تجمیع برنامه نویسی شبکه ی ژنتیک و مسئله ی کوله پشتی برای پشتیبانی از خوشه بندی رکورد در پایگاه های داده ی توزیع شده

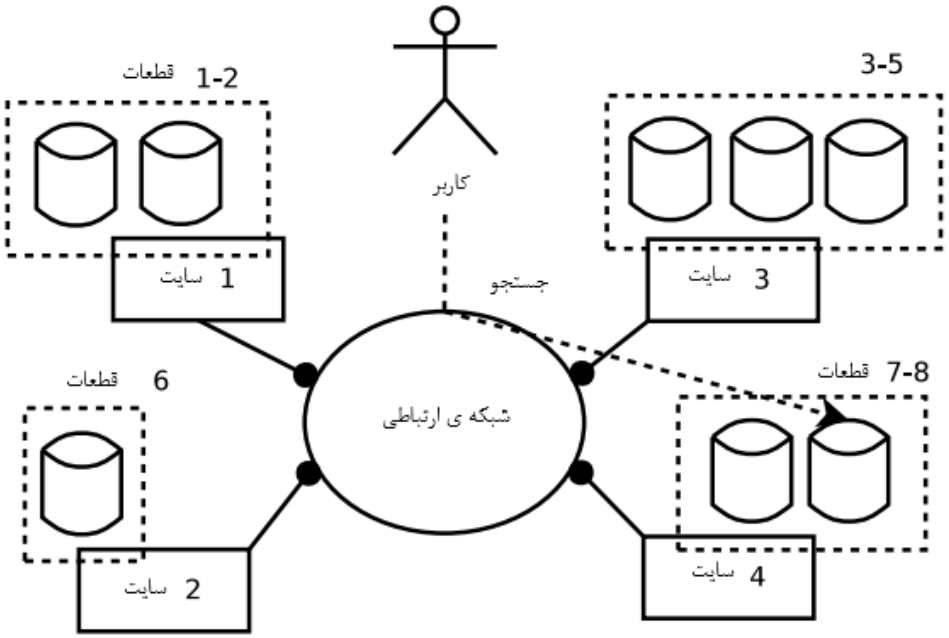
نکات برجسته

- یک الگوریتم پشتیبان تصمیم برای خوشه بندی رکورد در پایگاه های داده ارائه می شود
- مسئله ی محدودیت ظرفیت نشان داده می شود تا یک برنامه ی خوشه بندی کلی را ایجاد کند
- استخراج قاعده از مجموعه های داده از طریق الگوریتم تکاملی ارائه شده انجام می گیرد.
- خوشه بندی قاعده که محدودیت ظرفیت را در نظر می یگرد، بوسیله ی مسئله ی کوله پشتی حل می شود.
- شبیه سازی های خوشه بندی رکورد، برخی مزایای روش ارائه شده را نشان می دهد.

چکیده

این تحقیق شمال پیاده سازیِ برنامه نویسی شبکه ی ژنتیک (GNP) و برنامه نویسی پویای استاندارد به منظور حل مسئله ی کوله پشتی (KP) به عنوان سیستم پشتیبانی تصمیم برای خوشه بندی رکورد در پایگاه های داده ی توزیع شده می شود. تخصیص قطعه با مسئله ی محدودیت ظرفیت انباره، پیش زمینه ای برای روش پیشنهاد شده است. مسئله ی ظرفیت انباره برای توزیع مجموعه ها در چندین سایت (خوشه) است. مقدار کل قطعه ها در هر سایت نباید از ظرفیت سایت تجاوز کند، در حالیکه روند توزیع باید رابطه (تشابه) ی بین قطعه ها در هر سایت را حفظ کند. هدف، توزیع داده ی بزرگ بوسیله ی لحاظ کردن شباهتِ داده ی توزیع شده در هر سایت، در سایتهای مشخصی با مقدار محدود ظرفیت است. GNP برای حل این مسئله به کار گرفته می شود تا قواعد را بوسیله ی

لحاظ کردن مشخصاتِ (محدوده ی مقدار) هر ویژگی در یک مجموعه ی داده، استخراج کند. روش پیشنهاد شده، روش استخراج قاعده ی انتخاب تصادفی جزیی در GNP را ارائه می کند تا الگوهای متداول در یک پایگاه داده را برای بهبود الگوریتم خوشه بندی (خصوصا برای مسائل داده ی بزرگ) شناسایی کند. مفهوم KP برای مسئله ی ظرفیت انباره به کار گرفته می شود و برنامه نویسی پویای استاندارد بوسیله ی لحاظ کردن شباهت (مقدار) و مقدار داده (وزن) ی متناسب با هر قاعده برای قواعد توزیع استفاده می شود تا ظرفیت های سایت را تطبیق دهد. از نتایج شبیه سازی مشخص می شود که روش پیشنهاد شده، برتری هایی نسبت به الگوریتم های خوشه بندی مرسوم نشان می دهد و از این رو روش پیشنهاد شده، روش خوشه بندی جدیدی با مسئله ی ظرفیت انباره ی اضافی فراهم می کند.



شکل ۱. یک محیط پایگاه داده ی توزیع شده

کلمات کلیدی: برنامه نویسی شبکه ی ژنتیک، خوشه بندی پایگاه داده، مسئله ی کوله پشتی، خوشه بندی رکورد

1. دیباچه

سیستم مدیریت پایگاه داده ی توزیع شده (DDBMS) می تواند راه حلی برای سیستم های اطلاعاتیِ مقیاس بزرگ با مقادیر بزرگِ رشد داده و دسترسی داده باشد. پایگاه های داده ی توزیع شده (DDB) مجموعه ای از داده است که به طور منطقی متعلق به همان سیستم می باشد اما در سراسر سایت های شبکه ی کامپیوتر (شکل 1) پخش شده است. پس از آن DDBMS به عنوان یک سیستم نرم افزاری تعریف می شود که امکان مدیریت DDB را فراهم می کند و توزیع داده بین پایگاه های داده و نرم افزار را برای کاربران شفاف می کند.

روش های دسترسی کارآمد و تکنیک های ذخیره ی داده به طور فزاینده ای برای مدیریتِ تکثیر داده، در جهت قابل قبول نگه داشتن زمان پاسخِ جست و جو مهم شده اند. یک راه برای بهبود زمان پاسخ جست و جو، کاهش دادنِ تعداد I/O های دسیک از طریق خوشه بندی عمودی (خوشه بندی ویژگی) و/یا افقی (خوشه بندی رکورد) پایگاه داده است. بهبود در زمان بازیابیِ رکورد های چند ویژگی می تواند بدست آید اگر ثیت های مشابه در فضای فایل به صورت نتیجه ی بازسازی نزدیک به هم گره بندی شده باشند. این موضوع به خاطر این است که هر چقدر احتمال مقیم شدنِ دو هدف یا بیشتر در همان صفحه ی انباره کاهش پیدا کند، انتقال های صفحه ی کمتری مورد نیاز می باشد.

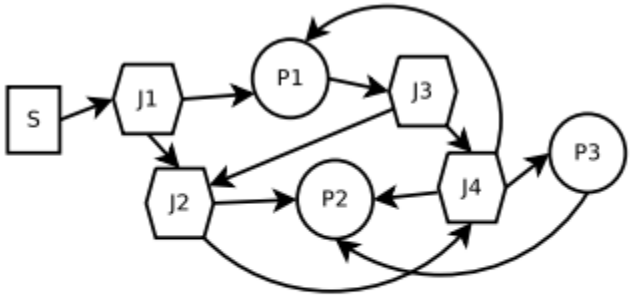


Fig. 2. Basic Implementation of GNP
S : start node, $[J_1,...,J_4]$: judgement node, $[P_1,...,P_3]$: processing node
is increased (Lowden & Kitsopanidis, 1993).

در این مقاله، یک روش جدیدکه برنامه نویسی شبکه ی ژنتیک و برنامه نویسی پویای استاندارد را برای خوشه بندی رکورد تجمیع می نماید تا مسائل کوله پشتی (KP) را حل کند، ارائه می شود. فرضیه ی این تحقیق این است که

پیاده سازیِ GNP برای داده کاوی می تواند خوشه های کارآمدی از مجموعه های داده ی پیچیده شده ایجاد کند و مفهوم KP می تواند با لحاظ کردن مقدار (شباهت داده) و جرم (اندازه ی داده) در DDBMS برای تعریف مسئله ی توزیع قطعات در سایت های متعدد استعمال شود. بنابراین، این روش می تواند راه حلی برای تخصیص قطعه و مسائل ظرفیت انباره ی سایت باشد.

