داده کاوی بزرگ مقرون به صرفه در زمینه ابر : یک مطالعه موردی با K-means

چکیده

کاوش داده ی بزرگ , اغلب نیازمند منابع محاسباتی فوق العاده می باشد. این امر به یک مانع عمده در رابطه با استفاده ی وسیع از تجزیه و تحلیل داده های بزرگ تبدیل شده است. محاسبات ابری به محققانی که در زمینه ی داده فعالیت می کنند , اجازه ی دسترسی به منابع محاسباتی , بر اساس تقاضای ساخت راه حل های تحلیلی داده ای بزرگ در ابر را می دهد. هر چند , هزینه ی پولی کاوش داده ی بزرگ در ابر , هنوز هم می تواند بر خلاف انتظار مان , بالا باشد.

برای مثال , اجرای مثال های Amazon EC2 100 m4- xlarge به مدت یک ماه هزینه ای در حدود $17,495,00 را به دنبال دارد. در این زمینه , مسئله ی حیاتی به منظور تجزیه و تحلیل هزینه ی بهره وری (هزینه ی مقرون به صرفه) داده کاوی بزرگ در ابر , چگونگی دستیابی به یک نتیجه ی رضایت بخش کافی با حداقل هزینه ی محاسباتی ممکن است. در سناریو های داده کاوی بزرگ حقیقی , دقت 100% غیر ضروری است. در عوض , اغلب , دستیابی به یک دقت کافی , برای مثال 99% , یا هزینه ی کمتر مانند 10% , نسبت به هزینه ی دستیابی با دقت 100% , ترجیح داده می شود.

در این مقاله , ما به کشف و نمایش داده کاوی بزرگ مقرون به صرفه به همراه یک مطالعه ی موردی و با استفاده از K-means اقدام می کنیم. با استفاده از مطالعه ی موردی , در می یابیم که دست یابی به دقت 99% تنها نیاز به هزینه ی محاسبتی 0.32%-46.17% مربوط به دقت 100% دارد. این یافته , سنگ بنای لازم را برای داده کاوی مقرون به صرفه در انواع دامنه ها قرار می دهد.

کلمات کلیدی : محاسبات ابری ; داده کاوی ; مقرون به صرفه ; داده بزرگ ; K- means

I . مقدمه

دوران داده های بزرگ آغاز شده است. امروزه , نود درصد از داده ها در طی دو سال اخیر تولید شده و 2.5 کوانتیلین از داده های جدید هر روزه تولید می شوند. برای مثال , هر ماهه در حدود 6 میلیارد عکس جدید به وسیله ی فیسبوک گزارش شده و در هر دقیقه 72 ساعت ویدئو به یوتیوب آپلود می شود. این رشد انفجاری داده , داده کاوی بزرگ را در رنج وسیعی از زمینه ها همانند تجارت , حکومت , مراقبت های بهداشتی و غیره فعال ساخته است.

بسیاری از الگوریتم های داده کاوی در پیچیدگی محاسباتی , نمایان هستند. در سناریو های داده ای بزرگ , به طول انجامیدن فرایند داده کاوی برای ساعت ها و یا حتی روز ها به منظور تکمیل , پدیده ی نادری نیست. از این رو , داده کاوی بزرگ اغلب نیازمند منابع محاسباتی عظیم است. بسیاری از کسب و کار ها و سازمان ها از عهده ی هزینه های زیر ساختی داخلی برای داده کاوی بزرگ , بخصوص کسب کار های با اندازه ی کوچک و متوسط , بر نمی آیند. محاسبات ابری راه حلی کاملی برای این سازمان ها و کسب و کار ها به حساب می آید. مدل " pay-as-you-go " که به و سیله ی محاسبات ابری رواج یافته است , دسترسی منعطف و مورد تقاضا برای منابع محاسباتی غیر محدود مجازی را فراهم می کند. این امر اجازه ی اجرای داده کاوی بزرگ را تنها با استفاده از منابع محاسباتی ضروری برای مدت زمان لازم می دهد. در حقیقت , بسیاری از کسب و کار ها و سازمان ها در حال حاضر , دارای داده های ذخیره شده در ابر هستند.

برای چنین کسب و کار ها و سازمان هایی , انجام داده کاوی در ابر , یک انتخاب طبیعی است. هر چند , هزینه ی پولی استفاده از منابع منابع محاسباتی در ابر (با عنوان هزینه ی محاسبات به آن اشاره شده است) در صورتی که به صورت مناسبی مدیریت نشوند , برای داده کاوی بزرگ , به صورت غیر منتظره ای بالا خواهد بود.

برای مثال , اجرای ماشین مجازی (VM) 100 m4-xlarge Amazon EC2 , هر روزه هزینه ای در حدود $583,00 را در پی دارد. بنابراین , هزینه ی بهره وری (هزینه ی مقرون به صرفه) در ابر , تبدیل به مانعی

عمده برای کاربرد های وسیع داده کاوی بزرگ شده است. در این زمینه , مسئله ی حیاتی به منظور تجزیه و تحلیل هزینه ی بهره وری داده کاوی بزرگ در ابر , چگونگی دستیابی به یک نتیجه ی رضایت بخش کافی در حداقل هزینه ی محاسباتی ممکن است. در بسیاری از سناریو های داده کاوی , دستیابی به نتیجه ی مطلوب , همانند دقت 100% ضروری نیست. برای مثال , در رابطه با بازاریابی می توان گفت که داده کاوی معمولا بر روی تعداد زیادی از مشتریان اجرا می شود. حاشیه ی معقولی از بی دقتی قابل قبول است. برای مثال , بازاریابان نیاز ندارند تا مشتریانشان در دسته بندی دقت 100% قرار گیرند. تا زمانی که آنان بتوانند تصویری عمومی را بدست آورند , قادر به تصمیم گیری خواهند بود. در حقیقت , در برخی از سناریو های داده کاوی , آنان دارای دقت 100% نخواهند بود. برای مثال , در پیش بینی آ ب و هوا و پیش بینی ترافیک , این قضیه صادق است.

دست یابی به هزینه ی بهره وری با استفاده از متوقف ساختن فرایند داده کاوی امکان پذیر است, چرا که اغلب دست یابی به یک دقت کافی همانند 99% یا 99.9% , در هزینه های پایین همانند 10% یا 20% نسبت به هزینه ی دستیابی به دقت , 100% از ارجحیت بالاتری برخوردار است.

هزینه ی بهره وری داده کاوی , به تحلیل داده های بزرگ اجازه کمک کرده و اجازه می دهد تا رنجی وسیعی از زمینه ها , به وسیله ی کسب و کار ها و سازمان ها, به ویژه سازمان هایی با اندازه ی کوچک و متوسط تحت پوشش این امر قرار گیرند. هر چند که این مورد به خوبی توسط جامعه ی پژوهشی کشف نشده است. در این مقاله , ما به مطالعه ی k-means , یکی از 10 الگوریتم داده کاوی برتر , به کشف و نمایش هزینه ی بهره وری داده کاوی در ابر می پردازیم.

بخش های باقی مانده ی مقاله به شکل زیر سازماندهی شده اند.

