Á»I¨ ¸T¶ oM Án»o¶

JCCEM2013-1530

مروری بر متن کاوی و روش‌های آن

محمد رضا فیضی درخشی[۱]، شیما رشیدی[۲]، فاطمه محمودلو[۳]

[۱]استادیار گروه مهندسی کامپیوتر، دانشکده مهندسی برق و کامپیوتر، دانشگاه تبریز، mfeizi@tabrizu.ac.ir

[۲]دانشجوی کارشناسی ارشد، دانشکده علوم کامپیوتر، دانشگاه تبریز، s.rashidi90@ms.tabrizu.ac.ir

[۳]دانشجوی کارشناسی ارشد، دانشکده علوم کامپیوتر، دانشگاه تبریز، f.mahmoodlu90@ms.tabrizu.ac.ir

چکیده

رشد فزاینده پایگاه داده‌ها در زمینه‌های مختلف از فعالیت انسان باعث شده است که نیاز به ابزارهای قدرتمند جدید، برای تغییر دادن داده به دانش مفید، افزایش یابد. جهت برآوردن این نیاز، محققان به کاوش در زمینه‌های مختلف برای یافتن روش‌ها و ایده‌های مناسب پرداختند. متن کاوی یکی از زمینه‌های است که به دنبال استخراج اطلاعات مفید، از داده‌های متنی بدون ساختار، به وسیله شناسایی و اکتشاف الگوها می‌باشد. ایده اصلی متن کاوی، یافتن قطعات کوچک اطلاعات ازحجم زیاد داده‌های متنی، بدون نیاز به خواندن تمام آن است. در این مقاله با توجه به اهمیت این روش مختصراً به متن کاوی، زمینه های مرتبط با آن و برخی روش‌های رایج طبقه بندی و خوشه بندی پرداخته شده است. اگرچه بیان همه روش‌ها و کاربردها ممکن نیست، اما این مقاله می‌تواند دید کلی از متن کاوی را در ذهن خواننده ایجاد کرده و در صورت علاقه برای مطالعه بیشتر، فرد را به منابع مناسب هدایت کند.

واژه های کلیدی

بازیابی اطلاعات- خوشه بندی- طبقه بندی- متن کاوی.

مقدمه

بیشتر تلاش و وقت کاربران در جستجوهای ناکارآمد میان منابع اطلاعاتی تلف می‌شود. این نوع مشکل اضافه بار اطلاعاتی به خاطر قالب غیر ساخت یافته اکثر داده‌ها تشدید شده است. از طرف دیگر، مقدار داده‌های متنی در دسترس ما به طور مستمر در حال افزایش است[۱]. به منظور بازیابی و استفاده از دانش ارزشمند موجود در این مجموعه مدارک، باید روش¬های خودکار، کارا و اثر بخشی برای تحلیل حجم زیاد متون غیر ساخت یافته طراحی شود.

در راستای اهداف مربوطه، متن کاوی برای اولین بار توسط فلدمن و همکاران[۲] مطرح شد. متن کاوی کشف اطلاعات جدید و از پیش ناشناخته، به وسیله استخراج خودکار اطلاعات از منابع مختلف نوشته شده، است. متن کاوی یک زمینه جوان میان رشته‌ای است[۳]، که به عنوان تجزیه و تحلیل هوشمند متن، داده کاوی متن یا کشف دانش در متن (KDT) نیز شناخته می‌شود[۱]. از آن جا که اکثر اطلاعات (بیش از ۸۰٪) به صورت متن ذخیره شده اند، و حاوی اطلاعات ارزشمند و نهفته‌ای می‌باشند، اعتقاد بر این است که متن کاوی ارزش بالقوه تجاری بالایی دارد.[۱].

زمینه‌های مرتبط با متن کاوی

داده کاوی، بازیابی اطلاعات، پردازش زبان طبیعی و استخراج اطلاعات از زمینه‌های مرتبط با متن کاوی هستند.

داده کاوی: روشی بسیار کارا برای کشف اطلاعات از داده‌های ساخت یافته است. متن کاوی مشابه داده کاوی است، اما ابزارهای داده کاوی طراحی شده‌اند تا داده‌های ساخت یافته از پایگاه داده را به کار ببرند. می‌توان گفت، متن کاوی یک راه حل بهتر برای شرکت‌ها است.[۱]

بازیابی اطلاعات: معمولاً در بازیابی اطلاعات با توجه به نیاز مطرح شده از سوی کاربر، مرتبط‌ترین متون و مستندات و یا در واقع کیسه‌ی کلمه از میان دیگر مستندات یک مجموعه بیرون کشیده می‌شود. بازیابی اطلاعات یافتن دانش نیست بلکه تنها آن مستنداتی را که مرتبط‌تر به نیاز اطلاعاتی جستجوگر تشخیص داده به او تحویل می‌دهد. این روش در واقع هیچ دانش و حتی هیچ اطلاعاتی را به ارمغان‌نمی آورد.

پردازش زبان طبیعی(NLP): هدف کلی آن رسیدن به یک درک بهتر از زبان طبیعی توسط کامپیوترهاست. تکنیک‌های مستحکم و ساده‌ای را برای پردازش سریع متن به کار می‌برد. همچنین از تکنیک‌های آنالیز زبان شناسی نیز برای پردازش متن استفاده می-کند. نقش NLP در متن کاوی فرآهم کردن یک سیستم در مرحله استخراج اطلاعات با داده‌های زبانی است.