این مقاله به صورت پیش رو سازمان دهی شده است: بخش 2 بررسی اجمالی از چارچوب پیشنهاد شده را تشریح می کند، بخش 3 مروری از ادبیات علمی ارائه می کند، بخش 4 الگوریتمِ دقیقِ چارچوب پیشنهاد شده را نشان می دهد، بخش 5 نتایج شبیه سازی را نشان می دهد و در نهایت بخش 6 به نتیجه گیری تخصیص داده شده است.

2. بررسیِ چارچوب پیشنهاد شده

2.1. برنامه نویسی شبکه ی ژنتیک

GNP یک تکنیک بهینه سازی تکاملی است که به جای رشته ها در الگوریتم ژنتیک یا درخت ها در برنامه نویسی ژنتیک از ساختارهای گراف مستقیم شده استفاده می کند. این کار منجر به تقویت توانایی ارائه با برنامه های فشرده می شود که از استفاده ی مجددِ گره ها در ساختار گراف استنتاج شده اند.

گره ها در GNP به عنوان واحد های کمینه ی داوری و عمل تفسیر می شوند. و انتقال گره، قواعد برنامه را ارائه می کند. GNP بعد از آغاز انتقال گره از گره شروع، در زمانی که فعالیت ها کامل شود به گره شروع باز نمی گردد. داوری و عمل بعدی همیشه تحت تاثیرِ انتقال گره قبلی می باشد. داوری و پردازشِ برنامه های GNP در سطح گره اجرا می شوند.

ساختار پایه ی GNP در شکل 2 نشان داده می شود که S گره شروع را معنی می دهد. دو نوع دیگر گره، گره های داور و گره های پردازش، به ترتیب توابع داوری Jp و پردازش Pq را دارند. Jp ، p امین تابع داوریِ ذخیره شده در کتابخانه برای گره های داوری را نشان می دهد در حالی که $P_q\ (q = 1, \ldots, m)$، q امین تابع پردازش ذخیره شده در یک کتابخانه برای گره های پردازش را نشان می دهد.

GNP در این مقاله، از طریق تحلیل رکورد ها برای اداره کردنِ قواعد استخراج از مجموعه های داده استفاده می شود. هر گره داوری یک ویژگی با محدوده ی مقدار را ارائه می کند. برای مثال، ویژگی قیمت می تواند به سه محدوده (پایین، وسط، بالا) تقسیم شود و یک محدوده به یک گره داوری تخصیص داده می شود. GNP قواعد را از طریق تکامل دادن تجمیع گره ها ایجاد می کند و پوشش قواعد استخراج شده را اندازه گیری می کند. پوشش بدین معنا است که هر قاعده چه میزان رکورد در یک مجموعه ی داده می تواند ارائه کند (پوشش). قواعدی که حداقل یک رکورد را پوشش می دهند، در مخزن قواعد ذخیره خواهند شد و بعد از آن قواعد ذخیره شده به منظور کاربردی برای فاز KP، در چندین سایت توزیع می شوند. هدف این مقاله توزیع قواعد است نه داده که با لحاظ کردنی تشابهات بین قواعد و داده در توزیع هر گونه داده در سایت ها مشارکت می کند. توضیحِ پیاده سازی GNP در استخراج قاعدهف به تفصیل در بخش 4.1 قابل دسترس خواهد بود.

2.2. مسئله ی کوله پشتی

KP یک مسئله ی بهینه سازی ترکیبی است که مجموعه ای از اقلام را مدیریت می کند. هر کدام از این اقلام با یک جرم و مقدار هستند که شماره ی هر قلم را برای شامل شدن در یک مجموعه مشخص می کند به طوریکه وزن کل کمتر یا برابر با محدودیت داده شده است و مقدار کل تا جای ممکن بزرگ است. KP به صورت پیش رو تعریف می شود:

بیشینه کردن $S = \sum_{i=1}^{n} v_i x_i$، مشروط به $\sum_{i=1}^{n} w_i x_i \leq W$

که S برابر با مقدار کل کوله پشتی (سایت)؛ i شماره ی قطعه $(1 \leq i \leq n)$؛ xi تعداد قطعات i ؛ vi مقدار (تشابه به قاعده ی رهبرِ سایت) قطعه ی i ؛ wi وزن (اندازه ی داده) قطعه ی i؛W برابر با ظرفیت سایت است.این بهینه سازی با اجازه دادن به هر قطعه (قلم) برای بیش از یک بار اضافه شدن به سایت ها، می تواند مسئله ی تکرار را مدیریت کند.

مسئله ی کوله پشتی در این مقاله از طریق برنامه نویسی پویای استاندارد برای مسئله ی کوله پشتی 0/1 حل می شود. به ما اجازه دهید تا آرایه ای دو بعدی $m[i,w]$ را با i ردیف و w ستون تعریف کنیم. $m[i,w]$ مقدار کوله پشتی را در زمان لحاظ کردن اقلام با شماره قلم $1,2,\ldots,i-1,i$ را نشان می دهد و وزن کلی انها $w.\ m[i,w]$ بوسیله ی معادله ی 2 محاسبه می شود.

$$m[i,\ w] = m[i-1,\ w] \ \text{ if } \ w_i > W$$
$$m[i,\ w] = \max(m[i-1,\ w],\ m[i-1, w-w_i] + v_i) \ \text{ if } \ w_i \leqslant W. \qquad (2)$$

قدم اول محاسبه کردن $m[0,w]$ است، سپس $m[1,w]$ بر مبنای مقادیر $m[0,w]$ محاسبه می گردد. روندی مشابه برای محاسبه ی $m[2,w],\ldots,m[n,w]$ تکرار می شود. بعد از اتمام محاسبه ی $m[i,w]$، مقدار بیشینه در میان همه ی $m[n,w](0 \leq w \leq W)$ به عنوان پاسخ مسئله انتخاب می شود.

در ان تحقیق، برنامه نویسی پویای استاندارد به کار گار گرفته شده تا KP را حل کند و توزیع قواعد توزیع شده بوسیله ی GNP در هر سایت را اداره نماید. قواعدی که با پوشش داده ی زیاد هستند، رهبرهای هر سایت خواهند بود و کاربردِ KP، تشابه بین قواعد رهبر و قواعد باقی مانده را در نظر می گیرد (که به عنوان مقدار قلم (قاعده) در KP در نظر گرفته می شود) و پوشش قواعد (که به عنوان وزن در KP در نظر گرفته می شود) باید مطابق با ظرفیت های سایت باشد.بنابراین، قواعد مشابه به یک قاعده ی رهبر، اساسا در یک سایت قرار داده می شوند. توضیحات پیاده سازی کاربرد KP در توزیع قاعده به تفصیل در بخش 4.2. قابل دسترسی است.

3. برررسی ادبیات علمی

روش پیشنهاد شده از الگوریتم GNP برای داده کاوی استفاده می کند که در Mabu ارائه شده است و روش ارائه شده برای مسئله ی ظرفیت انباره ی تخصیص قطعه در پایگاه های داده ی توزیع شده ای که در Ozsu and Valduriez نشان داده شده به کار گرفته می شود. این تحقیق شامل پیاده سازی برنامه نویسی شبکه ی ژنتیک (GNP) برای داده کاوی و برنامه نویسی پویای استاندارد می شود تا مسئله ی کوله پشتی (KP) را برای قاعده ای که

بر مبنای خوشه بندی است، حل کند. مسئله ی ظرفیت انباره، خوشه بندی پایگاه داده را معرفی می کند و معرفی مفهوم KP برای حل کردنِ مسئله یکی از نکات منحصر به فردِ روش ارائه شده است. علاوه بر این، روش ارائه شده، گزینش تصادفی جزیی ویژگی را در استخراج قواعد فراهم می کند که می تواند الگوهای متداول در پایگاه داده را شناسایی کرده و کیفیت خوشه بندی را بهبود ببخشد. روش ارائه شده با توجه به ویژگی های بالا یک خوشه بندی رکورد خودکار ارائه می کند که قصد دارد تا یک سیستم پشتیبانی تصمیم برای خوشه بندی رکورد در پایگاه های داده باشد.

ادبیات علمی کنونی مرتبط به تخصیص قطعه Rahimi,Parand and Riahi است. این تحقیق یک رویکرد ارائه می کند که به طور همزمان به طور عمودی قطعات داده ایجاد می کند و در سایت قطعات را به سایت های مناسب تخصیص می دهد. الگوریتم انرژی پیوند (BEA) یا اندازه ی همبستگی بهتری به کار گرفته می شود که کیفیت خوشه های ویژگی ایجاد شده را بهبود می بخشد. BEA می تواند از طریق شناسایی اقلام متداول بین رکورد ها در پایگاه داده روابط خوبی بین ویژگی ها یافته شوند. روش پیشنهاد شده مجموعه های الگوی متداول را نیز شناسایی می کند اما برای شناسایی یک قطعه بندی افقی خودکار یا خوشه بندی رکورد است نه برای قطعه بندی عمودی (همان طور که بوسیله ی این ادبیات علمی ارائه شده).

