دانشکده مهندسی کامپیوتر

خوشه‌بندی متون فارسی

پایان‌نامه برای دریافت درجه کارشناسی

در رشته مهندسی کامپیوتر گرایش نرم افزار

محسن ایمانی

استاد راهنما:

دکتر مینایی

شهریورماه ۱۳۹۱

تقدیم به:

شهید حمیدرضا شکارچی

شهید علیرضا رسول خمینی

شهید سعید علاف جوادی

شهید امرالله نادران

دانشجویان شهید دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران

چکیده

خوشه‌بندی یکی از تکنیک‌های بسیار قدرتمند برای کشف گروه‌ها و وابستگی‌های طبیعی در یک مجموعه داده و همچنین شناخت الگوهای ساختاری و موضوعی موجود در آن، بدون داشتن هر گونه پیش‌زمینه‌ی شناختی در مورد مشخصات و ویژگی‌های داده، می‌باشد.[۱]

خوشه‌بندی اسناد، به عنوان یکی از روش‌های یادگیری ماشین بدون ناظر[1]، در زمینه‌های مختلف پردازش زبان‌های طبیعی از قبیل بازیابی اطلاعات[2] ، خلاصه سازی چند متنی خودکار و ... کاربرد گسترده ای دارد. به عنوان مثال در موتورهای جستجو، خوشه‌بندی اسنادی که از نتایج موتور جستجو به دست می‌آید تأثیر قابل ملاحظه ای در بهبود دقت بازیابی اطلاعات خواهد داشت. [۲]

در این پژوهش به بررسی روش‌های موجود برای خوشه‌بندی اسناد و همچنین پیاده‌سازی یکی از این روش‌ها برای متون فارسی پرداخته شده است.

اساس خوشه‌بندی در اسناد یافتن و دسته‌بندی سندهایی می‌باشد که با یکدیگر شباهت دارند. در واقع خوشه‌بندی به روشی گفته می‌شود که یک مجموعه‌ی بزرگ از اسناد را گرفته و به صورت خودکار به چند مجموعه‌ی کوچک‌تر از اسناد مشابه تقسیم می‌کند در واقع اسناد موجود در یک خوشه از لحاظ موضوعی و یا مفهومی یکسان می‌باشند.[۳]

در حالت کلّی دو روش خوشه‌بندی وجود دارد: ۱- روش سلسله مراتبی[3] ۲- روش افرازی[4]

در روش سلسله مراتبی هر سند ابتدا به صورت یک خوشه در نظر گرفته می‌شود و سپس فاصله‌ی بین جفت خوشه‌ها محاسبه شده و در گام بعدی هر جفت خوشه با کم‌ترین فاصله ادغام می‌شوند. این کار آن قدر تکرار می‌شود تا آن که تعداد خوشه مورد نظر به دست آید.[۴]

در الگوریتم‌های افرازی اسناد به نحوی به چند بخش تقسیم می‌شوند، مثلاً در الگوریتم‌های خانواده‌ی K-means، ابتدا k مرکز مشخص شده که هر مرکز به عنوان شاخص یک خوشه می‌باشد، سپس هر سندی براساس اندازه فاصله (بین هر سند و k مرکز) به یک خوشه تخصیص می‌یابد. سپس k مرکز مجدداً محاسبه شده و این گام آنقدر تکرار می‌شود تا یک مجموعه از k خوشه بر اساس تابع معیار به صورت بهینه به دست آید.

واژه‌های کلیدی: خوشه‌بندی متون، متن‌کاوی، پردازش زبان طبیعی، زبان فارسی

[1] unsupervised
[2] Information retrieval
[3] Hierarchical
[4] Partitioning

فهرست مطالب

فصل اول:

مقدمه

۱. مقدمه

در عصر حاضر، انسان به این نتیجه رسیده است که اطلاعات، بزرگ‌ترین و مهم‌ترین نقش را در زندگی وی ایفا می‌کند. مقادیر عظیم داده که به روش‌های مختلف جمع‌آوری می‌شوند امروزه نیازمند پردازش‌اند تا بتوان از آن‌ها دانش استخراج کرد. سؤال مهم اینجاست: "چگونه نمونه‌های مهم را پردازش کنیم؟" به وضوح نیازمند روش‌هایی قدرتمند برای پردازش این اطلاعات هستیم، زیرا انجام دادن این کار توسط انسان کاری بس طاقت فرسا و حتی غیرممکن است. مطالعاتی که در زمینه‌ای بسیار کوچک‌تر، مطالعه‌ی متن انجام شده خود زمینه ای برای پیدایش چندین رشته‌ی مختلف در علم و صنعت شده است، رشته‌هایی نظیر زبان شناسی محاسباتی، داده‌کاوی متنی، بازیابی اطلاعات و ...

یافتن معنا از میان انبوه کلمات و حروف، بیشتر شبیه پیدا کردن ذرات ریز طلا در زیر خروارها سنگ معدن در زیر زمین است. حتی این سؤال مطرح است که آیا پیدا کردن مفهوم صحیح، آن هم در صورت وجود توسط ماشین‌ها، کاری عملی است؟ یا اینکه انسان این کار را تنها بر اساس احساس خود و پیش‌زمینه‌ای که در ذهن خویشتن ثبت کرده است این کار را انجام می‌دهد.[۵]

در حالی که ما مفهوم را در کوچک‌ترین اجزاء زبان می‌یابیم و متداول‌ترین روش‌های پردازش متن امروزه کلمات و در نهایت همسایگان آن‌ها در متن را مورد بررسی قرار می‌دهند چاره‌ای نداریم تا از کنار مهم‌ترین حامل‌های مفهوم یعنی ساختار متن، دستور زبان آن و معنی بگذریم.

یکی از ابتدایی‌ترین روش‌های بشر برای تحلیل مسئله‌های مختلف استفاده از الگوها و بررسی آن‌هاست، این سؤال برای ما پیش خواهد آمد که چه چیزی بین این داده‌ها مشترک است؟ جواب این سؤال ما را به موضوع اصلی این نوشتار یعنی خوشه‌بندی اسناد رهنمون خواهد کرد. خوشه‌بندی تنها ابتدای پردازش داده‌هاست زیرا خود به تنهایی پردازش را به عهده نمی‌گیرد و تنها کار را برای مراحل بعدی ساده می‌کند. خوشه‌بندی ما را قادر می‌سازد تا به جای پردازش حجم انبوه اطلاعات، تنها اطلاعاتی را بررسی و تجزیه و تحلیل کنیم که بسیار به هم شبیه هستند و این خود یک گام بلند برای ساده کردن مسئله است.

از خوشه‌بندی اسناد می‌توان به عنوان یکی از تکنیک‌های حوزه متن‌کاوی در هوش مصنوعی نام برد. متن‌کاوی، خود افرازی از داده‌کاوی است که تمرکز آن بر استخراج داده‌های مفید و کشف دانش در داده‌های سندی می‌باشد. خوشه‌بندی اسناد، به عنوان یکی از روش‌های یادگیری ماشین بدون ناظر[۵]، در زمینه‌های مختلف پردازش زبان‌های طبیعی از قبیل بازیابی اطلاعات[۶] ، خلاصه سازی چندمتنی خودکار و ... کاربرد گسترده ای دارد. به عنوان مثال در موتورهای جستجو، خوشه‌بندی اسنادی که از نتایج موتور جستجو به دست می‌آید تأثیر قابل ملاحظه ای در بهبود دقت بازیابی اطلاعات خواهد داشت.

[5] Unsupervised machine learning
[6] Information retrieval

۱٫۱. داده‌کاوی چیست؟

امروزه با گسترش سیستم‌های پایگاهی و حجم بالای داده‌های ذخیره شده در این سیستم‌ها ، نیاز به ابزاری است تا بتوان داده های ذخیره شده پردازش کرد و اطلاعات حاصل از این پردازش را در اختیار کاربران قرار داد .

با استفاده از پرسش‌های ساده در SQL و ابزارهای گوناگون گزارش گیری معمولی ، می‌توان اطلاعاتی را در اختیار کاربران قرار داد تا بتوانند به نتیجه گیری در مورد داده‌ها و روابط منطقی میان آن‌ها بپردازند اما وقتی که حجم داده‌ها بالا باشد ، کاربران هر چند زبر دست و با تجربه باشند نمی‌توانند الگوهای مفید را در میان حجم انبوه داده‌ها تشخیص دهند و یا اگر قادر به این کار هم باشند ، هزینه عملیات از نظر نیروی انسانی و مادی بسیار بالا است .

از سوی دیگر کاربران معمولاً فرضیه ای را مطرح می‌کنند و سپس بر اساس گزارشات مشاهده شده به اثبات یا رد فرضیه می‌پردازند ، در حالی که امروزه نیاز به روش‌هایی است که اصطلاحاً به کشف دانش[۷] بپردازند یعنی با کمترین دخالت کاربر و به صورت خودکار الگوها و رابطه های منطقی را بیان نمایند .

داده کاوی[۸] یکی از مهم‌ترین این روش‌ها است که به وسیله آن الگوهای مفید در داده‌ها با حداقل دخالت کاربران شناخته می‌شوند و اطلاعاتی را در اختیار کاربران و تحلیل گران قرار می‌دهند تا بر اساس آن‌ها تصمیمات مهم و حیاتی در سازمان‌ها اتخاذ شوند .

در داده کاوی از بخشی از علم آمار به نام تحلیل اکتشافی داده‌ها[۹] استفاده می‌شود که در آن بر کشف اطلاعات نهفته و ناشناخته از درون حجم انبوه داده‌ها تاکید می‌شود . علاوه بر این داده کاوی با هوش مصنوعی و یادگیری ماشین نیز ارتباط تنگاتنگی دارد ، بنابراین می‌توان گفت در داده کاوی تئوری‌های پایگاه داده‌ها ، هوش مصنوعی ، یادگیری ماشین و علم آمار را در هم می‌آمیزند تا زمینه کاربردی فراهم شود .

باید توجه داشت که اصطلاح داده کاوی زمانی به کار برده می‌شود که با حجم بزرگی از داده‌ها ، در حد مگا یا ترابایت ، مواجه باشیم . در تمامی منابع داده کاوی بر این مطلب تاکید شده است .

هر چه حجم داده‌ها بیشتر و روابط میان آن‌ها پیچیده تر باشد دسترسی به اطلاعات نهفته در میان داده‌ها مشکل‌تر می‌شود و نقش داده کاوی به عنوان یکی از روش‌های کشف دانش ، روشن‌تر می‌گردد .

[7] Knowledge Discovery
[8] Data Mining
[9] Exploratory Data Analysis

۱،۱،۱. مفاهیم پایه در داده‌کاوی

در داده‌کاوی معمولاً به کشف الگوهای مفید از میان داده‌ها اشاره می‌شود . منظور از الگوی مفید ، مدلی در داده‌ها است که ارتباط میان یک زیر مجموعه از داده‌ها را توصیف می‌کند و معتبر ، ساده ، قابل فهم و جدید است .

۱،۱،۲. تعریف داده کاوی

در متون آکادمیک تعاریف گوناگونی برای داده کاوی ارائه شده‌اند . در برخی از این تعاریف داده کاوی در حد ابزاری که کاربران را قادر به ارتباط مستقیم با حجم عظیم داده‌ها می‌سازد معرفی گردیده است و در برخی دیگر ، تعاریف دقیق‌تر که درآنها به کاوش در داده‌ها توجه می‌شود موجود است . برخی از این تعاریف عبارتند از :

- داده کاوی عبارت است از فرایند استخراج اطلاعات معتبر ، از پیش ناشناخته ، قابل فهم و قابل اعتماد از پایگاه داده های بزرگ و استفاده از آن در تصمیم گیری در فعالیت‌های تجاری مهم. [۶]
- اصطلاح داده کاوی به فرایند نیم خودکار تجزیه و تحلیل پایگاه داده های بزرگ به منظور یافتن الگوهای مفید اطلاق می‌شود[۷]
- داده کاوی یعنی جستجو در یک پایگاه داده‌ها برای یافتن الگوهایی میان داده‌ها .[۷]
- داده کاوی یعنی استخراج دانش کلان ، قابل استناد و جدید از پایگاه داده‌های بزرگ .
- داده کاوی یعنی تجزیه و تحلیل مجموعه داده های قابل مشاهده برای یافتن روابط مطمئن بین داده‌ها .

همان‌گونه که در تعاریف گوناگون داده کاوی مشاهده می‌شود ، تقریباً در تمامی تعاریف به مفاهیمی چون استخراج دانش ، تحلیل و یافتن الگوی بین داده‌ها اشاره شده است .

۱،۲. متن‌کاوی چیست؟

بخش قابل توجهی از اطلاعات قابل دسترس به صورت متنی که شامل مجموعه بزرگی از اسناد منابع مختلف (مثلاً مقالات خبری، مقالات، کتاب‌ها، ایمیل‌ها، صفحات وب و ...) ذخیره شده‌اند. پایگاه داده‌های متنی به علت افزایش مقدار اطلاعات موجود به فرم الکترونیکی سریع رشد می‌کنند. امروزه بیشتر اطلاعات در صنعت، کسب و کار[۱۰] و سازمان‌های دیگر به صورت الکترونیکی و به فرم پایگاه داده متنی ذخیره شده‌اند.

داده‌های ذخیره شده به صورت الکترونیکی، اغلب داده‌های نیمه ساختاریافته هستند چون نه به طور کامل غیرساختیافته هستند و نه به طور کامل ساخت‌یافته هستند. به رای مثال یک سند شامل تعدادی فیلد

[10] business

ساخت‌یافته مانند عنوان، نویسندگان، تاریخ انتشار، رده[11] و ..... و از طرف دیگر شامل برخی عناصر متنی غیرساختاریافته مانند چکیده و محتویات است. روش‌های بازیابی اطلاعات مانند (متدهای ایندکس کردن متن) برای مدیریت سندهای غیر ساختاریافته ایجاد شده‌اند. روش‌های بازیابی اطلاعات قدیمی برای مقدار زیادی داده متنی که به طور فزاینده افزایش می‌یابند، ناکارآمد هستند. بدون دانستن محتویات سندها، فرمول‌بندی کردن سؤال‌های[12] مناسب برای استخراج اطلاعات مفید از داده، مشکل است. کاربرها نیاز به ابزارهایی برای مقایسه سندهای مختلف، مرتب کردن سندها بر اساس مربوط بودن آن‌ها و یافتن الگوها دارند. بنابراین یکی از جدیدترین زمینه‌های مورد تحقیق در داده‌کاوی، متن‌کاوی برای این منظور گسترش یافت. متن‌کاوی یعنی جستجوی الگوها در متن غیرساختیافته. متن‌کاوی برای کشف خودکار دانش مورد علاقه یا مفید از متن نیمه ساخت‌یافته استفاده می‌شود. چندین تکنیک برای متن‌کاوی پیشنهاد شده است عبارتند از ساختار مفهومی[13]،کاوش قوانین انجمنی[14]، درخت تصمیم‌گیری، روش‌های استنتاج قوانین[15]، همچنین روش‌های بازیابی اطلاعات برای کارهایی مانند تطبیق دادن سندها، مرتب کردن، خوشه‌بندی و... .

از جمله مشکلاتی که در زمینه متن‌کاوی وجود دارد کشف کردن دانش مفید از متن نیمه ساخت‌یافته یا غیرساختیافته است که توجه زیادی را به خود جلب کرده است. روش‌های داده‌کاوی سنتی فرض می‌کنند که اطلاعات به فرم پایگاه داده‌های رابطه‌ای هستند به همین دلیل برای بسیاری از کاربردها مانند اطلاعات الکترونیکی قابل دسترس به فرم نیمه ساخت‌یافته یا غیرساختیافته مفید نیستند. بدون عمل متن‌کاوی پردازش کردن پایگاه داده‌های متنی غیرساختیافته باید به صورت دستی توسط کاربران انجام شود که این امر بسیار طاقت‌فرساست. بنابراین می‌توان گفت هدف متن‌کاوی خودکارسازی مقدار زیادی از کار کاربران است. گاهی اوقات به جای واژه متن‌کاوی از واژه‌های "کاوش داده های متنی"[16] و نیز نام معروف "کشف دانش در متن"[17] یا KDT، استفاده می‌شود. متن‌کاوی تکیه‌اش روی پیدا کردن دانش[18] جدید از متن است (معمولاً دانشی که به طور ضمنی در سندهاست) در حالی که بازیابی اطلاعات سندهایی که بیشترین ارتباط را دارند می‌یابد متن‌کاوی را می‌توان به عنوان یک روش میان رشته ای برای بازیابی اطلاعات، یادگیری ماشین، آماری[19]، زبان‌دانان محاسباتی [20]و به ویژه داده‌کاوی در نظر گرفت. از آنجا که متن-کاوی، در تکنولوژی‌های متفاوتی ریشه دارد، از این رو تعاریف زیادی نیز برای آن وجود دارد. افرادی که دارای پیشینه کار در زمینه‌ی داده کاوی[21] بودند می‌خواستند که همان مفاهیم و روش‌های موجود در داده کاوی را بر متون اعمال کنند و تعاریفشان نیز منطبق بر همین زمینه بود. اما کسانی که از جامعه‌ی زبان‌دانان

[11] category
[12] Query
[13] conceptual
[14] association rule
[15] Rule induction
[16] Text data Mining
[17] Knowledge Discovery in Text
[18] knowledge
[19] statistic
[20] Computational linguistics
[21] Data mining

محاسباتی[۲۲] آمده بودند، قصد داشتند که این توانایی را به کامپیوتر بدهند که بتوانند متن را بفهمند و این غایت چیزی است که از متن کاوی مورد انتظار است.

در ادامه مطالب در این بخش ابتدا به بیان مفهوم کشف دانش و ارتباط آن با متن‌کاوی می‌پردازیم. سپس تعاریف مختلفی که در ناحیه‌های مختلف سرچ برای متن‌کاوی وجود دارد بیان می‌شود.