استخراج اطلاعات: هدف استخراج اطلاعات خاص از سندهای متنی است و می‌تواند به عنوان یک فاز پیش پردازش در متن کاوی به کار رود. استخراج اطلاعات عبارتند از نگاشت متن‌های زبان طبیعی به یک نمایش ساخت یافته و از پیش تعریف شده یا قالب-هایی که وقتی پر می‌شوند، منتخبی از اطلاعات کلیدی از متن اصلی را نشان می‌دهند. این سیستم‌های استخراج اطلاعات به شدت بر داده‌های تولید شده توسط سیستم‌های NLP تکیه دارند[۴].

فرآیند متن کاوی

در [۵] دو فاز اصلی برای فرآیند متن کاوی تعریف شده است: پیش پردازش مستندات و استخراج دانش.

اولین فاز پیش پردازش مستندات است. خروجی این فاز می‌تواند دو شکل مختلف داشته باشد:۱) مبتنی بر سند ۲) مبتنی بر مفهوم. در فرمت اول، نحوه‌ی نمایش بهتر مستندات مهم است، برای مثال تبدیل اسناد به یک فرمت میانی و نیمه ساخت یافته، یا به کار بردن یک ایندکس بر روی آن‌ها یا هر نوع نمایش دیگری که کار کردن با اسناد را کاراتر می‌کند. هر موجودیت در این نمایش در نهایت باز هم یک سند خواهد بود. در فرمت دوم، نمایش اسناد بهبود بخشیده می‌شود، مفاهیم و معانی موجود در سند و نیز ارتباط میان آن‌ها و هر نوع اطلاعات مفهومی دیگری که قابل استخراج است، از متن استخراج می‌شود. در این حالت نه با خود موجودیت بلکه با مفاهیمی که از این مستندات استخراج شده‌اند، مواجه هستیم.

قدم بعدی استخراج دانش از این فرم‌های میانی است که بر اساس نحوه‌ی نمایش هر سند متفاوت می‌باشد. نمایش مبتنی بر سند، برای گروه بندی، طبقه بندی و تجسم سازی استفاده می‌شود، در حالی که نمایش مبتنی بر مفهوم برای یافتن روابط میان مفاهیم، ساختن اتوماتیک آنتولوژی و... به کار می‌رود.

روش‌های متن کاوی

در این بخش دو روش طبقه بندی، خوشه بندی و الگوریتم‌های پرکاربرد آن‌ها را بررسی می‌کنیم.

طبقه بندی

هدف از طبقه‌بندی متون نسبت دادن کلاس‌های از پیش تعریف شده به اسناد متنی است.

در طبقه‌بندی یک مجموعه‌ی آموزشی از اسناد، با کلاس‌های معین وجود دارد. با استفاده از این مجموعه، مدل طبقه‌بندی معین شده و کلاس سند جدید مشخص می‌گردد. برای اندازه‌گیری کارایی مدل طبقه‌بندی، یک مجموعه تست، مستقل از مجموعه آموزشی در نظر گرفته می‌شود. برچسب‌های تخمین زده شده با برچسب واقعی اسناد مقایسه می‌شود. نسبت اسنادی که به درستی طبقه‌بندی شده‌اند به تعداد کل اسناد، دقت (accuracy) نامیده می‌شود.

مجموعه‌ای از اسناد برچسب دار از منبع $D=[d_1,d_2,\ldots, d_n]$ را در نظر بگیرید که به مجموعه‌ای از کلاس‌های $C= [c_1,c_2,\ldots,c_p]$ متعلق باشد. کار طبقه بندی متن این است که طبقه بندی کننده را با استفاده از این اسناد آموزش داده و دسته‌ها را به اسناد جدید اختصاص دهد. در مرحله آموزش، n تا از اسناد در p پوشه جدا قرار می‌گیرند که هر پوشه به یک کلاس انتساب داده شده است. در مرحله بعد مجموعه داده‌های آموزشی از طریق فرآیند انتخاب ویژگی ساخته می‌شوند.[۱]

درخت‌های تصمیم

برای ساختن این درخت‌ها از یک استراتژی تصمیم و غلبه استفاده می‌شود. برای یک مجموعه آموزش M با اسناد برچسب‌گذاری شده، باید کلمه t_i جهت تقسیم‌بندی مجموعه انتخاب شود. M بر اساس کلمه t_i به دو زیر مجموعه تقسیم می‌شود. زیر مجموعه M_i^+ شامل اسنادی است که حاوی t_i هستند و M_i^- شامل اسنادی است که کلمه t_i در آن‌ها نیست. همین فرآیند را برای M_i^+ و M_i^- تکرار می‌کنیم. این کار را تا زمانی که همه اسناد موجود در یک زیر مجموعه متعلق به یک کلاس باشند ادامه می‌دهیم، در این صورت برچسب آن گره برابر با کلاس متناظر با اسناد می‌شود و آن گره تبدیل به برگ می‌گردد. یا فرآیند را تا زمانی ادامه می‌دهیم که دیگر کلمه‌ای برای تقسیم‌بندی زیر مجموعه‌ها وجود نداشته باشد. سپس بر چسبی را انتخاب می‌کنیم که اکثریت اسناد آن بخش، آن برچسب را دارند.[۷]