عنوان خوشه بندی مرتبط کنونی یک یادگیری وزن ویژگی خودکار است که بوسیله ی Saha and Das ارائه شده است. این مقاله نوعی جدید از الگوریتم خوشه بندی حالات k فازی را برای داده ی مطلق با یادگیری وزن ویژگی خودکار، ارائه و بررسی می کند. این روش به طور خودکار با وزن های زیاد ویژگی ها همراه است که در شناسایی الگوهای خوشه بندیِ داده در الگوریت حالات k فازیِ مطلق مفید هستند. روش پیشنهاد شده در این مقاله مجموعه های الگوی متداول برای ویژگی ها (خصوصیات) را نیز شناسایی می کند تا عملکرد خوشه بندی را بهبود ببخشد که در بخش 4.1.3. توضیح داده می شود و ضمنا، روش ارائه شده می تواند مسئله ی ظرفیت انباره را اداره کند که در این ادبیات علمی حل نشده است.

تنظیمات مناسب انقلابیِ سیستم های یادآوری معنایی خودکار موضوع مرتبط دیگری است که بوسیله ی Cuzzola, Jovanovic, Bagheri, & Gasevic ارائه شده است. این مقاله یک ساختار تنظیم پارامتر (PTA) برای خودکار کردنِ کار تنظیم کردن مقادیر پارامتر ابزار یادگیری معنایی با محاسبات تکاملی ارائه می کند. استفاده از محاسبه ی تکاملی در جهت یافتن تجمیع های مناسبِ ویژگی ها برای حل کردن مسئله و استفاده از گزینش وزن ویژگی، تشابه با روش ارائه شده است. اما مشکل این مقاله (یادگیری معنایی) متفاوت از روش ارائه شده در این مقاله است. مسئله ی هدفی این مقاله یک خوشه بندی رکورد با یک مسئله ی محدودیت ظرفیت انباره ی اضافی است.

4. تجمیع مسئله ی کوله پشتی و GNP

پیاده سازی خوشه بندی رکورد به دو بخش تنظیم می شود: استخراج قاعده ی GNP و توزیع قاعده بر مبنای برنامه نویسی پویای استاندارد برای حل کردن مسئله ی کوله پشتی که در بخش 4.1 و 4.2 توضیح داده می شود. علاوه بر این، تحلیل پیچیدگیِ روند خوشه بندی کل در بخش 4.3 تشریح می شود.

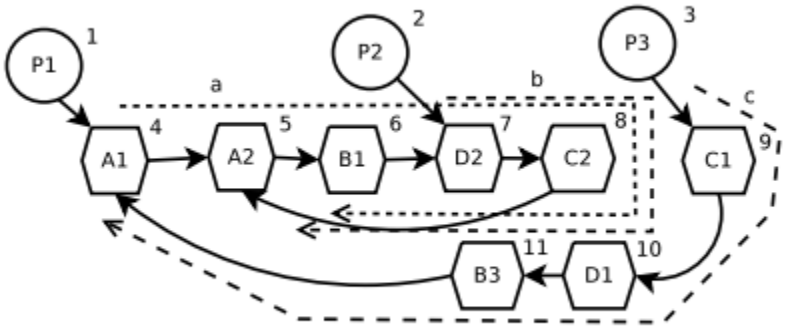


Fig. 3. GNP Implementation on Data Mining

4.1. استخراج قاعده ی GNP

GNP برای استخراج قواعد از پایگاه داده، بوسیله ی تحلیلِ ساختار پایگاه داده به کار گرفته می شود و شامل موارد زیر می شود:

مقدار ویژگی ها: تعداد ویژگی ها در مجموعه ی داده. هر ویژگی به چندین گره تقسیم می شود که وابسته به تنوع و محدوده های مقدار (فاصله ی بین مقدار کمینه تا مقدار بیشینه) آن هستند..

مقدار داده: تعداد رکوردها در مجموعه ی داده

تنوع داده: رکوردهای مختلف تا چه میزان در مجموعه ی داده موجود هستند. اگر همه ی رکوردها در مجموعه ی داده مختلف باشد، تنوع 100٪ است، اگر نصف این رکوردها در این مجموعه ی داده مختلف باشند، تنوع 50٪ . است و اگر همه ی رکوردها در مجموعه ی داده یکسان باشند، تنوع برابر با 1 روی ضرب تعداد داده در 100٪ است. برای مثال، در جدول 4 که در صفحه ی بعد نشان داده خواهد شد، شش نوع داده در مجموع 310 داده وجود دارد پس تنوع برابر است با: $(6/300) \times 100 = 1.94\%$.

GNP برای استخراج قواعد از مجموعه داده از طریق تحلیل همه ی رکوردها به کار گرفته می شود. ساختارهای ژنوتیپ و فنوتیپِ GNP به ترتیب در شکل 3 و جدول 1 تشریح می شوند. هر گره در شکل 3 دارای شماره ی گره (1-11) مخصوص به خودش است و در جدول 1، اطلاعات گره برای هر شماره گره تشریح می شود. اندازه ی برنامه وابسته به تعداد گره ها است که روی مقدار قواعد ایجاد شده بوسیله ی برنامه تاثیر می گذارد.

گره داوری در پیاده سازی داده کاوی، یک ویژگی مجموعه داده را ارائه می کند که بوسیله ی Ai ارائه می شود و یک شاخص ویژگی مانند قیمت، سهام و غیره را نشان می کند و Ri شاخص محدوده ی مقدار ویژگی را نشان می دهد. برای مثال، Ai=A نشان دهنده ی ویژگی قیمت است و Ri=1 محدوده ی مقدار [0,50] و Ri=2 محدوده ی [51,80] را نشان می دهد. گره های پردازش، نقطه ی شروعِ بخش گره های داوری را نشان می دهد که از طریق اتصالشان به صورت بخش بخش اجرا می شود. بخش های گره ها که از هر گره پردازش شروع می شوند (P_1, P_2, P_3) از طرقی خط نقطه چین a,b,c ارائه می شوند. یک بخش گره جریان می یابد تا زمانی که پشتیبان برای تجمیع بعدی، آستانه را ارضا نمی کند. گره ها با ویژگی هایی که پیش از این در بخش ظاهر شده اند، کنار گذاشته خواهد شد. قواعد کاندید استخراج شده بوسیله ی برنامه ی شکل 3 از مجموعه داده ی جدول 2 در جدول 3 نشان داده می شود. در جدول 3، سه قاعده بوسیله ی بخش گره از هر گره پردازش استخراج می شود.

جدول ۱
ساختار ژن جی ان پی شکل ۳

i	NT_i	A_i	R_i	C_i
1	1	0	0	4
2	1	0	0	7
3	1	0	0	9
4	2	A	1	5
5	2	A	2	6
6	2	B	1	7
7	2	D	2	8
8	2	C	2	5
9	2	C	1	10
10	2	D	1	11
11	2	B	3	4

i : تعداد گره

NT_i : انواع گره- ۱. پردازش ۲. داوری

A_i : شاخص ویژگی

R_i : شاخص محدوده ی ویژگی

C_i : ارتباط

امتیاز قاعده به صورت زیر تعریف می شود:

$$\begin{cases} 0 \text{ if } sup(r) = 0 \\ 10 * sup(r) + 10 * (n_{con}(r) - 1) \text{ if } sup(r) > 0, \end{cases} \quad (3)$$

امتیاز قاعده ی r =

که $sup(r)$ پشتیبان قاعده ی r و $n_{con}(r)$ طول قاعده ی r است.