بخش II به توضیح آثار مربوطه می پردازد , بخش III به معرفی روش شناسی اتخاذ شده در این مطالعه می پردازد. بخش IV به ارائه و تحلیل نتایج تجربی, بخش V بیشتر به توضیح یافته های این مطالعه , بخش VI به تحلیل و بررسی تهدید های اعتبار آزمایشات ما و بالاخره, بخش VII به نتیجه گیری این مقاله و به توضیح کار های آینده می پردازد.

II . آثار مرتبط

مدل pay-as-you-go که به وسیله ی محاسبات ابری معرفی و ترویج داده شده است , به صورت قابل توجهی مسیر زیرساخت IT را تغییر داده و مورد استفاده قرار می گیرد. از آن جایی که بسیاری از مزیت های عمده ی ارائه شده توسط محاسبات ابری , پیرامون قابلیت انعطاف این مدل هزینه ای به وجود آمده اند , بهره وری هزینه , توجه بسیاری از محققان را به عنوان یک مسئله ی اصلی تحقیق , در محاسبات ابری به خود جذب کرده است.

مطالعات بسیاری در رابطه با محاسبات بهره وری هزینه در ابر صورت گرفته است. Ostermann et al , تاثیر و هزینه ی بهره وریEC2 آمازون را با استفاده از micro-obenchmark و kernel ها مورد تجزیه و تحلیل قرار داد. دو مطالعه ی مشابه , که یکی از آن ها توسط Mehrotra et al به همراه بار های کاری NASA HPC انجام شده و دیگری به وسیله ی Iosup et al , به همراه بار های کاری Many-Task Computing (MTC) انجام شده , و هر دو تحقیق به چنین نتیجه ی مشترکی رسیدند که کارایی سرویس های ابری عمومی برای برنامه های HPC کافی نیست. همان طور که vendor های ابری در طی چند سال اخیر , سرویس های ابری خود را به صورت پیوسته بهبود بخشیده اند , مطالعات بسیاری پیرامون کارایی و همچنین بهره وری هزینه ی سرویس های ابری عمومی انجام گرفته و نتایج رضایت بخشی به دست آمده است.

Berriman et al , به مطالعه ی بهره وری هزینه ی برنامه های محاسباتی علمی در EC2 آمازون , از طریق انجام یک مقایسه بین EC2 آمازون و Cluster کارای Abe در مرکز ملی محاسبات ممتاز در ایالات متحده پرداخت. مطالعات وی نشان داد که ابر EC2 آمازون کارایی بهتری را ارائه کرده و مقدار پردازنده و برنامه های حافظه ی محدود نسبت به برنامه های I/O-Bound بیشتر است.

Carlyle et al , مطالعه ی مشابهی را انجام داد که بااستفاده از برنامه ی " HPC community cluster " دانشگاه پورودا , به مقایسه ی هزینه های محاسبات کارا در محیط های سنتی HPC و محیط های EC2 آمازون پرداخت. مطالعات وی نشان داد , زمانی که سازمان 3 شرط زیر را برآورده می سازد , یک کلاستر (خوشه) in-house بهره وری هزینه ی بالاتری را نشان خواهد داد:

1) دارا بودن تقاضای کافی که به طور کامل از cluster بهره ببرد ; 2) دارا بودن بخش فناوری اطلاعات که قادر به حفظ زیر ساخت های IT باشد ; 3) دارا بودن تحقیقاتی فعال در زمینه ی سایبر به عنوان یک اولویت .

این محدودیت ها , در حقیقت به تثبیت قابلیت انعطاف و بهره وری هزینه ی اجرای برنامه های Computation-intensive در ابر های تجاری کمک می کند. Deelman et al , مصالحه ی بین هزینه ی اجرای برنامه هایe Computation-intensiv و برنامه های data- intensive و کارایی آن ها در ابر را انجام داد. یافته ی اصلی آنان بدین صورت بود که اجرای برنامه های Computation-intensive دارای بهره وری هزینه ی بالاتری نسبت به برنامه های data- intensive در ابر داراست. Gupta et al , به ارزیابی و بررسی کارایی برنامه های HPC در ابر پرداخت. آزمایشات وی نشان داد که سرویس های ابری موجود نمی توانند جایگزین Super کامپیوتر ها شوند اما می توانند به صورت موثری مکمل آنان گردند. Wang et al , به ارائه ی یک چارچوب multi-tanent تصادفی برای بررسی زمان پاسخ سرویس های ابری به عنوان یک اندازه ی تصادفی به همراه یک توزیع احتمالی عمومی پرداخت. Hwang et al , به امتحان کارایی سرویس های ابری آمازون به همراه 5 برنامه , با تمرکز بر روی مقایسه ی بین استراتژی های scaling out و scaling up پرداخت.

تحقیق موجود , رشد سریع محبوبیت اجرای برنامه های Computation-intensive در ابر را نشان داده و به ارائه ی تصویری کلی در رابطه با بهره وری هزینه ی داده کاوی بزرگ در ابر از طریق مقایسه بین محیط ابری و یک محبط کلاستر سنتی اقدام می کند. در این مطالعه , ما به موضوع بهره وری هزینه از دیدگاهی مهم و مختلف با هدف دستیابی به دقتی رضایتبخش در نسبت نسبتا کوچکی از هزینه ی کلی دستیابی به دقت 100% با متوقف ساختن فرایند داده کاوی در برخی از مراحل قبل از تکمیل , خواهیم پرداخت.

III . روش شناسی

در این بخش , ما روش مورد استفاده در مطالعه ی موردی خود را که در بردارنده ی تکنیک داده کاوی , مجموعه داده ها , روش محاسبه ی دقت , مدل هزینه و پروسه ی مطالعه است را توضیح می دهیم.

A . K-means

رنج وسیعی از تکنیک های داده کاوی که به منظور کاوش و نمایش بهره وری هزینه ی داده کاوی بزرگ در ابر اتخاذ شود , موجود است. Wu et al , به صورت سیستماتیک , 10 تکنیک برتر داده کاوی را مطابق با تاثیر آن ها در جامعه ی پژوهشی مطرح ساخته است. K-means نسبت به C4.5 در جایگاه دوم قرار دارد. K-means الگوریتم data clustering ساده ای است که به صورت intensive مورد مطالعه قرار گرفته و به صورت وسیع در صنعت و دانشگاه به کار برده شده است. علاوه بر این , k-means همگرا بوده - و به تکرار نتیجه ی نهایی (و بهینه) نزدیک است. این امر به ما اجازه می دهد تا به محاسبه و نمایش دقت نتیجه ی clustering متوسط و همچنین هزینه وارده , در هر تکرار فرایند clustering بپردازیم.