درخت‌های تصمیم در تعداد متغیرها و اندازه مجموعه آموزشی، مقیاس پذیرند. البته دارای اشکالاتی مانند وابستگی تصمیم نهایی به تعداد نسبتاً کمی از کلمه‌ها می‌باشند. بهبود قطعی ممکن است با تقویت درخت‌های تصمیم گیری به دست آید.[۷]

درخت تصمیم متوالی بر پایه طبقه بندی

در این مدل هر یک از گره‌های داخلی به عنوان تصمیم گیرنده و هر یک از برگ‌ها به عنوان یک برچسب کلاس می‌باشند. این مدل از دو مرحله تشکیل شده است: ۱) القای درخت- که از مجموعه آموزشی داده شده القا می شود.۲) هرس درخت- درخت القا شده را با از بین بردن هر وابستگی آماری روی مجموعه داده آموزشی خاص، کوتاه تر و قوی تر می‌کند.[۴]

روش Hunt:

ساخت درخت به صورت بازگشتی و با استفاده از راهبرد حریصانه تقسیم و حل اول عمق می‌باشد. فرض کنید یک مجموعه آموزشی T از اسناد متنی تفکیک شده با کلاس‌های $C= C_1, C_2,\ldots, C_k$ داده شده است. روش به شرح زیر می‌باشد:

T شامل مواردی است که همگی متعلق به یک کلاس C_j هستند. درخت تصمیم برای T، برگی است که کلاس C_j را مشخص می‌کند.

۱) T شامل مواردی است که متعلق به یک کلاس یا بیشتر هستند. آزمون بر پایه‌ی یک ویژگی منفرد T انتخاب می‌شود که دارای یک یا بیشتر نتایج متقابل منحصر به فرد $O_1, O_2, \ldots, O_n$ است. T به زیر مجموعه‌های $T_1, T_2, \ldots, T_n$ تقسیم می‌شود در این جا T_i شامل تمام موارد در T می‌باشد که دارای خروجی O_i از مجموعه انتخابی هستند.

۲) T شامل موردی نمی‌باشد. درخت تصمیم برای T یک برگ می‌باشد، اما کلاس وابسته به برگ باید با اطلاعاتی بیش از T معین شود.[۸]

الگوریتم C4.5:

مراحل کلی الگوریتم C4.5 برای ساخت درخت تصمیم:[۹]

۱) انتخاب ویژگی برای گره ریشه

۲) ایجاد شاخه برای هر مقدار از آن ویژگی

۳) تقسیم موارد با توجه به شاخه‌ها

۴) تکرار روند برای هر شاخه تا زمانی که تمام موارد شاخه، کلاس یکسان داشته باشند.

انتخاب هر ویژگی به عنوان ریشه بر پایه بالاترین حصول از هر صفت است.

الگوریتم SPRINT:

SPRINT (القای مقیاس پذیر قابل موازی سازی از الگوریتم درخت تصمیم گیری) یک درخت تصمیم طبقه بندی کننده سریع و مقیاس پذیر است. این الگوریتم مجموعه داده آموزشی را به صورت بازگشتی با استفاده از تکنیک حریصانه اول پهنا تقسیم می‌کند تا وقتی که هر قسمت متعلق به گره برگ یا کلاس یکسان باشد. این روش، از مرتب سازی داده‌ها استفاده می‌کند و محدودیتی برای حجم داده ورودی نداشته و می‌تواند بر روی الگوهای سریال یا موازی برای جایگزینی داده‌های خوب و توازن بار اجرا شود. دو ساختار داده‌ای را به کار می-گیرد: لیست داده‌ها و پیشینه نما، که مقیم در حافظه نیستند و این مسئله SPRINT را برای مجموعه داده‌های بزرگ مناسب می‌سازد. بنابراین همه محدودیت‌های حافظه بر داده‌ها را حذف می‌کند. این الگوریتم صفت‌های پیوسته و طبقه‌ای را به کار می‌برد.[۸]

مزایای درخت‌های تصمیم گیری ارزان بودن ساخت، تفسیر آسان، و ادغام آسان با پایگاه داده تجاری است. همچنین دقت بهتری را نتیجه می‌دهند. از معایب آن عدم توانایی رسیدگی به مجموعه داده‌های بزرگ است چون از محدودیت حافظه رنج می‌برند و سرعت محاسباتی پایینی دارند.[۴]

فرمول بندی موازی از درخت تصمیم بر پایه طبقه بندی

هدف این روش مقیاس پذیری در زمان اجرا و حافظه مورد نیاز است. فرمول بندی موازی بر محدودیت حافظه که برای الگوریتم‌های ترتیبی مشکل ساز است غلبه می‌کند، بدین صورت رسیدگی به مجموعه داده‌های بزرگ تر بدون نیاز به دیسک I/O افزونه را ممکن می‌سازد. همچنین فرمول بندی موازی سرعت بالاتری نسبت به الگوریتم سریال ارائه می‌کند. انواع فرمول بندی‌های موازی برای ساخت درخت تصمیم طبقه بندی:[۴]

- رویکرد ساخت درخت همزمان
- رویکرد ساخت درخت قسمت بندی شده
- فرموله بندی موازی ترکیبی

رویکرد ساخت درخت همزمان

همان طور که در [۱۰] می‌بینید، در این رویکرد، تمام پردازنده‌ها یک درخت تصمیم همزمان را با ارسال و دریافت اطلاعات توزیعی کلاس از داده‌های محلی می‌سازند. گام‌های اصلی این رویکرد در زیر نشان داده شده است:

۱) یک گره را برای گسترش با توجه به استراتژی توسعه (اول عمق یا اول پهنا) انتخاب کنید، و آن را گره جاری بنامید. در ابتدا گره ریشه به عنوان گره جاری انتخاب می‌شود.