۱,۲,۱. کشف دانش[۲۳] و ارتباط آن با متن‌کاوی

کشف دانش در پایگاه داده‌ها[۲۴] (KDD)در اوایل دهه ۸۰ در مراجعه به مفهوم کلی،سطح بالا و به دنبال جستجوی دانش در اطلاعات شکل گرفته است. این لغت به بیان دیگر به همه شیوه‌هایی اشاره دارد که هدف آن‌ها پی بردن به ارتباط و نظم بین اطلاعات قابل مشاهده است. لغت KDD برای توصیف همه مراحل استخراج اطلاعات از پایگاه داده و نیز بیان اهداف کارهای اولیه کاربرد قوانین تصمیم گیری است. این واژه به طور رسمی اولین بار توسط Usama Fayaad در اولین کنفرانس بین‌المللی داده کاوی و کشف دانش که در سال ۱۹۹۵ در مونترال برگزار شده بود، معرفی شد که به بیان ارتباط تکنیک‌های آنالیز در چندین مرحله با هدف استخراج دانش‌های ناشناخته قبلی از داده های در دسترس می‌پرداخت. داده‌هایی که ارتباط منظم و پراهمیت آن‌ها قبلاً به نظر نمی‌رسید. کم کم واژه داده کاوی جای خود را پیدا کرد و مترادفی برای همه مراحل استخراج دانش شد. گاهی داده‌کاوی را به عنوان مترادفی برای KDD به کار می‌برند. بدین معنا که داده‌کاوی شامل همه ابعاد فرآیند کشف دانش است. گاهی داده‌کاوی به عنوان افرازی از فرآیندهای KDD در نظر گرفته می‌شود و فاز مدل کردن را توصیف می‌کنند. در کل KDD فرآیند یافتن اطلاعات و الگوهای مفید از داده را گویند و داده کاوی بهره‌گیری از الگوریتم‌هایی برای یافتن اطلاعات مفید در فرآیند KDD است. تعاریف مختلفی برای کشف دانش یا کشف دانش در پایگاه داده وجود دارد. مثلاً در بیان شده است که KDD فرآیند شناسایی کردن الگوهای قابل فهم، مفید، جدید و معتبر در داده است. در واقع هدف پیدا کردن الگوها و روابط پنهان در این داده‌هاست. از جمله خصوصیاتی که برای اندازه-گیری کیفیت الگوهای پیدا شده در داده می‌توان استفاده کرد عبارتند از: قابلیت فهم انسان، اعتبارسنجی با معیارهای آماری، تازگی و مفید بودن. کشف دانش در پایگاه داده را می‌توان به عنوان یک فرآیند که به وسیله چندین گام پردازش کردن[۲۵] تعریف می‌شود، در نظر گرفت. این گام‌ها به منظور استخراج الگوهای مفید باید بر روی مجموعه داده‌ای اعمال شوند. این گام‌ها به صورت تکراری اجرا می‌شوند و برخی گام‌ها نیاز به بازخورد[۲۶] از کاربر دارند. یک کاربر سیستم KDD به منظور انتخاب زیر مجموعه صحیحی از داده‌ها باید درک بالایی از قلمرو داده‌ها، رده مناسبی از الگوها و معیار خوبی برای الگوهای جالب داشته باشد. بنابراین سیستم KDD باید ابزارهایی با اثر تعاملی داشته باشد نه سیستم‌های تجزیه و تحلیل خودکار. طبق گام‌ها را می‌توان به صورت زیر بیان کرد: ۱) درک کردن کسب و کار[۲۷] ۲) درک کردن داده ۳)آماده سازی داده ۴)

[22] Computational linguistics community
[23] Knowledge discovery
[24] Knowledge discovery in database
[25] processing
[26] feedback
[27] Understanding business

مدل کردن ۵) ارزیابی ۶) deployment. مرحله پیش پردازش غالباً یکی از مراحل زمان بر و در عین حال بسیار مهم در کسب نتیجه مطلوب است. مخصوصاً در متن‌کاوی که نیاز به متدهای پیش‌پردازش کردن خاصی برای تبدیل داده متنی به فرمتی که برای الگوریتم‌های داده‌کاوی مناسب است، داریم.

۱,۲,۲.تعاریف متن‌کاوی

برخی نویسندگان[۸] داده‌کاوی را به عنوان ابزاری برای جستجو کردن اطلاعات ارزشمند در مقدار زیادی داده در نظر می‌گیرند. داده‌کاوی در ناحیه‌های گوناگونی از تحقیق مطرح می‌شود. مثلاً پایگاه داده، یادگیری ماشین و آماری. پایگاه داده‌ها برای تحلیل کردن حجم زیادی از داده‌ها ضروری هستند. یادگیری ماشین، یک ناحیه هوش مصنوعی است که با ایجاد تکنیک‌هایی امکان یادگیری به وسیله تحلیل مجموعه‌های داده-ای[۲۸] را به کامپیوترها می‌دهند. تمرکز این روش‌ها روی داده سمبولیک است. آماری پایه‌اش در ریاضیات است و با آنالیز داده‌های تجربی سر و کار دارد. پایه آن تئوری آماری است. در این تئوری عدم قطعیت[۲۹] و شانس[۳۰] به وسیله تئوری احتمال مدل می‌شوند. امروزه بسیاری از روش‌های آماری در فیلد KDD استفاده می‌شوند.

متن‌کاوی یا کشف دانش از متن برای اولین بار در [۹]بیان شد. می‌توان گفت که متن‌کاوی از تکنیک‌های بازیابی اطلاعات، استخراج اطلاعات همچنین پردازش کردن زبان طبیعی[۳۱] استفاده می‌کند و آن‌ها را به الگوریتم‌ها و متدهای KDD، داده‌کاوی، یادگیری ماشین و آماری مرتبط می‌کنند. با توجه به ناحیه‌های تحقیق گوناگون، بر هر یک از آن‌ها می‌توان تعاریف مختلفی از متن‌کاوی در نظر گرفت در ادامه برخی از این تعاریف بیان می‌شوند:

متن‌کاوی = استخراج اطلاعات: در این تعریف متن‌کاوی متناظر با استخراج اطلاعات در نظر گرفته می‌شود (استخراج واقعیت‌ها[۳۲] از متن).

متن‌کاوی = کشف داده متنی: متن‌کاوی را می‌توان به عنوان متدها و الگوریتم‌هایی از فیلدهای یادگیری ماشین و آماری برای متن‌ها با هدف پیدا کردن الگوهای مفید در نظر گرفت. برای این هدف پیش پردازش کردن متون ضروری است. در بسیاری از روش‌ها، متدهای استخراج اطلاعات، پردازش کردن زبان طبیعی یا برخی پیش پردازش‌های ساده برای استخراج داده از متون استفاده می‌شود. سپس می‌توان الگوریتم‌های داده‌کاوی را بر روی داده‌های استخراج شده اعمال کرد.

متن‌کاوی = فرآیند KDD: که در بخش قبلی به طور کامل توضیح داده شده است و در اینجا دیگر بیان نمی‌شود.

---

[28] Data sets
[29] uncertainty
[30] Randomness
Natural language processing[31] (NLP)
[32] facts

در این مقاله ما بیشتر متن‌کاوی را به عنوان کشف داده متنی در نظر می‌گیریم و بیشتر تمرکز می‌کنیم روی روش‌های استخراج الگوهای مفید از متن برای خوشه‌بندی مجموعه‌های متنی یا استخراج اطلاعات مفید.

۱،۲،۳. حوزه‌های مرتبط با متن‌کاوی

در دنیای کنونی این کمبود اطلاعات نیست که مسئله است بلکه کمبود دانشی است که از این اطلاعات می‌توان حاصل کرد. میلیون‌ها صفحه‌ی وب، میلیون‌ها کلمه در کتابخانه های دیجیتال و هزاران صفحه اطلاعات در هر شرکت تنها چند دست از این منابع اطلاعاتی هستند. اما نمی‌توان به طور مشخص منبعی از دانش را در این بین معرفی کرد. دانش خلاصه‌ی اطلاعات است و نیز نتیجه گیری و حاصل فکر و تحلیل بر روی اطلاعات.

داده کاوی، یک روش بسیار کاراست برای کشف اطلاعات از داده های ساخت‌یافته ای که در جداول نگهداری می‌شوند. داده کاوی، الگوها را از تراکنش‌ها[۳۳]، استخراج می‌کند، داده را گروه بندی می‌کند و نیز آن‌را دسته بندی می‌کند. به وسیله‌ی داده کاوی می‌توانیم پی به وجود روابطی میان اقلام داده ای که دیتابیس را پر کرده‌اند ببریم. در عین حال ما با داده کاوی مشکلی داریم و آن عدم وجود عامیت در کاربرد آن است. بیشتر دانش ما اگر به صورت غیر دیجیتال نباشند، کاملاً غیر ساخت‌یافته اند. کتابخانه های دیجیتال، اخبار، کتاب‌های الکترونیکی، بسیاری از مدارک مالی، مقالات علمی و تقریباً هر چیزی که شما می‌توانید در داخل وب بیابید، ساختیاقته نیستند. در نتیجه ما نمی‌توانیم آموزه های داده کاوی را در مورد آن‌ها به طور مستقیم استفاده کنیم. با این حال، سه روش اساسی در مواجهه با این حجم وسیع از اطلاعات غیر ساخت‌یافته وجود دارد که عبارتند از: بازیابی اطلاعات[۳۴]، استخراج اطلاعات[۳۵] و پردازش زبان طبیعی

بازیابی اطلاعات: اصولاً مرتبط است با بازیابی مستندات و مدارک. کار معمول در بازیابی اطلاعات اینست که با توجه به نیاز مطرح شده از سوی کاربر، مرتبط‌ترین متون و مستندات و یا در واقع بقچه‌ی کلمه را از میان دیگر مستندات یک مجموعه بیرون بکشد. این یافتن دانش نیست بلکه تنها آن بقچه ای از کلمات را که به نظرش مرتبط‌تر به نیاز اطلاعاتی جستجو گر است را به او تحویل می‌دهد. این روش به واقع هیچ دانشی و حتی هیچ اطلاعاتی را برایمان به ارمغان نمی‌آورد.

پردازش زبان طبیعی: هدف کلی NLP رسیدن به یک درک بهتر از زبان طبیعی توسط کامپیوترهاست. تکنیک‌های مستحکم و ساده‌ای برای پردازش کردن سریع متن به کار می‌روند. همچنین از تکنیک‌های آنالیز زبان شناسی نیز برای پردازش کردن متن استفاده می‌شود.

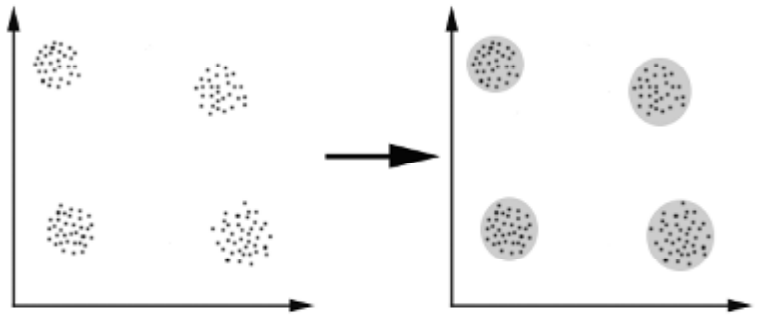
استخراج اطلاعات: هدف روش‌های استخراج اطلاعات، استخراج اطلاعات خاص از سندهای متنی است. استخراج اطلاعات می‌تواند به عنوان یک فاز پیش پردازش در متن کاوی بکار برود. استخراج اطلاعات

[33] Transactions
[34] Information Retrieval
[35] Information Extraction

عبارتند از نگاشت کردن متن‌های زبان طبیعی (مثلاً گزارش‌ها، مقالات journal، روزنامه‌ها، ایمیل‌ها، صفحات وب، هر پایگاه داده متنی و.....) به یک نمایش ساخت‌یافته و از پیش تعریف شده یا قالب‌هایی که وقتی پر می‌شوند، منتخبی از اطلاعات کلیدی از متن اصلی را نشان می‌دهند. یک‌بار اطلاعات استخراج شده و سپس اطلاعات می‌توانند در پایگاه داده برای استفاده‌های آینده، ذخیره شوند.

۱,۳. مقدمه‌ای بر خوشه‌بندی

خوشه‌بندی را می‌توان به عنوان مهم‌ترین مسئله در *یادگیری بدون نظارت* در نظر گرفت. خوشه‌بندی با یافتن یک ساختار درون یک مجموعه از داده‌های بدون برچسب درگیر است. خوشه به مجموعه‌ای از داده‌ها گفته می‌شود که به هم *شباهت* داشته باشند. در خوشه‌بندی سعی می‌شود تا داده‌ها به خوشه‌هایی تقسیم شوند که شباهت بین داده‌های درون هر خوشه حداکثر و شباهت بین داده‌های درون خوشه‌های متفاوت حداقل شود.



شکل ۱ در این شکل نمونه‌ای از اعمال خوشه‌بندی روی یک مجموعه از داده‌ها مشخص شده است که از معیار فاصله(Distance) به عنوان عدم شباهت(Dissimilarity) بین داده‌ها استفاده شده است.

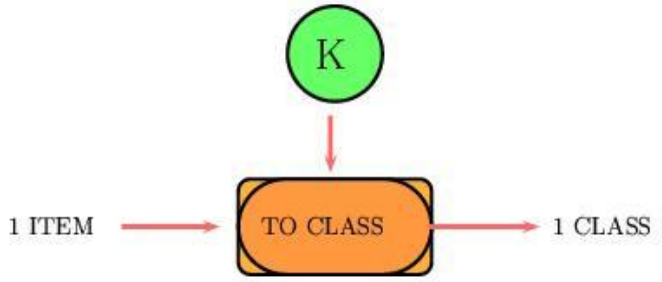
۱,۳,۱. خوشه‌بندی در مقابل رده‌بندی

در رده‌بندی[۳۶] دسته‌ها(کلاس) از پیش مشخص شده‌اند و هر داده به یک رده(کلاس) از پیشین مشخص شده تخصیص می‌یابد. ولی در خوشه‌بندی هیچ اطلاعی از کلاس‌های موجود درون داده‌ها وجود ندارد و به عبارتی خود خوشه‌ها نیز از داده‌ها استخراج می‌شوند. در شکل زیر تفاوت بین خوشه‌بندی و رده‌بندی بهتر نشان داده شده است. [۱۰]

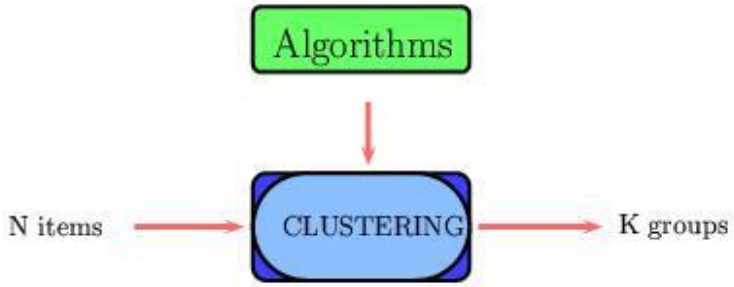
[36] Classification

۱,۳,۲. یادگیری با نظارت در مقابل یادگیری بدون نظارت

در یادگیری با نظارت[۳۷] از ابتدا دسته‌ها مشخص هستند و هر یک از داده‌های آموزشی به دسته‌ای خاص نسبت داده شده است و اصطلاحاً گفته می‌شود ناظری وجود دارد که در هنگام آموزش اطلاعات علاوه بر داده‌های آموزش در اختیار یادگیرنده قرار می‌دهد. ولی در یادگیری بدون نظارت[۳۸] هیچ اطلاعاتی بجز داده‌های آموزشی در اختیار یادگیرنده قرار ندارد و این یادگیرنده است که بایستی در داده‌ها به دنبال ساختاری خاص بگردد.[۱۰]



(الف)



(ب)

شکل ۲ الف) در رده‌بندی با استفاده یک سری اطلاعات اولیه داده‌ها به دسته‌های معلومی نسبت داده می‌شوند. در خوشه‌بندی داده‌ها با توجه به الگوریتم انتخاب شده به خوشه‌هایی نسبت داده می‌شوند.
ب) در خوشه‌بندی یادگیری به صورت بدون ناظر انجام می‌شود یعنی نیازی به داده‌های آموزشی ابتدایی نداریم.

[37] Supervised learning
[38] Unsupervised learning

۱،۳،۳. کاربردهای خوشه‌بندی

خوشه‌بندی رایج‌ترین روش یادگیری بدون ناظر است و ابزار خوبی در طیف وسیعی از کاربردها در حوزه‌های متفاوت تجارت و علم است. در این جا به ذکر چند مورد از این کاربردها به طور مختصر می‌پردازیم:

- **یافتن سندهای مشابه**

این ویژگی اغلب زمانی مورد استفاده قرار می‌گیرد که کاربر در نتایج جستجوی خود یک سند مفید را پیدا می‌کند و حالا می‌خواهد سندهای شبیه به آن را ببیند. جالب اینجاست که خوشه‌بندی بر خلاف روش‌های بر مبنای جستجو، که تنها قادر به کشف اشتراک کلمات در سندها هستند، قادر به کشف سندهایی هم است که به صوت مفهومی با هم شبیه باشند.

- **سازماندهی مجموعه‌های عظیم سند**

تمرکز اصلی در بازیابی اسناد بر روی سندهایی است که مربوط به سؤال(Query) خاصی باشند، اما زمانی که تعداد زیادی سند دسته بندی نشده داشته باشیم، بازیابی سند چندان مفید نخواهد بود. چالش اصلی در این مسئله سازماندهی این اسناد در یک طبقه بندی، مشابه با آنچه یک انسان به صورت دستی می‌تواند ایجاد کند، است. در این حالت به راحتی می‌تواند این مجموعه را مرور کرده و سند مورد نظر خود را بیابد.

- **تشخیص محتوای تکراری و تقلب**

در بسیاری از موارد نیاز داریم که در بین تعداد زیادی از اسناد، سندهای تکراری و یا نزدیک به کپی را تشخیص دهیم. خوشه‌بندی در مورد و در واقع در زمینه تشخیص تقلب و سرقت ادبی کاربرد موثری دارد، همچنین این کاربرد از خوشه‌بندی در گروه بندی اخبار جدید و یا رتبه بندی نتایج جستجوها (به این صورت که نتایج متنوع در بین لیست بهترین جواب‌ها باشند) نیز مفید می‌باشد. البته این نکته نیز باید مورد توجه قرار گیرد که در این کاربرد توصیف خوشه‌ها کمتر مورد نیاز بوده و بیشتر بر مبنای شباهت بین اسناد می‌باشد.

- **سیستم‌های پیشنهاد دهنده (Recommendation Systems)**

در این کاربرد سیستم بر مبنای سندی که کاربر مشاهده نموده است، سندهایی را به کاربر پیشنهاد می‌دهد. خوشه‌بندی اسناد این امر را ممکن خواهد نمود که به صورت بی‌درنگ (real time) این کار صورت گرفته و همچنین کیفیت آن هم به صورت قابل توجهی افزایش یابد.