تناسب برای ارزیابی یک مورد به صورت زیر تعریف می شود:

$$\text{Fitness} = \sum_{r \in R} \{sup(r) + 10(n_{con}(r) - 1) + \alpha_{new}(r)\}, \quad (4)$$

اگر قاعده ی r به تازگی استخراج شده باشد، $\alpha_{new}(r)$ یک مقدار اضافی است

جدول ۲

مثال از مجموعه داده

A_1	A_2	B_1	D_2	C_2	C_1	D_1	B_3
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
0	1	0	1	1	0	0	0
0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0
1	0	0	0	0	1	1	1

جدول ۳

مثال از مجموعه داده و پشتیبانش برای قواعد استخراج شده

گره های پردازش	قواعد استخراج شده	پشتیبان	Score	
			Rule	Template
1	$A_1 \wedge B_1$	3/6	15.00	6.00
	$A_1 \wedge B_1 \wedge D_2$	1/6	21.66	3.67
	$A_1 \wedge B_1 \wedge D_2 \wedge C_2$	1/6	31.66	4.67
2	$D_2 \wedge C_2$	2/6	11.66	4.33
	$D_2 \wedge C_2 \wedge A_2$	1/6	21.66	3.67
	$D_2 \wedge C_2 \wedge A_2 \wedge B_1$	0/6	0	0
3	$C_1 \wedge D_1$	2/6	13.33	4.33
	$C_1 \wedge D_1 \wedge B_3$	1/6	21.66	3.67
	$C_1 \wedge D_1 \wedge B_3 \wedge A_1$	1/6	31.66	4.67
			199.95	

امتیاز الگو در بخش ۴.۱.۳ ارائه میشود

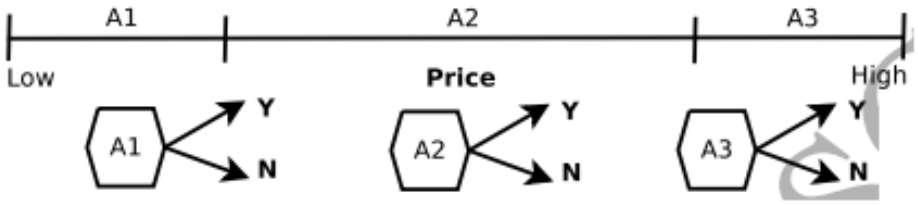
جدول 3 طول و پشتیبانِ قواعد استخراج شده را نشان می دهد. امتیاز قاعده که بوسیله ی معادله ی 3 نشان داده شده، نه تنها از طریق پشتیبانش $(sup(r))$ محاسبه می شود بلکه از طریق طول آن $(n_{con}(r))$ نیز محاسبه می گردد. لحاظ کردنِ طول قاعده، قواعد را اطمینان پذیر تر می کند چون قواعد بلندتر می تواند ترکیب های متنوعی از ویژگی ها را پوشش دهد. برای مثال، $A_1 \wedge B_1$ دارای پشتیبان نسبتا بالایی 3/6 است اما تنها طول دو

را دارد، پس امتیاز قاعده تنها 15.00 است. از سوی دیگر، $C_1 \wedge D_1 \wedge B_3 \wedge A_1$ تنها پشتیبان 1/6 است اما طول چهار است. بنابراین، امتیاز 31.66 می شود. $\alpha_{new}(r)$ نیز در تناسب وجود دارد چون هدف استخراج قاعده، شناسایی کردنِ قواعد جدید از یک مجموعه داده تا جای ممکن است.

آماده سازی گره برای استخراج قاعده ی GNP از دو فاز تشکیل می شود: تعریف گره و ترتیب گره. ضمنا، دو نوع از روش های ترتیب گره ارائه می شود: یکی ترتیب انتخاب تصادفی کامل است و دیگری ترتیب انتخاب تصادفی جزیی است.

جدول ۴
مثال از جدول تناوب ویژگی های قیمت

x	f	xf
10	30	300
25	25	625
50	30	1500
80	140	11200
100	65	6500
150	20	3000
Total	310	23125



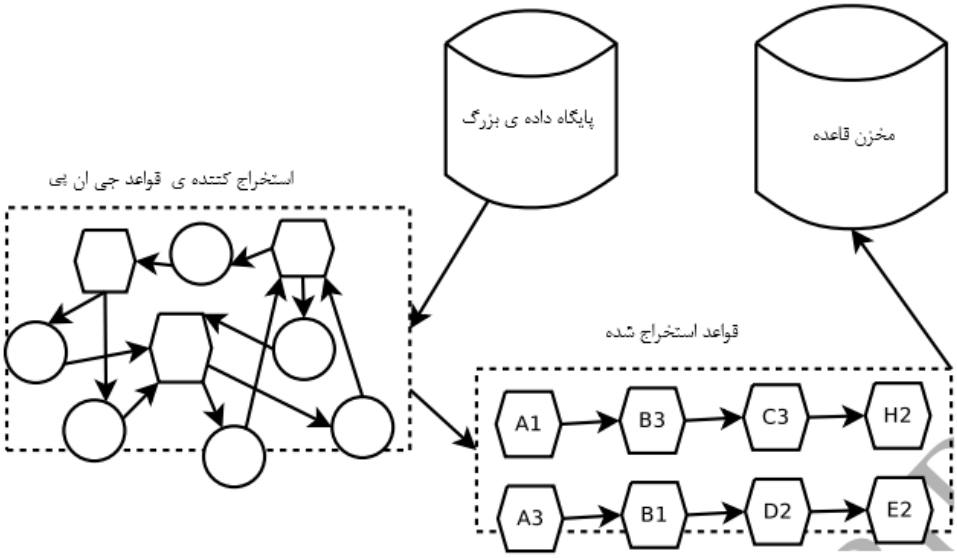
شکل ۴. گره برای قضاوت ویژگی ها

4.1.1. تعریف گره

هدف اصلی از تعریف گره، آماده کردنِ گره های داوری است که برای ایجاد قواعد ترکیب می شوند. قدم اول یافتن مقادیر کمینه و بیشینه ی هر ویژگی است. برای مثال، در مجموعه داده ای با 310 رکورد، مقدار کمینه ی ویژگی قیمت 10 و مقدار بیشینه 150 می باشد. در ادامه، همان طور که در شکل 4 نشان داده شده، یک جدول تناوب به ازای هر ویژگی ایجاد می شود. x قیمت محصول ا نشان می دهد و f تعداد دفعات رکورد شدن محصول با همان قیمت در مجموعه داده را نشان می دهد. برای مثال، محصول (ها) با قیمت x=10 ، 30 دفعه ظاهر شده است. سپس، مقدار میانگین $(\overline{xf})$ از طریق معادله ی 5 محاسبه می شود.

$$\overline{xf} = \frac{\sum xf}{\sum f} = 74.60 \qquad (5)$$

داده برای تعریف گره ها از جدول 4 باید برمبنای مقدار داده به طور مساوی تقسیم شده باشد. برای مثال، همان طور که در شکل 3 نشان داده شده، بوسیله ی تقسیم محدوده ی مقدار و با در نظر گرفتن تناوب رخداد، می توان سه گره ایجاد کرد . در این مثال، سه محدوده برابر هستند با $x = \{10, 25, 50\}\ (85\ \text{data})$، $x = \{80\}\ (140\ \text{data})$ و $x = \{100, 150\}\ (85\ \text{data})$. گره اول و گره سوم بیشتر از یک قیمت را شامل می شوند چون هر تک رکورد $(10,25,50,100,150)$ تناوب کافی برای تعریف شدن به عنوان گره را ندارد. میانگین $(\overline{xf} = 75.42)$ برای اندازه گیری پوشش کمینه برای گره شدن استفاده می شود. گره دوم از طریق اندازه گیری می تواند از تک رکورد $(x = \{80\})$ ایجاد شود چون $f = 140$ از $\overline{xf}$ تجاوز می کند.



شکل ۵. استخراج قاعده ی جی ان پی

4.1.2. ترتیب گره: انتخاب تصادفی کامل

هدف ترتیب گره، انتخاب گره های لازم برای استخراج کارآمدِ تعداد زیادی قاعده است. روش انتخاب تصادفی کامل، به طور تصادفی گره ها را از گره های تعریف شده در بخش 4.1.1. انتخاب می کند و ساختارهای گراف را ایجاد می کند. GNP از ساختارهای گراف ایجاد شده، تعداد زیادی از قواعد مهم را استخراج می کند و آنها را در مخزن قاعده قرار ذخیره می کند (شکل 5). چارچوب اصلیِ استخراج قاعده به تفصیل در Shimada et al تشریح می شود.

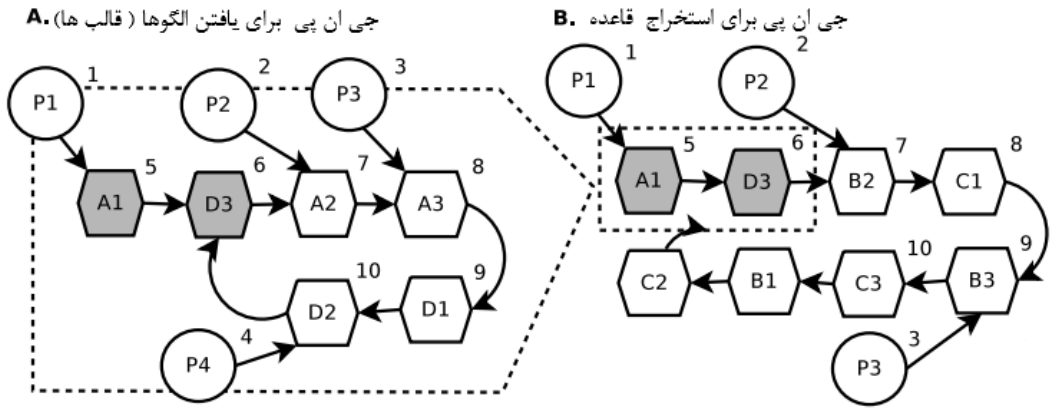
GNP بعد از اینکه قواعد استخراج می شوند، مقدار پوشش بدست آمده بوسیله ی قواعد را اندازه گیری خواهد کرد. در این تحقیق، پوشش قاعده ی r به معناب تعداد رکوردهایی است که با قاعده ی r تطتابق می کند (بوسیله ی آن پوشش داده می شود). اگر یک قاعده حداقل یک داده را پوشش داده باشد، این قاعده به مخزن قاعده اضافه می شود، در غیر اینصورت، قاعده حذف می شود. قواعد با پوشش بالا به عنوان قواعد الیت تعریف خواهد شد و رهبرانِ هر خوشه (سایت) در روند KP می شوند. روند استخراج قاعده تا زمانی که همه ی رکوردها در یک مجموعه داده پوشش داده شوند، ادامه پیدا می کند.