۲) برای هر صفت داده مجموعه اطلاعات توزیعی کلاس از داده‌های محلی در گره جاری جمع آوری می‌شود.

۳) اطلاعات توزیعی کلاس محلی با استفاده از کاهش سراسری در میان پردازنده‌ها مبادله می‌شوند.

۴) به طور همزمان دستاوردهای آنتروپی هر صفت در هر یک از پردازنده‌ها محاسبه و بهترین صفت برای گسترش گره فرزند انتخاب می‌شود.

۵) با توجه به فاکتور انشعاب، برای تعداد بخش‌هایی از مقادیر صفت گره‌های فرزند را ایجاد کرده و بر طبق آن موارد آموزشی را تقسیم می‌کنیم.

۶) مراحل فوق (۱-۵) را تا زمانی که دیگر گره‌ای برای گسترش وجود نداشته باشد تکرار می‌کنیم.

مزیت این رویکرد این است که نیازمند جا به جایی آیتم‌های داده‌های آموزشی نیست. یکی از معایب این الگوریتم، هزینه بالای ارتباطات به دلیل همگامی پردازنده‌ها و مبادله اطلاعات می‌باشد. عیب دیگر آن عدم تعادل بار به دلیل وجود تعداد قابل ملاحظه‌ای گره مشابه میان پردازنده‌ها است.

رویکرد ساخت درخت قسمت بندی شده

همان طور که در [۱۰] می بینید، در این رویکرد پردازنده‌های متفاوت روی قسمت‌های متفاوت از درخت طبقه بندی کار می‌کنند. اگر بیش از یک پردازنده برای گسترش یک گره با هم همکاری کنند آن هنگام این پردازنده‌ها برای گسترش جانشینان این گره تقسیم بندی می‌شوند. حالتی را در نظر بگیرید که در یک گروه از پردازنده-های P_n برای گسترش گره n با هم همکاری می‌کنند. الگوریتم شامل مراحل زیر است:

۱) پردازنده‌ها در P_n برای گسترش گره n با استفاده از روش بیان شده در قسمت قبل با هم همکاری می‌کنند.

۲) هنگامی که گره n به گره‌های جانشین $n_1, n_2, \ldots, n_k$ گسترش یافت سپس پردازنده گروه P_n نیز تقسیم می‌شود و جانشین گره‌ها مطابق روند زیر اختصاص داده می‌شوند:

مورد ۱: اگر تعداد جانشین گره‌ها بیش از $|P_n|$ است،

۱-جانشین گره‌ها را به $|P_n|$ گروه تقسیم بندی کنید به گونه‌ای که تعداد کل موارد آموزشی مربوط به هر گروه از گره‌ها تقریباً برابر باشد. هر پردازنده را به یک گروه از گره‌ها اختصاص دهید.

۲-داده‌های آموزشی را بر هم بزنید به نحوی که هر پردازنده اقلام داده‌هایی از گره‌ها را دارا باشد که مسئولیت آن‌ها (گره‌ها) را برعهده دارد.

۳-حال زیر درختان ریشه‌دار در یک گروه گره، مستقلاً در هر پردازنده و مانند الگوریتم سریال گسترش داده می‌شوند.

مورد ۲: در غیر این صورت (اگر کمتر از $|P_n|$ باشد)،

۴-زیر مجموعه‌ای از پردازنده‌ها را به هر گره اختصاص دهید (تعداد متناسب با تعداد موارد آموزشی مربوط به گره باشد).

۵-موارد آموزشی را برهم بزنید به نحوی که هر زیر مجموعه از پردازنده‌ها موارد آموزشی متعلق به گره‌هایی را داشته باشد که مسئولیت آن (گره)ها را بر عهده دارد.

۶-زیر مجموعه‌های از پردازنده را به گره‌های متفاوت اختصاص می-دهیم تا زیر درخت‌ها را به صورت مستقل توسعه دهند. زیر مجموعه‌های پردازنده که شامل تنها یک پردازنده است از الگوریتم ترتیبی برای گسترش بخشی از درخت طبقه بندی استفاده می-کند. زیر مجموعه‌هایی از پردازنده‌ها که شامل بیش از یک پردازنده هستند مراحل فوق را به صورت بازگشتی ادامه می‌دهند.

- مزایا: مزیت این رویکرد این است که یک پردازنده صرفاً مسئول یک گره می‌شود، و می‌تواند زیر درخت را مستقلاً بدون هر گونه سربار ارتباطات توسعه دهد.
- معایب: وقتی یک پردازنده مسئول یک زیر درخت کامل باشد به جابه جایی داده بعد از گسترش هر گره نیاز دارد. هزینه ارتباطات در گسترش قسمت فوقانی درخت گران است. عیب دوم توازن بار ضعیف ذاتی در الگوریتم است. واگذاری گره به پردازنده بر پایه تعداد موارد آموزشی در گره‌های جانشین انجام می‌شود. با این حال تعداد موارد آموزشی مرتبط با گره لزوماً با مقدار کار مورد نیاز برای پردازش زیر درخت ریشه دار در گره متناظر نمی‌باشد.

