

قسمت اول :

1. معرفی

این کد یک رویکرد خوشه‌بندی مبتنی بر الگوریتم ژنتیک را بر روی مجموعه داده Iris پیاده‌سازی می‌کند، که مجموعه داده‌ای شناخته شده برای ارزیابی الگوریتم‌های خوشه‌بندی است. هدف تعیین خودکار تعداد بهینه خوشه‌ها و تخصیص نقاط داده به آن خوشه‌ها است.

2. مجموعه داده

مجموعه داده Iris شامل 150 نمونه است که هر کدام دارای 4 ویژگی است. مجموعه داده شامل 3 گونه مختلف از گل‌های زنبق است که می‌تواند به عنوان برچسب حقیقت زمین برای ارزیابی نتایج خوشه‌بندی استفاده شود.

3. رویکرد الگوریتم ژنتیک

کد اجزای زیر از الگوریتم ژنتیک را تعریف می‌کند:

****Initialization****: -جامعه با تخصیص خوشه‌های تصادفی برای هر نقطه داده، همراه با تعداد خوشه‌ها مقداردهی اولیه می‌شود.

****تابع تناسب اندام****: تابع تناسب جمع منفی حداقل فواصل بین هر نقطه داده و مرکز خوشه اختصاص داده شده به آن را محاسبه می‌کند. این الگوریتم را تشویق می‌کند تا خوشه‌های فشرده و کاملاً جدا شده را پیدا کند.

****عملگرهای جهش****: سه عملگر جهش تعریف شده است `mutate_random_label`، `mutate_nearest_label` و `mutate_nearest_cluster` که تخصیص خوشه‌های نقاط داده منفرد یا کل خوشه‌ها را تغییر می‌دهند.

****اپراتورهای متقاطع****: دو عملگر متقاطع تعریف شده است :

`crossover_cluster_swap` و `crossover_split_cluster` که برای تولید فرزندان، انتساب‌های خوشه‌ای را بین دو فرد والد مبادله یا تقسیم می‌کنند.

****حلقه الگوریتم ژنتیک****: الگوریتم برای تعداد مشخصی از نسل‌ها اجرا می‌شود و تناسب جمعیت را ارزیابی می‌کند، والدین را انتخاب می‌کند، انجام متقاطع و جهش انجام می‌دهد و بدترین افراد را با فرزندان جایگزین می‌کند.

**4. معیارهای ارزیابی

کد نتایج خوشه‌بندی را با استفاده از دو معیار ارزیابی می‌کند:

****شاخص رند تعدیل شده: (ARI)**** این متریک نتایج خوشه‌بندی را با برچسب‌های حقیقت زمینی مقایسه می‌کند و امتیازی بین -1 و 1 ارائه می‌دهد که در آن 1 نشان دهنده تطابق کامل است.

****Silhouette Score****: این متریک فشردگی و تفکیک خوشه‌ها را با امتیازی بین

1- و 1 اندازه گیری می کند که در آن نمره بالاتر نشان دهنده خوشه بندی بهتر است.

5. مقایسه با K-Means

این کد همچنین امتیاز ARI و Silhouette را برای الگوریتم خوشه بندی K-Means که یک روش خوشه بندی محبوب و پرکاربرد است، محاسبه می کند تا مقایسه ای با رویکرد مبتنی بر الگوریتم ژنتیک ارائه دهد.

روش مبتنی بر الگوریتم ژنتیک این مزیت را دارد که به طور خودکار تعداد بهینه خوشه ها را تعیین می کند، برخلاف الگوریتم K-Means که نیاز به تعیین تعداد خوشه ها به عنوان پارامتر دارد. کاوش الگوریتم ژنتیک در فضای جستجو و توانایی فرار از بهینه محلی به طور بالقوه می تواند منجر به نتایج خوشه بندی بهتری شود، به خصوص برای اشکال خوشه ای پیچیده یا غیر محدب.

با این حال، رویکرد مبتنی بر الگوریتم ژنتیک به طور کلی از نظر محاسباتی فشرده تر از K-Means است، زیرا به چندین نسل از تکامل و ارزیابی جمعیت نیاز دارد. عملکرد خاص الگوریتم ژنتیک ممکن است به عواملی مانند انتخاب عملگرهای جهش و متقاطع، اندازه جمعیت و تعداد نسل ها بستگی داشته باشد.