مشکل clustering , پارتیشن بندی یک مجموعه داده ی ارائه شده D به تعدادی cluster است , به طوری که کل فاصله ی اقلیدسی بین هر نقطه ی داده و مرکز پنهان آن به حداقل برسد. راه حل این مسئله دقیقا NP-Hard است. ما در مطالعه ی موردی خود , از راه حل جستجوی محلی که توسط Lloyd ارائه شده است , استفاده می کنیم. تاکنون , الگوریتم های clustering بسیار مشهوری در برنامه های علمی و صنعتی مورد استفاده قرار گرفته است. k-means فرایند ساده ی زیر را دنبال می کند:

1. انتخاب مراکز دلخواه k $C=\{c_1, c_2, \ldots, c_k\}$.
2. برای هر $i \in \{1, \ldots, k\}$, کلاستر C_i به عنوان مجموعه ای از داده ها تنظیم گردد.
3. برای هر $i \in \{1, \ldots, k\}$, c_i به عنوان مرکز C_i تنظیم گردد.

$$c_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \ ;$$

4. گام های 2 و 3 را تا زمانی که C تغییر نکند , تکرار کنید. در طول این فرایند , فاصله ی اقلیدوسی بین هر نقطه ی داده و مرکز پنهان آن, تکرار یکنواخت را یکی پس از دیگری کاهش می دهد. این امر تضمین می کند که هیچ تخصیص کلاستری در طول فرایند تکرار نمی شود. بنابراین , این فرایند همواره خاتمه می یابد. با توجه به نقاط داده ای موجود در D , تنها مجموعه تخصیص کلاستر k^n وجود دارد. پیچیدگی زمانی الگوریتم K-

means متعلق به Lloyd , به شکل $O(nkdi)$ است . به گونه ای که n تعداد نقاط داده ای در K , D تعداد کلاستر ها و i تعداد فعل و انفعالات مورد نیاز برای الگوریتم , به منظور تکمیل کار است.

بهره وری الگوریتم k-means متعلق به Lloyd , نسبت به مراکز اولیه انتخاب شده ی دلخواه k حساس است. مراکز اولیه ی نامناسب منجر به تکرار های بیش از حد و زمان محاسبه در سناریو های داده ای بزرگ می گردد. چندین قسمت کاری پیرامون نحوه ی انتخاب k مرکز اولیه ی مناسب وجود دارد. متاسفانه , هیچ راه حل ساده و همه جانبه ای برای این مشکل وجود ندارد. در این مطالعه , اثر تصادفی در انتخاب مرکز اولیه پیرامون تاثیر الگوریتم , بایستی با حصول اطمینان از وجود ثبات در انتخاب مراکز اولیه , در سراسر آزمایشات محدود گردد.

بنابراین , ما از یک روش مشابه , با انتخاب مرکز اولیه استفاده شده به وسیله ی Erisoglu et al استفاده می کنیم که بیشترین فاصله ی اقلیدوسی بین مراکز اولیه را دنبال می کند. روش مرحله ی 1 موجود در الگوریتم , k- means متعلق به Lloyd را با فرایند ساده ای که مراکز اولیه را گسترش می دهد , جایگزین می سازد :

1. انتخاب مراکز k $C = \{c_1, c_2, \ldots, c_k\}$.

a . انتخاب نقطه داده به همراه بیشترین فاصله ی اقلیدوسی از مرکز به عنوان اولین مرکز c_1 .

b . محاسبه ی فاصله ی اقلیدوسی بین هر نقطه ی داده و کلیه ی مراکز انتخاب شده ی m :

$$d(c_i) = \sum_{r=1}^{m} \sum_{j=1}^{p} (x_{r,j} - x_{i,j})^2 \quad i = 1, 2, \ldots, n \quad (1)$$

به صورتی که $x_{i,j}$ مختصات d_i بر روی محور j^{th} (همه ی p ها) است .

انتخاب نقطه ی داده به همراه بالاترین $d(c_i)$ به عنوان مرکز بعدی .

مراحل 1.b و 1.c تا زمان انتخاب کلیه ی مراکز اولیه ی k تکرار شود.

B . ارزیابی دقت

دقت , اندازه گیری مهمی برای ارزیابی بهره وری K-means می باشد. الگوریتم k-means متعلق به Lloyd که در این مطالعه به کار گرفته شده است , الگوریتمی ابتکاری بوده و بنابراین , راه حلی بهینه را برای مشکل

clustering , تضمین نمی کند. به منظور نشان دادن رشد تدریجی در دقت مربوط به نتیجه ی clustering تکرار شونده , ما از پارتیشن نهایی به دست آمده توسط الگوریتمK-means متعلق به Lloyd به عنوان پارتیشن مرجع استفاده می کنیم. توجه داشته باشید که به وسیله ی P_f و از طریق مقایسه ی بین پارتیشن به دست آمده در هر تکرار الگوریتم K-means متعلق به Lloyd , می توانیم نشان دهیم که چگونه دقت پارتیشن متوسط $P_1, \ldots, P_{r-1}$, r-1 کاهش می یابد.

در این جا , دقت به وسیله ی شباهت بین $P_1, \ldots, P_{f-1}$ و P_f مقایسه می شود. در این مطالعه , ما از شاخص rand که در [26] ارئه شد به منظور اندازه گیری شباهت بین $P_1, \ldots, P_{r-1}$ و P_f استفاده می کنیم. شاخص rand به صورت گسترده , به منظور محاسبه ی دقت پارتیشن ها از یک دیدگاه ریاضی به کاربرده می شود. شاخص Rand شباهت بین دو پارتیشن P_1 و P_2 مربوط به مجموعه داده های مشابه D را اندازه گیری می کند. هر پارتیشن به عنوان جمعی از تصمیمات دو طرفه ی $n \times (n - 1) / 2$ مشاهده می شود , به صورتی که n اندازه ی مربوط به D است. برای هر جفت از نقاط داده ای داده ای d_i و d_j در D , یک پارتیشن به کلاستر مشابه و یا به کلاستر های مختلف اختصاص می یابد , بنابراین , شباهت بین P_1 و P_2 به شکل زیر تعریف می گردد:

$$Rand(P_1, P_2) = \frac{a+b}{n \times (n-1)/2} \qquad (2)$$

به گونه ای که a , تعداد تصمیمات می باشد. این در شرایطی است که d_i در کلاستری مشابه با عنوان d_j در P_1 , P_2 قرار گرفته و b تعداد تصمیمات است . d_i , d_j موجود در هر دو P_1 , P_2 در کلاستر های مختلفی واقع شده اند. برای مثال , P_1 و P_2 موجود در شکل 1 را در نظر بگیرید. دو کلاستر وجود دارد که در

P_1: $\{d_1, d_2, d_3\}$ and $\{d_4, d_5, d_6, d_7, d_8\}$ و در آن ها

P_2: $\{d_1, d_2, d_3, d_4\}$ and $\{d_5, d_6, d_7, d_8\}$ وجود دارد.

نقاط داده ی d_1 و d_2 به یگ کلاستر مشابه , به هر دو P_1 و P_2 اختصاص یافته اند. بنابراین , جفت (d_1, d_2) متعلق به a است. در مجموع 9 جفت بدین شکل با نام های $(d_1, d_2), (d_1, d_3), (d_2, d_3), (d_5, d_6), (d_5, d_7), (d_5, d_8), (d_6, d_7), (d_6, d_8)$ و (d_7, d_8) وجود دارد. بنابراین , a=9 است.

