دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضیات و علوم کامپیوتر

تحلیل قابلیت های خوشه بندی NMF

پروژه پایانی درس جبر خطی عددی

آرمان صالحی

استاد درس:

دکتر مهدی دهقان

زمستان ۱۴۰۴

چکیده

چکیده در این پروژه، به بررسی و پیاده‌سازی الگوریتم تجزیه نامنفی ماتریس (NMF) به عنوان یک روش قدرتمند برای کاهش ابعاد و خوشه‌بندی داده‌ها پرداخته شده است. هدف اصلی، اعمال این الگوریتم بر روی مجموعه داده ارقام دست‌نویس (MNIST) و استخراج الگوهای پایه سازنده این ارقام است. برخلاف روش‌های سنتی مانند K-means که بر اساس میانگین‌گیری از کل داده عمل می‌کنند، NMF با تجزیه داده‌ها به مولفه‌های نامنفی، رویکردی "بخش محور" ارائه می‌دهد که تفسیرپذیری بالایی دارد. در بخش عملی، نتایج حاصل از NMF با الگوریتم K-means مقایسه شده است. نتایج بصری نشان می‌دهد که پایه‌های به دست آمده از NMF شبیه به اجزای تشکیل‌دهنده اعداد (مانند خطوط و کمان‌ها) هستند، در حالی که مراکز خوشه‌های K-means شبیه به میانگین محو شده اعداد کامل می‌باشند. همچنین دقت خوشه‌بندی هر دو روش به صورت کمی محاسبه و مقایسه گردید.

کلمات کلیدی: جبر خطی عددی، تجزیه نامنفی ماتریس(NMF) ، خوشه‌بندی K-means ، داده‌کاوی، مجموعه داده MNIST

فهرست مطالب

فصل اول: مقدمه و کلیات

۱-۱ مقدمه

در عصرحاضر، با رشد روزافزون داده‌های دیجیتال، نیاز به روش‌های خودکار برای تحلیل و استخراج اطلاعات از این داده‌ها بیش از پیش احساس می‌شود. یکی از حوزه‌های اصلی در هوش مصنوعی و یادگیری ماشین، یادگیری نظارت‌نشده (Unsupervised Learning) است که در آن الگوریتم بدون داشتن برچسب‌های از پیش تعیین‌شده، سعی در کشف ساختارهای پنهان و الگوهای موجود در داده‌ها دارد.

خوشه‌بندی (Clustering) به عنوان یکی از مهم‌ترین تکنیک‌های یادگیری نظارت‌نشده، فرآیندی است که در آن داده‌ها به گروه‌هایی تقسیم می‌شوند که اعضای هر گروه بیشترین شباهت را به یکدیگر و بیشترین تفاوت را با اعضای سایر گروه‌ها داشته باشند. این تکنیک در کاربردهای متنوعی از جمله پردازش تصویر، فشرده‌سازی داده‌ها، و تشخیص الگو کاربرد دارد.

در این پروژه، تمرکز ما بر روی تحلیل و خوشه‌بندی تصاویر ارقام دست‌نویس است. چالش اصلی در پردازش این تصاویر، ابعاد بالای داده‌ها و تغییرات زیاد در نحوه نوشتن ارقام توسط افراد مختلف است. برای غلبه بر این چالش، از روش‌های کاهش ابعاد و تجزیه ماتریس استفاده می‌شود.

۱-۲ بیان مسئله

مسئله‌ی مورد بررسی در این پژوهش، خوشه‌بندی تصاویر مجموعه داده MNIST است. این مجموعه داده شامل تصاویر سیاه و سفید از ارقام دست‌نویس (۰ تا ۹) است که هر کدام در یک شبکه پیکسلی مشخص (۲۸ در ۲۸ پیکسل) قرار دارند.

هدف اصلی این است که با استفاده از الگوریتم‌های جبر خطی عددی، این تصاویر را پردازش کرده و الگوریتم بتواند بدون دانستن برچسب واقعی تصاویر (یعنی بدون اینکه بداند کدام عکس عدد ۵ است یا ۷)، آن‌ها را بر اساس ویژگی‌های ظاهری دسته‌بندی کند. همچنین، استخراج ویژگی‌های پایه‌ای که این ارقام را تشکیل می‌دهند (مانند خطوط عمودی، افقی و حلقه‌ها) بخش مهمی از این پژوهش است.