همگذری و دگرگونی برای ایجاد تعداد زیادی از قواعد خوب استخراج می شوند.

همگذری: تبادل یگ گره یا بیشتر بین والدین برای ایجاد قواعد جدید

دگرگونی: تغییر یک گره یا بیشتر برای ایجاد ترکیبات مختلف از گره ها

همگذری برای تعویض گره های ضعیفِ والدین (گره هایی با تناوب داده ی کمتر) با گره های قوی (گره هایی با تناوب داده ی بیشتر) کارآمد است. دگرگونی برای تعویض گره های ضعیفِ منفرد با گره های قوی کارآمد است.



شکل ۶. بهینه سازی ترتیب گره در جی ان پی

4.1.3. ترتیب گره: انتخاب تصادفی جزیی

روش انتخاب تصادفی جزیی دارای دو روند ترتیبی GNP می باشد. روند اول، یافتن قواعد الگو و روند دوم اجرای استخراج قاعده ی کلیِ GNB که با الگوهای ایجاد شده در روند اول ترکیب شده اند ، می باشد. الگوها برای بدست آوردن ترکیباتِ ویژگی هایی که متداولا در مجموعه داده اتفاق می افتند، استخراج می شوند. امتیاز الگو بوسیله ی معادله ی 6 محاسبه می شود و الگوهایی که با امتیاز بالا هستند، در روند دومو استفاده می شوند.

$$\text{Score of template } t = \begin{cases} 0 \text{ if } sup(t) = 0 \\ 10 * sup(t) + (n_{con}(t) - 1) \text{ if } sup(t) > 0 \end{cases} \quad (6)$$

امتیاز الگو، برخلاف امتیاز قاعده (معادله ی 3) که وزن بیشتری به طول گره می دهد، همان ئطور که بوسیله ی معادله ی 6 نشان داده شد، به پشتیبان وزن بیشتری می دهد. برای مثال، امتیازهای الگوها در جدول 3 نشان داده می شوند که نتیج در آن نسبتا مخالفِ امتیاز قاعده است. اگر چه طول گره تنها 2 است، اما $A_1 \wedge B_1$ بالاترین

امتیازِ الگو را دارد. انتخاب تصادفی جزیی در زمانی که $A_1 \wedge B_1$ به عنوان یک الگو استفاده می شود، بوسیله ی انتخاب تصادفی ویژگی های باقی مانده مانند C و D پیاده سازی خواهد شد.

در روند استخراج الگو تنها تعداد کمی از ویژگی ها در استخراج قاعده ی GNP قرار دارند. این روند قصد دارد تا احتمالِ بدست آوردن الگوهایی با پشتیبان بالا را افزایش دهد. برای مثال، ترکیبِ ویژگیِ A و D به عنوان الگویی منتج شده از محاسبات امتیاز(معادله ی 6) ، در "A. یافتن الگو"، در شکل 6، تعریف می شود. این موضوع، احتمال یافتن ترکیبات خوب با ویژگی های A و D را افزایش خواهد داد. در "B. استخراج قاعده"، الگو و ویژگی های باقی مانده (که B و C هستند) لحاظ شده اند. روند استخراج قاعدهمیتوان قواعد را با طول بلندتری از الگوها بدست آورد.

شکل ۵
مثال ترکیب قالب ها با ویژگی های باقی مانده

قالب ها	ویژگی های باقی مانده	پوشش	امتیاز قاعده
$A_2 \wedge D_3$	$B_1 \wedge C_2$	0	0
$A_2 \wedge D_3$	$B_3 \wedge C_2$	10	40.4
$A_1 \wedge D_3$	B_3	24	34.5
$A_3 \wedge D_3$	$B_1 \wedge C_2$	14	40.5

جدول 5، مثال ساده ای از انتخاب تصادفی جزیی را برای توضیحات نشان می دهد. هر الگو شامل ویژگی A و D می شود و با ویژگی های باقی مانده (B و C) ترکیب می شوند. قاعده ی تولید شده ی $A_3 \wedge D_3 \wedge B_1 \wedge C_2$ بالاترین امتیازی قاعده (معادله ی 3) را بدست می آورد چون دارای طول قاعده ی بلند و پوشش زیاد است.

4.2. توزیع قاعده بر مبنای برنامه نویسی پویای استاندارد برای حل کردن مسئله ی کوله پشتی

بعد از اینکه همه ی رکوردها از طریق قواعد استخراج شده بوسیله ی GNP در مجموعه داده شناسایی می شوند، برنامه نویسی پویای استاندارد برای حل مسئله ی کوله پشتی استفاده می شود تا قواعد را در سایت های متعدد توزیع کند. قواعدی که پوشش بالای دارند (الیت)، رهبرهای هر سایت خواهند شد، در ادامه، کاربرد، شباهت های

قواعد باقی مانده نسبت به قواعد رهبر (مقدار) و پوشش قواعد (وزن) را به منظور توزیع قواعد باقی مانده در سایت ها در نظر می گیرد. تشابه قاعده ی باقیمانده r1 نسبت به قاعده ی رهبر r2 بوسیله ی معادله ی 7 محاسبه می شود.

$$S(r_1, r_2) = \frac{N_{match}(r_1, r_2)}{Max\{N_{ante}(r_1), N_{ante}(r_2)\}} \qquad (7)$$

$S(r_1, r_2)$ برابر با تشابه بین قاعده ی r1 و r2 ، $N_{match}(r_1, r_2)$ برابر با تعداد ویژگی های تطبیق داده شده بین r1 و r2 و $N_{ante}(r)\ (r \in \{r_1, r_2\})$ تعداد ویژگی ها در قاعده ی r می باشد.

$Max\{N_{ante}(r_1), N_{ante}(r_2)\}$ بدین معنی است که طول قاعده ی بلندتر تبدیل به یک تقسیم کننده برای تعداد ویژگی های تطبیق داده شده بین دو قاعده می شود $(N_{match}(r_1, r_2))$. در زمانی که قاعده ی بلندتر شامل ویژگی هایی می شود که در قاعده ی کوتاه تر قرار ندارد، این ویژگی ها تطبیق داده شده فرض می شوند. مثال های محاسبه ی تشابه در جدول 6 نشان داده می شود. قاعده ی 2 از جدول 6 ، بالاترین تشابه به رهبر را نشان می دهد. قاعده ی رهبر دارای ویژگی D نیست پس هر ویژگی D در قواعد باقی مانده، تطبیق داده شده فرض می شود.

جدول ۶
مثال محاسبه ی تشابه بین قواعد رهبر وباقی مانده

Rule	A	B	C	D	$N_{match}(r_1, r_2)$	$S(r_1, r_2)$
Leader	A_1	B_3	C_2	-	-	-
1	$*A_1$	B_2	C_1	$*D_2$	2	$2/4$
2	A_2	$*B_3$	$*C_2$	$*D_1$	3	$3/4$
3	$*A_1$	B_1	$*C_2$	-	2	$2/3$

$*$ ویژگی تطبیق داده شده

4.3. تحلیل پیچیدگی

روندهای اصلیِ روش ارائه شده با تحلیل پیچیدگی آنها به صورت زیر خلاصه می شوند:

1) بخش استخراج قاعده

a) تعریف گره: این روند گره های داوری را آماده می کند که برای ایجاد قواعد تجمیع خواهند شد. پیچیدگی در این روند متناسب به تعداد داده و ویژگی ها است. تعداد زیاد ویژگی ها روی تعداد گره هایی که باید تعریف شوند تاثیر می گذارد. تعداد زیاد داده روی ویژگی های پیچیدگیِ ایجاد یک جدول تناوب به ازای هر ویژگی تاثیر می گذارد.

b) ترتیب گره: این روند، گره های لازم را برای استخراج کارآمدی تعداد زیادی از قواعد، انتخاب می کند. پیچیدگی در این روند متناسب به تعداد ویژگی ها است. تعداد زیادِ ویژگی ها روی تعداد ترکیبات محتملِ ویژگی هایی که می توانند استخراج شوند تاثیر می گذارد. روند استخراج قاعدهتا زمانی که همه ی داده در یک مجموعه داده شناسایی شود، ادامه می یابد، بنابراین، تعداد زیادِ ترکیبات محتمل نیازمندِ تکرر بیشتر هستند تا همه ی داده را پوشش دهد. روش انتخاب تصادفیِ جزیی برای اداره کردنِ کارآمدِ این پیچیدگی طراحی می شود تا الگوی مرسوم را با پوشش بالا نگه دارد تا در تکرار بعدی استفاده شود.

c) اندازه گیری قواعد استخراج شده: این روند، پوشش بایگانی شده بوسیله ی قواعد استخراج شده رااندازه گیری می کند. پیچیدگی در این روند متناسب به تعداد داده است. تعداد زیاد داده روی تعداد روندهای اندازه گیریِ هر قاعده تاثیر می گذارد.