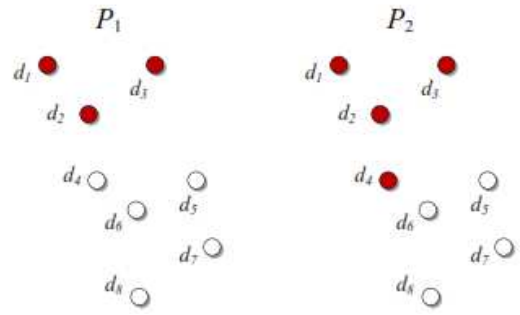
نقاط داده ای d_1 و d_5 به دو کلاستر مختلف , به دو P_1 و P_2 اختصاص یافته اند. بنابراین , جفت (d_1, d_5) متعلق به b است. 12 جفت با نام های

$(d_1, d_5), (d_1, d_6), (d_1, d_7), (d_1, d_8), (d_2, d_5), (d_2, d_6), (d_2, d_7), (d_2, d_8), (d_3, d_5), (d_3, d_6), (d_3, d_7), (d_3, d_8)$

وجود دارد. بنابراین , $b = 12$. با توجه به این که مقادیر a=9 , b=12 و n=8 به ما داده شده است , می توانیم شباهت بین P_1 و P_2 را بدین شکل محاسبه کنیم :

$$Rand(P_1, P_2) = (9 + 12) / (8 \times (8 - 1) / 2) = 0.75$$

فرض کنید که P_2 پارتیشن نهایی به دست آمده توسط الگوریتم k-means متعلق به Lloyd می باشد , در این صورت P_1 به دقت 75% دست می یابد. در فعل و انفعال (r^{th}) نهایی فرایند k-means متعلق به Lloyd , $P_r = P_f$ می باشد . بنابراین , $Rand(P_r, P_f) = 1.0$ نشان می دهد که فرایند با دقت 100% تکمیل شده است.



شکل 1. پارتیشن های P_1 و P_2.

C . مدل هزینه

ما در مطالعه ی خود , به اندازه گیری هزینه ی محاسباتی وارد آمده در طی فرایند Clustering , تا زمان تکمیل آن , می پردازیم. Vendor های ابری مختلف , مدل های هزینه ای مختلفی را به منظور نیاز های مختلف کاربران ارائه می دهند. برای مثال , آمازون را در نظر بگیرید که 3 مدل هزینه ای EC2 را به شرح زیر ارائه می دهد:

- بر اساس تقاضا . این مدل به کاربران اجازه می دهد که هزینه را به صورت ساعتی و بدون هر گونه تعهد طولانی مدت یا پیشاپیش انجام دهند.
- حالت نقطه . این مدل به کاربران اجازه می دهد تا به پیشنهاد پیرامون منابع EC2 ذخیره شده بپردازند.
- موارد رزرو شده . این مدل به کاربران اجازه می دهد تا هزینه ها را به همراه تعهدی طولانی مدت پرداخت نمایند (1 تا 3 سال) .

مدل هزینه ای بر پایه ی تقاضا . یک مدل هزینه ای پایه و قابل انعطاف می باشد که از طریق vendor های ابری مختلف که شامل میکروسافت , گوگل و غیره است , در دسترس قرار می گیرند.

بنابراین , ما در این مدل ما از مدل هزینه ای مورد تقاضا به منظور اندازه گیری هزینه ی محاسباتی وارد آمده , در طی فرایند k-means استفاده می کنیم.

(3) هزینه ی محاسبه = قیمت واحد * زمان محاسبه

زمان محاسبه , مدت زمان به طول انجامیدن فرایند k-means را نشان می دهد که می تواند به آسانی مورد محاسبه قرار گیرد. هر چند , هزینه ی واحد بسته به منبع محاسباتی به کار برده شده به منظور اجرای الگوریتم , به صورت قابل توجهی متفاوت می باشد. برای مثال , آمازون EC2 را در نظر بگیرید. 6 دسته بندی عمده از نمونه های EC2 VM وجود دارد: لینوکس , RHEL (Red Hat Enterprice Linux) , SLES (SUSE Linux Enterprice Server) , ویندوز , ویندوز با استاندارد SQL , ویندوز با استاندارد SQL Web . در هر یک از این دسته بندی ها , انواع مختلفی از نمونه های EC2 VM در قیمت های واحد مختلف قابل دسترس است. تنها در دسته بندی لینوکس , 45 EC2 VM نمونه از 5 نوع وجود دارد : هدف کلی , محاسبه ی بهینه, دست یابی به نمونه های GPU , حافظه ی بهینه و ذخیره سازی بهینه می باشد.

قیمت واحد این نمونه های EC2 VM از محدوده ی $0.0065 تا $16.006 در هر ساعت می باشد. علاوه بر این , این قیمت ها در مراکز داده ای آمازون در 12 منطقه ی مختلف متعلق به آمازون در سراسر جهان متفاوت است. برای مثال , نمونه ی x1.32xarge EC2 VM , هزینه ی $19.341 را در هر ساعت , در منطقه ی سنگاپور را داراست اما هزینه ی یک ساعت این نمونه در منطقه ی ویرجینیای شمالی تنها $13.338 بر آورد می شود.

به منظور دستیابی به سادگی و حالت کلی , ما از زمان محاسباتی به عنوان یک شاخص هزینه ی محاسبات استفاده می کنیم. دلایل به 2 دسته تقسیم می شوند :

1) با توجه به نمونه ی خاصی از منبع ابر همانند : نمونه ی خاص EC2 VM آمازون , هزینه ی محاسبات و زمانی محاسبات دارای همبستگی مثبتی با یکدیگر بوده - زمان محاسباتی طولانی تر موجب هزینه ی محاسباتی بالا تری می شود ; 2) با توجه به الگوریتم داده کاوی ویژه و ورودی مشابه , نمونه ای از منبع ابر با کارایی بالا , معمولا نیاز به زمان محاسباتی کمتری به منظور تکمیل کار دارد , چرا که یک نمونه منبع محاسباتی قدرتمند بسیار دارای قیمت بالاتری نسبت به نمونه ی کم قدرت تر است. هزینه های دیگر ممکن است که برای اجرای

الگوریتم K-means پیش بیایند. برای مثال , مجموعه ی داده به منظور پارتیشن بندی , بایستی در ابر ذخیره شده یا از قبل به ابر منتقل شود. هر چند , هزینه ی وارد شده به وسیله ی ذخیره ی داده و انتقال داده مستقل از فرایند k-means است . ازین رو , ما در این مطالعه تنها بر روی هزینه ی وارد شده به وسیله ی فرایند محاسبات k-means تمرکز کرده و آن را از هزینه های دیگر جدا میکنیم.

D . پروسه ی مطالعه ی موردی

پروسه ی مطالعه ی موردی ما شامل 7 مرحله می باشد:

1. آماده سازی مجموعه داده . مجموعه ی داده به منظور پارتیشن بندی در آزمایشات , آماده می شوند.

2. پارتیشن مجموعه داده ها. مجموعه داده ها با استفاده از الگوریتم K-means متعلق به Lloyd که در بخش III.A به آن اشاره شد , به همراه K متفاوت در آزمایشاتی متفاوت , پارتیشن بندی می شوند. در طی پارتیشن بندی یک مجموعه داده , پارتیشن متوسط و زمان محاسبات در هر تکرار الگوریتم ضبط می شود.

3. محاسبه ی دقت . برای هر دسته از آزمایشات , شباهت بین پارتیشن های متوسط و پارتیشن نهایی با استفاده از فرمول (2) در بخش III.B به منظور به دست آوردن دقت پارتیشن های متوسط , محاسبه می شود.

4. مقایسه ی زمان دقیق. برای هر دسته از آزمایشات , دقت پارتیشن های متوسط حاصل از الگوریتم در هر فعل و انفعال , برخلاف زمان محاسباتی که به وسیله ی الگوریتم و توسط انتهای هر فعل و انفعال مبتنی بر مدل نقاضای توضیح داده شده در بخش III.C , نشان داده شده است.