- **بهینه سازی جستجو(Search Optimization)**

خوشه‌بندی کمک بسیار زیادی به بهبود کیفیت و کارایی موتورهای جستجو می‌کند، به این صورت که سؤال کاربر به جای این که مستقیماً با اسناد مقایسه شود، در ابتدا با خوشه مقایسه شود. همچنین با استفاده از خوشه‌بندی به سادگی می‌توان نتایج جستجو را مرتب نمود.

۱،۳،۴. چالش‌های پیش رو در خوشه‌بندی موجود

خوشه‌بندی اسناد دهه‌های بسیاری است که در حال مطالعه و تدریس می‌باشد اما هنوز تا رسیدن به جایی که تبدیل به یک مسئله حل شده شود، فاصله زیادی وجود دارد. چالش‌های اصلی خوشه‌بندی اسناد به طور اختصار عبارتند از:

۱. انتخاب خصوصیت‌های مناسبی از اسناد که باید در خوشه‌بندی استفاده می‌شود.
۲. انتخاب یک معیار محاسبه شباهت مناسب بین سندها.
۳. انتخاب یک روش مناسب برای خوشه‌بندی که از معیار شباهت بالا استفاده کند.
۴. پیاده‌سازی الگوریتم خوشه‌بندی به صورت کارا که اجرای آن متناسب با حافظه و پردازنده‌های موجود عملی باشد.
۵. یافتن روش‌هایی برای ارزیابی کیفیت روش‌های مختلف خوشه‌بندی.

علاوه بر این، با مجموعه‌های متوسط و بزرگ اسناد (۱۰۰۰۰ سند و بیشتر)،پیچیدگی محاسباتی الگوریتم‌های موجود بالا خواهد بود و امکان پذیر بودن استفاده از آن در مسائل دنیای واقعی به عنوان یک چالش مطرح می‌شود. وقتی که اطلاعات مربوط به ترم‌ها در یک ماتریس انبوه نگهداری شود، این ماتریس ممکن است به راحتی آنقدر بزرگ شود که قادر به نگهداری آن در حافظه نباشیم، برای مثال اگر ۱۰۰۰۰۰ سند داشته باشیم که هر کدام ۱۰۰۰۰۰ ترم داشته باشند، آنگاه برای نگهداری این تعداد مقادیر ممیز شناور، حدود ۴۰ گیگابایت حافظه نیاز است. همچنین اگر از مدل فضای بردار استفاده شود، به این ترتیب ابعاد فضای بردار چنین مسئله ای بسیار بالا (بیش از ۱۰۰۰۰) خواهد بود. به این معنا که یک عملیات ساده، مانند پیدا کردن فاصله اقلیدسی بین دو سند در فضای برداری، تبدیل به یک کار بسیار زمان بر می‌شود.[۱۱]

فصل دوم:

مدل‌های نمایش

۲. مدل‌های نمایش

مدل‌ها طراحی می‌شوند تا طبیعت یک شیء را به نمایش بگذارند.[۱۲] یک سند دارای عنوان، جملات و کلمات است و همه‌ی این‌ها در کنار هم یک سند را نمایش می‌دهند. یک فرد باسواد به خوبی درک می‌کند که یک سند چه چیز را نمایش می‌دهد، مفهوم آن، چیزی که به صورت طبیعی انتظار داریم یک سند نمایش دهد. هر چند که ما امروزه رایانه‌ها را قادر کرده‌ایم تا یک سند تجزیه کنند و آن را دوباره برای کاربر بخوانند، هنوز راه زیادی در پیش است تا بتوان ادعا کرد رایانه اسناد را درست مانند انسان می‌خواند و درک می‌کند. در واقع این موضوع چالش جدّی است زیرا هنوز درک کاملی از نحوه‌ی مطالعه‌ی انسان‌ها و اینکه آن‌ها چگونه اسناد را درک می‌کنند در دست نیست، اما در اینجا ما به این بحث نخواهیم پرداخت.

حال که ما مدلی که انسان‌ها به وسیله‌ی آن متن را درک می‌کنند نداریم، ناچاریم تا مدلی برای رایانه طراحی کنیم تا متن را برای آن نمایش دهیم. اما سؤال اینجاست که این کار چگونه انجام می‌شود؟ آیا می‌توان اسناد را درست شبیه یک کیسه‌ی پر از کلمات در نظر گرفت و آن‌ها را به وسیله‌ی مقایسه‌ی محتوای کیسه‌ها با یکدیگر مقایسه کرد؟ این فرض کار را بسیار ساده خواهد کرد، اما سؤال اینجاست که آیا این نحوه‌ی نمایش کافی است؟ متن حاوی ساختار، جملات و عباراتی است که مطمئناً بر روی مفهوم آن تأثیرگذارند، آیا می‌توان این خصوصیات متن‌ها را کنار گذاشت و انتظار داشت تا رایانه بتواند مفهوم را درک کند؟ در این بخش به معرفی انواع روش‌های مدل کردن اسناد می‌پردازیم تا به جواب این سؤال‌ها برسیم.

۲,۱. فضای برداری

پس از اینکه این مدل برای فهرست‌گذاری خودکار در دهه‌ی هفتاد توسط سالتون[۳۹] مطرح شد به مدل استاندارد برای خوشه‌بندی اسناد مبدل گشت.

در این مدل یک سند d به عنوان مجموعه‌ای از جمله‌ها $[t_1 \dots t_n]$ در نظر گرفته می‌شود. به هر کدام از این جمله‌ها وزنی متناسب با یک معیار انتخابی (مثل اهمیت، تکرار و غیره) داده می‌شود. سپس به وسیله‌ی یک بردار n بُعدی w می‌توان یک الگوی وزن دار $W(d)$ برای نمایش d در نظر گرفت.

مهم‌ترین انگیزه برای چنین مدل نمایشی این است که فرموله کردن و کار با آن بسیار ساده است.

در اینجا با یک مثال نشان می‌دهیم که چگونه سه سند $[d_1, d_2, d_3]$ می‌توانند به وسیله‌ی الگوی وزن‌دهی بسامد جمله‌ها نمایش داده شوند.

[39] Salton

مثال ۲,۱ :

d_1: روباه خرگوش را تعقیب کرد

d_2: خرگوش هویج را خورد

d_3: روباه خرگوش را گرفت

این سه سند می‌توانند به وسیله‌ی یک ماتریس رویداد به نمایش درآیند.

	d_3	d_2	d_1
را	۱	۱	۱
روباه	۱	۰	۱
خرگوش	۱	۱	۱
تعقیب	۰	۰	۱
کرد	۰	۰	۰
هویج	۰	۱	۰
خورد	۰	۱	۰
کرد	۰	۰	۱

جدول ۱ ماتریس رویدادهای مشترک جمله-سند

این مدل علی‌رغم سادگی، معایبی نیز دارد، برای مثال در جمله‌های بالا هم "را" و هم "خرگوش" در همه-ی اسناد دارای یک بسامد هستند و به همین خاطر کمکی به شناسایی تفاوت‌های اسناد نمی‌کنند. در نظر بگیرید که چند سکه در یک کیسه هستند و همگی از یک جنس‌اند اما بر روی آن‌ها قیمت آن‌ها حک شده است. مسلماً در اینجا برای شناسایی تفاوت‌ها جنس کمکی به ما نمی‌کند و این قیمت حک شده بر روی سکه‌هاست که مشخص می‌کند کدام سکه‌ها شبیه به یکدیگرند.

برای رفع این مشکل باید یک الگوی وزن‌دهی دقیق‌تر انتخاب کرد. راهی که ما برای بازیابی اطلاعات استفاده می‌کنیم بسامد جمله در مقابل بسامد سند یا tf-idf (term frequency inverse document frequency) است. این مقدار برای یک جمله در موقعیت i که در سند j قرار دارد به صورت زیر محاسبه می‌شود:

$$(f\text{-}idf)_{ij} = ft_{ij} \times idf_i$$ رابطه ۱

که در این رابطه tf_{ij} بسامد جمله i در سند j است و idf_i از رابطه‌ی زیر محاسبه می‌شود:

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$ رابطه ۲

$|D|$ تعداد کل اسناد پیکره است و مخرج کسر تعداد اسنادی است که در آن‌ها جمله‌ی t_i وجود دارد.

این روش باعث می‌شود تا عباراتی که در یک مجموعه‌ی کوچک از اسناد وجود دارند وزن بیشتری پیدا کنند و آن‌هایی که در اغلب اسناد موجودند وزن کمتری داشته باشند. این روش یکی از چندین روشی است که در فصل‌های بعد به توضیح آن‌ها خواهیم پرداخت. [۱۳, ۱۴]

در اینجا یک نکته‌ی قابل توجه وجود دارد و آن اینکه اندازه‌ی ماتریس ربطی به تکرار عبارات ندارد و به مرور با اضافه شدن یک کلمه‌ی جدید اندازه‌ی ماتریس به اندازه‌ی کل پیکره افزایش خواهد یافت. بنابراین ما دچار مشکل کمبود فضا برای نگهداری این ماتریس خواهیم شد. اما با توجه به جدول ۱ درمی‌یابیم که تعداد صفرها در ماتریس مذکور بیشتر از عناصر غیر صفر است و این خود ناشی از آن است که در یک زبان کلماتی که از نظر بسامد استفاده در جایگاه اول قرار دارند به طور معمول دو برابر بیشتر از عناصر رده‌ی دوم استفاده می‌شوند و این روند به همین شکل ادامه می‌یابد[۱۵]. این مسئله در سال ۱۹۴۹ توسط زیپف[۴۰] مطرح شد و به قانون زیپف نیز معروف است. بنابراین ما تنها عناصر غیر صفر این ماتریس را نگهداری خواهیم کرد و از عناصر صفر صرف‌نظر خواهیم کرد.

مشکل دیگری که با آن مواجه خواهیم شد این است که هر چقدر ابعاد فضا بیشتر باشند نقاط پراکنده‌تر به نظر خواهندرسید و پیدا کردن نقاط اشتراک بین آن‌ها سخت‌تر خواهد بود. ما برای حل این مشکل سعی خواهیم کرد تا فضای صفات را محدودتر کنیم.

۲,۱,۱. توسعه‌ها

مدل بردار استاندارد بر این فرض استوار است که تمام محورها دو به دو بر یکدیگر عمودند، یعنی عبارات مستقل خطی‌اند. این فرض مدل کردن را ساده خواهد کرد اما زبان‌های طبیعی را بد منعکس می‌کند، جایی که عبارات به صورت کلی مستقل خطی نیستند.

برای مثال در صورتی که ما بردارهای سه کلمه‌ی حیوان، خرگوش و چکش را در فضا رسم کنیم، خرگوش همان‌قدر به چکش ارتباط دارد که به حیوان، در صورتی که می‌دانیم خرگوش یک حیوان است و این یعنی رابطه‌ای فراتر از رابطه‌ی خرگوش و چکش!

این مسئله باعث خواهد شد تا اسنادی که در واقع درباره‌ی یک موضوع هستند اما از واژگانی متفاوت استفاده می‌کنند، متفاوت قلمداد شوند. ایده‌ای که برای حل این مشکل مطرح شد استفاده از وزن

[40] Zipf

همبستگی *عبارات* به یکدیگر بود. چیزی که باعث به وجود آمدن *مدل فضای برداری عمومی* شد.[۱۶] در این مدل، ما هنگامی که اسناد را بررسی می‌کنیم وزن همبستگی بین آن‌ها را نیز مطرح می‌کنیم تا شباهت‌ها در نظر گرفته‌شوند و مشکل مستقل درنظرگرفتن عبارات برطرف شود.

مدل برداری مبحث-محور[۴۱] *(TBVM)* اصول مباحث بنیادین را مطرح می‌کند. مباحث بنیادین مباحثی هستند که مستقل تعریف شده‌اند.

در اینجا یک تفاوت عمده با مدل برداری استاندارد وجود دارد و آن این است که TBVM اسناد را به یک فضای d بُعدی R می‌برد و برای هر عبارت $t_i \in T$ در برابر هر کدام از این مباحث بنیادین یک رابط تعریف می‌شود و طبق آن هر عبارت بوسیله‌ی یک بردار مبحث بنیادین t_i در R نمایش داده می‌شود.

۲,۲. مدل گراف

در حالی که مدل فضای برداری مدل استاندارد به شمار می‌رود، برخی دیگر از مولفان روش دیگری را پیشنهاد می‌کنند، روشی اساس آن بر گراف استوار است. [۱۷, ۱۸]

در مقابل دیدگاه کیسه‌ی کلمات که در مدل برداری مدنظر بود، دید مشترک بین روش‌های مختلفی که بر اساس گراف مسئله را مدل می‌کنند، جامعیت ساختار اسناد است. انگیزه‌ای که در پس تحلیل ساختار نهفته‌است پس از بررسی این مثال مشخص می‌شود:

مثال ۲,۲ :

"یک روباه خرگوش را دنبال می‌کند."

"یک خرگوش روباه را دنبال می‌کند."

این دو جمله معنای کاملاً متفاوتی دارند در صورتی که دارای کلمات یکسانی هستند.

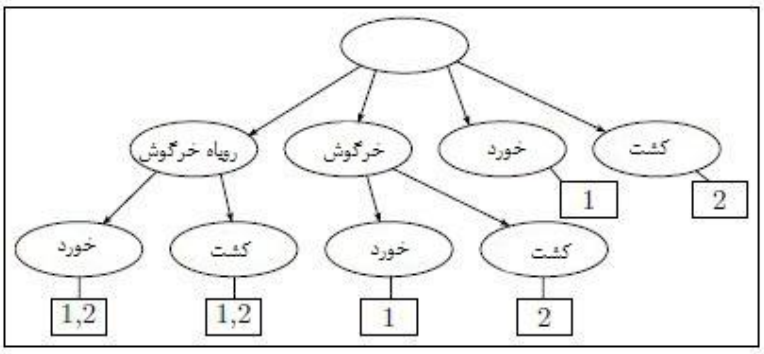
هاموندا[۴۲][۱۹] گراف اندیس سند[۴۳] را مطرح می‌کند. DIG یک گراف جهت‌دار است که هر راس آن نمایش دهنده‌ی یک کلمه‌ی منحصر به فرد در پیکره می‌باشد. بین دو راس v_i , v_j یک یال وجود دارد اگر و تنها اگر در سند v_j به دنبال v_i آمده باشد. برای هر سند این گراف به صورت مجزا نگهداری می‌شود و مشخص می‌کند که چه ساختارهایی در آن سند به کار رفته‌است. به طور مثال یک مسیر از v_1 به v_n نمایش دهنده‌ی جمله‌ای به طول n در سند است. شباهت بین اسناد به وسیله‌ی زیر ساختارهای مشترک آن‌ها شناسایی می‌شود.

[41] Topic Based Vector Model
[42] Hammonda
[43] Document Index Graph

خوشه‌بندی درختی پسوندی روشی مشابه DIG است. در این روش نیز ایده‌ی اصلی کار با ساختار عبارات در سند است[۱۸]. یک درخت پسوندی درختی جهت‌دار با یک ریشه است. اسناد نیز رشته‌های از کلمات در نظر گرفته می‌شوند و این ساختار خصوصیات زیر را به ارمغان می‌آورد:

- هر گرهی داخلی حداقل دو فرزند دارد.
- هر یال در درخت برچسبی با یک عبارت یکتا از سندی در پیکره دارد.
- هیچ دو یالی که از یک راس شروع می‌شوند با یک کلمه مشترک شروع نمی‌شوند.
- برای هر پسوندی مانند s در سند، یک گرهی پسوندی با برچسب s وجود دارد.
- در برگ‌ها اطلاعاتی در مورد اینکه عبارت از کجا منشا دارد وجود دارد.

برای مثال اگر دو جمله‌ی "روباه خرگوش کشت " و "روباه خرگوش خورد" را در نظر بگیریم گراف زیر به دست می‌آید:



شکل ۳ نمونه‌ای از مدل گراف برای نمایش متن

شنکر[۴۴] [۱۷]مدلی دیگر ارائه کرد که در آن هر سند توسط گرافی جهت‌دار نمایش داده می‌شد. در این مدل از راس i به راس j یالی وجود دارد اگر و تنها اگر عبارت t_j به دنبال عبارت t_i بیاید. در مدل ساده‌ی شنکر وابستگی‌ها بر اساس ترتیب کلمات تعریف می‌شوند.

مدل n-gram که توسعه‌ای از مدل ساده‌ی شنکر است به n کلمه‌ی بعدی توجه می‌کند. در گراف، برچسب هر یال نمایش دهنده‌ی فاصله‌ی راس بعدی از ابتدای جمله است.

۲,۳. مدل مبحث احتمالی:

یک رویکرد دیگر راجع به مدل کردن اسناد، این است که در نظر بگیریم هر سند در طی یک پروسه‌ی اتفاقی به وجود آمده‌است. تصور کنید سند در ابتدا با انتخاب یک توزیع از مباحث که توسط پیکره پوشش

[44] Schenker

داده می‌شوند ساخته شده‌است. سپس از میان این توزیع یک مبحث به صورت اتفاقی برگزیده می‌شود. در این صورت هر مبحث نمایشگر توزیعی از کلمات است. فرض کنید ما به صورت تصادفی مبحث *روباه* را انتخاب می‌کنیم و سپس کلمات زیرک، خز و ... را بر می‌گزینیم تا سند را تشکیل دهند. در کل اسناد می‌توانند توسط مخلوطی از مباحث ساخته‌شوند.

در واقع ما هنگام بررسی اسناد نه از توزیع مباحث و نه از توزیع مبحث-کلمات اطلاعی نداریم و در اصطلاح این مشخصه‌ها را متغیرهای مخفی می‌نامیم. با استفاده از مشاهداتمان که همان کلمات موجود در سند هستند و هم‌چنین استفاده از روش‌های استنتاج موخر این ساختار مخفی را کشف کنیم.

در ابتدا به فرموله‌کردن مسئله می‌پردازیم:

$P(z)$ را توزیع مباحث z در سند d در نظر می‌گیریم. طبق آنچه که قبلاً گفته‌شد فرض کنید که $P(z_i = j)$ را احتمال اینکه j-اُمین مبحث برای انتخاب i-اُمین کلمه در سند انتخاب شده باشد در نظر بگیریم. برنابراین احتمال اینکه کلمه‌ی w_i در مبحث j-اُم به عنوان i-اُمین کلمه انتخاب شده باشد برابر $P(w_i | z_i = j)$ خواهد بود. بنابراین توزیع احتمال کلمات در یک سند از رابطه‌ی زیر به دست می‌آید:

$$P(w_i) = \sum_{j=1}^{k} P(w_i|z_i = j)P(z_i = j)$$

رابطه ۳

برخی نمایش‌های بصری مثل نمودار صفحه‌ای نیز ارائه شده‌اند تا راحت‌تر بتوان این مدل را نمایش داد. در واقع این مدل نیز اسناد به شکل کیسه‌های کلمات در نظر می‌گیرد اما توسعه‌هایی نیز به وجود آمده‌اند تا برای ترتیب کلمات نیز اهمیت قائل شوند. بعدها این قابلیت که مولف و سال نوشته‌شدن سند را نیز بتوان از آن استخراج کرد نیز به این روش‌ها افزوده شد.