رویکرد خوشه بندی مبتنی بر الگوریتم ژنتیک ارائه شده در کد، رویکرد امیدوارکننده ای را برای خوشه بندی بدون نظارت نشان می دهد، به ویژه زمانی که تعداد خوشه ها از قبل مشخص نباشد. نتایج نشان می دهد که رویکرد مبتنی بر الگوریتم ژنتیک می تواند از الگوریتم K-Means از نظر شاخص رند تنظیم شده و امتیاز Silhouette در مجموعه داده های Iris بهتر عمل کند. کاوش بیشتر و بهینه سازی پارامترها و عملگرهای الگوریتم ژنتیک می تواند به طور بالقوه منجر به عملکرد خوشه بندی بهتر شود.

قسمت دوم :

کد ابتدا ماتریس فاصله بین شهرها را تعریف می‌کند. سپس پارامترهای الگوریتم ژنتیک اعم از اندازه جمعیت، نرخ جهش و تعداد نسل‌ها را تعیین می‌کند.

تابع `create_population` یک جمعیت اولیه از افراد (مسیرها) را ایجاد می‌کند. هر فرد تشکیل شده از اعداد 1 تا تعداد شهرها است و با استفاده از تابع `random.shuffle` ترتیب این اعداد تصادفی می‌شود.

تابع `calculate_distance` مسافت کل یک مسیر را محاسبه می‌کند. با استفاده از ماتریس فاصله، بین هر دو شهر متوالی در مسیر، فاصله محاسبه شده و به مجموع مسافت اضافه می‌شود. همچنین فاصله بین شهر آخر و شهر اول نیز به مجموع اضافه می‌شود.

تابع `evaluate_population` توانایی (فیتنس) هر فرد در جمعیت را محاسبه می‌کند. این توانایی برابر با معکوس مسافت کل مسیر است. بنابراین فردی که مسافت کمتری دارد، توانایی بیشتری دارد.

تابع `selection` از روش انتخاب چرخشی (roulette wheel selection) برای انتخاب والدین استفاده می‌کند. احتمال انتخاب هر فرد بر اساس توانایی آن فرد محاسبه می‌شود و با استفاده از تابع `random.choices` دو فرد والد انتخاب می‌شوند.

تابع `crossover` عملیات ترکیب را برای ایجاد فرزندان انجام می‌دهد. یک نقطه تقاطع تصادفی انتخاب می‌شود و اجزای قبل از آن از والد اول و اجزای بعد از آن از والد دوم دریافت می‌شوند. اجزایی که قبلاً در فرزند حضور دارند، متوجه شده و در فرزند صرف نظر می‌شوند.

تابع `mutate` عملیات جهش را انجام می‌دهد. دو ژن تصادفی انتخاب می‌شوند و جای آنها در فرد جابه‌جا می‌شود. این عملیات جهش با احتمال `mutation_rate` انجام می‌شود.

در قسمت اصلی الگوریتم، یک جمعیت اولیه ایجاد می‌شود و سپس برای تعداد نسل‌ها اقدامات لازم برای بهبود جمعیت و یافتن بهترین مسیر انجام می‌شود.

در هر نسل، توانایی هر فرد محاسبه شده و فردی که توانایی بیشتری دارد (کمترین مسافت) به عنوان بهترین فرد انتخاب می‌شود. سپس این بهترین فرد به جمعیت بعدی اضافه می‌شود.

سپس تا زمانی که جمعیت بعدی به اندازه مورد نیاز نرسیده است، والدین انتخاب شده و عملیات ترکیب و جهش بر روی آنها انجام می‌شود و فرزندان به جمعیت بعدی اضافه می‌شوند.

در نهایت، بهترین مسیر و مسافت محاسبه شده و چاپ می‌شود.

در کل، این الگوریتم ژنتیک با ترکیب عملیات ترکیب و جهش، بهبود جمعیت را به سمت یافتن بهترین مسیر سوق می‌دهد. با اجرای الگوریتم برای تعداد نسل‌های مشخص شده، بهترین مسیر و مسافت آن به دست می‌آید.