دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

گزارش سمینار داده‌کاوی

استاد راهنما:
دکتر امیرمزلقانی

دانشجویان:
علی نظری – ۹۶۳۱۰۷۵
عرفان عابدی – ۹۶۳۱۰۴۲۷

زمستان ۱۳۹۹

فهرست مطالب

صفحه

مقاله‌ی اول

A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining

۱-۱- چکیده

یکی از معروف‌ترین الگوریتم‌های خوشـــه‌بندی، k-means اســـت که برای داده‌های عددی به نسبت پاسخ مناسبی می‌دهد ولی در داده‌هایی که مقادیرشان categorical بوده، پاسخ مناسبی را به همراه نـدارد. در این مقـالـه الگوریتم k-modes را معرفی می‌کنیم تـا بتوانیم این نوع داده‌هـا را نیز خوشه‌بندی کنیم و از لحاظ مقیاس‌پذیری هم عملکرد مناسبی داشته باشیم.

۱-۲- معرفی

روش‌های خوشـه‌بندی به صـورت کلی بسـیار کمک‌کننده هسـتند و ما را در جهت خوشـه‌بندی داده‌های حجیم در خوشـــه‌هایی همگن بســـیار یاری می‌کنند تا بتوانیم بر روی آن‌ها آنالیز بهتری انجام دهیم. همـان‌طور کـه می‌دانیم، در داده‌کـاوی بـا داده‌هـای حجیمی مواجـه هســـتیم پس مقیـاس‌پذیری الگوریتم‌های مورد استفاده نیز برای ما اهمیت دارد.

در این‌جا الگوریتمی ارائه می‌شـود که روی داده‌های حجیم عملکرد مناسـبی دارد و می‌تواند در خوشـــه‌بندی داده‌های categorical ما را یاری کند. یکی از روش‌هایی که برای خوشـــه‌بندی داده‌های categorical استفاده می‌شده به این صورت بوده که این داده‌ها را به مقادیر عددی تبدیل می‌کردیم و با آن‌ها مثل مقادیر عددی برخورد می‌کردیم ولی این نوع خوشـــه‌بندی نتایج معناداری به مـا نمی‌داده مخصوصا زمانی که این categoryها ترتیب خاصی نداشتند. الگوریتم k-modes که ارائه می‌شود، تلاش می‌کند که علاوه بر حفظ مزایای الگوریتم k-means، محدودیت‌های این روش در خوشـه‌بندی داده‌های categorical را نیز مرتفع سازد.

پیش از این، الگوریتم دیگری تحت عنوان k-prototype ارائه شـــده بوده که در آن هم معیاری برای عدم شباهت ویژگی‌های عددی وجود داشته و هم معیاری برای عدم شباهت داده‌های categorical در آن وجود داشـــته اســـت. در قســـمت عددی، فاصـــله‌ی اقلیدســـی محاســـبه می‌شـــده و برای قســـمت categorical هم به این صورت بوده که تعداد ویژگی‌های categoricalی که بین دو داده با هم متفاوت بودند رو محاسـبه می‌کرده و در نهایت نیز این دو معیار را با هم جمع می‌کرده اسـت. البته برای اینکه در

این جمع، توازن برقرار شود؛ یک وزن به معیار عدم شباهت در قسمت categorical داده می‌شده که مشکل این روش انتخاب همین وزن مناسب بوده است.

الگوریتم k-modes که در اینجا ارائه می‌شود تغییراتی در الگوریتم k-prototype داده‌است به این صورت که تمامی ویژگی‌های داده‌ها را categorical فرض کرده و اگر مقادیر عددی وجود داشت نیز آن‌ها را به صورت categorical در می‌آورد و مهم‌ترین ویژگی آن مقیاس‌پذیر بودن آن است.