2) بخش توزیع قاعده: برنامه نویسیِ پویای استاندارد برای حلی مسئله ی KP استفاده می شود. بدین معنی که، قواعد استخراج شده در چندین خوشه با لحاظ کردنِ تشابه بین قواعد (مقدار) و پوشش قواعد (وزن) توزیع می شوند. هر خوشه نمی تواند همه ی قواعد را در زمانی که جمع پوششِ قواعد از محدودیت ذخیره تجاوز می کند، ذخیره نماید. پیچیدگی در این روند متناسب به تعداد قواعد و خوشه ها و محدودیت های ذخیره ی هر خوشه است. تعداد زیادِ قواعد، پیچیدگی را بوسیله ی ترکیبات محتملِ توزیع قاعده افزایش می دهد در حالیکه تعداد زیادِ خوشه ها و محدودیت ذخیره ی کم نیز پیچیدگی را از طریق ترکیب چندین هدفِ روند توزیع افزایش می دهد.

جدول ۷

امتیاز میانگین قواعد

Crossover rate	Average Score of Rules	تکرار
0.01	20.31	28
0.05	20.29	25
0.1	20.24	23
0.2	20.12	23
0.5	19.78	22

جدول ۷

جدول ۸

مقایسه نرخ دگرگونی

Mutation rate	Average Score of Rules	Iteration
0.01	20.29	28
0.05	20.13	26
0.1	19.98	24
0.2	18.45	20
0.5	14.34	18

5. شبیه سازی ها

اول، روش های انتخاب تصادفی کامل و انتخاب تصادفی جزیی در استخراج قاعده ی GNP مقایسه می شوند. سپس، توزیع قاعده ی کوله پشتی انجام داده می شود و نتایج آن بررسی می گردند. در نهایت، شبیه سازی های خوشه بندی که شش مجموعه داده ی دانلود شده از مخزنِ یادگیری ماشین UCI را استعمال می کنند، انجام داده می شوند و نتایج آنها با پنج الگوریتم خوشه بندی مرسوم مقایسه می شوند.

5.1. استخراج قاعده ی GNP

استخراج قاعده ی GNP در این زیر بخش انجام داده می شود و تحلیل پارامتر برای نرخ همگذری و نرخ دگرگونی برای یافتن پارامترهای بهینه اجرا می گردد سپس عملکرد دو روشِ ترتیب گره مقایسه می گردند.

جدول ۹
نتایج استخراج قاعده ی جی ان پی با نتخاب کاملا تصادفی در شش مجموعه داده

Attr	انتخاب کاملا تصادفی					انتخاب تصادفی جزئی				
	Itr	n	Rule	Cvrg	Score	Itr	n	Rule	Cvrg	Score
3	34	2.33	34	29	10.15	25	3.00	39	23	20.29
3	78	2.23	12	808	10.01	45	3.00	18	526	20.02
8	564	3.45	23	42	20.23	435	6.62	43	22	50.26
8	1056	2.76	52	182	10.07	786	5.43	57	145	40.13
15	6290	2.46	34	20	10.15	5987	7.35	45	21	60.23
15	987	4.23	12	833	30.02	789	11.25	8	1110	100.03
					90.63					290.96

Attr : تعداد ویژگی ها Itr : تعدا تکرار ها برای پوشش همه رکوردها
n : طول میانگین برای هر قاعده Rule : تعداد میانگین قاعده
Cvrg : پوشش میانگین Score : امتیاز میانگین قواعد

5.1.1. تحلیل پاراکترِ نرخ همگذری و نرخ دگرگونی

پارامترهای اصلیِ روش ارائه شده که روی کیفیت قواعد استخراج شده و زمان تکرار تاثیر می گذارد، نرخ همگذری و نرخ دگرگونی هستند. بنابراین، مقایسه ی چندین تنظیمات پارامترِ نرخ همگذری و نرخ دگرگونی با استفاده از مجموعه های داده با سه ویژگی و 1000 نمونه استخراج می شود.

جدول 7 امتیاز میانگینِ قواعد و تکرارهای مورد نیاز را برای پوشش دادن همه ی داده ها در زمانی که نرخ همگذری در مقادیر متعددی تنظیم می شود، نشان داده است. جدول 7 نشان می دهد که افزایش نرخ همگذری، زمان تکرار را اندکی کاهش می دهد و میانگینِ امتیاز میانگینِ قاعده را کاهش می دهد. نرخ همگذری 0.01 در این مقاله استفاده می شود تا بهترین امتیاز میانگین قواعد را با اینکه افزایش زمان تکرار اندک است، بدست آورد. بهرحال، امتیاز میانگین قواعد چندان وابسته به نرخ همگذری نیست، از این رو، عملکرد روش ارائه شده می تواند پایدار باشد.

جدول 8 مقایسه ای مشابه به جدول 7 را در زمانی که نرخ دگرگونی روی چندین مقدار تنظیم می شود را نشان می دهد. جدول 8 نشان می دهد که افزایش نرخ دگرگونی نسبت به نرخ همگذری، تاثیر بیشتری روی کاهش زمان تکرار دارد و میانگین امتیاز میانگین قواعد را کاهش می دهد.

نرخ دگرگونی در محاسبه ی تکاملی به طور کلی بین 0.01 و 0.1 تنظیم می شود و 0.5 مقداری بسیار بزرگ است. با توجه به همین موضوع، اگر نرخ دگرگونی بین 0.01 و 0.1 تنظیم شده باشد، تاثیر تنظیمات پارامتر روی امتیاز میانگین قادعده بزرگ نیست. با توجه به این قیاس، تصمیم گرفتیم تا 0.01 را به عنوان نرخ دگرگونی استفاده کنیم تا بهترین امتیاز میانگین قواعد را با اینکه این مقدار زمان تکرار را افزایش می دهد، بدست آوریم.

جدول ۱۰
نتیجه ی مسئله ی کوله پشتی (مقادیر سیلونت)

k	توازن ظرفیت خوشه	میانگین	Max	Min
8	1:1:1:1:1:1:1:1	0.97	0.98	0.92
8	4:2:4:6:4:2:7:5	0.91	0.97	0.88
6	1:1:1:1:1:1	0.87	0.91	0.86
6	1:5:2:6:3:2	0.82	0.88	0.78
4	1:1:1:1	0.75	0.81	0.70
4	1:4:2:1	0.72	0.79	0.68

5.1.2. مقایسه ی روش های ترتیب گره

نتیجه ی قیاس بین دو روش ترتیب گره که انتخاب تصادفی کامل و انتخاب تصادفی جزیی هستند، در جدول 9 نشان داده می شود. شش مجموعه داده برای قیاس استفاده می شود که تعداد داده ()5000 و تنوع داده (50٪) یکسان هستند، اما تعداد ویژگی ها متفاوت هستند. ارزیابی عملکرد برای مقایسه ی تعداد تکرر مورد نیاز در جهت پوشش همه ی داده، طول قاعده ی اصلی، تعداد قواعد استخراج شده و امتیاز میانگین قواعد اجرا می شوند. تکرار در اینجا، به معنی تعداد منفرد های ایجاد شده در استخراج قاعده تا زمان پوشش همه ی رکوردها است.

زمانی که تعداد ویژگی ها افزایش می یابد، معمولا تعداد تکرار های مورد نیاز برای پوشش همه ی داده افزایش پیدا می کند. بهر حال، با مقایسه ی زمان مورد نیاز از طرف انتخاب تصادفی کامل و انتخاب تصادفی جزیی، انتخاب تصادفی جزیی نتایج بهتری نشان می دهد (تکرار کمتری مورد نیاز است). قواعد تا زمانی که همه ی رکوردها در مجموعه داده پوشش داده شوند، استخراج می گردند اما رکوردهایی که پیش از این پوشش داده شده اند، مجددا شامل نمی شوند. اختلاف برجسته بین انتخاب تصادفی کامل و انتخاب تصادفی جزیی در طول گره میانگین است که انتخاب تصادفی جزیی اساسا طول بلندتری نشان می دهد. انتخاب تصادفی جزیی با یافتن مجموعه اقلامِ رایج می تواند طول کمینه ی قواعد در هر خوشه را برقرار کند. انتخاب تصادفی جزیی اساسا تعداد بیشتری از قواعد بلندتر را به نسبت با انتخاب تصادفی کامل استخراج می کند، از همین رو، امتیازهای میانگین قواعدی که بوسیله ی انتخاب تصادفی جزیی استخراج شده اند، نتایج بهتری را نشان می دهند. با توجه به بخش بعدی، روش انتخاب تصادفی جزیی در شبیه سازی ها استفاده می شود.