۳-۱ معرفی روش های مورد استفاده

برای انجام این پروژه، از دو رویکرد متفاوت استفاده و نتایج آن‌ها مقایسه خواهد شد:

۱. تجزیه نامنفی ماتریس (NMF): این روش یکی از تکنیک‌های قدرتمند در جبر خطی است که ماتریس داده‌ها را به دو ماتریس با درایه‌های نامنفی تجزیه می‌کند. ویژگی منحصربه‌فرد NMF این است که به دلیل قید نامنفی بودن، اجزای سازنده داده‌ها را به صورت «بخش‌محور (Parts-based) «یاد می‌گیرد. این یعنی NMF سعی می‌کند تصاویر را به عنوان ترکیبی از اجزای کوچکتر (مانند تکه‌های قلم) بازسازی کند.

۲. الگوریتم K-means: این الگوریتم یکی از کلاسیک‌ترین روش‌های خوشه‌بندی است که بر اساس فاصله اقلیدسی و میانگین‌گیری عمل می‌کند. برخلاف NMF، روش K-means معمولاً رویکردی «کل‌محور» دارد و مراکز خوشه‌ها در آن شبیه به میانگین کلی تصاویر یک دسته هستند.

شکل ۱-۱: نمونه‌هایی از تنوع دست‌خط‌ها در مجموعه دادهMNIST

٤-١ ساختار گزارش

این گزارش در چهار فصل تدوین شده است. پس از این مقدمه، در فصل دوم به مبانی نظری و ریاضیاتی تجزیه NMF و الگوریتم K-means خواهیم پرداخت. در فصل سوم، پیاده‌سازی عملی الگوریتم‌ها بر روی داده‌های MNIST شرح داده شده و نتایج کمی و کیفی (شامل تصاویر پایه‌ها و دقت خوشه‌بندی) تحلیل می‌شوند. در نهایت، در فصل چهارم نتیجه‌گیری کلی پروژه و پیشنهادات ارائه خواهد شد.

فصل دوم: مبانی نظری و پیشینه تحقیق

۲-۱ مقدمه

در این فصل به بررسی دقیق ریاضیاتی الگوریتم‌های مورد استفاده در پروژه می‌پردازیم. ابتدا الگوریتم تجزیه نامنفی ماتریس (NMF) و فرمول‌بندی آن در جبر خطی را تشریح کرده و سپس به الگوریتم خوشه‌بندی K-means خواهیم پرداخت. در نهایت، تفاوت بنیادین این دو روش در نحوه نمایش داده‌ها بررسی می‌شود.

۲-۲ تجزیه نامنفی ماتریس (NMF)

تجزیه نامنفی ماتریس (Non-negative Matrix Factorization) گروهی از الگوریتم‌ها در تحلیل چندمتغیره و جبر خطی است که در آن یک ماتریس X به دو ماتریس H و W تجزیه می‌شود، با این قید مهم که تمام درایه‌های این سه ماتریس باید نامنفی (بزرگتر یا مساوی صفر) باشند.