البته یک روش دیگر نیز برای خوشه‌بندی داده‌های categorical با کمک k-means وجود داشته به این صورت که تمامی categoryها را به صورت یک ویژگی دودویی در میاورده و بعد به این صورت عمل می‌کرده که آیا این داده دارای این category می‌باشد یا خیر و بعد با این ویژگی‌ها به صورت عددی (۰ یا ۱) برخورد می‌کرده که البته مشکلات زیادی داشته‌است؛ به عنوان مثال وقتی که این categoryها زیاد باشند، ابعاد بسیار زیادی به دیتاست داده‌ها اضافه می‌شده. مورد دیگری که وجود داشته این بوده که محاسبه‌ی میانگین بین تعداد زیادی ۰ و ۱ می‌توانسته بی‌معنی باشد و نتوان آن را به عنوان یک مشخصه در نظر گرفت.

استفاده از خوشه‌بندی سلسله‌مراتبی نیز برای این دیتاست‌ها به علت حجم بالای داده‌ها، پیشنهاد نمی‌شده؛ پس استفاده از k-modes پیشنهاد می‌شود زیرا می‌توان از خود داده‌ها و همان categoryها استفاده کرد و در عین حال مقیاس‌پذیری را نیز حفظ کرد.

۳-۱- الگوریتم k-modes

رویکرد این الگوریتم به k-means شباهت زیادی دارد و البته نمونه‌ی ساده شده‌ی k-prototype نیز محسوب می‌شود. این الگوریتم سه تفاوت اصلی با الگوریتم k-means دارد. اول، معیار عدم شباهت است که در این الگوریتم فرق دارد، دوم اینکه mean را با mode جایگزین کردیم و سوم اینکه نحوه بروزرسانی modeها است که به کمک روشی بر پایه فرکانس categoryها انجام می‌شود.

۱-۳-۱- معیار عدم شباهت

این معیار به این صـــورت اســـت که تمامی ویژگی‌های بین دو نمونه را نگاه می‌کند و آن تعداد ویژگی‌هایی که category یکسـانی ندارند را به عنوان عدم شـباهت، محسـوب می‌کند. البته این روش به صـورتی که تعریف شـد، دارد به تمامی categoryها وزن یکسـانی میدهد؛ پس حالا یک وزنی هم ایجاد شـده و برای هر مقایسـه‌ای این وزن را هم درگیر می‌کند، به این صـورت که مجموعِ معکوسِ فرکانس این دو categoryی را در معیار عدم شـــباهت قبلی‌ای که محاســـبه کردیم، ضـــرب می‌کنیم. اگر دقت کنیم، می‌بینیم که ما با این کار داریم وزن بیشـــتری به داده‌هایی که فرکانس کمتری دارند می‌دهیم و مثلا در جایی که می‌خواهیم کلاه‌برداری را در داده‌های بیمه پیدا کنیم، به کار ما می‌آیند.

۲-۳-۱- پیدا کردن mode دیتاست

Mode دیتاســـت، وکتوری اســـت که ما معیار عدم‌شـــباهت داده‌ها را نســـبت به آن اندازه‌گیری می‌کنیم و تلاش می‌کنیم که این فاصله و معیار عدم شباهت را کمینه کنیم.

این تابع وقتی کمینـه می‌شـــود که تعداد تکرار ویژگی‌های مشـــابـه بـا mode در داده‌ها از ســـایر categoryهای داده‌ها بیشتر باشد.

۳-۳-۱- روند اجرای الگوریتم

روند اجرای الگوریتم به این صـورت اسـت که دیتاسـت را به بخش‌هایی (خوشـه) افراز می‌کنیم و mode هر کدام از این بخش را انتخاب می‌کنیم. تابع هزینه‌ای به وجود می‌آید که حاصل از ترکیب جمع عدم‌شباهت‌های هر داده با mode همان خوشه‌ای است که در آن قرار گرفته است.

