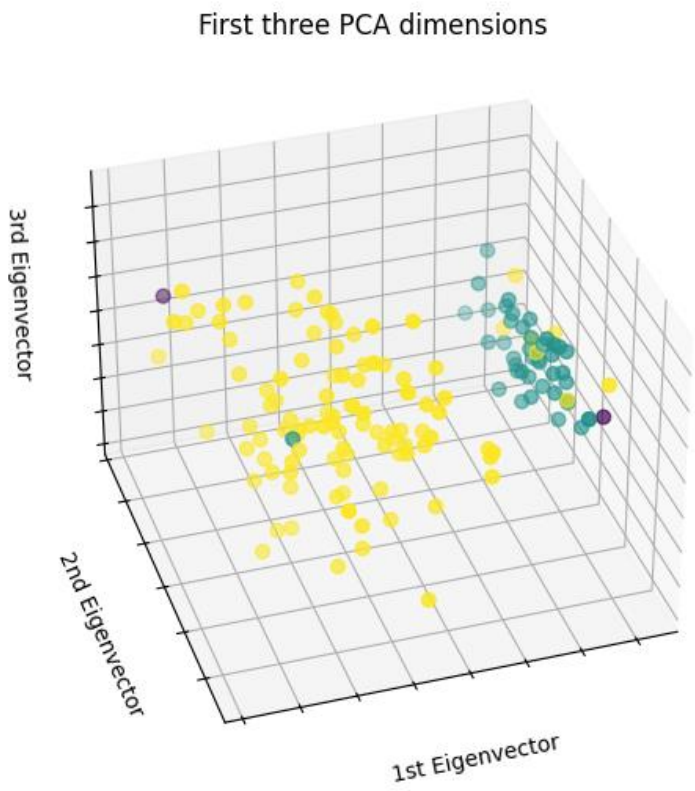
تابع clustering مسئولیت خوشه بندی داده ها را دارد

تعداد نسل ها،منظور از تعداد iteration ها و تعداد crossover های کل ژنوم ها هست که به طور کلی یعنی چند بار میخواهیم روی ژنوم ها crossover انجام شود

همچنین این تابع درصد کار انجام شده را نیز در ترمینال نشان میدهد و بعد از اتمام دسته بندی،از آخرین نسل بهترین ژنوم را بر حسب تابع fitness انتخاب کرده و برمیگرداند

نمایش دسته بندی به صورت سه بعدی از 3 ویژگی داده ها است

نتیجه نهایی:



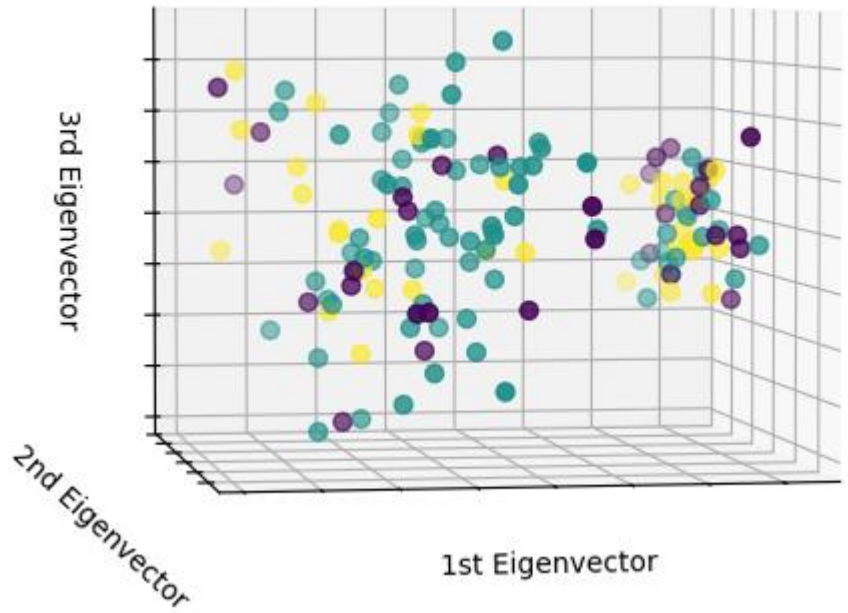
نتیجه نهایی:

با دادن تعداد بالای iteration ها،دقت دسته بندی بالاتر رفته اما معمولا دو دسته درست میکند

به دلیل ماهیت رندوم بودن الگوریتم ژنتیک،در هر بار اجرای کد،ممکن است نتایج متفاوت باشد و هرچه تعداد iteration کوچکتر باشد،ممکن است خوشه بندی انجام شده خیلی دقیق نباشد

اما معمولا در تعداد iter بالا،کلیت خوشه بندی ثابت و درست است و فقط در جزئیات تفاوت وجود دارد

در مقایسه با kmeans باید گفت که kmeans خیلی بهتر از GA در خوشه بندی است و با توجه به کمتر بودن زمان predict آن،دقتش بیشتر و تحت اجرا های مختلف پایدار تر است و خروجی ثابتی میدهد



(این حالت جز دفعه هایی است که تقریبا به سه دسته تقسیم کرده اما دسته بندی آن دقت بالایی ندارد)