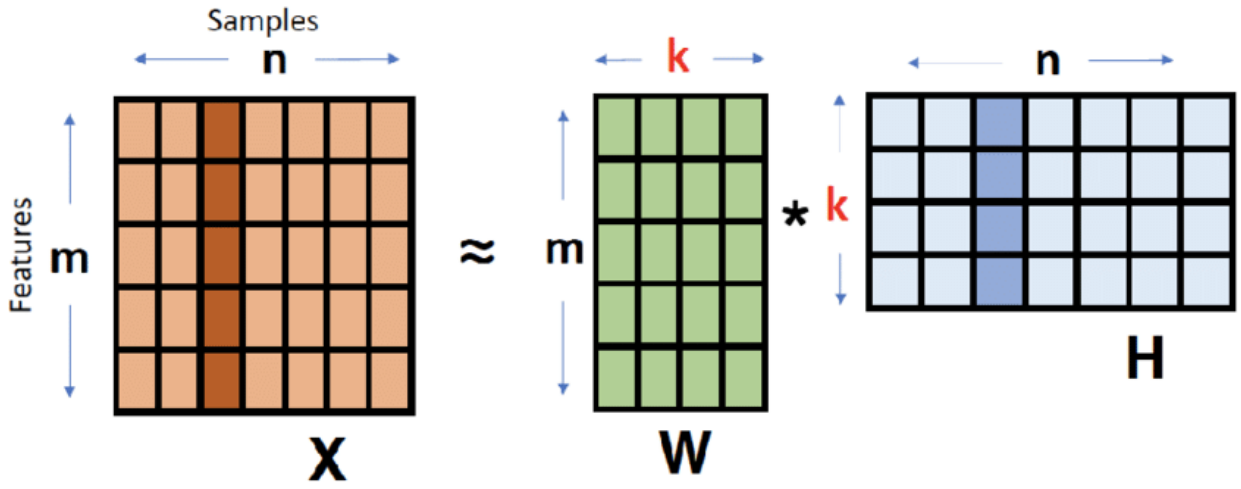
۱-۲-۲ فرمول‌بندی ریاضی

فرض کنید مجموعه داده‌های ما به صـورت ماتریس x با ابعاد $n \times m$ باشـد که در آن n تعداد ویژگی ها (پیکسل ها) و m تعداد نمونه ها (تصاویر) است. هدف NMF یافتن دو ماتریس H و W است بطوریکه:

$$X \approx WH$$

که در آن:

- ماتریس $X \in \mathbb{R}^{n \times m}$: ماتریس داده‌های ورودی (شامل درایه‌های نامنفی).
- ماتریس $W \in \mathbb{R}^{n \times r}$: ماتریس ویژگی‌های پایه. سـتون‌های این ماتریس همان «الگوهای پایه» یا اجزای سازنده تصاویر هستند.
- ماتریس $H \in \mathbb{R}^{r \times m}$: ماتریس ضرایب یا کدگذاری. هر سـتون نشـان می‌دهد که چه ترکیبی از پایه‌ها برای ساختن تصویر اصلی نیاز است.
- پارامتر r: رتبه تجزیه یا تعداد مولفه‌ها که معمولاً بسیار کوچکتر از n و m انتخاب می‌شود.



۲-۲-۲ قید نامنفی بودن و تفسیرپذیری

تفاوت اصلی NMF با روش‌هایی مانند PCA (تحلیل مولفه‌های اصلی) در قید زیر است:

$$X_{ij} \geq 0, \qquad W_{ik} \geq 0, \qquad H_{kj} \geq 0$$

این قید باعث می‌شود که بازسازی داده‌ها تنها از طریق جمع مولفه‌ها صورت گیرد و هیچ تفریق کردنی وجود نداشته باشد. به عبارت دیگر، یک تصویر به صورت ترکیبی از روی هم قرار گرفتن چندین لایه (پایه) ساخته می‌شود. این ویژگی باعث می‌شود پایه‌های به دست آمده W شبیه به اجزای معنادار تصویر (مانند خطوط، گوشه‌ها و منحنی‌ها در ارقام) باشند.

۲-۲-۳ تابع هزینه و به‌روزرسانی

برای پیدا کردن بهترین ماتریس‌های Wو H، باید فاصله بین ماتریس اصلی X و ماتریس بازسازی شده WH را کمینه کنیم. رایج‌ترین تابع هزینه، نرم فروبنیوس است:

$$\min_{W,H} \frac{1}{2} ||X - WH||_F^2$$

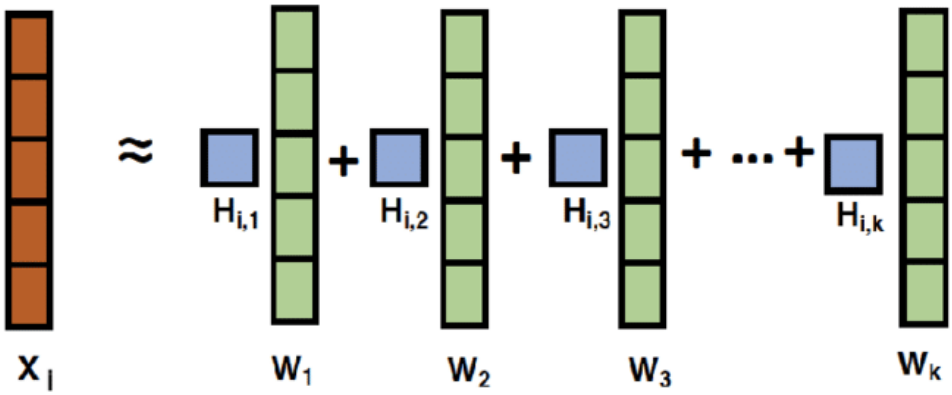
$$subject\ to\ W, H \geq 0$$

از آنجا که این تابع نسبت به هر دو متغیر W و W محدب نیست، راه‌حل بسته ریاضی ندارد. برای حل آن معمولاً از روش‌های تکرارشه (Iterative) استفاده می‌شود. یکی از معروف‌ترین الگوریتم‌ها، «قوانین به‌روزرسانی ضربی‌پیشنهاد شده توسط لی و سونگ (Lee & Seung) است:

$$H_{kj} \leftarrow H_{kj} \frac{(W^T X)_{kj}}{(W^T WH)_{kj}}$$

$$W_{ik} \leftarrow W_{ik} \frac{(XH^T)_{ik}}{(WHH^T)_{ik}}$$

این به‌روزرسانی‌ها تضمین می‌کنند که اگر ماتریس‌های اولیه نامنفی باشند، در تمام تکرارها نامنفی باقی بمانند و تابع هزینه به صورت یکنواخت کاهش یابد.



۲-۳ الگوریتم خوشه بندی K-means

الگوریتم K-means یکی از ساده‌ترین و پرکاربردترین الگوریتم‌های خوشه‌بندی است که هدف آن تقسیم m داده ورودی به k خوشه مجزا است، به طوری که واریانس درون هر خوشه کمینه شود.

۲-۳-۱ تابع هدف

این الگوریتم سعی می‌کند تابع هدف J که مجموع مربعات خطای فاصله هر نقطه تا مرکز خوشه‌اش است، کمینه کند:

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left|\left|x_i - \mu_j\right|\right|^2$$

که در آن:

- k : تعداد خوشه‌ها
- C_j : مجموعه نقاط متعلق به خوشه j
- μ_j : میانگین نقاط خوشه j
- $||.||$: فاصله اقلیدسی استاندارد

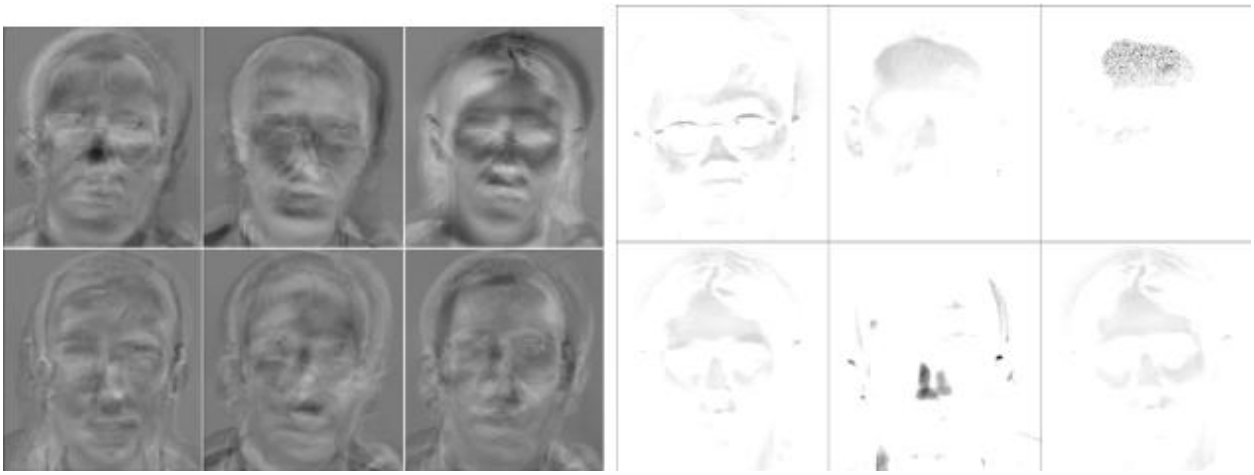
در زمینه پردازش تصویر، مراکز خوشه‌ها (μ_j) نماینده «میانگین کلی» تصاویری هستند که در آن خوشه قرار گرفته‌اند. برخلاف NMF که تصویر را از ترکیب اجزا می‌سازد، K-means برای هر تصویر فقط یک مرکز (نماینده) در نظر می‌گیرد.

٤-٢ مقایسه نظری: جزءنگر در برابر کل‌نگر

مهم‌ترین تفاوت نظری بین NMF و K-means در نحوه نمایش اطلاعات است:

۱. **رویکرد K-means (کل‌نگر):** این روش از نوع Vector Quantization است. هر داده تنها با یک بردار مرجع (مرکز خوشه) تقریب زده می‌شود. بنابراین مراکز خوشه‌ها باید شبیه به یک نسخه کامل (اما میانگین‌گیری شده) از داده‌ها باشند.