فصل سوم:

پیش پردازش متن و کاهش ابعاد

۳. پیش‌پردازش متن و کاهش ابعاد

پیش پردازش شامل مراحلی است که در آن سندهای متن خام به عنوان ورودی داده شده و خروجی آن مجموعه ای از کلمات است (که می‌توان شامل یک کلمه و یا n-gram[۴۵] باشد) که می‌تواند در مدل فضای بردار استفاده شود. در این قسمت ابتدا به بررسی روش‌های ابتدایی پیش پردازش متن می‌پردازیم. در قسمت دوم این فصل به بررسی روش‌های کاهش ابعاد می‌پردازیم. در پردازش متن در مدل فضای بردار هر کلمه به عنوان یک ویژگی محسوب شده و با بالا رفتن تعداد اسناد و همچنین تعداد کلمات، ابعاد مسئله به شدت بالا خواهد رفت و پیچیدگی فراوانی را موجب می‌گردد. برای ساده تر شدن مسئله از روش‌های کاهش ابعاد استفاده می‌شود که با استفاده از این روش‌ها سعی می‌شود به صورتی که کمترین مقدار ممکن اطلاعات مفید از دست برود، ابعاد فضای مسئله را کاهش داد.

۳,۱. پیش پردازش

۳,۱,۱.فیلتر کردن

در این مرحله کاراکترهای خاصی که به نظر می‌رسد در مدل فضای بردار نمی‌تواند اطلاعات مفیدی را در اختیار قرار دهند حذف می‌شوند. همچنین این مرحله برای سندهای دارای ساختار خاص مانند صفحات وب بسیاری حیاتی می‌باشد زیرا باید تگ‌های اضافی حذف شده و یا شناخته شده و با توجه به مقدارشان وزنشان مشخص شود.

۳,۱,۲. نرمال سازی و اصلاح نویسه‌ها

یکی از مشکلات زبان فارسی وجود چند نمونه‌ی مختلف از یک نویسه است، که کار جستجو در متون فارسی را مشکل می‌کند. در این مرحله کاراکترهای غیر استاندارد با کاراکترهای استاندارد جایگزین می‌شوند و کاراکترهای اضافی نیز بسته به نوع پردازش از بین می‌روند تا واژه‌های یکسان در تمامی متن به یک صورت نوشته شده باشند.

[۴۵] دنباله ای از یک یا چند کلمه

۳,۱,۳. تکه تکه کردن

در این مرحله جملات تکه تکه شده و به صورت مجموعه ای از کلمات در می‌آید. روش‌های پیچیده تکه تکه کردن متن از پردازش زبان طبیعی برای این کار بهره می‌گیرد، به این صورت که از تجزیه ساختار گرامری متن برای به دست آوردن کلمات پرمعناتر از قبیل اسم‌ها استفاده می‌کنند.

۳,۱,۴.ریشه یابی

پروسه ریشه یابی کلمات را به صورت عبارت پایه آن در می‌آورد. برای مثال، کلمه "روش‌هایم"، "روشی"، "روشمند" هر سه از ریشه روش هستند. روش‌های ریشه یابی اغلب مبتنی بر زبان هستن. الگوریتم Porter یک الگوریتم ریشه یاب استاندارد برای زبان انگلیسی است.

۳,۱,۵. حذف کلمات توقف[۴۶]

کلمه ایست به کلمه ای گفته می‌شود که به تنهایی معنای خاصی را نمی‌رساند و در واقع به عنوان یک عنصر از مدل فضای بردار اطلاعات مفیدی در بر ندارد. یک روش ابتدایی برای حذف کلمات توقف مقایسه هر کلمه با مجموعه ای از کلمات توقف شناخته شده است. روش دیگر این است که ابتدا عملیات برچسب زنی اجزای سخن[۴۷] صورت گرفته و سپس تمامی تکه‌هایی که اسم، فعل و یا صفت نیستند را حذف کرد.

۳,۱,۶. هرس کردن

در این مرحله کلماتی که تکرار آن‌ها در پیکره اسناد بسیار نادر است را حذف می‌کنیم. پیش‌فرض انجام این عملیات این است که این کلمات، حتی اگر قدرت تمایز زیادی داشته باشند، تنها تعداد به ساخت تعداد بسیار اندکی از خوشه‌ها کمک می‌کنند. برای میزان تکرار معمولاً از یک حدّ از قبل تعیین شده، درصد کمی از تعداد کل کلمات موجود در پیکره اسناد، استفاده می‌شود. بعضی اوقات کلماتی هم که تکرار بسیار زیادی دارند (مثلاً ۴۰ درصد و بیشتر از سندها) حذف می‌شوند.

۳,۱,۷.یکسان سازی کلمات هم معنی

علاوه بر این مراحل که بسیار رایج می‌باشند، از یک پایگاه داده لغوی، به نام WordNet، برای از بین بردن واژه‌های هم معنی و به جای آن وارد کردن مفاهیم عام‌تر استفاده می‌شود.[۱۱]

[46] Stop Words
[47] Part-of-speech tagging

۳,۲. کاهش ابعاد[۴۸]

داده‌های با ابعاد زیاد باعث پیچیدگی محاسبات خواهند شد. در طول خوشه‌بندی N سند ممکن است M خصوصیت مختلف را در نظر بگیریم در حالی که M >> N. اما مسئله اینجاست که آیا واقعاً نیازی به بررسی این تعداد خصوصیت هست؟ ممکن است که بررسی همه‌ی این خصوصیات لازم نباشد؟ جواب این سؤال موجب ارائه‌ی راهکارهایی با عنوان کاهش ابعاد می‌شود.

در صورتی که کاهش ابعاد در یک فرآیند با ناظر انجام پذیرد، به آن انتخاب خصوصیات گوییم. در این فرآیند ناظر خصوصیاتی که معیارهای مشخصی داشته باشند انتخاب می‌کند.

در حالت دیگر می‌توانیم با استفاده از یک فرآیند بدون ناظر ویژگی‌ها را استخراج کنیم تا معیار بهینه‌سازی حاصل شود. به این فرآیند استخراج خصوصیات گفته می‌شود. استخراج خصوصیات طی یک تبدیل از فضای M-بُعدی به یک فضای K-بُعدی صورت می‌پذیرد که K < M. این تبدیل می‌تواند هم به صورت خطی و هم به صورت غیرخطی انجام شود.

۳,۲,۱.انتخاب خصوصیات[۴۹]

ادبیات مملو از ایده‌های مختلف برای انتخاب خصوصیات مناسب است.

شاید بتوان فیلتر کردن کلمات توقف را محبوب‌ترین و ساده‌ترین روش‌ها دانست. کلمه توقف به کلمه‌ای اطلاق می‌شود که یا بسامد حضورش در متن خیلی زیاد است، نظیر کلماتی چون "و"، "یا"، "چند" و ... یا حاوی معنای خاصی نیست یا حداقل معنای کمی در متن دارد مثل پیشوندها و ...

در حقیقت هنگامی که کلمه توقف را از سند حذف می‌کنیم کلماتی را حذف می‌کنیم که اطلاعاتی راجع به محتوای یک سند ندارند. با ایجاد یک لیست چند صد کلمه‌ای از کلمات توقف لطمه‌ای به باقی‌مانده‌ی زبان که چند برابر کلمه دارد نخواهد خورد.

می‌توان یک گام جلوتر رفت و علاوه بر کلماتی که بسامد زیادی دارند، رده‌ای از کلمات که خصوصیت مشابهی دارند را نیز حذف کنیم. برای مثال در جمله‌ی "روباه قرمز خرگوش خاکستری را گرفت" علاوه بر کلمه‌ی "را" که یک کلمه توقف است می‌توان کلمات "قرمز" و "خاکستری" را نیز حذف کرد زیرا بدون آن‌ها نیز معنا کامل است. بنابراین می‌توان جمله‌ی بالا را به صورت زیر در نظر گرفت: "روباه خرگوش گرفت."

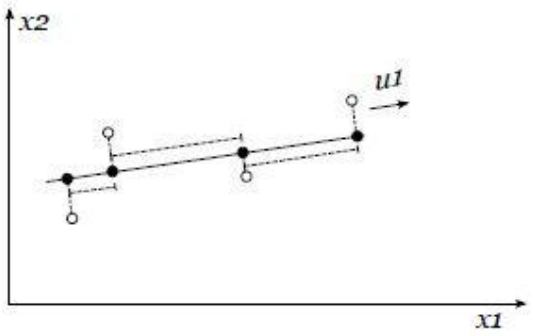
[48] Dimension reduction
[49] Feature Selection

این سبک از انتخاب رده‌های کلمات را انتخاب علامت افرازی از سخن[۵۰] می‌نامیم. از دیدگاه زبان‌شناسانه می‌دانیم که حداقل جزء با معنی شامل عبارت فعلی و عبارت اسمی است. دیگر بخش‌ها تنها به معنی می-افزایند. به همین دلیل بود که در جمله‌ی بالا تنها فعل و اسم‌ها را باقی گذاشتیم.

می‌توان نقش کلمات مثل فاعل یا مفعول و ... را از متن استخراج کرد. این کار را انتخاب علامت کاهش آماری[۵۱] می‌نامیم. این روش یک روش بسیار ساده است، به این ترتیب که کلماتی با بسامد خیلی زیاد و یا خیلی کم را از متن حذف می‌کند. حذف کردن کلماتی با بسامد بالا درست شبیه به tf-idf است، در حالی که حذف کلماتی با بسامد بسیار کم همانند حذف نویز از صدا است.

۳,۲,۲. تحلیل اجزای اصلی[۵۲]

یکی دیگر از روش‌های متداول برای کاهش دامنه، روش تحلیل اجزای اصلی یا PCA است که به عنوان تبدیل کاروهن-لاو[۵۳] نیز مشهور است. در [۲۰]دو تعریف مختلف برای PCA وجود دارد. در تعریف اول PCA را می-توان یک تصویرسازی خطی در نظر گرفت که میانگین هزینه‌ی تصویرسازی را حداقل می‌سازد، هزینه‌ای که میانگین مجذور فاصله‌ی بین نقاط تا تصویر آن‌ها در نظر گرفته می‌شود. در تعریف دوم PCA تصویر متعامد داده به یک فضای خطی با ابعاد کمتر است، به گونه‌ای که واریانس داده‌ی منعکس شده حداکثر است.



شکل ۴: PCA از یک فضای دو بُعدی به یک بُعدی. کمینه کردن فاصله بین نقاط و تصویر آن‌ها یا بیشینه کردن واریانس بین نقاط در تصویر.

PCA خلاصه می‌شود در پیدا کردن مجموعه‌ای از k بردار مستقل که توسط آن‌ها داده‌ها را تصویر کنیم، اما کدام بردارها باید انتخاب شوند؟ با استفاده از مشتق‌گیری‌های A.1 نیاز داریم که یک ماتریس کواریانس که

[50] Part-of-speech tag selection
[51] Statistical reduction tag selection
[52] Principal Component Analisys(PCA)
[53] Karhunen-Loeve

نمایان‌گر خطاهای کمینه‌سازی در تعریف اول هستند را به دست آوریم. بعد از محاسبه‌ی ماتریس باید k مقدار ویژه با بیشترین مقدار و بردارهای ویژه‌ی منطبق با آن‌ها را محاسبه کنیم. این‌ها k ویژگی اصلی اول هستند.

محاسبه‌ی تمام مقادیر ویژه برای یک ماتریس D×D هزینه‌ی زیادی دارد و از پیچیدگی $O(D^3)$ است، اما ما تنها نیاز داریم تا k جزء اصلی را پیدا کنیم، بنابراین پیچیدگی محاسباتی معادل با $O(Kd^2)$ است.

۳,۲,۳. تجزیه به مقادیر منحصر به فرد[۵۴]

تجزیه به مقادیر منحصر به فرد (SVD) به عنوان یک روش تقریب مرتبه پایین برای ماتریس‌ها استفاده می‌شود.

قضیه زیر SVD را تشریح می‌نماید:

فرض کنید r مرتبه یک ماتریس M x N به نام C باشد. بنابراین یک تجزیه به مقادیر منحصر به فرد از C به این صورت وجود دارد:

$$C = U\Sigma V^T$$

که در آن U یک ماتریس M x M است که ستون‌های آن بردارهای ویژه متعامد ماتریس CC^T است. V یک ماتریس N x N است که ستون‌های آن بردارهای ویژه ماتریس C^TC است. Σ یک ماتریس Σ یک ماتریس M x N قطری است که $\Sigma_{ii} = \sigma_i$ برای مقادیر i = 1 ... r غیر از ۰ که در آن $\sigma_i = \sqrt{\lambda_i}$ و $\lambda_i > \lambda_{i+1}$. به σ_i مقدار منحصر به فرد C می‌گوییم. در بعضی از نوشته‌ها Σ به صورت ماتریسی با ابعاد r x r که مقادیر صفر آن و همچنین مقادیر مشابه در U و V از آن خارج شده است. به این نمایش از SVD ، SVD کاهش یافته[۵۵] گفته می‌شود.

با استفاده از قضیه اکهارت-یانگ[۵۶] که در شکل۵ آمده است، از SVD می‌توان در ساخت یک تقریب مرتبه پایین از D به اسم D_k استفاده نمود،

برای ماتریس‌های خلوت با ابعاد *M x N* که داری *c* عنصر غیر صفر باشند. یک *SVD* در زمانی با مرتبه *O(cMN)* قابل محاسبه است.

برای اثبات بهینگی D_k به عنوان تقریبی از *D* رجوع کنید به [۲۱].

[54] Singular Value Decomposition (SVD)
[55] Reduced SVD
[56] Eckhart-Young

Construct the SVD of D_r as in equation …
From Σ_r identify the k largest similar singular values (k first, since $\lambda_i > \lambda_{i+1}$), and set the r – k other values to 0, yielding Σ_k
Compute D_K from $U\Sigma_k V^T$. The reduced form is obtained by pruning row vectors of length 0 in D_K

شکل ۵: قضیه اکهارت-یانگ

۳,۲,۴. تصویرسازی تصادفی[۵۷]

در تصویرسازی تصادفی هدف تصویر کردن داده‌های M-بُعدی به k-بُعد - با استفاده از یک ماتریس تصادفی k×m که ستون‌هایش طول یکسانی دارند - است. در صورتی که D ماتریس M×N اصلی باشد داریم:

$$D_{RP} = R \times D$$ رابطه ۶

D_{RP} مجموعه‌ی داده‌های کاهش یافته‌ی k×N است.

در حالی که دو روش تجزیه‌ی مقادیر تکین و تجزیه‌ی اجزای اصلی داده‌آگاه بودند یعنی عمل کاهش را با استفاده از داده‌های اصلی انجام می‌دادند روش تصویرسازی تصادفی به داده‌ها توجهی ندارد و هیچ‌گونه اطلاعاتی در مورد آن نیاز ندارد.

تصویرسازی تصادفی به لم جانسون- لیندنستراوس[۵۸] تکیه دارد که طبق آن در صورتی که نقاط یک فضای برداری به یک زیرفضای تصادفی تصویر شوند با احتمال زیادی فاصله‌ی بین نقاط ثابت باقی خواهد ماند. در صورتی که یک نفر بخواهد تنها خصوصیات اقلیدسی ماتریس را نگه دارد، می‌توان نشان داد که اندازه‌ی ابعاد را می‌توان به تا اندازه‌ی لگاریتم ماتریس اصلی کم کرد.

در واقع تصویرسازی تصادفی تصویرسازی نیست، چرا که R متعامد نیست. در حالی که اطمینان یافتن از متعامد بودن R برای یک ماتریس N×N به اندازه‌ی $O(N^3)$ هزینه دارد. متعامد نبودن R می‌تواند به پیدایش اعوجاج در مجموعه‌ی داده‌ها منجر شود. در هر صورت با توجه به استنتاج‌های هِشت- نیلسن [۲۲] که توسط آن اثبات می‌شود در یک فضا با ابعاد بالا، تعداد نسبتاً بیشتری بردار تقریباً متعامد نسبت به بردارهای متعامد وجود دارد همچنان می‌توان از تصویرسازی تصادفی را با خطای قابل قبول استفاده کرد.

برای ساختن R روش‌های زیادی پیشنهاد می‌شود. متداول‌ترین روش انتخاب k بردار تصادفی یکه‌ی M-بُعدی از یک توزیع (معمولاً توزیع گاوس یا یکنواخت) است.

[57] Random Projection
[58] Johnson-Lindenstrauss

با انتخاب R به روش یاد شده، اعمال کردن R بر روی هر بردار هزینه‌ی O(kN) دارد. برای حل شدن این مشکل آکلیوپتاس [۲۳] توزیع ساده‌ی زیر را معرفی می‌کند:

$$r_{ij} = \begin{cases} -\frac{\sqrt{3}}{N} with\ probablity \frac{1}{6} \\ 0 \qquad\qquad\qquad\quad \frac{2}{3} \\ \frac{\sqrt{3}}{N} \qquad\qquad\quad \frac{1}{6} \end{cases}$$

رابطه ۷

در عمل هر توزیع واریانس واحد با میانگین صفر از r_{ij} نگاشتی است که شرایط جانسون- لیندنستراوس لما را ارضا می‌کند. بالا رفتن سرعت ناشی از نحیف بودن R است که باعث می‌شود بتوان عناصر غیر صفر R را به صورت هوشمندانه‌ای ساماندهی کرد. این کار باعث می‌شود تا سرعت عمل در مقایسه با یک توزیع گاوسی تا سه برابر بیشتر شود بدون اینکه از کارایی چندان کاسته شود. متأسفانه این نحیف بودن برای تمام ورودی‌ها نتیجه‌ی مناسبی در بر نخواهد داشت. در صورتی که ورودی نیز خیلی نحیف باشد، D_{RP} ممکن است تهی شود، زیرا حاصل‌ضرب‌های $r_{ij}D_j$ صفر خواهند شد.