5.2. توزیع قاعده ی کوله پشتی

در اینجا از مقدار سیلوئت برای ارزیابی نتایج خوشه بندی استفاده می شود. سیلوئت یک ارائه ی گرافیکی مختصر از اینکه هر شیء چقدر خوب در خوشه اش قرار گرفته فراهم می کند. مقدار سیلوئت از طریق معادله ی 8 محاسبه می شود.

$$s = \frac{b-a}{\max\{a,b\}} = \begin{cases} 1 - a/b, & \text{if } a < b \\ 0, & \text{if } a = b \\ b/a - 1, & \text{if } a > b \end{cases} \quad (8)$$

S: مقدار سیلوئت برای تک نمونه. مقدار سیلوئت برای یک مجموعه ی نمونه به عنوان میانگینِ مقادیر سیلوئتِ هر نمونه داده می شود.

a: تفاوت میانگینِ (فاصله) داده در خوشه ای یکسان.

b: کمترین تفاوت میانگین (فاصله) تا خوشه های دیگر

نتایج توزیع قاعده در جدول 10 نشان داده می شود. همه ی شبیه سازی ها با تعداد داده (5000) و تنوع داده (50%) یکسان انجام داده می شود. k تعداد خوشه ها (سایت ها) است، "تعادلِ ظرفیت خوشه ها" تناسب ظرفیت هرسایت را نشان می دهد. برای مثال، 1:1:1:1 بعنای این است که هر چهار سایت اندازه ای یکسان دارند و 1:4:2:1 یعنی سایت دوم (اندازه ی چهار) چهار برابر بزرگتر از اندازه ی سایت اول (اندازه ی یک) است. "میانگین، حداکثر و حداقل" داده را روی مقدار سیلوئتی نشان می دهد که از طریق خوشه های تولیدشده، بدست آمده اند. روش ارائه شده، در هنگام k و ظرفیت خوشه ی متعادل شده ی بزرگ تر، توانایی خوشه بندی خوبی براساس مقادیر سیلوئت نشان می دهد. هر چقدر که k کاهش یابد، مقادیر سیلوئت نیز کاهش می یابند و ظرفیت خوشه نامتعادل می شود. این وضعیت به علت ناسازگاری ظرفیت بین پوشش قاعده و ظرفیت خوشه، رخ می دهدبرای مثال، در زمانی که ظرفیت خوشه تنها 100 داده ی باقی مانده و پوشش قاعده ی مشخصی، 120 داده باشد 20 داده در خوشه ی دیگر توزیع خواهد شد. این مسئله روی نتیجه ی سیلوئت تاثیر می گذارد. اگر تعداد سایت ها k بزرگتر باشد، انواع مختلفی از قواعد می تواند در سایت های متعددی توزیع شود، پس از آن، قواعد نزدیک تر (شبیه تر) می تواند در هر خوشه قرار داده شود که به بهتر شدن مقدار سیلوئت کمک می کند. اگر ظرفیت خوشه نامتعادل باشد، برخی از سایت ها ظرفیت بزرگترو برخی ظرفیت کوچکتری دارند. سایتهایی که ظرفیت بزرگتری دارند باید انواع مختلفی از قاعده را شامل شوند (که در برخی از اوقات فاصله ی کمی از یکدیگر دارند)، بنابراین، مقدار سیلوئت کوچکتر می شود.

5.3. مقایسه با دیگر روش ها

شش مجموعه داده از مخزن یادگیری ماشین UCI (نشان داده شده در جدول 11) از طریق مقدار سیلوئت و نرخ دقت، برای مقایسه و عملکرد خوشه بندی ارزیابی می شوند.

جدول ۱۱

مجموعه داده UCI

	ویژگی	کلاس ها	نمونه ها	نوع داده
کیفیت شراب	12	2	4898	Real
ارزیابی ماشین	6	4	1728	Int
قطعه بندی تصویر	19	7	2100	Int, Real
شانل	9	7	54600	Int
نوع پوشش	54	8	581012	Int
مخمر	8	10	1484	Real

جدول ۱۲

مقایسه روش ها با ارزیابی سیلونت

Dataset	Methods Comparison with Silhouette						
	OCKC	KAP	FCM	K-means	HC	GNP	mean
Wine Quality	0.172	0.182	0.227	0.123	0.224	0.241*	0.195
Car Evaluation	0.795	0.789	0.809	0.801	0.752	0.812*	0.793
Segmentation	0.234	0.265	0.303	0.253	0.296	0.305*	0.276
Shuttle	0.324	0.314	0.398*	0.312	0.354	0.352	0.342
Covertype	-0.214	-0.453	-0.167	-0.254	-0.346	-0.125*	-0.260
Yeast	0.634	0.622	0.779	0.626	0.786	0.788*	0.706
میانگین	0.324	0.287	0.392	0.310	0.344	0.396*	

پنج روش برای مقایسه با روش ارائه شده (میانگین های k، خوشه بندی سلسله مراتبی، میانگین های C فازی، راه حل ترتیب محدود شده در خوشه بندی میانگین های K (OCKM) و تکثیر همبستگی K) استفاده شده است. همه ی این روش ها که در مقایسه استفاده شدند، روش های خوشه بندی بدون نظارت هستند و به جز خوشه بندی سلسله مراتبی از فاصله ی اقلیدسی به عنوان یک معیار فاصله استفاده می کنند. تنظیمات پارامتر برای هر روش به شرح زیر مشخص می شود:

1) میانگین K: فاصله ی اقلیدسی به عنوان مقیاس فاصله استفاده می شود. مقدار K به صورت تعداد کلاس های هر مجموعه داده تنظیم می شود.

2) خوشه بندی سلسله مراتبی: متراکم به عنوان استراتژی سلسله مراتبی انتخاب می شود و تک پیوند به عنوان یک روش خوشه بندی استعمال می گردد. فرایند خوشه بندی در زمانی که تعداد گروه ها به تعداد کلاس های هر مجموعه داده می رسد پایان می یابد.

3) میانگین C فازی: بهبود کمینه ی فازی ساز m که سطح فازی بودن خوشه را تعیین می کند، در 1.0×10^{-5} تنظیم می شود. مقدار K همانند کلاس های هر مجموعه داده تعیین می شود.

4) راه حل ترتیب محدود شده در خوشه بندی میانگین های K (OCKM): فاصله ی اقلیدسی به عنوان مقیاس فاصله استفاده می شود و استراتژی برنامه نویسی پویای بازگشتی برای بهبود کیفیت خوشه بندی استفاده می شود. مقدار K همانند تعداد کلاس های هر مجموعه داده قرار داده می شود.

5) تکثیر همبستگی K: فاصله ی اقلیدسی به عنوان مقیاس فاصله استفاده می شود و تکثیر همبستگی برای بهبود کیفیت خوشه بندی به کار گرفته می شود. مقدار K همانند تعداد کلاس های هر مجموعه داده قرار داده می شود.

جدول ۱۳
مقایسه با روش ها با ارزیابی نرخ دقت

Dataset	Methods Comparison with Accuracy Rate						
	OCKC	KAP	FCM	K-means	HC	GNP	mean
Wine Quality	0.642	0.613	0.786	0.771	0.695	0.787*	0.716
Car Evaluation	0.689	0.678	0.699	0.701	0.698	0.701*	0.694
Segmentation	0.678	0.724	0.776	0.725	0.712	0.792*	0.735
Shuttle	0.812	0.787	0.864*	0.839	0.818	0.824	0.824
Covertype	0.675	0.646	0.705	0.676	0.622	0.708*	0.672
Yeast	0.667	0.704	0.812	0.692	0.801	0.856*	0.755
mean	0.694	0.692	0.774	0.734	0.724	0.778*	

6) روش پیشنهاد شده: پارامترهای اصلیِ روش پیشنهاد شده برمبنای نتایج نشان داده شده در جدول 7و8 در بخش 5.1.1. مشخص می شود. در بخش 5.1.1. چندین تنظیماتِ نرخ های همگذری و نرخ های دگرگونی به لحاظ امتیاز میانگین و تکرارهای مورد نیاز برای پوشش همه ی داده ارزیابی می شود.