۲. **رویکرد NMF (جزءنگر):** این روش داده را به صورت ترکیب خطی چندین بردار پایه تقریب می‌زند($x \approx \sum w_i h_i$). چون ضرایب منفی مجاز نیستند، الگوریتم مجبور است ویژگی‌هایی را یاد بگیرد که «اجزای سازنده» داده‌ها باشند. این تفاوت باعث می‌شود که در کاربرد تشخیص ارقام، NMFپایه‌هایی شبیه به قلم‌موی نوشتن تولید کند، در حالی که K-means تصاویری شبیه به ارقام کامل اما مات تولید می‌کند.



مقایسه بصری پایه‌های استخراج شده. (چپ) پایه‌های PCA که به صورت چهره‌های کامل اما محو هستند. (راست) پایه‌های NMF که به صورت اجزای موضعی و تفکیک‌شده مثل بینی یا ابرو ظاهر شده‌اند.

فصل سوم: پیاده‌سازی و تحلیل نتایج

۳-۱ معرفی مجموعه دادهMNIST

در این پروژه از مجموعه داده استاندارد MNIST استفاده شده است. این مجموعه شامل ۷۰,۰۰۰ تصویر از ارقام دست‌نویس (۰ تا ۹) است که به طور گسترده برای آموزش و تست سیستم‌های یادگیری ماشین استفاده می‌شود.

- **تعداد نمونه‌ها:** ۷۰,۰۰۰ تصویر (۶۰,۰۰۰ داده آموزش و ۱۰,۰۰۰ داده تست).
- **ابعاد تصاویر:** هر تصویر سیاه و سفید و دارای ابعاد ۲۸ در ۲۸ پیکسل است.
- **فرمت داده:** هر تصویر به صورت یک بردار مسطح شده با طول ۷۸۴ (28×28) نمایش داده می‌شود تا بتواند به عنوان ورودی به الگوریتم‌های ماتریسی داده شود.

۳-۲ ابزارها و محیط پیاده‌سازی

برای پیاده‌سازی این پروژه از زبان برنامه‌نویسی **Python** استفاده شده است. جهت مدیریت وابستگی‌ها و محیط مجازی، از ابزار **Poetry** استفاده گردید تا تکرارپذیری نتایج تضمین شود. کتابخانه‌های اصلی مورد استفاده عبارتند از:

۱. **Scikit-learn:** برای اجرای الگوریتم‌های NMF و K-means و محاسبه معیارهای ارزیابی.

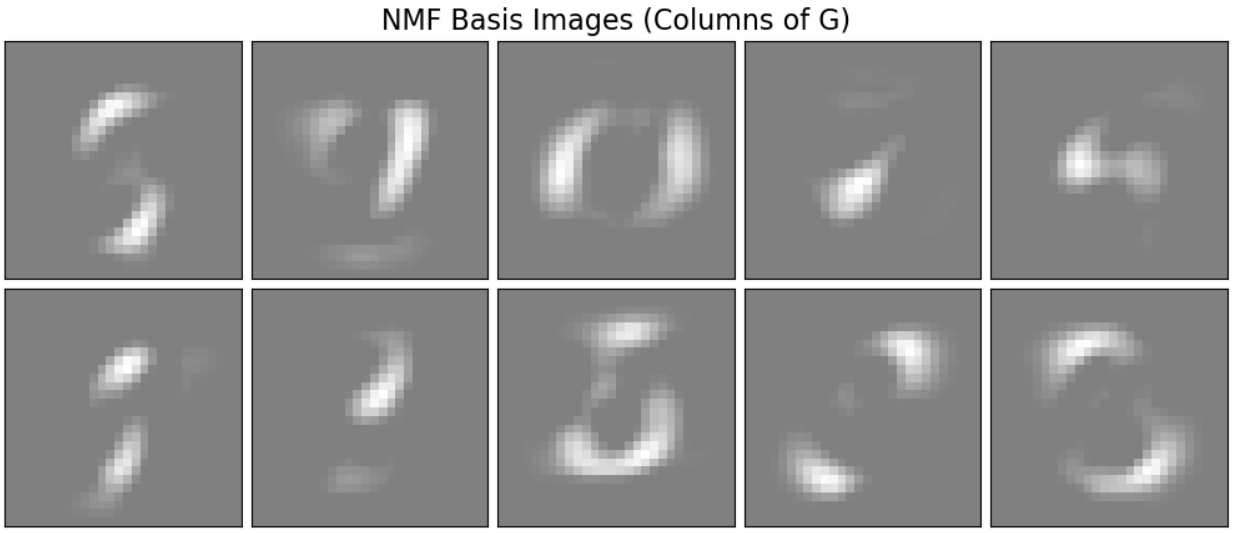
۲. **NumPy:** برای عملیات ماتریسی و برداری.