فصل چهارم:

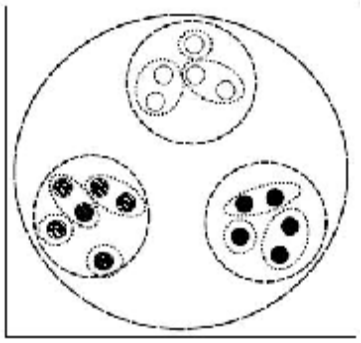
روش‌های خوشه‌بندی

۴. روش‌های خوشه‌بندی

برای دسته‌بندی یک گروه از عناصر راه‌های زیادی وجود دارد. هر مجموعه از خوشه‌ها می‌تواند به وسیله‌ی تعدادی از ویژگی‌ها توصیف شود. عضویت می‌تواند انحصاری[۵۹] یا هم‌پوشان[۶۰] باشد. در این‌جا عضویت بدین معناست که هر عنصر در نهایت می‌تواند عضو یک مجموعه باشد در حالی که مجموعه‌های هم‌پوشان دارای اعضای تکراری هستند. عضویت می‌تواند فازی[۶۱] یا دودویی نیز باشد. عضویت فازی یعنی نسبت دادن مقادیری در بازه‌ی [۰,۱] به کمیت‌های دسته‌بندی شده[۶۲] که این مقدار می‌تواند یک احتمال برای کلاس-بندی یا نزدیکی به خوشه‌ای خاص باشد.

خوشه‌بندی را می‌توان یک مسئله‌ی بهینه‌سازی در نظر گرفت که در آن هدف کمینه کردن انحراف هر خوشه به عنوان بردار کوانتیزه کردن اعضای آن است. این مسئله یک مسئله npسخت است یعنی به دست آوردن جواب بهینه زمان فوق‌العاده زیادی طلب می‌کند و ما ناچاریم تا به جواب‌های تقریبی قناعت کنیم. هر چند تمام الگوریتم‌های پیش رو از نوع الگوریتم‌های ابتکاری[۶۳] هستند اما برخی از آن‌ها کارایی بهتری دارند. در اکثر این الگوریتم‌ها یک پارامتر k وجود دارد که بیان‌گر تعداد مورد جستجوی خوشه‌هاست.

از یک منظر می‌توان خوشه‌ها را در یک ساختار سلسله‌مراتبی تصور کرد (شکل ۴-۱) که مجموعه‌های بالایی خود شامل زیرمجموعه‌هایی هستند که هر کدام جزئیات متفاوتی نسبت به دیگری دارند. الگوریتم‌ها معمولاً



شکل ۶

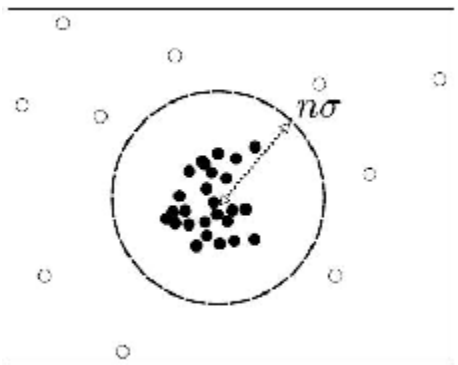
به یکی از صورت‌های بالا به پایین یا کف به سقف[۶۴] بر روی داده‌ها اعمال می‌شوند.

[59] Exclusive
[60] Overlapping
[61] Fuzzy
[62] Graduate
[63] Heuristic
[64] Bottom up

۴,۱. مقادیر دورافتاده

هنگامی که داده‌ها را در فضای n-بُعدی بررسی می‌کنیم به داده‌هایی برخورد می‌کنیم که فاصله‌ی زیادی از دیگر داده‌های موجود دارند. این مسئله می‌تواند نمایان‌گر آن باشد که قسمتی از داده‌های ورودی از دست رفته‌است یا اینکه این داده‌ها به هیچ دسته‌ای تعلق ندارند. این استثنائات بسته به نیاز کاربر ممکن است حاوی نکات مهمی باشند. برای مثال در سیستم‌های تشخیص نفوذ یا تقلب از این قبیل داده‌ها نقش تعیین کننده‌ای را ایفا می‌کنند.

البته این داده‌ها هنگامی که هدف یک خوشه‌بندی کامل باشد مشکلاتی را ایجاد می‌کنند زیرا داده‌های صحیح با داده‌های حاوی خطا در یک دسته قرار گرفته‌اند. این مسئله می‌توان باعث افزایش مرزهای خوشه-ها شود و یا خوشه‌های متفاوت به دلیل وجود مقادیر دورافتاده با یکدیگر ترکیب شوند و یا اینکه داده‌ها در خوشه‌های دیگر قرار گیرند. در روش‌های حریصانه، مثل پیوند یگانه و یا پیوند کامل که در قسمت ۴,۵ بررسی خواهند شد الگوریتم‌ها بیشترین تأثیر را می‌پذیرند. این خطا را می‌توان با استفاده از توابعی ملاک متوسط محور کم کرد چرا که آن را بر روی تمامی اعضای داخلی پخش می‌کنند. در صورتی که از وجود نویز بسیار در داده‌ها آگاهی وجود داشته باشد می‌توان به صورت هوشمندانه از روش‌هایی که برای همین منظور تعبیه شده‌اند (مثل DBSCAN[14]) استفاده کرد و یا اینکه این نویز را در مرحله‌ی اول از داده‌ها حذف کرد.



شکل ۷

۴,۲. دسته‌ای[۶۵]، برخط، جریان[۶۶]

راه‌های متفاوتی برای قضاوت در مورد چگونگی خوشه‌بندی وجود دارد. در صورتی که همه‌ی داده‌ها را به یک‌باره خوشه‌بندی کنیم، مثل روش K-معدل‌های[۶۷] استاندارد، خوشه‌ها را به صورت دسته‌ای تعیین کرده-ایم.

[65] Batch
[66] Stream
[67] K-means

در حالتی دیگر می‌توان خوشه‌ها را به منظور رقابت آزاد گذاشت. با وارد شدن هر داده خوشه‌ها بر سر این داده رقابت می‌کنند. این رقابت معمولاً بر اساس یک معیار مثل فاصله در نظر گرفته می‌شود. خوشه‌ای که در این رقابت بنده شود می‌تواند خود را به گونه‌ای تغییر دهد تا قرابت بیشتری با ورودی جدید داشته باشد و ورودی‌های مشابه دیگر را نیز جذب کند. خوشه‌بندی به این روش خوشه‌بندی برخط نام دارد. در مقایسه با روش دسته‌ای روش برخط حساسیت کمتری نسبت به خوشه‌های ابتدایی دارد و به گونه‌ای می‌توان گفت که این روش‌ها هرگز به یک حالت پایان نمی‌رسند.

در یک حالت دیگر می‌توان در نظر گرفت که داده‌ها به صورت جریان وارد می‌شوند. یک مقدار معین از داده در بافر ذخیره می‌شود و الگوریتم خوشه‌بندی بر روی آن اجرا می‌شود. خوشه‌بندی را می‌توان بعداً در یک بردار تاریخ که در مجموعه‌ی بعدی داده وزن‌دهی شده‌است خلاصه کرد.

۴,۳. معیارهای مشابهت

انتخاب تابع مجاورت کافی است زیرا تعیین می‌کند که خوشه‌ها در فضای n-بُعدی که سند ما در آن قرار دارد چگونه تفسیر شوند. هدف از خوشه‌بندی رسیدن به یک جدایی خوب در عین عام بودن است. بنابراین نباید در حین خوشه‌بندی به تعداد خیلی کم یا خیلی زیاد خوشه برخورد کرد. شهودی‌ترین معیار سنجش را می‌توان فاصله‌ی اقلیدسی در نظر گرفت که فاصله‌ی مستقیم دو شیء در فضای خطی است و از رابطه‌ی زیر به دست می‌آید:

$$d_2(p,q) = \sqrt{\sum_{k=1}^{n}(p_k - q_k)^2}$$

رابطه ۸

که در آن p و q می‌توانند نقطه یا بردار باشند.

دیگر معیارهای مبتنی بر فضای چند بُعدی هر چند کمتر مورد استفاده قرار می‌گیرند عبارتند از فاصله‌ی منهتن[۶۸] (یا فاصله‌ی تاکسی) و فاصله‌ی صفحه‌ی شطرنج (یا فاصله‌ی چبیشِو[۶۹]).

فاصله‌ی منهتن:

$$d_1(p,q) = \sum |p_k - q_k|$$

رابطه ۹

[68] Manhattan distance
[69] Chebyshev

فاصله‌ی صفحه‌ی چبیشِو:

$$d_{Chebyshev}(p,q) = max|p_k - q_k|$$

رابطه ۱۰

رابطه‌های پیچیده‌تر و سطح بالاتری نیز وجود دارند. مانند رابطه‌ی ماهالانوبیس[۷۰] که همبستگی مجموعه‌ی داده‌ها را استخراج می‌کند و متأسفانه برای پیاده‌سازی آن هزینه‌ی گزافی را باید متحمل شد. تمام روش‌های یاد شده تمایز بین بردارها را استخراج می‌کنند که مثالی از اصل کلی واگرایی برگمن[۷۱] است.

یک معیار کمی خبره تر برای پیدا کردن شباهت استفاده از ضریب همبستگی پیرسون[۷۲] است. ضریب همبستگی معیاری است برای این که دو مجموعه داده چقدر روی یک خط راست قرار دارند. فرمول محاسبه این ضریب از فاصله اقلیدسی پیچیده تر است، اما نتایج بهتری را برای زمانی که داده نرمال شده نیست می‌دهد. برای مثال در خوشه‌بندی متون وقتی سندها دارای تعداد کلمات متفاوتی باشند، ممکن است فاصله اقلیدسی نتیجه درستی به ما ندهد اما ضریب همبستگی نتایج بهتری در این شرایط خواهد داد.[۲۴]

رابطه ضریب همبستگی پیرسون به صورت زیر است:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

رابطه ۱۱

که در آن X و Y دو بردار داده هستند.

مشهورترین مثال یک تابع مشابهت واقعی مشابهت کسینوسی است که با ضرب نقطه‌ای تمام بردارهای خصوصیت محاسبه می‌شود:

$$s_c(p,q) = p.q = \sum_{k=1}^{n} p_k q_k$$

رابطه ۱۲

یک خصوصیت جذاب هنگام انجام بهینه‌سازی خرد ساده بودن آن یا بردارهای تُنُک است که می‌توان از هر مختصات غیر صفر از هر بردار گذر کرد. در اسناد تعداد عناصر غیر صفر بسته به پیکره کمتر از ۰٫۱٪ است و باعث می‌شود این معیار بسیار سبک باشد. تفسیر هندسی آن کسینوس زاویه‌ی بین بردارها در هر نقطه است.

[70] Mahalanobis
[71] Bregman
[72] Pearson Correlation Score

یک اصل که از احتمالات اقتباس شده‌است اندیس جاکارد[۷۳] یا ضرایب مشابهت جاکارد است که شباهت بین مجموعه‌هایی از مثال‌ها را استخراج می‌کند. ایده‌ای اصلی این طرح تعیین مشابهت بر اساس اشتراک مجموعه‌هاست که با اجتماع آن‌ها نرمال‌سازی شده‌است.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

رابطه ۱۳

با اندکی تغییر در روش وزن‌دهی یک روش احتمالاتی دیگر به نام ضرایب مشابهت سورنسن[۷۴] که معمولاً ضرایب تاس نامیده می‌شود.

$$S(A,B) = \frac{2|A \cap B|}{|A| + |B|}$$

رابطه ۱۴

این ایده در مورد مجموعه‌ها را می‌توان به بردارهای خصوصیت تعمیم داد. در صورتی که هر خصوصیت غیر صفر از هر بردار را به عنوان یک مشخصه عضویت دودویی در نظر بگیریم. اشتراک شامل ابعادی خواهد شد هر دوی خصوصیات غیر صفر هستند در حالی که اجتماع شامل ابعادی می‌شود که یکی یا هر دو در آن غیر صفرند. با ترکیب شباهت کسینوسی و این روش تفکر به ضرایب تانیموتو[۷۵] می‌رسیم:

$$T(p,q) = \frac{|p.q|^2}{|p|^2 + |q|^2 - |p.q|^2}$$

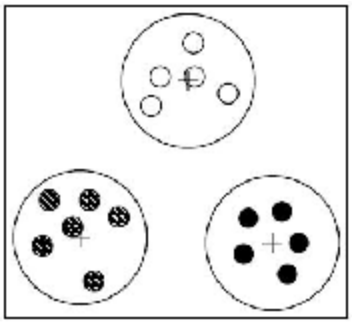
رابطه ۱۵

که نوع دیگری از شباهت جاکارد برای بردارهای مشخصه‌ی دوبُعدی است.

۴,۴. K-معدل‌های پایه

پایه‌ای‌ترین و احتمالاً شهودی‌ترین روش خوشه‌بندی تفسیر خصوصیات m-بُعدی به عنوان نقاط فضاست. ایده‌ی پایه معرفی نقاط گرانیگاه در فضا و به دنبال آن نسبت دادن نقاط به این مراکز بر اساس نزدیکی آن‌ها است. پس از آن این نقاط گرانیگاه بر اساس فاصله‌ی تمام عناصر مجموعه‌شان به روز می‌شوند و این فرآیند تا زمانی که دیگر هیچ انتصابی صورت نپذیرد ادامه می‌یابد. نقاط گرانیگاه در واقع هیچ نشان‌گر هیچ شیئی در ورودی ما نیستند. در شکل ۴-۳ نقاط گرانیگاه با علامت + مشخص شده‌اند. در صورتی که نقاط گرانیگاه به اجبار اشیاء واقعی در نظر گرفته شوند الگوریتم k-medioids نامیده می‌شود.

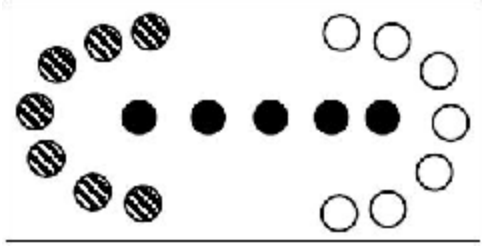
[73] Jaccard
[74] Sørensen
[75] Tanimoto



شکل ۸: شمایی از خوشه‌بندی به روش k-معدل‌های پایه

با توجه به الگوریتم، بیشتری زمان اجرای این الگوریتم مربوط به محاسبه‌ی بردار فاصله است. هر فاصله یا تشابه یک بار محاسبه می‌شود و پس از آن در هر مرحله‌ی انتصاب به بهترین نقطه منتسب می‌شود. در آخر نقاط جدیدی به عنوان گرانیگاه نسبت به تمام اعضای مجموعه انتخاب می‌شوند. به این ترتیب می‌توان الگوریتم را نسبت به تعداد اسناد و ابعاد خطی دانست. تعداد گرانیگاه‌ها نیز به صورت خطی با k مشخص می‌شود. در صورتی که الگوریتم با تعداد تکرار ثابت i اجرا شود پیچیدگی زمانی آن O(iknd) خواهد بود. در عمل این پیچیدگی برای خوشه‌بندی اسناد بسیار سبک خواهد بود چرا که بردارها بسیار تُنُک هستند و ابعاد هر بردار بسیار کمتر از d خواهد بود.

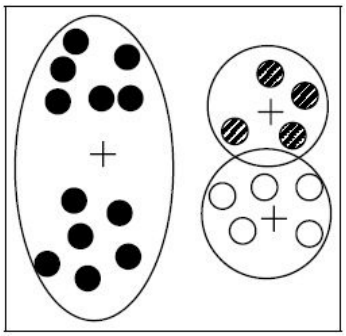
به انتساب شعاع محور این الگوریتم خوشه‌هایی کروی و در بعضی ابعاد فوق کروی تشکیل می‌دهد. در شکل ۴-۴ گروه‌هایی که شکل کروی ندارند مشخص شده‌اند. این امر نشان‌گر آن است که الگوریتم k-معدل‌ها موفق نمی‌شود به صورت عناصر مشابه را به صورت زیر تکه‌ای گروه‌بندی کند.



شکل ۹

هیچ تضمینی وجود ندارد که یک خوشه‌بندی بهینه از این طریق به دست آید. تنها می‌توان گفت که نتیجه از حالت ابتدایی بدتر نخواهد بود. به این ترتیب می‌توان حدس زد که یک مقداردهی اولیه‌ی نامناسب می‌تواند به بهینه‌سازی محلی مانند شکل منجر شود.

راه‌های متفاوتی برای غلبه بر این مشکل پیشنهاد شده‌اند، برخی از آن‌ها یک خوشه‌بندی سلسله‌مراتبی روی تعداد محدودی از ورودی‌ها انجام می‌دهند و نقاط مرکزی خوشه‌های به دست‌آمده را پایه‌ی الگوریتم اصلی در نظر می‌گیرند. می‌توان یک جستجوی محلی نیز انجام داد و با جابجا کردن نقاط بین خوشه‌ها بررسی کرد آیا این جابجایی باعث کمتر شدن انحراف می‌شود یا خیر.



شکل ۱۰

```
Initialize k centroids
Repeat
        For all objects in input do
                Assign each element to its closest centroid
        End for
        For all centroids do
                Compute the mean of the assigned points
                This mean now becomes the new centroid
        End for
Until all centoids remains unchanged or other termination criteria
```

شکل ۱۱: الگوریتم k-معدل‌های پایه

۴,۵. k-معدل‌های میانی[۷۶]

این شاخه از الگوریتم k-معدل‌ها به این صورت عمل می‌کند که در ابتدا همه‌ی داده‌ها را در یک خوشه در نظر می‌گیرد، پس از آن شروع به تقسیم کردن این خوشه‌ها می‌کند به این صورت که در هر مرحله بدترین خوشه انتخاب شده و به دو قسمت تقسیم می‌شود. انتخاب بدترین خوشه بر اساس عوامل مختلفی مثل

[76] Bisecting K-means

اندازه‌ی خوشه و یا میزان انحراف اعضای آن انجام می‌پذیرد. انتخاب کردن اندازه به عنوان این عامل، اندازه‌ی خوشه‌های متعادل را به دنبال خواهد داشت.