5. بررسی و بحث . نتایج مقایسات مورد بررسی و بحث قرار می گیرد.

IV . آزمایشات

در این بخش , ما به ارائه ی و بحث و گفتگو در رابطه با پلت فرم آزمایش , مجموعه داده ها و آزمایشات متناظر انجام شده پیرامون مطالعه ی موردی خود, می پردازیم.

A . پلت فرم

الگوریتم K-means بر روی MATLAB R2014b اجرا شد. کلیه ی آزمایشات بر روی ماشین به همراه یک پردازنده ی 3.40 GHz Intel Core i5 و 8 GB حافظه اجرا شد. سیستم عامل ویندوز 7 enterprise , 64 bit بود.

B . مجموعه داده . ما در آزمایشات خود , از دو مجموعه داده استفاده کردیم :

- مجموعه داده ی Gaussian . این مجموعه داده به صورت مصنوعی بر مبنای توزیع Gaussian به وجود آمده , و دنبال کننده ی یک طراحی برای تولید داده مشابه با [27] , [28] می باشد , همچنین دو بخش پذیرفته شده ی وسیع در رابطه با کار بر روی k-means را در بر می گیرد. اولا 72 نقطه ی مرکزی به صورت تصادفی تولید می شود . پس از آن , با تکیه بر هر نقطه ی مرکزی , مطابق با یک توزیع Gaussian با انحراف معیار $\sigma = 0.5$ در امتداد هر مختصات , 13,889 نقطه ی داده ای 2 بعدی تولید شد. در مجموع , 1000.008 نقاط داده ی دو بعدی در این مجموعه داده ی 32 Mb موجود بود.
- مجموعه داده ی Road network (شبکه جاده ای) . این مورد مجموعه داده ی عمومی ارائه شده توسط مرکز آموزش مکانیک و سیستم های هوشمند در دانشگاه کالیفرنیا , آروین است. این مجموعه داده , حاوی اطلاعات طول جغرافیایی , عرض جغرافیایی و ارتفاع در رابطه با یک شبکه ی جاده ای است که یک منطقه ی 135 k m^2 *185 در شمال جوتلند , دانمارک را اداره می کند. 434,874 نقاط داده ای 3 بعدی در این مجموعه داده ای با میزان 20Mb وجود دارد.

C . آزمایشات بر روی مجموعه داده ی Gaussian

در این مجموعه آزمایشات , به منظور ترسیم این مورد , ما الگوریتم K-means متعلق به Lloys را به منظور پارتیشن بندی مجموعه داده ی Gaussian به همراه k= 2,4,8 ,16,32,64 اجرا کردیم . مجموعه داده ی مشابه با Gaussian برای انجام یک مقایسه ی عادلانه , در سراسر آزمایشات مورد استفاده قرار گرفت. مطابق با روش انتخاب مرکز اولیه که در بخش III.A به آن اشاره شد , در مجموع 64 مرکز انتخاب شد , $C = \{c_1, c_2, \ldots, c_{64}\}$.

در آزمایش اول با K=2 , K-means به منظور پارتیشن بندی مجموعه داده ی Gaussian به دو کلاستر با استفاده از c_1 و c_2 به عنوان مراکز اولیه اجرا شد. $\{c_1, c_2, c_3, c_4\}$ به عنوان مراکز اولیه در آزمایش دوم به همراه $k = 4, \{c_1, c_2, \ldots, c_8\}$ در آزمایش سوم و غیره انتخاب شدند. به این ترتیب , ما مجددا تاثیر اتفاق افتاده را با انتخاب تصادفی مرکز اولیه , پیرامون پارتیشن بندی مجموعه داده ها در سراسر آزمایشات مختلف در این دسته , محدود می کنیم.

شکل 2 , افزایش دقت پارتیشن های متوسط در زمان (چند ثانیه) را در مجموعه آزمایشات نشان می دهد. هر نشانگر موجود بر روی خطوط , پارتیشن متوسط را در یک تکرار نشان می دهد. همان طور که در بخش III.A , گفته شد , افزایش K موجب افزایش پیچیدگی زمانی الگوریتم K-means خواهد شد . بر اساس نتایج , این الگوریتم به منظور تکمیل , نیاز به تکرار و زمان بیشتری دارد. شکل 2 به اثبات این قضیه می پردازد. مهمتر از همه این که , شکل 2 نشان می دهد که الگوریتم k-means در ابتدا تعداد نسبتا کمی از تکرار ها را به منظور دست یابی به دقت بالا مورد استفاده قرار داده و سپس تعداد بالایی از تکرار ها را به منظور همگرایی با دقت 1.0 , همانند دقت 100% صرف می کند. ما این پدیده را long tail می نامیم. در این دسته از آزمایشات , ما همچنین الگوریتم k-means را به همراه k دیگر و مفهوم long tail مشابه اشاره شده , اجرا می کنیم. مفهوم long tail , نشان می دهد که الگوریتم k-means , بسیاری از زمان محسباتی را در مراحل وسط و پایانی مصرف کرده است.

آزمایشی را که در آن k=64 است را در نظر بگیرید. الگوریتم k-means از 3 تکرار (2.27 ثانیه) به منظور دستیابی به دقت 0.99 , یعنی , 99% استفاده می کند . پس از 84 تکرار بیشتر (33.20 ثانیه) تکمیل را ادامه می دهد.

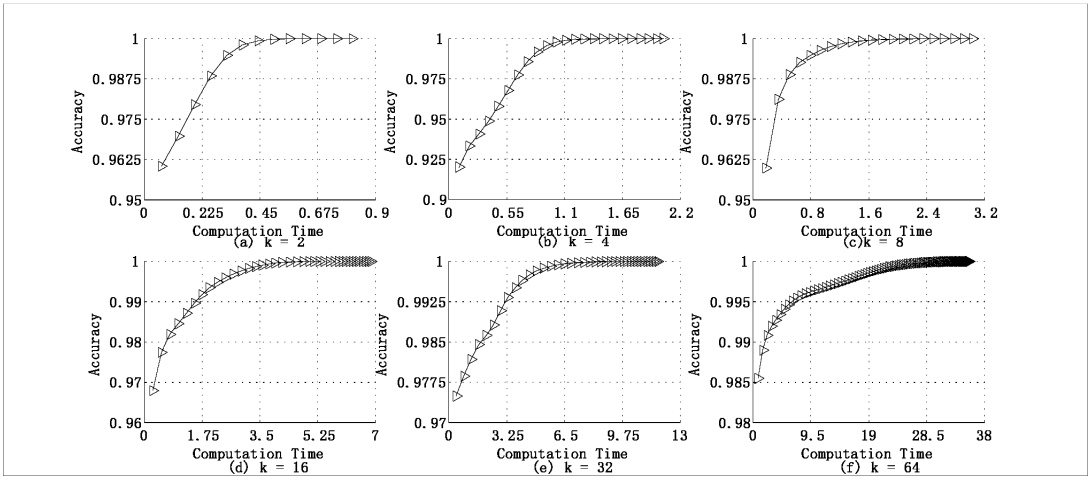
جدول I زمان محاسباتی استفاده شده توسط الگوریتم K-means را به منظور دستیابی به دقت بالای 0.99 , 0.999 و0.9999 در این دسته از آزمایشات , خلاصه کرده است. این جدول نشان می دهد که همزمان با افزایش k , الگوریتم k-means تمایل به صرف زمان بیشتری به منظور همگرایی از دقت 0.99 تا دقت 1.0 را داراست.