۳. **Matplotlib:** برای رسم نمودارها و نمایش تصاویر خروجی.

۳-۳ نتایج تجزیه نامنفی ماتریس (NMF)

در این بخش، الگوریتم NMF با تعداد مولفه‌های ۱۰ ($n_{components} = 10$) بر روی کل داده‌های نرمال‌سازی شده اجرا گردید. هدف از این کار، یافتن ۱۰ الگوی پایه‌ای است که تمامی ارقام دیگر از ترکیب آن‌ها ساخته می‌شوند.

شکل زیر خروجی حاصل از ماتریس H (تصاویر پایه) را نشان می‌دهد:



شکل ۱-۳: تصاویر پایه استخراج شده توسط الگوریتم NMF

تحلیل کیفی نتایج NMF

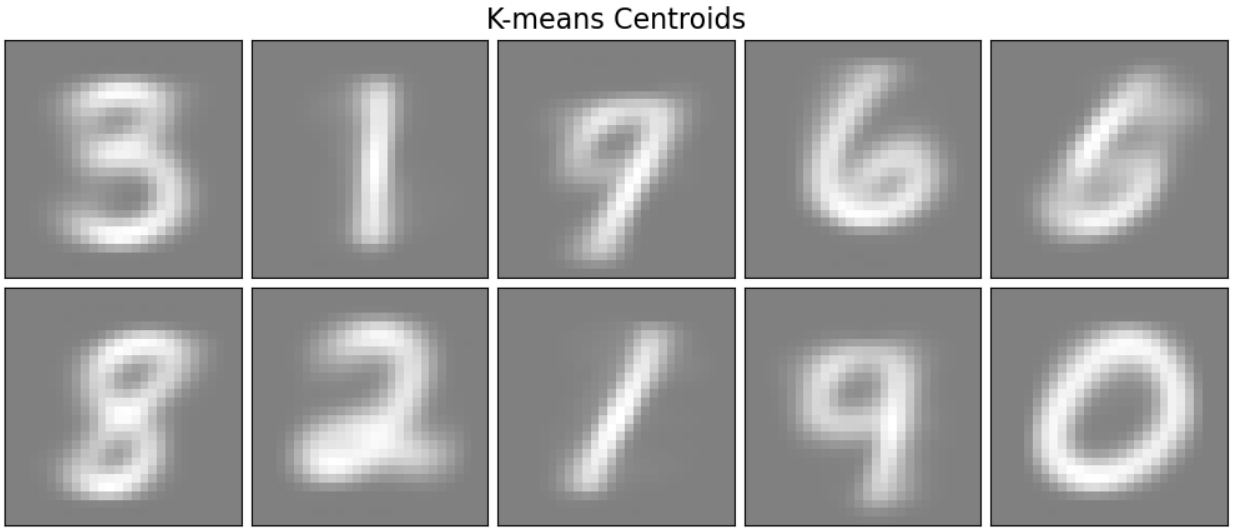
همان‌طور که در شکل ۱-۳ مشاهده می‌شود، پایه‌های به دست آمده شبیه به ارقام کامل نیستند، بلکه شبیه به اجزای سازنده ارقام هستند. برای مثال:

- برخی پایه‌ها تنها یک خط کج یا افقی را نشان می‌دهند.
- برخی دیگر شبیه به کمان‌ها یا حلقه‌هایی هستند که در اعداد ۸، ۰ یا ۹ دیده می‌شوند.

این نتیجه دقیقاً خاصیت Parts-based بودن NMF را تایید می‌کند. الگوریتم یاد گرفته است که برای ساختن یک عدد، باید این اجزا را با هم جمع کند.