اگر بخواهیم موارد بالا را بیشتر توضیح دهیم، می‌توانیم این‌گونه بگوییم که:

- K خوشه را که از افراز دیتاست به وجود آمده را در نظر می‌گیریم.
- برای هر خوشـــه یک وکتور به عنوان mode در نظر می‌گیریم (که در ادامه نحوه انتخاب آن را بیشتر توضیح می‌دهیم)

- به سراغ داده‌ها می‌رویم و اولین داده را به هر کدام از خوشه‌ها با محاسبه‌ی معیار عدم شباهت، نسبت می‌دهیم و mode را برابر همان داده‌ی اول در نظر می‌گیریم و بعد از آن به سراغ داده‌های دیگر می‌رویم و آن‌ها به خوشه‌ی نزدیک‌تر مربوط می‌کنیم و پس از هر نسبت‌دهی، mode را بروزرسانی می‌کنیم. این روند را تا زمانی که تمامی داده‌ها به یک خوشه نسبت داده‌شوند، ادامه می‌دهیم.
- زمانی که تمامی داده‌ها به یک خوشه نسبت‌داده شدند، دوباره تمامی داده‌ها را بررسی می‌کنیم که آیا با بروزرسانی modeها لازم است که در خوشه‌ی دیگری قرار گیرند یا خیر و این روند را تا زمانی که دیگر تغییری در این نسبت بین داده‌ها و خوشه‌ها اتفاق نیفتد ادامه می‌دهیم.

این الگوریتم نیز به مانند k-means ممکن است در مینیمم محلی بیفتد و این بستگی به حالتی دارد که با آن، الگوریتم را آغاز می‌کنیم.

همان‌طور که اشاره شد، روش‌هایی برای انتخاب mode برای شروع الگوریتم وجود دارد که در اینجا به دوتا از آن‌ها اشاره شده است.

اول اینکه k داده‌ی اول را در نظر بگیریم و هر کدام از آن‌ها را به یکی از خوشه‌ها به عنوان mode نسبت دهیم.

دوم اینکه فرکانس هر کدام از categoryهای هر ویژگی داده را محاسبه کنیم و آن‌ها را به صورت نزولی مرتب کنیم و بعد به صورت مساوی این‌ها را به عنوان mode به یک خوشه نسبت دهیم و بعد به سراغ داده‌ها میرویم و نزدیک‌ترین داده به هر خوشه را پیدا می‌کنیم و آن داده را با modeی که قبلا انتخاب کردیم، جایگزین می‌کنیم زیرا این modeها به صورت مصنوعی انتخاب شده بودند و ممکن بود موجب ایجاد خوشه‌های خالی شوند.

این الگوریتم مقیاس‌پذیری بالایی هر در تعداد خوشه‌ی بالا و هم در تعداد داده‌ی زیاد در خوشه‌های زیاد دارد و با افزایش تعداد این پارامترها، مقیاس‌پذیری خود را از دست نمی‌دهد و به صورت خطی زمان اجرای الگوریتم با افزایش این مقادیر، افزایش می‌یابد.

مقاله‌ی دوم

LoOP: Local Outlier Probabilities

۲-۱- مقدمه

شیوه‌های متفاوتی برای مساله‌ی تشخیص داده‌های Outlier ارائه شده‌اند. این شیوه‌ها به دو دسته‌ی کلی (Global) و محلی (Local) تقسیم می‌شوند. شیوه‌های کلی برای تشخیص Outlierها متکی به بررسی کل داده‌های موجود هستند، در حالی که شیوه‌های محلی به بررسی داده‌های نزدیک داده‌ی مورد بررسی اکتفا می‌کنند. (روش‌هایی مشابه با k-nearest neighbors). خصیصه‌ی دیگری که در عمده‌ی این شیوه‌ها دیده می‌شود این است که شیوه‌های کلی صفر و یکی عمل کرده و داده‌های مورد بررسی را به دو دسته‌ی «Outlier» و «داده‌ی عادی» تقسیم‌بندی می‌کنند، در حالی که شیوه‌های محلی به داده‌ی مورد بررسی یک فاکتور «Outlierness» می‌دهند که نشان‌دهنده‌ی میزان حدودی Outlier بودن داده‌ی مورد بررسی است. هرچند که این امر همیشه صادق نیست.