در صورتی که بدترین شرایط را در نظر بگیریم، یعنی در هر مرحله یک سند در یک خوشه قرار بگیرد و بقیه‌ی اسناد در خوشه‌ی دیگر، k-1 تقسیم خواهیم داشت و محاسبات مشابهت از پیچیدگی O(nk) خواهد بود که همان پیچیدگی k-معدل‌های معمولی است. اما در عمل این اتفاق به ندرت رخ می‌دهد و معمولاً خوشه‌ها به صورت متعادل‌تری وجود دارند که باعث می‌شود الگوریتم کمی سریع‌تر از حالت عادی اجرا شود.

```
Repeat
        pick a cluster to split according to a criterion
        for I = 1 to N do
                Bisect into two sub-cluster using k-means for k=2
                Keep track of best candidate
        End for
```

شکل ۱۲: الگوریتم k-معدل‌های میانی

۴,۶. الگوریتم Isodata

الگوریتم Isodata را می‌توان به عنوان یک بهبود بر روشی k-means در نظر گرفت. این الگوریتم نیز شبیه k-means سعی دارد هر نمونه را به خوشه ای با نزدیک‌ترین مرکز الحاق کند. اما بر خلاف آن، isodata تعداد خوشه‌ها را ثابت در نظر نمی‌گیرد و در این روش تعداد خوشه می‌توانند مقداری بیشتر و یا کمتر از مقدار تعیین شده توسط کاربر باشند. این الگوریتم خوشه‌های با تعداد سندهای بسیار کم حذف می‌کند. اگر تعداد خوشه‌ها بسیار زیاد شوند و یا این که اگر دو خوشه بسیار به هم نزدیک باشند، خوشه با یکدیگر ادغام می‌شوند. همچنین در صورتی که تعداد خوشه بسیار کم شوند و یا خوشه ای دارای سندهای غیر شبیه باشد، این خوشه می‌تواند بشکند و به دو خوشه مجزا تبدیل گردد.

الگوریتم Isodata دارای پارامترهای زیر می‌باشد:

- تعداد خوشه‌ها
- حداقل تعداد مجاز سندهای هر خوشه
- حداقل فاصله مجاز بین مرکز دو خوشه

- حداکثر واریانس مجاز سندها برای یک خوشه
- حداکثر تعداد تکرار الگوریتم
- حداکثر تعداد ادغام خوشه‌ها در هر دور

الگوریتم Isodata (گام‌های شماره ۱ تا ۴ همان الگوریتم k-means است):

۱. مرکز خوشه‌ها را مشخص کن.
۲. برای هر نمونه، نزدیک‌ترین مرکز خوشه را به آن پیدا کن. نمونه را در خوشه مربوط به آن مرکز قرار بده.
۳. مرکز خوشه‌های ایجاد شده را محاسبه کن.
۴. اگر مرکز حداقل یکی از خوشه‌ها تغییر کرد به گام ۲ برگرد.
۵. خوشه‌هایی را که نمونه‌های کمتر از حداقل تعداد مجاز سندها دارند را حذف کن. همچنین نمونه‌های موجود در این خوشه‌ها را نیز حذف کن.
۶. اگر تعداد خوشه‌ها بزرگ‌تر یا مساوی با ۲برابر پارامتر تعداد خوشه‌هاست یا شماره این تکرار زوج است، به گام ۷ برو وگرنه به گام ۸ برو.
۷. اگر فاصله بین دو مرکز خوشه کمتر از حداقل فاصله مجاز است، دو خوشه را ادغام کن و مرکز خوشه جدید را محاسبه کن. این گام را تا رسیدن به حداکثر تعداد ادغام ادامه بده.
۸. اگر تعداد خوشه‌ها کمتر یا مساوی نصف پارامتر تعداد خوشه‌ها بود یا شماره تکرار فرد بود، گام ۹ را اجرا کن وگرنه به گام ۱۰ برو.
۹. اگر واریانس خوشه ای بیشتر از حداکثر مقدار مجاز بود. خوشه را به دو خوشه بشکن و مرکز هر یک از آن دو را محاسبه کن.
۱۰. اگر شماره تکرار مساوی با حداکثر تعداد تکرار الگوریتم بود و یا هیچ خوشه تغییر نکرد خارج شو وگرنه به گام ۵ برگرد.

باید به این نکته توجه داشت که الگوریتم isodata تضمین نمی‌کند که همگرا شود پس باید تعداد مشخصی برای تکرار آن در نظر گرفت.

۴,۷. خوشه‌بندی متراکم سلسله‌مراتبی[۷۷] (HAC)

با رفتار کردن درباره‌ی هر شیء به عنوان یک خوشه و پس از آن ادغام کردن خوشه‌ها با یکدیگر برای رسیدن به یک ریشه، داده‌ها به صورت یک درخت در می‌آیند. گروه‌بندی دوتایی نیازمند دانستن بهترین خوشه‌ها برای ادغام است که تنها در اثر محاسبه‌ی مشابهت در هر دور یا به یادسپاری مشابهت در یک مرحله به دست می‌آید. روش اول در واقع عملی نیست چرا که محاسبه‌ی این ماتریس مشابهت در هر مرحله نیازمند محاسباتی با پیچیدگی $O(n^2)$ است.

```
Compute the similarity matrix
Repeat
        Find two best candidates according to criterion
        Save these two in the hierarchy as sub clusters
        Insert new cluster containing elements of both clusters
        Remove the old two from the list of active clusters
Until k or one cluster remains
```

شکل ۱۳: الگوریتم سلسله مراتبی متراکم

خوشه‌بندی پیوند یگانه ارزان‌ترین و سرراست‌ترین مقیاس برای ادغام است. به صورت محلی با در نظر گرفتن نزدیک‌ترین نقاط بین خوشه‌ها میزان مشابهت آن‌ها مشخص می‌شود. به بیان دیگر مشابه‌ترین عناصر میزان مشابهت کل خوشه‌ها را تعیین می‌کنند. با مرتب کردن مقادیر ماتریس مشابهت مرحله‌ی ادغام در زمان خطی انجام می‌پذیرد. یعنی زمان کل خوشه‌بندی پیوند یگانه به وسیله‌ی ماتریس مشابهت از $O(n^2)$ است.

در خوشه‌بندی پیوند کامل قاعده‌ی حریصانه به دنبال کمینه کردن قطر خوشه‌هاست. این مسئله باعث می‌شود تا دورترین نقاط هر خوشه نقاط مورد علاقه باشند. این یک قابلیت کلی است و به ساختار جاری بستگی دارد و در نهایت باعث مقداری محاسبات اضافی در فاز ادغام می‌شود. زمان محاسبه‌ی قاعده‌ی پیوند کامل از پیچیدگی $O(n^2 \log n)$ است.

نام‌های پیوند یگانه و پیوند کامل ریشه در تفسیر گرافی این الگوریتم‌ها دارند. ما می‌توانیم s_i را به عنوان مشابهت بین خوشه‌های ادغام شده در مرحله‌ی i در نظر بگیریم و $G(s_i)$ را گرافی که در آن تمام خوشه‌هایی که مشابهت آن‌ها حداقل s_i است به یکدیگر مرتبط شده‌اند. در خوشه‌بندی پیوند یگانه در مرحله‌ی i تمام عناصر $G(s_i)$ به یکدیگر متصل هستند و در پیوند کامل گروه‌های بیشینه‌ی $G(s_i)$.

[77] Hierarchical Agglomerative Clustering

خوشه‌بندی پیوند متوسط معیار را بر اساس تمام مشابهت‌های خوشه‌های تحت بررسی قرار می‌دهد نه فقط نقاط لبه‌ی خوشه‌ها. به بیان دیگر الگوریتم حریصانه به دنبال حداکثر کردن همبستگی عناصر خوشه است نه معیارهایی مثل قطر یا شباهت محلی. برای رسیدن به این هدف به اطلاعاتی بیشتر از ماتریس مشابهت نیاز است چرا که باید میانگین هر خوشه محاسبه شود. به این الگوریتم خوشه‌بندی گروه متوسط یا روش جفت‌گروه بدون وزن با متوسط حسابی (UPGMA[۷۸]) هم گفته می‌شود. UPGMA به صورت زیر محاسبه می‌شود:

$$\frac{1}{|c_1|.|c_2|}\sum_{x\in c_1}\sum_{y\in c_2} dist(x,y)$$

رابطه ۱۴

که در آن c_1 و c_2 دو خوشه‌ی مجزا هستند. زمان اجرای UPGMA از $O(n^2 \log n)$ است.

یک قدرت بزرگ HAC قطعی بودن آن است، یعنی با ورودی یکسان همیشه خروجی یکسان عرضه خواهد کرد. بدین ترتیب می‌توان مطمئن بود که با مقداردهی اولیه‌ی بد نتیجه‌ی بدی به دست نخواهدآمد. در درخت تولید شده خوشه‌بندی با k=1,2,…,n وجود دارد و دیگر نیازی به تعیین این مقیاس نیست. اما پیچیدگی حجم $O(n^2)$ باعث می‌شود که در خوشه‌بندی اسناد نتوان از این روش بهره‌مند شد چرا که برای پیکره‌های وسیع پاسخگو نیست. برخی اصلاحات نظیر الگوریتم باکشات[۷۹] معرفی شده‌اند که از پیچیدگی $O(\sqrt{nk})$ است.

۴,۸. گاز عصبی رونده[۸۰]

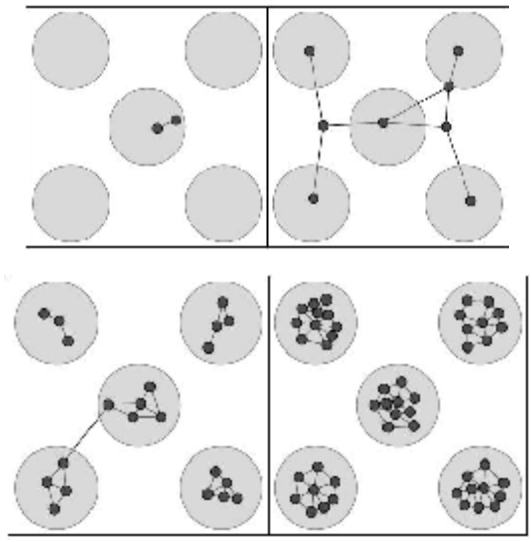
این روش دسته‌ای از گروه الگوریتم‌هایی است که اصول خود را از شبکه‌های عصبی خود سازمانده می‌گیرند، شبکه‌هایی که می‌توانند یک نگاشت از سیگنال با فضای حاوی ابعاد زیاد به ساختارهایی با ابعاد کمتر ارائه کنند. گاز عصبی رونده (GNG) به عنوان یک تعمیم گاز عصبی (NG) یک روش رقابتی است که یک توزیع را با رسم نمونه‌هایی فرا می‌گیرد و بعد بهترین جزء منطبق را تقویت می‌کند و آن را به سیگنال نزدیک‌تر می‌کند.

اجزا به عنوان گره‌هایی در گراف نمایش داده می‌شوند که یال‌ها مربوط به نزدیکی توپولوژیک هستند. اکتشاف در یک روش افزایشی انجام می‌شود و به جای شروع با k جزء مانند NG تنها با دو جزء شروع می‌شود و در هر α مرحله یک جزء اضافه می‌شود. این جزء جدید هنگامی اضافه می‌شود که بیش‌ترین نیاز به آن هست یعنی گره‌ای با بیشترین تراکم خطا و بدترین همسایه‌ی آن.

با نقاط دور افتاده به وسیله‌ی یک سن که به هر یال نسبت داده شده و ساختارهای قدیمی شبکه را رد می‌کند رفتار می‌شود. هنگامی که یک گره به عنوان بهترین نماینده‌ی سیگنال داده شده انتخاب می‌شود سن

[78] Unweighted Pair Group Method with Arithmetic Mean
[79] Buckshot
[80] Growing neural gas

تمام یال‌های همسایه‌ی آن افزایش می‌یابد. نقاط دور افتاده به ندرت به عنوان بهترین انطباق انتخاب می‌شوند و به این ترتیب به مرور زمان کاملاً منزوی می‌شوند تا اینکه الگوریتم به صورت کامل آن‌ها را از بین ببرد. این مسئله روش مذکور را قادر می‌سازد تا توزیع‌های متحرک را فراگیری کند هرچند که برای به دست آوردن بهترین نتیجه مقداری اصلاح مورد نیاز است.



شکل ۱۵

Start with two neurons
Repeat
Draw signal from distribution
Find best and second best matching units (bmu) and connect them
Move bmu and neighbors closer to signal
Age all touched local edges
Remove edges older than T_{max} and unconnected neurons
If iteration is multiple of α **then**
Find node with largest accumulated error
In this node find its worst neighbor
Insert new node between the connected nodes
Connect these nodes to the inserted node and remove old edge
Set error of new edges to half the largest and age to zero
End if
Decay error
Until termination criterion

شکل ۱۴: الگوریتم گاز عصبی روبنده

۴,۹. k-معدل‌های کروی برخط

این توسعه‌ی k-معدل‌ها از بردارهای نرمال شده بر اساسِ طولِ واحدِ اسناد و مشابهت کسینوسی استفاده می‌کنند تا مشابهت اسناد را اندازه‌گیری کنند. وقتی که بردارها همگی دارای یک طول باشند، بیشینه کردن مشابهت کسینوسی و کمینه کردن ریشه‌ی میانگین فاصله معادل هستند. برای اطمینان یافتن از اینکه گرانیگاه‌ها در اَبَر کره می‌مانند باید آن‌ها را در مرحله‌ی انتساب نرمال کرد. به صورت عمومی گرانیگاه‌ها تُنُک نیستند، بنابراین مرحله‌ی نرمال‌سازی هزینه‌ی O(m) خواهد داشت.

این اصلاحات را می‌توان با یک شِمای برخط به جای روش دسته‌ای انجام داد.

با این دو اصلاح در k-معدل‌های استاندارد k-معدل‌های کروی برخط (OSKM) را خواهیم داشت. گرانیگاه μ_i که نزدیک‌ترین گرانیگاه به بردار ورودی x_i است به صورت زیر به‌روز می‌شود:

$$\mu_i' \leftarrow \frac{\mu_i + \gamma(t)x_i}{|\mu_i + \gamma(t)x_i|}$$

رابطه ۱۵

که در آن $\gamma(t)$ نرخ یادگیری است و به تکرار t بستگی دارد.

Initialize unit length k centroid vectors $\{\mu_1, \dots, \mu_k\}$, $t \leftarrow 0$
While termination criteria not reached **do**
For each data vector x_i **do**
Find closest centroid $\mu_i = \arg\max x_i^T \mu_k$
Update μ_i accordingly
$t \leftarrow t + 1$
end for
end while

شکل ۱۶: الگوریتم k-معدل‌های کروی برخط

OSKM لزوماً یک الگوریتم افزایشی متغیر صعودی است.

ژونگ[۸۱][۲۵] سه نرخ مختلف یادگیری را تعریف می‌کند. یکی از آن‌ها با اندازه‌ی خوشه‌ها نسبت معکوس دارد و باعث به وجود آمدن خوشه‌های متعادل می‌شود. دومی یک نرخ ثابت ($\gamma = 0.5$) است. سومی یک شمای نمایی کاهشی به شکل زیر است:

[81] Zhong

$$\gamma(t) = \gamma_0 \left(\frac{\gamma_f}{\gamma_0}\right)^{\frac{t}{NM}}$$

رابطه ۱۶

که در آن N تعداد مقادیر ورودی و M تعداد تکرارهای دسته‌ای است. در [۲۵]دومی به صورت تجربی برای به دست آوردن نتایج بهتر در خوشه‌بندی نشان داده‌شده است.

گلوگاه محاسباتی نرمال‌سازی μ است. در صورتی که تمام M تکرار دسته‌ای انجام شوند کل پیچیدگی زمانی برای به دست آوردن تمام گرانیگاه‌ها O(MNm) است. به هر صورت با این امکان وجود دارد تا با اعمال اصلاحاتی در نرمال‌سازی زمان کل اجرای OSKM به $O(MN_{nz}k)$کاهش یابد که در آن N_{nz} تعداد عناصر غیر صفر در ماتریس عبارت-سند است. برای افزایش دوباره‌ی سرعت OSKM ژونگ پیشنهاد می‌کند که از یک نمونه-گیری استفاده شود. در هر m دسته، تعداد $\frac{mN}{M}$ نقطه‌ی نمونه‌گیری می‌شوند و بعد گرانیگاه‌های با توجه به آن‌ها به روز می‌شوند. انگیزه‌ی چنین کاری این است که با یک نرخ یادگیری گداختگی[۸۲] (همان طور که قبلاً ذکر شد)، گرانیگاه‌ها بیشتر به نرمی به سمت ساختار داده‌های محلی میل می‌کنند. با تکنیک نمونه-گیری، ژونگ زمان اجرا را به نصف کاهش داد بدون آنکه از کیفیت خوشه‌بندی کم شود.

۴,۱۰. خوشه‌بندی طیفی

با توجه به پیش‌زمینه‌ی افراد این روش را می‌توان به چند شکل مختلف تفسیر کرد. به هر صورت مشخص است که این روش بر روی یک گراف/ماتریس دوگانگیِ مشابهت دوبه‌دو عمل می‌کند. گراف مشابهت به چند روش مختلف می‌تواند تولید شود که هر روش گراف خاص خود را تولید می‌کند و این گراف‌ها یکسان نخواهند بود.

با متصل کردن نقاطی که مشابهتی بیشتر از یک مقدار حدی مثل ε گراف ε همسایگی به دست می‌آید. چون مشابهت‌های نتیجه بسیار هم‌گن هستند معمولاً آن‌ها را نادیده می‌گیرند و یک گراف بدون وزن تشکیل می‌دهند.

هم‌چنین ممکن است k عنصر منطبق برای هر رأس را نگه داشته شده و یک گراف به نام گراف k-نزدیک-ترین همسایه تشکیل شود. این گراف جهت‌دار خواهد شد چرا که این رابطه متقابل نیست. در روشی دیگر جهت رئوس در نظر گرفته نمی‌شود یا تنها رئوسی با k-بهترین مشابهت متقابل به یکدیگر متصل می‌شوند.

آخرین مورد، گراف کاملاً مرتبط است که در آن تمامی سلول‌های ماتریس مشابهت نگه داشته می‌شوند و نزدیکی بر اساس وزن یال‌ها به دست می‌آید. مفهوم انتخاب روش تشکیل گراف هنوز یک سؤال است هر چند هر کدام از روش‌ها موارد استفاده‌ی خود را بر اساس[۲۶] دارند.

[82] Annealing learning rate

مشخص شده‌است که یک مطابقت بین بردارهای ویژه‌ی گراف لاپلاس و دو افرازی‌ها[۸۳] وجود دارد. گراف لاپلاسی، از گراف ماتریس مشابهت و یک ویژگی دیگر به نام ماتریس درجه که یک ماتریس قطری است که در آن درجه‌ی هر رأس در درایه‌ی مربوط به آن وجود دارد به دست می‌آید.