۳-٤ نتایج خوشه‌بندیK-means

برای مقایسه، الگوریتم K-means نیز با ۱۰ خوشه روی همان داده‌ها اجرا شد. مراکز خوشه‌های به دست آمده در شکل زیر نمایش داده شده‌اند:



شکل ۲-۳: مراکز خوشه‌های حاصل از الگوریتمK-means

تحلیل کیفی نتایج K-means

در شکل ۲-۳ مشاهده می‌شود که مراکز خوشه‌ها شبیه به میانگین محو شده ارقام کامل هستند. برخلاف NMF، در اینجا هر تصویر نماینده‌ی یک عدد کامل (مثل میانگین تمام صفرهای موجود در دیتاست) است. این تفاوت نشان‌دهنده ماهیت کل‌نگر روش K-means است.

۵-۳ مقایسه کمی (دقت خوشه‌بندی)

علاوه بر تحلیل بصری، دقت هر دو روش در دسته‌بندی داده‌های بدون برچسب اندازه‌گیری شد. از آنجا که خوشه‌بندی یک روش نظارت‌نشده است، برای محاسبه دقت از الگوریتم تطبیق اریب (Hungarian) استفاده شد تا بهترین تناظر بین خوشه‌ها و برچسب‌های واقعی پیدا شود.

نتایج حاصل از اجرا به شرح زیر است:

الگوریتم	دقت
NMF	0.3418
K-means	0.5323

همان‌طور که مشاهده می‌شود، روش K-means در این آزمایش عملکرد بهتری نسبت به روش دیگر داشته است. البته مزیت اصلی NMF در اینجا صرفاً دقت نیست، بلکه قابلیت تفسیرپذیری بالای ویژگی‌های استخراج شده است.

فصل چهارم: نتیجه‌گیری و پیشنهادات

٤-١ نتیجه‌گیری

در این پژوهش، قابلیت‌های الگوریتم تجزیه نامنفی ماتریس (NMF) در استخراج ویژگی و خوشه‌بندی تصاویر ارقام دست‌نویس (MNIST) مورد بررسی قرار گرفت و عملکرد آن با روش کلاسیک K-means مقایسه شد. نتایج به دست آمده نشان می‌دهد که:

١. **نمایش بخش‌محور:** مهم‌ترین ویژگیNMF ، توانایی آن در تجزیه تصاویر به «اجزای سازنده» است. تصاویر پایه استخراج شده توسط NMF (مانند خطوط و کمان‌ها) نشان دادند که این الگوریتم به درستی ساختار درونی داده‌ها را یاد گرفته است. این ویژگی باعث می‌شود NMF در کاربردهایی که تفسیرپذیری مدل اهمیت دارد (مانند تشخیص چهره یا تحلیل متن)، برتری قابل توجهی داشته باشد.

٢. **تفاوت با K-means:** در مقابل، الگوریتم K-means رویکردی «کل‌نگر» داشت و مراکز خوشه‌ها را به صورت میانگین کلی ارقام نشان داد. اگرچه K-means از نظر محاسباتی سریع است، اما فاقد قابلیت تفکیک اجزای داده‌ها می‌باشد.

٣. **دقت خوشه‌بندی:** از نظر کمی، هر دو الگوریتم توانستند با دقتی قابل قبول (با توجه به ماهیت نظارت‌نشده مسئله) ارقام را دسته‌بندی کنند. این نشان می‌دهد که فضای ویژگی‌های کاهش‌یافته توسطNMF ، اطلاعات کافی برای تمایز کلاس‌های مختلف را حفظ کرده است.

۴-۲ پیشنهادات برای کارهای آتی

برای بهبود نتایج و گسترش این پژوهش، موارد زیر پیشنهاد می‌شود:

- استفاده از نسخه‌های پیشرفته‌تر NMF مانند Sparse NMF که با اعمال جریمه تنکی(Sparsity) ، پایه‌هایی تمیزتر و تفکیک‌شده‌تر تولید می‌کند.
- اعمال الگوریتم بر روی داده‌های پیچیده‌تر مانند تصاویر چهره (مانند دیتاست Olivetti Faces) برای مشاهده دقیق‌تر قابلیت تجزیه اجزای صورت (چشم، بینی، لب).
- استفاده از روش‌های مقداردهی اولیه هوشمند به جای مقداردهی تصادفی برای افزایش سرعت همگرایی الگوریتم.

مراجع

[1] Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755), 788-791.

[2] Aggarwal, C. C., & Reddy, C. K. (Eds.). (2013). *Data Clustering: Algorithms and Applications*. CRC Press.

[3] LeCun, Y., Cortes, C., & Burges, C. J. (2010). MNIST handwritten digit database.