فرمولاسیون موازی ترکیبی

با توجه به [۱۰] فرمولاسیون موازی ترکیبی عناصری از هر دو طرح را داراست. طرح ترکیبی تا وقتی که هزینه ارتباطاتی خیلی بالا نباشد با رویکرد اول دنبال می‌شود. هنگام بالا رفتن هزینه‌ها پردازنده‌ها به دو قسمت تقسیم می‌شوند. طبق فرض تعداد پردازنده‌ها توانی از ۲ می‌باشد و این پردازنده‌ها در یک پیکربندی ابر مکعب (hypercube) به یکدیگر متصل‌اند. اگر P توانی از دو نباشد الگوریتم می‌تواند به اقتضا تغییر کند. همچنین این الگوریتم می‌تواند به هر معماری موازی با تعبیه ساده یک ابر مکعب مجازی در معماری نگاشت شود.

طبقه بندی کننده ساده بیزی

یک روش طبقه‌بندی احتمالی است. کلاس یک سند متناسب با کلماتی است که در یک سند ظاهر شده‌اند. در این روش [۳] برای تخمین کلاس سند از فرمول زیر استفاده می‌شود:

$$P(L_c|t_1, \dots t_{n_i}) = \frac{P(t_1,\dots t_{n_i}|L_c)p(L_c)}{P(t_1,\dots,t_{n_i})} \qquad (۱)$$

در این فرمول L_c نشان دهنده کلاس c و t_iها کلمه‌های موجود در یک سند هستند. این نکته قابل ذکر است که هر سند دقیقا به یک کلاس تعلق دارد. از آنجا که $P(t_1, \dots, t_{n_i})$ برای همه کلاس‌ها مساوی در نظر گرفته می‌شود، می‌توان این احتمال را از فرمول بالا حذف نمود و همچنین برای سادگی می‌توان از فرض استقلال Na¨ıve استفاده کرد. طبق این فرض احتمال رخداد کلمات در یک سند مستقل از یکدیگر است. یعنی داریم:

$$P(t_1, \dots t_{n_i}|L_c) = \prod_{j=1}^{n_i} p(t_j|L_c) \qquad (۳)$$

طبقه‌بندی کننده Na¨ıve Bayes یک گام یادگیری دارد که در آن احتمالات $p(t_j|L_c)$ (تعداد اسنادی که در مجموعه آموزش شامل کلمه t_j هستند و برچسب کلاس آن‌ها L_c است تقسیم بر کل اسناد مجموعه آموزش) تخمین زده می‌شود. در گام طبقه‌بندی، احتمالات تخمین زده شده برای طبقه‌بندی کردن یک نمونه جدید مطابق با قانون بیز استفاده می‌شوند. اگرچه این روش به علت فرض استقلال تا حدی ممکن است غیر واقعی باشد اما در عمل نتایج خوبی از آن حاصل می‌شود.

برخی نویسندگان، به دلیل طاقت فرسا بودن برچسب زنی دستی، از اسناد بدون برچسب برای آموزش استفاده می‌کنند. فرض کنید که در یک مجموعه آموزشی کوچک کلمه t_i با کلاس L_c بسیار مرتبط باشد. اگر از اسناد بدون بر چسب به توان معین کرد که کلمه t_j با t_i مرتبط است پس t_j برای کلاس L_c یک پیش بینی کننده خوب است. به این ترتیب ممکن است کارآیی طبقه بندی را بهبود دهند. در [۱۱] از ترکیب EM و طبقه بندی کننده بیز استفاده شده است و به این طریق خطای طبقه بندی تا ۳۰٪ کاهش یافته است.

طبقه بندی کننده K نزدیکترین همسایه

راه دیگر این است که اسنادی از مجموعه آموزش انتخاب شوند که مشابه سند جاری هستند. کلاس سند جاری، کلاسی است که اکثریت اسناد مشابه، دارند. در این روش، k تا سند از مجموعه آموزش که بیشترین شباهت (بر اساس معیار شباهت تعریف شده) را به سند جاری دارند به عنوان همسایگان آن سند انتخاب می‌شوند.[۳] این طبقه بندی به سه مورد اطلاعاتی نیاز دارد: ۱) مقدار k، یعنی بر اساس شباهت داده‌ی آزمایش با چند تا از نزدیکترین همسایه‌ها برچسب آن را مشخص کنیم، ۲) مجموعه‌ای از داده‌های برچسب‌دار، که به عنوان داده‌های آموزشی مورد استفاده قرار گیرند و ۳) یک معیار شباهت.