با ساختن گراف مشابهت قدم بعدی یافتن شکاف‌های این ماتریس برای جدا کردن k خوشه است. بسته به اینکه کدام شکاف انتخاب شود، قدم آخر متفاوت خواهد بود. یک الگوریتم k-میانگین‌ها تمام k بردار خوشه را به خوشه‌های نهایی تبدیل می‌کند.

```
Compute the similarity matrix
Compute the laplacian maxrix L
Find the k first eigenvectors of L
Construct matrix A using eigenvectors as columns
        Partition into k clusters with k-means using the rows of A as initial centroids.
```

شکل ۱۷: الگوریتم خوشه‌بندی طیفی

خلاصه فصل

در این فصل روش‌های مختلف خوشه‌بندی را مطالعه نمودیم. به طور خلاصه این روش‌ها را از منظر نوع ورود داده‌ها می‌توان به سه گروه دسته‌ای، جریانی و برخط تقسیم نمود که توضیحات آن در خلال فصل بیان شد.

از منظر تعلق نمونه‌ها به خوشه‌ها نیز روش‌های خوشه بندی در دو دسته مختلف قرار می‌گیرند. دسته اول روش‌هایی هستند که فضای نمونه‌ها را به چند خوشه افراز می‌کنند و هر نمونه تنها به یک خوشه تعلق دارد از قبیل روش k-معدل‌های پایه و توسعه‌های آن. دسته دوم مربوط به روش‌هایی می‌شود که در آن هر نمونه می‌تواند به خوشه‌های متفاوتی تعلق داشته باشد مانند روش سلسله مراتبی که بسته به این که در کدام رده از سلسله مراتب به نمونه‌ها نگاه می‌کنیم، خوشه‌های متفاوتی خواهیم داشت.

از منظر دیگر می‌توان روش‌های خوشه بندی را به دو دسته دارای پارامتر و بدون پارامتر نیز تقسیم نمود. در روش‌های دارای پارامتر تعداد خوشه‌ها به عنوان پارامتر مسئله از قبل مشخص هستند اما در روش‌های بدون پارامتر مانند خوشه‌بندی سلسله مراتبی، لازم نیست تعداد خوشه‌ها از قبل در مسئله معین شوند.

[83] Bipartition

فصل پنجم:

روش‌های اندازه گیری اعتبار خوشه‌ها

۵. روش‌های اندازه گیری اعتبار خوشه‌ها

نتایج حاصل از اعمال الگوریتم‌های خوشه‌بندی روی یک مجموعه داده با توجه به انتخاب‌های پارامترهای الگوریتم‌ها می‌تواند بسیار متفاوت از یکدیگر باشد. هدف از اعتبارسنجی خوشه‌ها یافتن خوشه‌هایی است که بهترین تناسب را با داده‌های مورد نظر داشته باشند. دو معیارِ پایه اندازه‌گیری پیشنهاد شده برای ارزیابی و انتخاب خوشه‌های بهینه عبارتند از:[۲۷]

- **تراکم**[۸۴]: داده‌های متعلق به یک خوشه بایستی تا حد ممکن به یکدیگر نزدیک باشند. معیار رایج برای تعیین میزان تراکم داده‌ها واریانس داده‌ها است.
- **جدایی**[۸۵]: خوشه‌ها خود بایستی به اندازه کافی از یکدیگر جدا باشند. سه راه برای سنجش میزان جدایی خوشه‌ها مورد استفاده قرار می‌گیرد که عبارتند از:
  - فاصله بین نزدیک‌ترین داده‌ها از دو خوشه
  - فاصله بین دورترین داده‌ها از دو خوشه
  - فاصله بین مراکز خوشه‌ها

همچنین روش‌های ارزیابی خوشه‌های حاصل از خوشه‌بندی را به صورت سه دسته تقسیم می‌کنند که عبارتند از:

- معیارهای خارجی[۸۶]
- معیارهای درونی[۸۷]
- معیارهای نسبی[۸۸]

هم معیارهای خارجی و هم معیارهای درونی بر مبنای روش‌های آماری عمل می‌کنند و پیچیدگی محاسباتی بالایی را نیز دارا هستند. معیارهای خارجی عمل ارزیابی خوشه‌ها را با استفاده از بینش خاص کاربران انجام می‌دهند. معیارهای درونی عمل ارزیابی خوشه‌ها را با استفاده از مقادیری که از خوشه‌ها و نمای آن‌ها محاسبه می‌شود، انجام می‌دهند.

پایه معیارهای نسبی، مقایسه بین شماهای خوشه‌بندی (الگوریتم به علاوه پارامترهای آن) مختلف است. یک و یا چندین روش مختلف خوشه‌بندی چندین بار با پارامترهای مختلف روی یک مجموعه داده اجرا می‌شوند و بهترین شمای خوشه‌بندی از بین تمام شماها انتخاب می‌شود. در این روش مبنای مقایسه،

[84] Compactness
[85] Seperation
[86] External Criteria
[87] Internal Criteria
[88] Relative Criteria

شاخص‌های اعتبارسنجی (Validity-Index) هستند. شاخص‌های ارزیابی بسیار متنوعی پیشنهاد شده‌اند که در این قسمت سعی می‌شوند تعدادی از رایج‌ترین آن‌ها معرفی شوند.

۵,۱. شاخص‌های اعتبارسنجی

شاخص‌های اعتبارسنجی برای سنجش میزان صحت[۸۹] نتایج خوشه‌بندی به منظور مقایسه بین روش‌های خوشه‌بندی مختلف یا مقایسه نتایج حاصل از یک روش با پارامترهای مختلف مورد استفاده قرار می‌گیرند.

در جدول زیر مجموعه‌ای از علائم استفاده شده در ادامه این بخش ارائه شده است:[۲۷]

علامت	مفهوم
n_c	تعداد خوشه‌ها
d	تعداد ابعاد
$d(x,y)$	فاصله بین دو داده
$\overline{X_j}$	امید ریاضی زامین بعد داده‌ها
$\|X\|$	$\sqrt{X^T X}$؛ X^T یک بردار ستونی است
n_{ij}	تعداد داده‌های درون زامین بعد از خوشه iام
n_j	تعداد عناصر زامین بعد از تمام داده‌ها
v_i	نقطه مرکز خوشه iام
c_i	iامین خوشه
$\|c_i\|$	تعداد داده‌های درون iامین خوشه

جدول ۲: علائم استفاده شده در این بخش

[89] goodness

۵,۱,۱. شاخص دون[۹۰]

این معیار توسط رابطه زیر تعریف می‌شود:

$$D = \min_{i=1\ldots n_c} \left\{ \min_{j=i+1\ldots n_c} \left(\frac{d(c_i, c_j)}{\max_{k=1\ldots n_c} (diam(c_k))} \right) \right\} \qquad \text{(رابطه ۱۶)}$$

که $d(x,y)$ و $diam(c_i)$ در آن به ترتیب با روابط ۱۷ و ۱۸ محاسبه می‌شوند.

$$d(c_i, c_j) = \min_{x \in c_i, y \in c_j} \{d(x, y)\} \qquad \text{(رابطه ۱۷)}$$

$$diam(c_i) = \max_{x, y \in c_i} \{d(x, y)\} \qquad \text{(رابطه ۱۸)}$$

اگر مجموعه داده‌ای، دارای خوشه‌هایی جداپذیر باشد، انتظار می‌رود فاصله بین خوشه‌ها زیاد و قطر خوشه‌های آن کوچک باشد. در **نتیجه مقداری بزرگ‌تر برای رابطه این معیار مقداری مطلوب‌تر** است. معایب این معیار عبارتند از:

- محاسبه زمان بر
- حساسیت به نویز (قطر خوشه‌ها در صورت وجود یک داده نویزی می‌تواند بسیار تغییر کند.)

۵,۱,۲. شاخص دیویس بولدین[۹۱]

این معیار از شباهت بین دو خوشه (R_{ij}) استفاده می‌کند که بر اساس پراکندگی یک خوشه (s_i) و عدم شباهت بین دو خوشه (d_{ij}) تعریف می‌شود. شباهت بین دو خوشه را می‌توان به صورت‌های مختلفی تعریف کرد ولی بایستی شرایط زیر را دارا باشد.

- $R_{ij} \geq 0$
- $R_{ij} = R_{ji}$
- اگر s_i و s_j هر دو برابر صفر باشند آنگاه R_{ij} نیز برابر صفر باشد.
- اگر $s_j > s_k$ و $d_{ij} = d_{ik}$ آنگاه $R_{ij} > R_{ik}$
- اگر $s_j = s_k$ و $d_{ij} < d_{ik}$ آنگاه $R_{ij} > R_{ik}$

معمولاً شباهت بین دو خوشه به صورت زیر تعریف می‌شود:

[90] Dunn index
[91] Davis Bouldin index

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$ (رابطه ۱۹)

که در آن d_{ij} و s_i با روابط زیر محاسبه می‌شوند.

$$d_{ij} = d(v_i, v_j)$$ (رابطه ۲۰)

$$s_i = \frac{1}{\|c_i\|} \sum_{x \in c_i} d(x, v_i)$$ (رابطه ۲۱)

با توجه به مطالب بیان شده و تعریف شباهت بین دو خوشه شاخص دیویس بولدین به صورت زیر تعریف می‌شود.

$$DB = \frac{1}{n_c} \sum_{i=1}^{n_c} R_i$$ (رابطه ۲۲)

که R_i در آن به صورت زیر محاسبه می‌شود.

$$R_i = \max_{j=1...nc, i \neq j} (R_{ij}), i = 1...n_c$$ (رابطه ۲۳)

این شاخص در واقع میانگین شباهت بین هر خوشه با شبیه‌ترین خوشه به آن را محاسبه می‌کند. می‌توان دریافت که **هرچه مقدار این شاخص بیشتر باشد، خوشه‌های بهتری** تولید شده است.

۵،۱،۳. شاخص‌های اعتبارسنجی ریشه میانگین مربع انحراف از معیار[۹۲] و ریشه R[۹۳]

هرچند این شاخص‌ها معمولاً در اعتبارسنجی الگوریتم‌های سلسله مراتبی مورد استفاده قرار می‌گیرند ولی قابلیت ارزیابی نتایج سایر تکنیک‌های خوشه‌بندی را نیز دارا می‌باشند. در شاخص اعتبارسنجی RMSSDT (root – mean– square standard deviation) از واریانس خوشه‌ها استفاده می‌شود که به شکل رسمی می‌توان از رابطه ۱۶ برای محاسبه آن استفاده کرد.

$$RMSSDT = \sqrt{\frac{\sum_{\substack{i=1...n_c \\ j=1...d}} \sum_{k=1}^{n_{ij}} (x_k - \overline{x_j})^2}{\sum_{\substack{i=1...n_c \\ j=1...d}} (n_{ij} - 1)}}$$ (رابطه۲۴)

[92] RMSSDT
[93] RS

با توجه به رابطه بالا و این‌که این معیار میزان همگنی خوشه‌ها را اندازه می‌گیرد، می‌توان دریافت که **هرچه مقدار آن کمتر باشد نشان دهنده خوشه‌بندی بهتر** داده‌ها است.

شاخص اعتبارسنجی RS (R Square) که با استفاده از رابطه ۲۵، ۲۶ و ۲۷ تعریف می‌شود، معیاری برای بیان عدم‌تشابه بین خوشه‌ها است. به این شاخص درج همگنی بین گروهی نیز گفته می‌شود. مقادیر آن به بازه اعداد بین ۰ تا ۱ محدود می‌باشد. که ۰ نشان دهنده نبودن هیچ تفاوتی بین خوشه‌ها و ۱ نشان دهنده وجود تفاوتی قابل توجه بین خوشه‌ها است.

$$RS = \frac{SS_t - SS_w}{SS_t} \qquad \text{(رابطه ۲۵)}$$

$$SS_t = \sum_{j=1}^{d}\sum_{k=1}^{n_j}(x_k - \overline{x_j})^2 \qquad \text{(رابطه ۲۶)}$$

$$SS_w = \sum_{\substack{i=1\ldots n_c \\ j=1\ldots d}}\sum_{k=1}^{n_{ij}}(x_k - \overline{x_j})^2 \qquad \text{(رابطه ۲۷)}$$

۵،۱،۴. شاخص اعتبارسنجی SD

اساس شاخص اعتبارسنجی SD، میانگین پراکندگی (Average Scattering) و جدایی کلی (Total Sepration) خوشه‌ها است. پراکندگی از طریق محاسبه واریانس خوشه‌ها و واریانس کل مجموعه داده‌ها به دست می‌آید. با توجه به اینکه این معیار هم از میزان همگنی داده‌ها و هم از میزان تراکم خوشه‌ها بهره می‌برد معیار مناسبی برای ارزیابی خوشه‌ها محسوب می‌شود. واریانس مجموعه داده‌ها را با روابط ۲۸ و ۲۹ و نیز واریانس یک خوشه را با روابط ۳۰ و ۳۱ می‌توان محاسبه کرد.

واریانس مجموعه داده‌ها

$$\sigma_x^p = \frac{1}{n}\sum_{k=1}^{n}(x_k^p - \overline{x^p})^2 \qquad (۲۸)$$

$$\sigma(X) = \begin{bmatrix} \sigma_x^1 \\ \vdots \\ \sigma_x^d \end{bmatrix} \qquad (۲۹)$$

واریانس یک خوشه

$$\sigma_{v_i}^p = \frac{1}{\|c_i\|}\sum_{k=1}^{n}(x_k^p - \overline{x_i^p})^2 \qquad (۳۰)$$

$$\sigma(v_i) = \begin{bmatrix} \sigma_{v_i}^1 \\ \vdots \\ \sigma_{v_i}^d \end{bmatrix} \qquad (۳۱)$$

با توجه به واریانس‌های محاسبه شده با روابط بالا، میانگین پراکندگی خوشه‌ها از رابطه زیر محاسبه می‌شود.

$$Scatt = \frac{1}{n_c}\sum_{i=1}^{n_c}\frac{\|\sigma(v_i)\|}{\|\sigma(x)\|} \qquad (۳۲)$$

همچنین میزان جدایی کلی داده‌ها که بر اساس فاصله مراکز خوشه‌ها از هم تعریف می‌شود، از رابطه زیر محاسبه می‌شود.

$$Dis = \frac{\max_{i,j=1\ldots n_c}\left(\left\|v_j - v_i\right\|\right)}{\min_{i,j=1\ldots n_c}\left(\left\|v_j - v_i\right\|\right)} \sum_{k=1}^{n_c}\left(\sum_{\substack{j=1 \\ i\neq j}}^{n_c}\left\|v_j - v_i\right\|\right)^{-1} \qquad (۳۳)$$

در نهایت شاخص SD با رابطه زیر تعریف می‌شود.

$$SD = \alpha\, Scatt + Dis \qquad (۳۴)$$

که α عمل وزنی برای رابطه است که برابر میزان جدایی خوشه‌ها در صورت داشتن حداکثر تعداد خوشه‌ها می‌باشد. **مقدار محاسبه شده توسط این معیار هرچه کوچک‌تر باشد به معنی خوشه‌بندی بهتر** است.

۵,۱,۵. شاخص اعتبارسنجی S_Dbw

همانند شاخص SD این معیار هم بر اساس تراکم درون خوشه‌ای و میزان جدایی خوشه‌ها اما در این شاخص سعی شده تا چگالی خوشه‌ها نیز دخیل شود. به شکل رسمی می‌توان گفت که شاخص S_Dbw از واریانس بین خوشه‌ای و واریانس درون خوشه‌ای استفاده می‌کند. واریانس بین خوشه‌ها مقدار میانگین پراکندگی خوشه‌ها را به دست می‌آورد که در رابطه ۳۲ نحوه محاسبه آن بیان شده است. مقدار چگالی درون خوشه‌ای نیز با رابطه زیر محاسبه می‌شود.

$$Dens\_bw = \frac{1}{n_c(n_c-1)} \sum_{i=1}^{n_c}\left(\sum_{\substack{j=1, \\ i\neq j}}^{n_c} \frac{density(u_{ij})}{\max\left\{density(v_i), density(v_j)\right\}}\right) \qquad (\text{رابطه } ۳۵)$$

که u_{ij} که در آن نقطه وسط خطی است که v_i و v_j را به هم وصل می‌کند. برای محاسبه تابع چگالی اطراف یک نقطه، تعداد نقاط درون ابر کره‌ای را که شعاع آن برابر میانگین انحراف از معیار خوشه‌ها است، شمارش می‌شود. میانگین انحراف از معیار خوشه‌ها به صورت زیر تعریف می‌شود.

$$stdev = \frac{1}{n_c}\sqrt{\sum_{i=1}^{n_c}\left\|\sigma(v_i)\right\|} \qquad \text{رابطه } (۳۶)$$

در نهایت معیار S_Dbw به صورت زیر تعریف می‌شود.

$$S\_Dbw = Scatt + Dens\_bw \qquad (\text{رابطه } ۳۷)$$

در شاخص S_Dbw سعی شده هر دو معیار خوبی خوشه‌ها با هم ترکیب شوند و تخمینی دقیق از خوشه‌های حاصل به دست آید. **مقدار کم برای این شاخص به معنی خوشه‌بندی بهتر** است.

خلاصه فصل

در این فصل با شاخص‌هایی برای سنجش اعتبار خوشه‌های حاصل از عملیات خوشه‌بندی روی اسناد آشنا شدیم. به طور کلی می‌توان معیارهای سنجش خوشه‌ها را به دو دسته معیار تقسیم کرد. اولین معیار میزان تراکم داده‌ها در خوشه است، یعنی هر چه داده های یک خوشه به هم نزدیک‌تر بوده و شباهت بیشتری داشته باشند، اعتبار خوشه‌ها بیشتر خواهد بود. در واقع هر چه واریانس خوشه‌ها کمتر باشد، این اعتبار بیشتر است.

معیار دوم مربوط به فاصله بین خوشه‌ها می‌باشد. این معیار به این معناست که هر چه فاصله بین دو خوشه بیشتر باشد، خوشه‌بندی بهتری صورت گرفته است. این فاصله به سه صورت قابل اندازه‌گیری می‌باشد.

۱- نزدیک‌ترین فاصله بین نمونه‌های دو خوشه

۲- دورترین فاصله بین نمونه‌های دو خوشه

۳- فاصله بین مرکز دو خوشه

فصل ششم:

پیاده‌سازی روش سلسله مراتبی

۶. پیاده‌سازی روش سلسله مراتبی

در این بخش به بررسی جزئیات مربوط به پیاده‌سازی روش خوشه‌بندی سلسله مراتبی بر روی متن فارسی می‌پردازیم. در بخش اول مجموعه داده مورد استفاده در این پیاده‌سازی را معرفی می‌نماییم. سپس به بررسی روش‌های مورد استفاده برای پیش پردازش متون پرداخته و در نهایت نتایج مربوط به خوشه‌بندی را مشاهده می‌نماییم.