[4] Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, 100-108.

پیوست

پروژه حاضر به زبان پایتون و با استفاده از کتابخانه‌های استاندارد داده‌کاوی پیاده‌سازی شده است. جهت مدیریت وابستگی‌ها و محیط مجازی از ابزار Poetry استفاده شده است.

دسترسی به کد منبع: نسخه کامل و قابل اجرای پروژه شامل فایل‌های پیکربندی (pyproject.toml) و تاریخچه تغییرات، در مخزن گیت‌هاب قابل دسترسی است.

در ادامه، کدهای اصلی فایل main.py برای بررسی منطق پیاده‌سازی آورده شده است:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import fetch_openml
from sklearn.decomposition import NMF
from sklearn.cluster import KMeans
from sklearn.metrics import accuracy_score, confusion_matrix
from scipy.optimize import linear_sum_assignment


def get_clustering_accuracy(y_true, y_pred):
    """
    Since clustering labels have no specific order (permutation invariance),
    this function finds the best match between cluster labels and true labels
    using the Hungarian algorithm, then calculates accuracy.
    """
    cm = confusion_matrix(y_true, y_pred)
    # Find the best assignment
    row_ind, col_ind = linear_sum_assignment(cm.max() - cm)
    return cm[row_ind, col_ind].sum() / np.sum(cm)


def plot_gallery(title, images, n_col=5, n_row=2):
    """Helper function to plot images (basis patterns or centroids)"""
    plt.figure(figsize=(2. * n_col, 2.26 * n_row))
    plt.suptitle(title, size=16)
    for i, comp in enumerate(images):
        plt.subplot(n_row, n_col, i + 1)
        vmax = max(comp.max(), -comp.min())
        plt.imshow(comp.reshape(28, 28), cmap=plt.cm.gray,
                   interpolation='nearest',
                   vmin=-vmax, vmax=vmax)
        plt.xticks(())
        plt.yticks(())
    plt.subplots_adjust(0.01, 0.05, 0.99, 0.93, 0.04, 0.)
```

```
# 1. Load MNIST Dataset
print("Loading MNIST dataset...")
mnist = fetch_openml('mnist_784', version=1, as_frame=False)
X = mnist.data
y = mnist.target.astype(int)

# Normalize data (NMF requires non-negative data)
# Pixel values are 0-255, scaling them to 0-1
X = X / 255.0

print(f"Data Matrix (X) shape: {X.shape}")

# 2. Run NMF Algorithm
# We use 10 components since there are 10 digits (0-9)
n_components = 10
print(f"Running NMF with {n_components} components on the full dataset...")

nmf = NMF(n_components=n_components, init='random',
          random_state=42, max_iter=500)

# W is the membership matrix (referred to as F in the project description)
W = nmf.fit_transform(X)

# H is the basis matrix (referred to as G in the project description)
# The rows of H contain the basis images
H = nmf.components_

print("NMF execution completed.")

# 3. Visualize the columns of Matrix G (Basis Images)
plot_gallery("NMF Basis Images (Columns of G)", H)
plt.show()

# 4. Clustering using Membership Matrix F (here W)
# Each data point belongs to the cluster corresponding to the highest weight in W
y_pred_nmf = np.argmax(W, axis=1)

# Calculate NMF clustering accuracy
acc_nmf = get_clustering_accuracy(y, y_pred_nmf)
print(f"\nNMF Clustering Accuracy: {acc_nmf:.4f}")

# 5. Compare with K-means
print("\nRunning K-means on the original data...")
kmeans = KMeans(n_clusters=n_components, random_state=42, n_init=10)
y_pred_kmeans = kmeans.fit_predict(X)

# Calculate K-means accuracy
acc_kmeans = get_clustering_accuracy(y, y_pred_kmeans)
print(f"K-means Clustering Accuracy: {acc_kmeans:.4f}")

# 6. Visualize K-means Centroids
plot_gallery("K-means Centroids", kmeans.cluster_centers_)
```

```
plt.show()


# Final Analysis
print("-" * 30)
print("Analysis of Results:")
if acc_nmf > acc_kmeans:
    print("In this run, NMF performed better than K-means.")
else:
    print("In this run, K-means performed better (or close) to NMF.")

print("Qualitative Difference:")
print("- NMF basis images typically show 'parts' of digits (strokes, loops).")
print("- K-means centroids show the 'average' of whole digits.")
```