۲-۲- تعریف ریاضی

در این مقاله، مدل تشخیص Outlier جدیدی معرفی شده که ایده‌های فاکتورهای محلی و مبتنی-بر-چگالی همانند LOF (و واریانت‌های آن، مثل LOCI) را با مفاهیم احتمالاتی ترکیب می‌کند تا میزان Outlier بودن یک داده را مدل کند. مزیت یک مدل احتمالاتی ایجاد نسبی یک تحمل (Tolerance) نسبت به داده‌های نویز است. مدل‌های سنتی حتی در مواردی تاثیر داده‌های نویز را زیاد می‌کنند. به عنوان مثال، متد LOF مبتنی بر محاسبه‌ی فاصله‌ی kتا داده‌ی نزدیک نسبت به داده‌ی فعلی است و انتخاب اشتباه عدد k می‌تواند منجر به نتایج ناپایدار شود. در ادامه، D به معنای مجموعه‌ای از n داده است و d به معنای تابع فاصله‌ای که برای تشخیص Outlierها محاسبه می‌شود. برای پایدارتر شدن نتایج محاسبات، مفهومی به نام «فاصله‌ی احتمالاتی» یا pdist(o, S) معرفی می‌شود که پارامتر o در آن یکی از داده‌های موجود در مجموعه‌ی D است و S یک مجموعه‌ی محلی که زیرمجموعه‌ای از D است. این فاصله، باید دارای خصوصیت زیر باشد:

$$\forall s \in S: P[d(o,S) \leq pdist(o,S)] \geq \psi$$

این خصوصیت به این معناست که یک کره حول o با شعاع pdist هر داده‌ای در S را با احتمال ψ پوشش می‌دهد. فاصله‌ی pdist(o,S) می‌تواند به عنوان «محدوده‌ی احتمالاتی» مجموعه‌ی محلی S در نظر گرفته شود. تفاوت اصلی این نوع محدوده با محدوده‌ی نرمال این است که محدوده‌ی احتمالاتی از عمد رخداد برخی خطاها را جایز می‌شمرد. معکوس فاصله‌ی احتمالاتی می‌تواند به عنوان تخمینی برای چگالی محدوده‌ی S در نظر گرفته شود:

$$pdens(S) = \frac{1}{pdist(o,S)}$$

اگر به جای ψ از $\lambda = \sqrt{2}\,\mathrm{erf}^{-1}(\psi)$ استفاده شود که در آن، erf نشان‌دهنده‌ی تابع ارور گاوسی است، در حین تخمین چگالی S، می‌توانیم Outlier ها را به معنای داده‌هایی تفسیر کنیم که فاصله‌ی آن‌ها با میانگین S، بیش‌تر از $\lambda.\sigma$ است که σ در این‌جا به معنای انحراف معیار است. مقادیر λ توسط قانون تجربی ۳-سیگما تعیین می‌شوند، به این معنا که:

$$\lambda = 1 => \psi \simeq 68\% \,, \lambda = 2 => \psi \simeq 95\% \,, \lambda = 3 => \psi \simeq 99.7\%$$

با فرض این‌که o مرکز S باشد و توزیع فاصله‌های $s \in S$ حول o تقریبا نیم-گاوسی باشد، می‌توان «فاصله‌ی استاندارد» یک داده در S نسبت به o را مشابه با انحراف معیار محاسبه کرد:

$$\sigma(o,s) = \sqrt{\frac{\sum_{s \in S} d(o,s)^2}{|S|}}$$

فرق این فاصله با انحراف معیار این است که فرض می‌کند میانگین فاصله‌ها نسبت به o صفر است. (و نه برابر با $E[d(o,S)]$). به طور خاص، این تفاوت به این معناست که نمی‌توان توزیع فاصله‌ها را به صورت

نرمال فرض کرد، در عوض تصـــور می‌کنیم S *حول* o به صـــورت نرمال توزیع شـــده اســـت. هم‌چنین، «فاصله‌ی احتمالاتی مجموعه‌ای» از داده‌ی o به S با میزان اهمیت λ را به صورت زیر تعریف می‌کنیم:

$$pdist(\lambda, o, S) := \lambda . \sigma(o, S)$$

این فاصله‌ی جدید، به ما قدرت کنترل تخمین چگالی S حول o را می‌دهد، اما بر روی ترتیب Outlierها تاثیری نخواهد گذاشت.