یک روش ساده برای معیار شباهت شماردن تعداد کلمات مشترک در دو سند است. این روش باید برای اسناد با طول مختلف نرمال سازی شود. یک روش استاندارد برای اندازه‌گیری شباهت، شباهت کسینوسی است. برای مشخص شدن کلاس سند d_i، شباهت $S(d_i,d_j)$ برای همه اسناد d_j در مجموعه آموزشی محاسبه می‌شود. سپس k تا از شبیه‌ترین اسناد مجموعه آموزش به عنوان همسایه‌های سند جاری انتخاب می‌گردند. کلاس سند d_i برابر با کلاسی است که اکثر سندهای همسایه آن دارای آن کلاس هستند. در این روش مقدار بهینه k را می‌توان از مجموعه آموزش دیگری به وسیله اعتبارسنجی ضربدری (cross-validation) تخمین زد. طبق مطالعات انجام شده روش k همسایه نزدیک در عمل کارایی خوبی دارد. مشکل این است که در طی طبقه‌بندی محاسبات زیادی لازم است.[۳]

شبکه‌های عصبی

در مسائل مربوط به طبقه بندی، شبکه عصبی با داشتن ورودی‌ها و خروجی‌های مشخص باید تشخیص دهد که هر ورودی با کدام طبقه از خروجی‌های تعریف شده بیشترین تطابق را دارد. در شبکه پرسپترون چند لایه (MLP) از روش آموزش با نظارت استفاده می‌شود. هدف از آموزش شبکه به حداقل رساندن خطای تولید شده می‌باشد که بر اساس تنظیم وزن‌های شبکه انجام می‌شود. معمولا از الگوریتم آموزش پس انتشار استفاده می‌شود. در این الگوریتم پس از محاسبه مقدار خطا در لایه خروجی مقادیر وزن‌ها در لایه پنهان در جهت کاهش خطا تنظیم می‌شوند.

مزایای استفاده از شبکه عصبی:

- روش های خود تطبیقی برای مبنای داده هستند.
- می‌توانند هر تابعی را با دقت دلخواه تخمین بزنند.
- مدل‌های غیر خطی هستند.

- با داده‌های ناقص یا گم شده به خوبی کار می‌کنند.

معایب شبکه عصبی:

- برآورد یا پیش بینی خطا انجام نمی‌شود.
- چگونگی برآورد شدن روابط میان لایه های پنهان را نمی‌توان معین کرد.

ماشین بردار پشتیبانی(SVM)

SVM یک الگوریتم طبقه‌بندی تحت نظارت موفق است. با توجه به [۱۲] معمولاً سند d به وسیله بردار $(t_{d_1}, ..., t_{d_n})$ از تعداد کلماتش نمایش داده می‌شود. SVM می‌تواند فقط دو کلاس را جدا کند: یک کلاس مثبت L_1 (y = +1) و کلاس منفی L2 (y = -1). در فضای بردارهای ورودی، ابر صفحه با تنظیم کردن y = 0 در زیر معادله خطی تعریف می‌شود.

$$y = f(\vec{t_d}) = b_0 + \sum_{j=1}^{N} b_j t_{dj} \qquad (۳)$$

SVM یک ابر صفحه که بین نمونه‌های مثبت و منفی مجموعه آموزش قرار می‌گیرد را مشخص می‌کند. پارامترهای b_j به گونه‌ای تنظیم می‌شوند که فاصله ε (که حاشیه نامیه می‌شود) بین ابر صفحه و نزدیکترین نمونه مثبت و منفی ماکزیمم شود. اسنادی که دارای فاصله ε از ابر صفحه‌اند را بردار پشتیبانی می‌نامند که محل واقعی ابر صفحه را مشخص می‌کنند. یک سند جدید با بردار کلمه $\vec{t_d}$ اگر مقدار $f(\vec{t_d}) > 0$ باشد در L_1 و در غیر اینصورت در L_2 طبقه‌بندی می‌شود. SVM می‌تواند با پیشگوهای غیرخطی نیز استفاده شود.

مهمترین ویژگی SVM این است که یادگیری تقریباً مستقل از ابعاد فضای ویژگی است. این روش نیاز به انتخاب صفت ندارد زیرا به طور ذاتی نقاطی از داده (بردار پشتیبانی) برای یک طبقه‌بندی خوب انتخاب می‌شوند. بنابراین برای طبقه‌بندی متون مناسب است. در حالت داده متنی، انتخاب تابع هسته اثر کمی بر روی دقت طبقه-بندی دارد.[۱۲]

ژنتیک

یک روش بهینه سازی اکتشافی است که از قوانین تکامل بیولوژیک طبیعی تقلید می‌کند. الگوریتم ژنتیک قوانین را بر روی جواب‌های مسأله (کروموزوم‌ها)، برای رسیدن به جواب‌های بهتر، اعمال می‌کند. در هر نسل به کمک فرآیند انتخابی متناسب با ارزش جواب‌ها و تولید مثل جواب-های انتخاب شده و به کمک عملگرهایی که از ژنتیک طبیعی تقلید شده‌اند، تقریب‌های بهتری از جواب نهایی بدست می‌آید. این فرایند باعث می‌شود که نسل‌های جدید با شرایط مساله سازگارتر باشند.