با در نظر گرفتن k=32 , الگوریتم 75.23% (24.77%- 100%) از زمان محاسباتی خود را صرف دستیابی به دقت 0.99 تا 1.0 می کند , این عمل هنگامی اتفاق می افتد که این الگوریتم 93.59% (6.41 %-100%) را به منظور انجام کاری مشابه با k=64 صرف می کند . این یافته نشان می دهد که در صورتی که کاربر از الگوریتم k-means در ابر استفاده کند , نیاز به دقت 100% نداشته , و می تواند در مرحله ی اولیه با دقتی رضایتبخش کار خود را متوقف کرده و در در مقدار زیادی از هزینه های پولی برای گرفتن یک دقت 1.0 صرفه جویی کند.

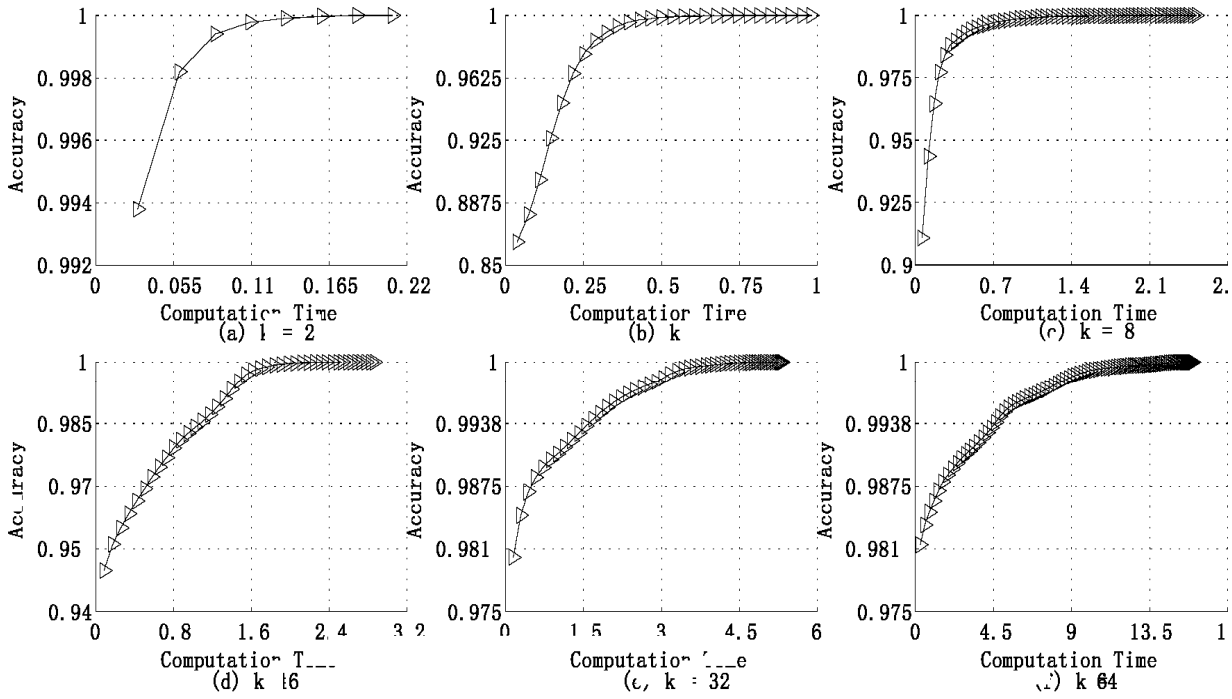
D . آزمایشات بر روی مجموعه داده ی شبکه ی جاده ای

در این دسته از آزمایشات , الگوریتم k-means به منظور پارتیشن بندی مجموعه داده ی شبکه ی جاده ای با k= 2,4,8,16,32,64 اجرا شد. مراکز اولیه برای الگوریتم k-means به روشی مشابه , همانند آزمایشات بر روی مجموعه داده ی Gaussian انتخاب شدند. شکل 3 نتایج حاصل را نشان می دهد. این شکل نشان می دهد که مفهوم long tail همچنین در طی پارتیشن بندی مجموعه داده ی شبکه ی جاده ای نیز وجود دارد. این امر یافته های ما را در آزمایشات انجام گرفته پیرامون مجموعه داده ی Gaussian تایید می کند.

جدول III زمان محاسبات استفاده شده به وسیله ی الگوریتم K-means را به منظور دست یابی به دقت 0.99 , 0.999 و 0.9999 در این دسته از آزمایشات , خلاصه می کند. به صورت میانگین , زمان محاسباتی مورد نیاز برای الگوریتم k-means به منظور دستیابی به این آستانه ها_مشابه با موارد ارائه شده در جدول I می باشد که 24.28% در مقابل 0.99 ≥ 26.43% , 54.62% در مقابل 0.999 ≥ 53.76% و 82.63% در مقابل 0.9999 ≥ 72.76% را ارائه می دهد. این یافته نشان می دهد که یک نقطه ی توقف اولیه به همراه یک دقت رضایت بخش به جای یک دقت 100% راه حل Clustering با بهره وری هزینه ی بالا ست.



شکل 2 . دقت و زمان محاسبه ی K-means بر روی مجموعه داده ی Gaussian



شکل 3. دقت و زمان محاسبه ی k-means بر روی مجموعه داده ی شبکه ی جاده ای.

جدول 1 . درصد زمان محاسبات برای k-means بر روی مجموعه داده ی Gaussian

	Accuracy		
k	**≥ 0.99**	**≥ 0.999**	**≥ 0.9999**
2	39.60%	54.74%	69.76%
4	41.18%	58.79%	71.79%
8	21.21%	49.26%	72.71%
16	25.39%	51.95%	69.86%
32	24.77%	48.75%	70.06%
64	6.41%	59.04%	82.38%
Average	26.43%	59.04%	72.76%

V . بحث

نتایج آزمایش ارائه شده در بخش های IV.C و IV.D نشان می دهد که هم چنانکه K افزایش می یابد , K-means به صورت کلی تکرار های زیاد (و بنابراین زمان بیشتری) را به منظور تکمیل کار صرف می کند. در نتیجه , مفهوم long-tail , معنی دار می شود. برای مثال , شکل 2 را در نظر بگیرید. این الگوریتم , در مجموع 13,23,21,38 , 38 و 87 تکرار را به منظور تکمیل به ترتیب به همراه k=2,4,8,16,32,64 استفاده می کند. مفهوم long tail نیز مربوط به مقیاس این سناریو است. نقاط داده که بایستی در اولین دسته از آزمایشات پارتیشن بندی شوند , بسیار بیشتر از نقاطی هستند که در دومین دسته از آزمایشات قرار دارند. بر این اساس , افزایش در تعداد تکرار ها و زمان محاسبات به همراه افزایش در k در اولین دسته از آزمایشات قابل توجه تر است. به صورت مشخص , در اولین دسته از آزمایشات , همان طور که k از 2 تا 64 افزایش می یابد , تعداد تکرار ها و کل زمان محاسباتی گرفته شده توسط الگوریتم , به ترتیب از 13 تا 87 و 0.81 تا 35.47 ثانیه افزایش می یابد. در دومین دسته از آزمایشات , این افزایشات به ترتیب از 8 تا 88 و از 0.21 تا 16.01 ثانیه خواهد بود. این مشاهده نشان می دهد که در یک سناریوی بزرگ داده کاوی در سطح واقعی , هزینه ی بهره وری بسیار مهم است. ما در آزمایشات , تاثیر تصادفی بودن را در انتخاب مراکز اولیه بر روی نتایج آزمایش با استفاده از روش ارائه شده در بخش III.A محدود کردیم.