معمولا، هر مدل‌سازی آماری بر اساس برخی مفروضات است. این مورد نیز مبتنی بر فرض‌های زیر است:

یک. زیرمجموعه‌ی محلی S به مرکزیت داده‌ی مدنظر o است.

دو. توزیع فاصله‌ها مشابه با قسمت مثبت یک توزیع نرمال است.

در مورد فرض یک، در صورتی که o از c_s (نقطه‌ی مرکزی مجموعه‌ی S) فاصله‌ی زیادی داشته‌باشد، نشان‌دهنده‌ی این است که S نسبت به o نامتقارن است و اندیکاتوری‌ست برای Outlier بودن نقطه‌ی o. فرض دو نیز از تقریبی حول قضیه‌ی آماری مقدار میانی استخراج شده و استفاده از این قضیه نشان می‌دهد که ما توزیع واقعی نقاط S را محدود به توزیع خاصی نکرده‌ایم و هر توزیعی که داشته باشند، با توجه به قضیه‌ی مقدار میانی و فرض این‌که از فاصله‌های مدل L_p (مثل منهتن یا اقلیدسی استفاده می‌کنیم) می‌توان آن‌ها را با توزیع نرمال تقریب زد.

با توجه به این دلایل برای تقریب زدن چگالی یک مجموعه حول یک نقطه، «معیار احتمالاتی محلی Outlier» نسبت به یک داده‌ی $o \in D$، یک میزان اهمیت λ و یک مجموعه‌ی محلی $S(o) \subset D$ به صورت زیر تعریف می‌شود:

$$PLOF_{\lambda,S}(o) := \frac{pdist\big(\lambda, o, S(o)\big)}{E_{a \in S(o)}[pdist\big(\lambda, s, S(s)\big)]} - 1$$

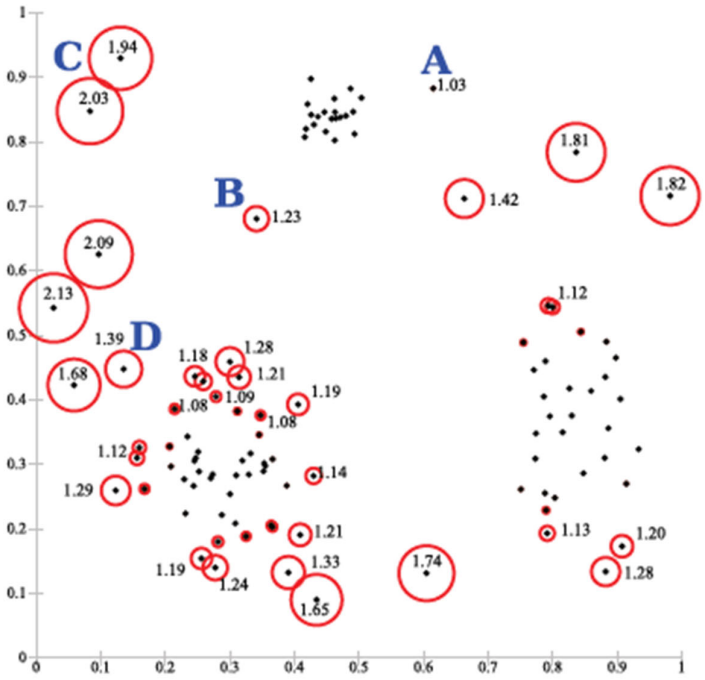
مقدار PLOF یک داده‌ی o نسبت تقریب چگالی حول o که بر اساس S(o) است به امیدریاضی تقریب چگالی حول تمامی داده‌ها در زیرمجموعه‌ی S را حساب می‌کند. این معیار که بسیار شبیه به معیار LOF است هنوز یک احتمال نیست و نیاز به نرمال‌سازی دارد. یک مقدار $0 \leq$ به معنای Outlier بودن نیست، اما مقادیر بالاتر میزان *Outlier* بودن بیش‌تری را نشان می‌دهند. برای نرمال‌سازی این مقدار فرای یک دیتاست خاص، مقدار متوسط $nPLOF$ بر اساس محاسبه‌ی $PLOF$ برای چند دیتاست به دست می‌آید:

$$nPLOF := \lambda . \sqrt{E[(PLOF)^2]}$$

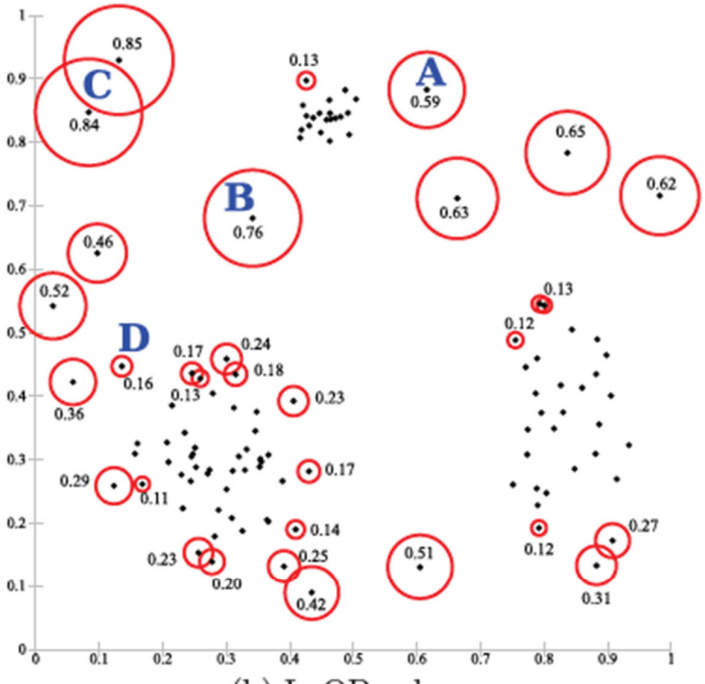
این مقدار می‌تواند به عنوان نوعی انحراف معیار نسبت به مقادیر $PLOF$ در نظر گرفته شود، یعنی $\lambda . \sigma(PLOF)$ با فرض ۰ بودن میانگین. ما در نهایت دوباره تابع ارور گاوسی را بر محاسبات اعمال می‌کنیم تا یک مقدار احتمال به نام «احتمال محلی *Outlier* بودن» را حساب می‌کنیم که مستقیما نشان دهنده‌ی احتمال *Outlier* بودن نقطه‌ی $o \in D$ است.

$$LoOP_{s(o)} := \max\{0, \operatorname{erf}\left(\frac{PLOF_{\lambda,s(o)}}{nPLOF . \sqrt{2}}\right)\}$$

این مقدار برای نقاطی که درون مناطق پرچگالی هستند نزدیک به صفر خواهد بود و برای نقاط دور افتاده نزدیک به یک. در نتیجه، بر خلاف مقادیری همانند LOF که بر روی دیتاست‌های مختلف نتایج متفاوتی ارائه خواهد کرد، این مقدار برای تمامی نقاط روی یک دیتاست و بر روی دیتاست‌های مختلف نتایج پایداری ارایه خواهد کرد. در جدول زیر نیز تفاوت این دو معیار برای یک دیتاست ساختگی را با مقادیر $k=20$ و $\lambda = 3$ مشاهده می‌کنید:



(a) LOF values.



(b) LoOP values.

Figure 1: Comparison of the interpretability of local, density-based outlier scoring (here: LOF) values and LoOP values on 2D synthetic data. Both algorithms were run with $k = 20$, for LoOP $\lambda = 3$.