به منظور حل هر مسئله، ابتدا باید یک تابع برازندگی برای آن ابداع شود. این تابع برای هر کروموزوم، عددی را بر می‌گرداند که نشان دهنده شایستگی آن کروموزوم است. در طی مرحله تولید نسل ازعملگرهای ژنتیکی استفاده می‌شود که با تأثیر این آن‌ها بر روی یک جمعیت، نسل بعدی تولید می‌شود. عملگرهای انتخاب, آمیزش و جهش معمولاً بیشترین کاربرد را در الگوریتم‌های ژنتیکی دارند. تعدادی شروط خاتمه برای الگوریتم ژنتیک وجود دارد از جمله: تعداد مشخصی نسل، عدم بهبود در بهترین شایستگی جمعیت در طی چند نسل متوالی و عدم تغییر بهترین شایستگی جمعیت تا یک زمان خاص.

ارزیابی الگوریتم‌های طبقه بندی

در طول سال‌های گذشته طبقه بندی کننده‌های متن با تعدادی از معیارهای مجموعه اسناد، ارزیابی شده اند. اما به نظر می‌رسد کارایی الگوریتم‌های طبقه بندی تا حد زیادی تحت تاثیر کیفیت منابع داده قرار دارند. ویژگی‌های نامربوط و افزونه از داده‌ها فقط هزینه فرآیند را افزایش نمی‌دهند، بلکه باعث کاهش کیفیت نتایج در برخی موارد نیز می‌شوند. هر الگوریتم دارای مزایا و معایب خود می‌باشند.[۱۳]

در [۱۲] عملکرد برخی طبقه بندی کننده‌ها برای ۲۰ مجموعه گروه خبری از خبرگزاری رویترز مقایسه شده است. جدول ۱ نتایج بدست آمده را نشان می‌دهد. همان طور که مشاهده می‌کنید در اکثر مواقع طبقه بندی کننده‌های SVM و k نزدیکترین همسایه کارآیی بالایی را ارائه می‌کنند و پس از آن‌ها شبکه عصبی، درخت‌های تصمیم و روش ساده بیزی قرار گرفته‌اند.

جدول ۱: عملکرد طبقه بندی کننده‌های مختلف برای مجموعه رویترز

متد	F-value
Naïve Bayes	۰,۷۹۵
Neural network	۰,۸۳۸
درخت تصمیم گیری C4.5	۰,۷۹۴
K تا نزدیکترین همسایه	۰,۸۵۶
SVM	۰,۸۷۰

خوشه بندی

خوشه بندی تکنیکی است برای گروه‌بندی اسناد [۱]،که امروزه نقش حیاتی در روش‌های بازیابی اطلاعات دارد. هدف آن قرار دادن اسناد مشابه در یک خوشه است به طوری که با اسنادی که در خوشه‌های دیگر قرار دارند، متفاوت باشند[۴]. برخلاف طبقه‌بندی در خوشه-بندی گروه‌ها از قبل مشخص نیست[۱] و همچنین معلوم نیست که برحسب کدام ویژگی گروه‌بندی صورت می‌گیرد. الگوریتم‌های خوشه‌بندی خوشه‌ها را براساس ویژگی داده‌ها و اندازه‌گیری شباهت-ها و یا عدم شباهت‌ها محاسبه می‌کنند[۳]. دو روش برای ارزیابی نتایج خوشه‌بندی وجود دارد.۱) اقدامات آماری، ۲) دسته‌بندی‌های استاندارد.[۳]

دسته‌های مختلف الگوریتم‌های خوشه بندی

الگوریتم‌های خوشه‌بندی غالباً به دو گروه روش‌های سلسله مراتبی و افراز بندی تقسیم می‌شوند.

روش‌های سلسله مراتبی

در این روش به خوشه‌های نهایی بر اساس میزان عمومیت آن‌ها ساختاری سلسله مراتبی که معمولا به صورت درختی است، نسبت داده می‌شود. به این درخت سلسله مراتبی دندوگرام (dandogram) می‌گویند. روش‌های خوشه‌بندی سلسله مراتبی معمولا به دو دسته تقسیم می‌شوند:[۱۴]

۱) بالا به پایین (top-down) یا تقسیم کننده (Divisive): ابتدا تمام داده‌ها در یک خوشه قرار دارند و در طی فرایند تکراری در هر

مرحله داده‌هایی که شباهت کمتری دارند به خوشه‌های مجزا تقسیم می‌شوند. این روال تا زمانی که خوشه‌ها دارای یک عضو شوند ادامه پیدا می‌کند.

۲) پایین به بالا (Bottom-up) یا متراکم‌شونده (Agglomerative): هر داده به عنوان خوشه جدا در نظر گرفته می‌شود و در طی فرایند تکراری در هر مرحله خوشه‌هایی که شباهت بیشتری به هم دارند با یکدیگر ترکیب می‌شوند. در نهایت تعداد مشخصی خوشه حاصل می‌شود. در عمل این روش نتایج چندان جالبی ندارد[۳].

روش‌های افرازبندی:

الگوریتم افراز بندی داده‌ها را به K قسمت تقسیم می‌کند، که در آن هر یک از بخش‌ها نشان دهنده یک خوشه است[۱۵]. این روش نیازمند حجم زیادی از محاسبات زمان بر فاصله یا معیارهای شباهت بین مجموعه داده‌ها و مراکز خوشه‌ها می‌باشد.

همچنین به دلیل عدم امکان بررسی همه زیر مجموعه‌ها، از برخی روش‌های تکراری حریصانه استفاده می‌شود تا کلمه‌ها را مکرراً بین k خوشه جا به جا کند. برخلاف روش‌های سلسله مراتبی که با ایجاد خوشه دوباره بازبینی نمی‌شود، این الگوریتم‌ها به تدریج خوشه‌ها را بهبود می‌بخشند[۳].