جدول II . درصد زمان محاسبات برای K-means در مجموعه داده ی شبکه جاده ای.

		Accuracy	
k	**≥ 0.99**	**≥ 0.999**	**≥ 0.9999**
2	14.14%	40.35%	76.25%
4	36.52%	57.94%	82.97%
8	16.08%	40.69%	80.55%
16	46.17%	63.96%	80.61%
32	16.26%	61.86%	84.26%
64	16.50%	62.94%	91.14%
Averag	24.28%	54.62%	82.63%

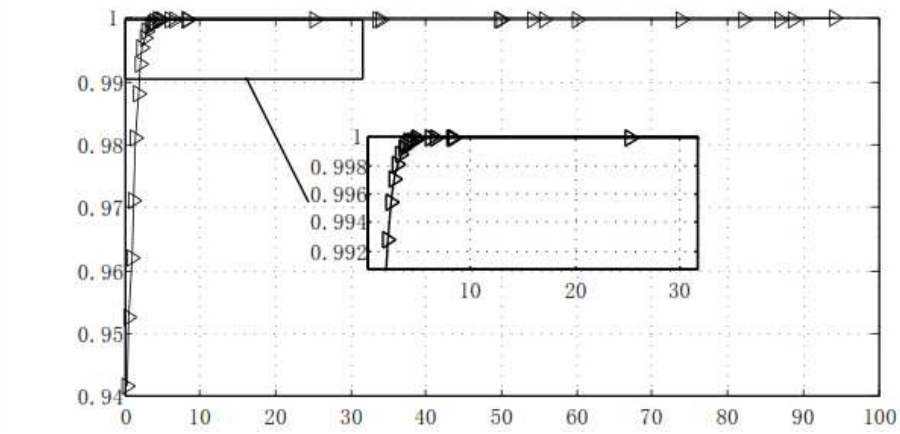
هر چند , ما از طریق آزمایشات انجام گرفته ی خود به این نتیجه رسیدیم که مراکز اولیه ی انتخاب شده به ندرت منجر به پدیده ی long tail قابل ملاحظه می شود. در این جا , اهمیت یک long tail با استفاده از نسبت بین دوره ی زمانی که دقت به سرعت افزایش می یابد و دوره ی زمانی که دقت به کندی افزایش می یابد اندازه گیری می شود.

برای مثال , یکی از آزمایشات بر روی مجموعه داده ی Gaussian مشابه به همراه k= 20 , اما با مراکز اولیه ی تصادفی اجرا شد. شکل 4 , نتایج مربوطه را نشان می دهد - یک long tail بسیار ملاحظه تر از مواردی است که در شکل 2 و شکل 3 ارائه شده است. نمودار درونی , بخش کوچکی از نمودار بیرونی را که دقت بیش از 0.99% را داراست , افزایش می دهد.

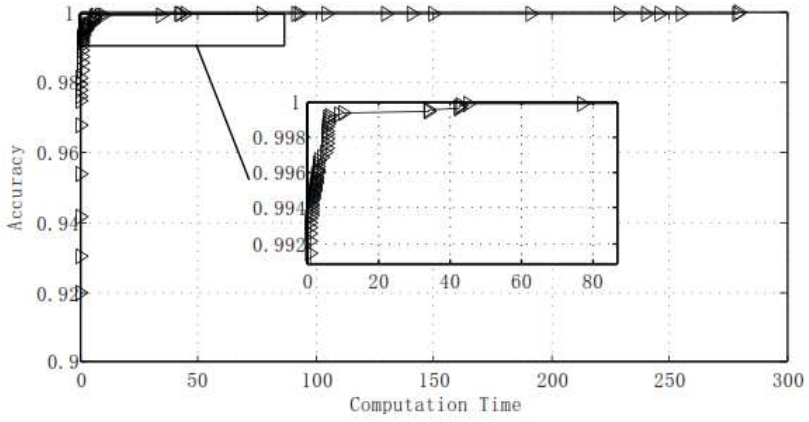
این الگوریتم تنها 2.23% از زمان کل محاسبات را به منظور دستیابی به دقت 0.99 (به صورت دقیق 0.9927) به کار می گیرد. به منظور دستیابی به دقت 0.999 و دقت 0.9999 , این الگوریتم به ترتیب تنها 3.79% و 6.50% از کل زمان محاسبات را به کار می گیرد. در شکل 4 می توان مشاهده کرد که زمان محاسبات به کار گرفته شده توسط الگوریتم برای هر تکرار بر روی long tail , بسیار ناهموار بوده , و برخی از آنان بلند تر از دیگری است. همان طور که در بخش IV.C وIV.D ارائه شد , این امر به این دلیل اتفاق افتاد که الگوریتم مجبور بود تا مراکز را در طول فرایند تغییر دهد , که در آزمایشات با مراکز انتخابی بهتر نیز این اتفاق نیفتاد. همچنین

مفهومی مشابه با آزمایشات ما , بر روی مجموعه داده ی شبکه ی جاده ای مطرح شد , به صورتی که مراکز اولیه به صورت تصادفی انتخاب شدند. شکل 5 نتایج یک آزمایش با K=10 را نشان می دهد. به طور مشخص , این الگوریتم 0.32% از کل زمان محاسبات را به منظور دستیابی به دقت 0.99 , 2.26% به منظور دستیابی به 0.999 و 33.43% به منظور دستیابی به 0.9999 به کار می گیرد.

Long tail های ارائه شده و توضیح داده شده در بخش های IV و V قابل توجه هستند. دلایل عمده به خاطر همگرایی سریع الگوریتم K-means اتفاق می افتد.



شکل 4 . دقت و زمان محاسبه ی k-means بر روی مجموعه داده ی Gaussian به همراه k=20 و مراکز اولیه تصادفی



شکل 5 . دقت و زمان محاسبه ی k-means بر روی مجموعه داده ی شبکه ی جاده ای به همراه k=10 و مراکز اولیه تصادفی

پدیده ی long tail ممکن است که به صورت مشخص در سناریو های داده کاوی به همراه تکنیک های داده کاوی در حال اجرا در ابر , وجود نداشته باشد. هر چند , چگونگی دستیابی به نتیجه ی کافی رضایت بخش در پایین ترین هزینه ی محاسباتی ممکن در این سناریو ها , تا زمانی که الگوریتم داده کاوی به تدریج به زمان مطلوبی دست پیدا کنند , هنوز هم بحرانی است.

IV . تهدید اعتبار

در این بخش , ما در رابطه با تهدیدات اصلی اعتبار مطالعه ی موردی خود بحث می کنیم .