۶,۱. مجموعه داده

در این پیاده‌سازی افرازی از نسخه شماره ۱ مجموعه پیکره متنی همشهری به عنوان مجموعه داده ورودی در خوشه‌بندی مورد استفاده قرار گرفته است.

مجموعه‌های متنی ابزارهای مهمی برای پیشبرد تحقیقات در تعدادی از شاخه‌های علوم کامپیوتر مانند بازیابی اطلاعات[۹۴]، زبانشناسی پیکره‌ای[۹۵] و زبانشناسی محاسباتی[۹۶] هستند. مجموعه آزمایش همشهری یکی از معتبرترین این منابع در زبان فارسی است. از این مجموعه در همایش‌های معتبر بین‌المللی Persian@CLEF2008 و Persian@CLEF2009 استفاده شده است.[۲۸]

یک مجموعه آزمایش[۹۷] دارای اجزاء زیر می‌باشد:

- یک مجموعه استاندارد باید به اندازه کافی بزرگ باشد تا بتوان آن را نماینده‌ای از متون فارسی در نظر گرفت و نتایج آزمایشات روی مجموعه را تعمیم داد.
- مجموعه‌ای از پرس‌وجوها
- داوری ارتباط اسناد مجموعه به پرس‌وجوها[۹۸]

مجموعه اسناد همشهری با خزش[۹۹] وب سایت همشهری و چندین مرحله پیش‌پردازش و برچسب‌گذاری حاصل آمده است. نسخه ۱ این مجموعه نمونه‌ای است که در همایش‌های CLEF در سال‌های ۲۰۰۸ و ۲۰۰۹ برای ارزیابی سامانه‌های ارزیابی سامانه‌های بازیابی اطلاعات تک‌منظوره[۱۰۰] مورد استفاده قرار گرفته است. نسخه ۲، آخرین نسخه مجموعه است که نسبت به نسخه ۱ بزرگ‌تر و جامع‌تر می‌باشد

[94] Information Retrieval
[95] Corpus Linguistics
[96] Computational Linguistics
[97] Test Collection
[98] Relevance Judgment
[99] Crawl
[100] Ad Hoc

۶,۲. پیش پردازش

پیش پردازش شامل مراحلی است که در آن سندهای متن خام به عنوان ورودی داده شده و خروجی آن مجموعه ای از کلمات است که می‌تواند در مدل فضای بردار استفاده شود.

در این پیاده‌سازی نیز مراحلی برای پیش پردازش متن پیکره مورد استفاده قرار گرفت تا داده برای خوشه‌بندی آماده شود. این مراحل در زیر شرح داده می‌شوند.

۶,۲,۱.حذف کاراکترهای اضافی و تکه تکه سازی

با توجه به این که پیکره مورد استفاده به صورت ساختاریافته و با ساختار XML بود، ابتدا متن خام سندها از داخل این ساختار استخراج شد و برچسب‌های XML حذف شدند. بعد از این مرحله اسناد تکه تکه شده و به مجموعه ای از کلمات تبدیل شدند. همچنین بعد از تکه تکه سازی تکه‌های یک و یا دو حرفی که عموماً اطلاعات مفیدی در اختیار نداشتند نیز حذف شدند.

۶,۲,۲. حذف کلمات توقف[۱۰۱]

کلمات توقف به کلماتی گفته می‌شود که به تنهایی اطلاعات مفید و متمایز کننده ای را در اختیار مدل فضای بردار قرار نمی‌دهند. در این مرحله این کلمات با استفاده از لیستی از کلمات توقف حذف شدند. در [29, 30]لیست کلماتی که به عنوان کلمات توقف شناخته شده و حذف شدند را مشاهده می‌نمایید:

[101] Stop Word

کلمات توقف فعلی:

آورد	بخواهد	بیاب	خواستن	کردی	گیرید
آوردم	بخواهم	بیابد	خواستند	کردید	گیریم
آوردن	بخواهند	بیابم	خواسته	کردیم	می‌شود
آوردند	بخواهی	بیابند	خواستی	کن	هست
آورده	بخواهید	بیابی	خواستید	کند	هستم
آوردی	بخواهیم	بیابید	خواستیم	کنم	هستند
آوردید	بکن	بیابیم	خواهد	کنند	هستی
آوردیم	بکند	بیاور	خواهم	کنی	هستید
آورم	بکنم	بیاورد	خواهند	کنید	هستیم
آورند	بکنند	بیاورم	خواهی	کنیم	یابد
آوری	بکنی	بیاورند	خواهید	گرفت	یابم
آورید	بکنید	بیاوری	خواهیم	گرفتم	یابند
آوریم	بکنیم	بیاورید	داد	گرفتن	یابی
آید	بگو	بیاوریم	دار	گرفتند	یابید
آیم	بگوید	بیاید	دارد	گرفته	یابیم
آیند	بگویم	بیایم	دارم	گرفتی	یافت
آیی	بگویند	بیایند	دارند	گرفتید	یافتم
آیید	بگویی	بیایی	داری	گرفتیم	یافتن
آییم	بگویید	بیایید	دارید	گفت	یافتند
باش	بگوییم	بیاییم	داریم	گفتم	یافته
باشد	بگیر	تواند	داشت	گفتن	یافتی
باشد	بگیرد	توانست	داشتم	گفتند	یافتید
باشم	بگیرم	توانستم	داشتن	گفته	یافتیم
باشند	بگیرند	توانستن	داشتند	گفتی	
باشی	بگیری	توانستند	داشته	گفتید	
باشید	بگیرید	توانسته	داشتی	گفتیم	
باشیم	بگیریم	توانستی	داشتید	گوید	

جدول ۳: کلمات توقف فعلی

لیست کلمات توقف غیرفعلی:

کلمه	اندازه	تکرار	کلمه	اندازه	تکرار
و	1	-	ای	2	2632
در	2	31438	نیز	3	2630
به	2	26871	تا	2	2441
از	2	19357	ما	2	2377
که	2	15985	باید	4	2304
این	3	15833	اند	3	1699
با	2	13041	هم	2	1635
می	2	12163	بود	3	1558
را	2	12081	نمی	3	1426
های	3	8955	هر	2	1424
برای	4	4733	یا	2	1333
ها	2	4118	دو	2	1321
آن	2	3684	آن‌ها	4	1282
وی	3	3673	اما	3	1262
یک	2	3253	دیگر	4	1259
خود	3	3221	اگر	3	1163
بر	2	3065	همچنین	6	1120

جدول۳: کلمات توقف غیرفعلی

۶،۲،۳. حذف کلمات پرتکرار و نادر

به راحتی می‌توان تعداد تکرار تمامی کلمات را در مجموعه داده‌ها به دست آورد. با داشتن این اطلاعات کلماتی که تعداد تکرار بسیار زیادی دارند را حذف می‌کنیم، زیرا می‌دانیم این کلمات در اکثر سندها تکرار شده‌اند و بار معنایی متمایز کننده ای نخواهند داشت. همچنین کلماتی که تعداد تکرار بسیار کمی دارند هم تاثیر ناچیزی در خوشه‌بندی می‌گذارند و آن‌ها را نیز حذف می‌کنیم.

تعیین حداقل و حداکثر میزان تکرار برای کلمات قاعده نظام‌مندی ندارد و با آزمون و خطا تعیین می‌شود.

۶,۲,۴. ریشه یابی

ریشه یابی به عملیاتی گفته می‌شود که طی آن کلمات به شکل ساده خود تبدیل شوند. با توجه به این که عملیات ریشه یابی وابسته به زبان است، باید از ریشه‌یاب‌های مختص به زبان فارسی استفاده نمود. متأسفانه در زبان فارسی ریشه یاب کارا و قدرتمند چندانی وجود ندارد. تنها به دو نمونه ابزار ریشه یابی فارسی Perstem [۳۱] و Virastyar را می‌توان نام برد که هر کدام از این ابزارها نیز دارای نقاط ضعف و مشکلاتی می‌باشند.

یکی از روش‌های ساده برای ریشه یابی کلمات این است که نشانه‌های جمع آن‌ها را حذف نماییم. به سادگی می‌توان کلمات نشانه جمع در انتهای کلمات از جمله "ها، ان، ات، های" را حذف نمود.

۶,۲,۵. یکسان‌سازی کلمات هم معنی

این عملیات به این معناست که واژه‌هایی که دارای معنای یکسان هستند همگی با یک واژه جایگزین شوند، به این ترتیب تأثیر معنای یکتایی که تمامی واژه‌های مختلف به آن اشاره می‌نمایند در خوشه‌بندی بیشتر خواهد شد. این عملیات با استفاده از یک شبکه واژگانی[۱۰۲] قابل انجام است.

"فارس نت" [۳۲]یک شبکه واژگانی برای زبان فارسی است که در این پیاده‌سازی از آن بهره برده شده است. نخستین نسخه از فارس نت شامل وردنت فارسی است که توسط آزمایشگاه پردازش زبان طبیعی دانشگاه شهید بهشتی و با حمایت مرکز تحقیقات مخابرات ایران ساخته شده است. این محصول در بسیاری کاربردهای پردازش زبان فارسی ازجمله ترجمه ماشینی، خلاصه سازی اخبار، جستجو و بازیابی اطلاعات، کشف هرزنامه‌ها، تحلیل اطلاعات متون و رمزنگاری معنایی نقش کلیدی بازی می‌کند. فارس نت در بردارنده زیر مجموعه ای از واژگان مورد استفاده در زبان فارسی نوشتاری معیار است که علاوه بر امکان استفاده در سیستم‌های پردازش زبان فارسی امکانات تبدیل دوزبانه را نیز فراهم می‌کند. نخستین نسخه از این واژ-هستان شناسی شامل ۱۰،۰۰۰ مجموعه هم معنا و ۱۸،۰۰۰ کلمه فارسی است. کلمات تحت پوشش این محصول دارای ۳ نوع مقوله نحوی (اسم، فعل و صفت) هستند و از بین پررخدادترین کلمات زبان فارسی انتخاب شده‌اند .

[102] Wordnet

۶,۳. روش خوشه‌بندی

در این پیاده‌سازی روش خوشه‌بندی سلسله مراتبی استفاده شده است که شرح آن در فصل۳ آمده است. این پیاده‌سازی به زبان python صورت گرفته است و در آن از معیار شباهت ضریب همبستگی پیرسون[۱۰۳] استفاده شده است. زیرا در این جا تعداد کلمات سندها با هم یکسان نیست. ضریب همبستگی پیرسون برای چنین شرایطی مناسب می‌باشد.

مقدار ضریب پیرسون عددی بین صفر و یک است به این صورت که اگر دو سند دقیقاً یکسان باشند یک خواهد بود و اگر دو سند رابطه ای با هم نداشته باشند عددی نزدیک به صفر خواهد بود. در این پیاده‌سازی چون می‌خواهیم از فاصله استفاده کنیم، مقدار ضریب مشابهت را از عدد یک کم می‌کنیم.

[103] Pearson Correlation Coefficient

۷. نتیجه‌گیری و کارهای آینده

خوشه‌بندی همان‌گونه که بیان شد، به کشف گروه‌هایی از داده‌های مشابه درون مجموعه‌ای از داده‌ها می‌پردازد، بدون هیچ اطلاع قبلی از کلاس‌های مربوط به داده‌ها. خوشه‌بندی متن مشابهت بین اسناد متنی را به دست آورده و از این طریق اسناد متنی را در گروه‌هایی دسته بندی می‌کند.

در این پژوهش ابتدا چگونگی نمایش و نگهداری اسناد را به عنوان مجموعه ای از داده‌ها مورد بررسی و مطالعه قرار دادیم. سپس روش‌های پیش پردازش متن در جهت آمادگی برای اجرای الگوریتم‌های خوشه‌بندی بررسی کردیم و روش‌های کاهش ابعاد را هم که از مدل‌های ریاضی و جبر خطی برای کاهش ابعاد فضای برداری مسئله استفاده می‌کرد بررسی نمودیم. در فصل بعدی روش‌های خوشه‌بندی را مطالعه کردیم. انواعی از روش‌های خوشه‌بندی تاکنون ارائه شده‌اند که وابسته به کاربرد می‌توان از آن‌ها استفاده کرد. در کل روش‌های خوشه‌بندی را می‌توان از منظری به دو بخش دارای پارامتر و بدون پارامتر تعبیر نمود که در روش‌های دارای پارامتر نیاز است تعداد خوشه‌ها را از قبل مشخص نماییم اما روش‌های بدون پارامتر، بدون این که تعداد خوشه‌ها را از قبل مشخص شده بدانند خوشه‌بندی می‌نمایند. در ادامه گروهی از این روش‌های که به الگوریتم‌های سلسله مراتبی خوشه‌بندی معروف هستند و یک نمودار که اولویت ترکیب داده‌ها برای تولید خوشه‌ها را ارائه می‌دهد، بررسی و پیاده‌سازی شد.

همچنین در طی این پیاده‌سازی روش‌های مختلف پیش پردازش برای زبان فارسی روی یک داده واقعی بررسی و آزمایش شده و ابزارهای مورد نیاز برای این روش مورد مطالعه قرار گرفت.

به عنوان کارهای آینده می‌توان از پیاده‌سازی مدل نمایشی که جایگاه معنایی جملات را نیز در بر گیرد و همچنین معیارهای مشابهتی که نقش‌ها و وابستگی‌های معنایی جملات را مورد مقایسه قرار می‌دهند، نام برد.

همچنین پیاده‌سازی روش‌های خوشه‌بندی متون به صورت توزیع شده به نحوی که بر روی داده‌های عظیم قابل انجام باشند به عنوان یکی دیگر از کارهای آینده می‌توان در نظر گرفت.

دیگر چالش قابل توجه در این پژوهش کمبود امکانات مناسب برای پیش‌پردازش متون به زبان فارسی از نظیر ریشه‌یاب کارا و .... در واقع پیش‌پردازش بهینه‌تر برای متون فارسی همچنان یک چالش پیش رو می‌باشد.

مراجع

1. Michael W.Berry, M.C., *Survey of Tect Mining: Clustering, Classification, and Retrieval, Second edition.* 2007.

2. Jacob Kogan, C.N., Marc Teboulle, *Clustering large and high dimensional data.* 2003.

3. گیری مشابهت جدید. ۲۰۱۰(چهارمین الهی, ع., بهبود خوشه بندی اسناد بر مبنای یک اندازه کنفرانس داده‌کاوی ایران)

4. Xu. Rui, W.D.I., *Survey of clustering algorithms. Neural Networks.* 2005.

5. Christopher Issal, M.E., *Document Clustering.* 2010(Master of Science Thesis).

6. *Introduction to Data Mining and Knowledge Discovery.*

7. Eamonn Keogh, S.L., Chotirat Ann Ratanamahatana, *Towards parameter-free datamining.* 2005.

8. M.Joshi, V.K.a., *What is datamining?* 2003.

9. Dagan, R.F.a.I., *Kdt - knowledge discovery in texts.* 1995(In Proc. of the First Int. Conf. on Knowledge Discovery (KDD).

10. Keller, F., *Clustering.*

11. Jajoo, P., *Document Clustering.* 2008.

12. G. Salton, A.W., and C. S. Yang, *A vector space model for automatic indexing. Communications of the ACM.* 1975.

13. Christopher D. Manning, P.N., Kasturi Varadarajan, *The planar k-means problem is np-hard.* 2009(proceedings of the 3rd international workshop of algorithms and computation): p. 274-285.

14. Man Lan, C.-L.T., Hwee-Boon Low, Sam-Yaun Sung, *A comprehensive Comparative study on term weighting schemes for text categorization with support vector machines.* ACM, 12005(special interest trucks and posters of the 14 international conference of World Wide Web): p. 1032-1033.

15. Sahlgren, M., *The word-Space Model: using distributional analysis to represent syntagnatic and paradigmatic relations between words in high dimensional vector space.* 2006.

16. George tsatsatonis, V.P., *A generalized vector space model for text retrieval based on semantic relatedness.* 2009(Proceedings of the 12th conference of the European chapter of association for computational linguistics: Student Research and Workshop): p. 70-78.

17. Adam Schenker, H.B., Mark Last, Abraham Kandel, *Clustering of web documents using graph representations. in Applied Graph Theory in Computer Vision and Pattern Recognition.* 2007: p. 247-265.

18. Oren Zamir, O.E., *Web Document Clustering: a feasibility demonstration*. 1998.

19. Hammouda, K.M. and M.S. Kamel, *Efficient phrase-based document indexing for web document clustering.* Knowledge and Data Engineering, IEEE Transactions on, 2004. 16(10): p. 1279-1296.

20. Bishop, C.M., *Pattern recognition and machine learning*. Vol. 4. 2006: springer New York.

21. Manning, C.D., P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Vol. 1. 2008: Cambridge University Press Cambridge.

22. Hecht-Nielsen, R., *Context vectors: general purpose approximate meaning representations self-organized from raw data.* Computational intelligence: Imitating life, 1994: p. 43-56.

23. Ailon, N. and B. Chazelle, *Faster dimension reduction.* Communications of the ACM, 2010. 53(2): p. 97-104.

24. Segaran, T., *Programming collective intelligence: building smart web 2.0 applications*. 2007: O'Reilly Media, Incorporated.

25. Zhong, S. *Efficient online spherical k-means clustering*. in *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*. 2005. IEEE.

26. Von Luxburg, U., *A tutorial on spectral clustering.* Statistics and computing, 2007. 17(4): p. 395-416.

27. Kovács, F., C. Legány, and A. Babos. *Cluster validity measurement techniques*. in *6th International Symposium of Hungarian Researchers on Computational Intelligence*. 2005. Citeseer.

28. مجموعه پیکره همشهری. [cited 2012; Available from: http://ece.ut.ac.ir/dbrg/hamshahri/fadownload.html.

29. Taghva, K., R. Beckley, and M. Sadeh, *A list of farsi stopwords.* Retrieved September, 2003. 7: p. 2006.

30. کیومرث شیخ اسماعیلی, ئ.س.ر., لیست کلمات ایست فارسی, in کنفرانس ملی داده‌کاوی, آزمایشگاه وب معنایی دانشگاه صنعتی شریف.

31. Jadidinejad, A., F. Mahmoudi, and J. Dehdari, *Evaluation of perstem: a simple and efficient stemming algorithm for Persian.* Multilingual Information Access Evaluation I. Text Retrieval Experiments, 2010: p. 98-101.

32. Shamsfard, M. *Developing FarsNet: A lexical ontology for Persian*. in *4th Global WordNet Conference, Szeged, Hungary*. 2008.