انواع الگوریتم‌های افرازبندی

در این قسمت به دو الگوریتم k-means و bi-seaction-k-means می پردازیم.

k-means

هدف این روش به حداقل رساندن میانگین مجذور فاصله بین اشیا یک خوشه و مرکز خوشه می‌باشد. در حالت ایده‌آل خوشه‌ها نباید با یکدیگر هم‌پوشانی داشته باشند. k-means با انتخاب تصادفی K خوشه مرکزی اولیه از اشیا شروع می‌شود. سپس آن مراکز خوشه‌ها را در فضای داده‌ها حرکت می‌دهد تا مجموع مجذور فاصله هر بردار از مرکز ثقل همه بردارها به حداقل برسد.[۱۶]

bi-section-k-means

این الگوریتم با یک خوشه که تمام اسناد را در بر دارد شروع می‌شود، گام‌های این الگوریتم به صورت زیر است:[۱۷]

۱. انتخاب یک خوشه برای تقسیم (مانند خوشه با واریانس بالا[۳])

۲. پیدا کردن ۲زیرخوشه توسط الگوریتم پایه (گام نیم‌ساز)

۳. تکرار مرحله ۲ تا مدت زمان معین شده، برای تولید خوشه‌هایی با بیشترین شباهت

۴. تکرار مراحل بالا برای رسیدن به تعداد مشخصی خوشه

نتیجه‌گیری

متن کاوی یک زمینه‌ی جوان و درحال رشد است که به ما کمک می‌کند از دانش موجود در متون غیر ساخت یافته بهره ببریم. طیف وسیعی از کاربردها برای متن کاوی قابل تصور است. در این مقاله مروری مختصر در زمینه گسترده متن کاوی و مهمترین روش‌های متن کاوی موجود به همراه مزایا و معایب آن‌ها، عرضه شده است. اگرچه بیان همه روش‌ها و کاربردها در این زمینه ممکن نیست اما این مقاله می‌تواند دید کلی از متن کاوی در ذهن خواننده ایجاد کرده و در صورت علاقه برای مطالعه بیشتر، فرد را به منابع مناسب هدایت کند.

مراجع

[1] V. Gupta and G. Lehal, 2009, "A Survey of Text Mining Techniques and Applications", *Journal of Emerging Technologies In Web Intelligence*,Vol. 1.

[2] R. Feldman and I. Dagan, 1995, "KDT-Knowledge Discovery in Texts," *Proc. of the First Int. Conf. on* Knowledge *Discovery KDD*, pp. 112-117.

[3] Hotho et al, 2005, "A Brief Survey of Text Mining Export", *LDV Forum*, Vol.20, pp.19-62.

[4] S. Ghosh, S. Roy and S. Bandyopadhyay, 2012, "A Tutorial Review on Text Mining Algorithms" *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 1.

[5] E. Liddy, 2000, "Text Mining,"*Bulletin of the American Society for Information Science*,pp 13-14.

[6] H. Zhuge et al., 2004, "An Automatic Semantic Relationships Discovery Approach," *The 13th International World Wide Web Conference (WWW2004)*, USA.

[7] H.F. Moed, M. May and G. Paab, 2004, "Handbook of Quantitative Science and Technology Research*", Kluwer Academic Publishers*, pp. 187-213.

[8] M. Anyanwu and S. Shiva, "Comparative Analysis of Serial Decision Tree Classification Algorithms" *International Journal of Computer Science and Security (IJCSS)*, Vol. 3.

[9] Kusrini and S. Hartati, 2007, "Implementation of C4.5 Algorithm to Evaluate The Cancellation Possibility of New Student Applicants at Stimik Amikom Yogyakarta", *Proc. Int. Conf. on Electrical Engineering and Informatics Institut Teknologi Bandung*, Indonesia.

[10]Srivastava, E. Han, V. Kumar and V. Singh, "Parallel Formulations of Decision-Tree Classification Algorithms*",* pp. 1-24.

[11]K. Nigam A. McCallum, S. Thrun, and T. Mitchell, 2000, "Text Classification from Labeled and Unlabeled Documents Using em". Machine Learning, pp. 103–134.

[12]F. Sebastiani, 2002, "Machine Learning in Automated Text Categorization" *ACM Computing Surveys*, Vol. 34, pp. 1–47.

[13]V. Korde and N. Mahender, 2012, "Text Classification And Classifiers: A Survey" *International Journal of Artificial Intelligence & Applications (IJAIA)*, Vol.3.

[14]Y. Zhao and G. Karypis, 2005, "Hierarchical Clustering Algorithms for document Datasets" *Data Mining and Knowledge Discovery*, pp. 141-158.

[15]S. Elavarasi, January 2011, "A Survey on Partition Clustering Algorithms", *International Journal of Enterprise Computing and Business Systems*,Vol. 1.

[16]G Manimekalai et al, 2011, "A Survey on Various Approaches in Document Clustering" *Int. J. Comp. Tech. Appl.(IJCTA)*, Vol 2 (5), 1534-1539.

[17]M. Steinbach, G. Karypis and V. Kumar, 2000, "A Comparison of Document Clustering Techniques", *In: KDD Workshop on Text Mining.*