تهدید نسبت به اعتبار ساخت . تهدید اصلی نسبت به اعتبار ساخت مطالعه ی موردی ما , یک روش اندازه گیری پذیرفته شده به منظور ارزیابی دقت یک پارتیشن متوسط در طی فرایند K-means , همانند شاخص Rand می باشد. همان طور که در بخش III.B ارائه شد , محاسبه ی شاخص Rand بر اساس پارتیشن نهایی انجام می پذیرد. بنابراین , این شاخص یک شاخص خارجی به حساب می آید. در بسیاری از سناریو های داده کاوی دنیای واقعی , بخصوص در سناریو های داده کاوی بدون نظارت , آن دسته از شاخص های داخلی که بر دانش پیشین از مجموعه داده ها , همانند پارتیشن نهایی تکیه نکرده اند , مورد پذیرش قرار گرفته اند. معیار های (اندازه های) داخلی مشهور شامل شاخص CH (C alinski-Harabasz) , شاخص DB (Dvies Bouldin) , شاخص Silhouette , شاخص Dunn و غیره می باشد. این شاخص های داخلی ممکن است که دامنه های مختلفی از شاخص Rand را دارا باشند.

برای مثال , شاخص DB و Dunn , محدود به چنین فاصله ای هستند [∞ , 0] . بر اساس عملکرد الگوریتم k-means افزایش یا کاهش مقدار آن ها ممکن است که به صورت پیوسته با شاخص Rand ارتباط نداشته باشد. از این رو , شاخص Rand , انتخابی معمول برای سناریو های داده کاوی دنیای واقعی بخصوص برای سناریو های داده کاوی نظارت نشده نیست. هر چند این تهدید نسبت به اعتبار , در مطالعه ی موردی ما در کمترین حد ممکن قرار دارد , چرا که در این مرحله , هدف ما کشف و نمایش احتمال متوقف سازی فرایند داده کاوی در برخی از مراحل در طول فرایند مربوط به آن , به منظور دستیابی به بهره وری هزینه ی بالاست. شاخص

های داخلی برای این قضیه مناسب نیستند , چرا که آن ها لزوما خوب بودن یک پارتیشن را نشان نمی دهند. در مقابل , شاخص Rand این هدف را به وسیله ی ارزیابی دقت روش نزدیک کردن یک پارتیشن متوسط به پارتیشن نهایی انجام می دهد.

تهدید نسبت به اعتبار خارجی . تهدید اصلی نسبت به اعتبار خارجی مطالعه ی موردی ما نشان دهنده ی مجموعه داده ی استفاده شده در آزمایش ها می باشد. ما در آزمایشات خود از مجموعه داده ی شبکه ی جاده ای , که به صورت گسترده در تحقیقات مختلف از آن استفاده می شود , استفاده کرده ایم. این مجموعه داده دارای ویژگی های مختص خود بوده و همه ی مجموعه ها را به صورت دقیق نشان نمی دهد.

آزمایشات صورت گرفته بر روی مجموعه داده ی مختلف , احتمالا نتایجی متفاوت از آن چه که در شکل 3 , شکل 5 , و جدول II ارائه شد , تولید خواهد کرد. هر چند , ویژگی های عمده ی شکل های مربوطه , برای مثال , یکنواختی در افزایش دقت , مفهوم long tail , و غیره مشابه خواهند بود. این تهدید نسبت به اعتبار خارجی , در آینده با استفاده از مجموعه داده ی تصادفی تولید شده مطابق با توزیع Gaussian به روشی مشابه همانند [27] و [28] به حداقل می رسد.

نتایج حاصل از آزمایشات بر روی بر روی این مجموعه داده ها به صورت عمومی بیشتر نمایان می شود. در حین حال , شباهت در نتایج به دست آمده از آزمایشات بر روی مجموعه داده Road Network و مجموعه داده ی Gaussian نشان می دهد که تهدید نسبت به اعتبار خارجی مطالعه ی موردی ما , به حداقل رسیده است.

تهدید نسبت اعتبار داخلی . تهدید اصلی نسبت به اعتبار داخلی مطالعه ی موردی ما , جامع بودن آزمایشات را نشان می دهد. پارتیشن های متوسط و نهایی مجموعه داده ای که توسط الگوریتم k-means به دست می آیند , بر مقدار پیشنهادی k تکیه میکنند , همچنین اطمینان از پارتیشن های متوسط با استفاده از فرمول 2 محاسبه شده است. شکل 2 و شکل 3 نتایج به دست آمده از آزمایشات را به همراه 6 مقدار مختلف k به ترتیب با مقادیر 2,4,8,16,32 و 64 نشان می دهد. آزمایشات بیشتر با استفاده از دیگر مقادیر k انجام شده است که با توجه به محدودیت فضا , در این مقاله به آن اشاره نشده است. هر چند , با توجه به ارتباط بین زمان محاسباتی و

دقت مطرح شده در این آزمایشات , این موارد شباهت زیادی به موارد ارائه شده در شکل 2 و شکل 3 دارند. بنابراین , تهدید نسبت به اعتبار داخلی آزمایشات ما , قابل توجه نیست.

تهدید نسبت به اعتبار نتیجه . تهدید اصلی برای نتیجه گیری اعتبار این آزمایش , قابلیت اطمینان پارتیشن نهایی مجموعه داده به عنوان پارتیشن بهینه ی آن است پاسخ به دست آمده به منظور cluster کردن مسئله , NP-Hard است. k-means استفاده شده در آزمایشات , همان طور که در بخش III.A به آن اشاره شد , برای تقسیم پارتیشن بهینه تلاش می کند. بنابراین , پارتیشن نهایی لزوما پارتیشن بهینه نیست. بر اساس نتایج , شکل 3 لزوما دقت پارتیشن متوسط مجموعه داده نزدیک به پارتیشن مطلوب واقعی را نشان نمی دهد .

اگر چه , ما معتقدیم که پارتیشن نهایی , که با الگوریتم K-means ارائه شده در بخش III.A به دست آمده است , برای نشان دادن مفهوم long tail در فرایند clustering به اندازه ی کافی قابل اعتماد است. الگوریتم واقعی بهینه سازی k-means به احتمال زیاد زمان بیشتری را در بر گرفته و منجر به پدیده ی long tail قابل توجهی می شود. بنابراین , تهدید نسبت به اعتبار نتیجه ی آزمایشات ما وجود دارد , هر چند که این تهدید قابل توجه نیست.

به منظور انجام داده کاوی بزرگ در ابر با استفاده از منابع محاسباتی ارائه شده توسط vendor های ابر , بهره وری هزینه , موضوعی حیاتی در این رابطه می باشد که معمولا توسط جامعه ی پژوهشی نادیده گرفته می شود . در این تحقیق , ما به بررسی این موضوع با استفاده از الگوریتم K-means به عنوان یک مورد مطالعاتی پرداختیم. نتایج تجربی اهمیت بهره وری هزینه ی داده کاوی بزرگ در ابر را تایید می کنند. در این آزمایشات , الگوریتم k-means تنها 46.17% - 0.32% از کل زمان محاسبات را به منظور دستیابی به یک دقت 99% به کار می گیرد. این مقدار بالای 99.68% صرفه جویی در هزینه ی پولی با امتیاز دقت تنها 1% را داراست. در آینده , ما به بررسی بهره وری هزینه ی داده کاوی در ابر به همراه شاخص های دقت داخلی و دسته ای از الگوریتم های داده کاوی که به صورت گسترده مورد استفاده قرار گرفته اند خواهیم پرداخت.