اگرچه روش های خوشه بندی مرسوم می توانند تعداد خوشه ها را برای ایجاد شدن تنظیم کنند اما تابعی برای اندازه گیری ظرفیت خوشه همانند روش خوشه بندی با تابعی برای حل KP، ندارند. از این رو، مسئله ی ظرفیت خوشه در این قیاس بحث نمی شود. روش ارائه شده می تواند خوشه بندی را با لحاظ کردنِ ظرفیت ها انجام دهد که این امر یکی از مزایای آن نسبت به الگوریتم های خوشه بندی مرسوم است.

نرخ دقت در شبیه سازی ها به عنوان مقیاس عملکرد خوشه بندی دیگری علاوه بر مقدار سیلوئت استفاده می شود. نرخ دقت، یک معیار رایج است که برای ارزیابی میزان خوب عملکردنِ الگوریتم های خوشه بندی در یک مجموعه داده با ساختاری آشنا استفاده می شود. نرخ دقت متناسب با مجموعه داده نتایج متفاوتی نسبت به سیلوئت نشان می دهد.

جدول 12 نتایج ارزیابی با سیلوئت و جدول 13 نتایج ارزیابی با نرخ دقت را نشان می دهد. علامت های ستاره (*) در کنار نتایج در هر دو جدول بهترین نتیجه در هر ردیف (مجموعه داده) را نشان می دهد. همان طور که در آخرین ردیفی جدول 12 و 13 نشان داده شده، روش ارائه شده بالاترین میانگین نتایج را نشان می دهد. روش ارائه شده، در جداول 12 و 13 نتایج خوشه بندی بهتری را نیز در پنج مورد از شش مجموعه داده نشان می دهد. روش ارائه شده تنها در مورد مجموعه داده ی شاتل در برابر دیگر روش های مرسوم شکست می خورد. ساختار مجموعه داده ی شاتل که در جدول 11 نشان داده شده، الگوی مستقیمی برای تشریح این موضوع که چرا روش پیشنهادی به دیگر روش ها می بازد نشان نمی دهد اما جدول 13 نشان می دهد که نرخ دقت میانگین برای همه ی روش ها (ستون آخر جدول 13) برای مجموعه داده ی شاتل بالاترین (0.824) است، به طوریکه دیگر روش های مرسوم نتایج خوشه بندی بهتری برای مجموعه داده ای نشان می دهند. که ساختن خوشه ها در مقایسه با دیگر مجموعه داده ها در آن نسبتا ساده است.

در اینجا، به آخرین ستون از جداول 12 و 13 توجه کنید که مقادیر میانگینِ سیلوئت (جدول 12) و نرخ دقت (جدول 13) را برای همه ی روش ها نشان می دهند. برای مثال، در جدول 12، مجموعه داده ی نوع پوشش، مقدار سیلوئت بسیار کمی نشان می دهد که به -0.26 می رسد اما نرخ دقت میانگین آن در جدول 13 0.672 است. در چنین موردی، مجموعه داده ی نوع پوشش دارای بیشترین تعداد ویژگی (54) است. مقدار سیلوئت نسبت تنوع داده بسیار حساس است، از این رو، مقدار میانگین سیلوئت برای همه ی روش ها کمتر از دیگر موارد (مجموعه های داده) می شود. نتایج مشابه برای مجموعه های داده ی "کیفیت شراب" و "بخش بندی تصویر" نیز نشان داده می شوند. می توانیم با تحلیل چنین نتایجی در جدول 12 بیابیم که تعداد زیاد ویژگی معمولا مقدار سیلوئت را کاهش می دهد چون پیچیدگیِ ترکیبات ویژگی را افزایش می دهد، در حالیکه تعداد زیاد کلاس ها، مقادیر سیلوئت را افزایش میدهد چون برای بسیاری از خوشه ها حفظ شباهت داده ساده تر می شود، به عبارت دیگر، شناسایی انواع متعدد مجزای بخش های داده برای تعداد کمی از خوشه ها دشوار است.

6. نتیجه گیری

این مقاله یک روش جدید خوشه بندی که برنامه نویسی شبکه ی ژنتیک و مسئله ی کوله پشتی را در جهت اداره ی خوشه بندی رکورد تجمیع می کند را ارائه می نماید. روش ارائه شده می تواند ترکیباتِ خوبی از ویژگی ها پیدا کند تا قواعد را برای خوشه بندی ایجاد نماید و همچنین ظرفیت سایت ها را برای توزیع قواعد در نظر می گیرد.

روش پیشنهاد شده، روش خوشه بندی جدیدی با مسئله ی ظرفیت انباره ی اضافی فراهم می کند که سازگار با داده ی بزرگ با تعداد زیادِ ویژگی، نمونه و خوشه است و عملکرد خوشه بندی با شش مجموعه داده از مخزن یادگیری ماشین UCI ارزیابی می شود و بهترین نتایج میانگین در مقایسه با پنج الگوریتم خوشه بندی مرسوم دیگر حاصل می شود.

روش ارائه شده به علت زمان تکامل در جهت بدست آوردنِ قواعد خوب، برای پردازش برخط کمتر مطلوب است. روش ارائه شده برای یک پردازش برون خطی که به جای زمان پردازش نیازمند نتایج بهینه است، مناسب می باشد.

در تحقیق آینده، اجرای شبیه سازی ها با DDBMAS حقیقی با برنامه های در حال اجرا ضروری است تا عملی بودنِ روش ارائه شده آزمایش شود. روش ارائه شده می تواند به عنوان یک میان افزار بین پایگاه های داده ی توزیع شده و کاربردی از مدیریت تخصیص قطعه ی پایگاه داده ساخته شود. لازم به ذکر است که این روش می تواند به ماتریس CRUD متعلق به پایگاه های داده دسترسی داشته باشد.

این الگوریتم باید برای اجرای پردازش های برخط بهبود داده شود. تجمیع با دیگر الگوریتم ها مانند منطق فازی و شبکه ی عصبی می تواند مشخص شود تا توانایی روش ارائه شده را بهبود ببخشد.

References

Ahmad, A., & Dey, L. (2007). A k-mean clustering algorithm for mixed numeric and categorical data. *Data & Knowledge Engineering*, *63*(2), 503–527.

Bezdek, J. C., Ehrlich, R., & Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, *10*(2), 191–203.

Bhuyar, P. R., Gawande, A. D., & Deshmukh, A. B. (2012). Horizontal fragmentation technique in distributed database. *International Journal of Scientific and Research Publications*, *2*(5).

Cuzzola, J., Jovanovic, J., Bagheri, E., & Gasevic, D. (2015). Evolutionary fine-tuning of automated semantic annotation systems. *Expert Systems with Applications*.

Guinepain, S., & Gruenwald, L. (2006). Automatic database clustering using data mining. In *Proc. of the 17th international workshop on database and expert systems applications (DEXA'06)* (pp. 124–128).

Guinepain, S., & Gruenwald, L. (2008). Using cluster computing to support automatic and dynamic database clustering. In *Proc. of the 2008 IEEE international conference on cluster computing* (pp. 394–401).

Holland, J. H. (1975). *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Karypis, G., Han, E.-H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, *32*(8), 68–75.

Koza, J. R. (1992). *Genetic Programming, on the programming of computers by means of natural selection.* Cambridge, Mass.: MIT Press.

Lai, K. (2006). The knapsack problem and fully polynomial time approximation schemes (FPTAS). *18.434: Seminar in Theoretical Computer Science, Prof. M. X. Goemans*.

Lowden, B. G., & Kitsopanidis, A. (1993). Enhancing database retrieval performance using record clustering. *Department of Computer Science, The University of Essex*.

Mabu, S., Chen, C., Lu, N., Shimada, K., & Hirasawa, K. (2011). An intrusion-detection model based on fuzzy class-association-rule mining using genetic network programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, *41*(1), 130–139.

Özsu, M. T., & Valduriez, P. (2011). *Principles of distributed database systems.* Springer Science & Business Media.

Rahimi, H., Parand, F.-A., & Riahi, D. (2015). Hierarchical simultaneous vertical fragmentation and allocation using modified bond energy algorithm in distributed databases. *Applied Computing and Informatics*.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Saha, A., & Das, S. (2015). Automated feature weighting in clustering with separable distances and inner product induced norms–a theoretical generalization. *Pattern Recognition Letters*, *63*, 50–58.

Shimada, K., Hirasawa, K., & Hu, J. (2006). Genetic network programming with acquisition mechanisms of association rules. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, *10*(1), 102–111.

Singh, R. P. (2011). Solving 0–1 knapsack problem using genetic algorithms. In *Proc. of the 2011 IEEE 3rd international conference on communication software and networks (ICCSN)* (pp. 591–595).

Steinley, D., & Hubert, L. (2008). Order-constrained solutions in k-means clustering: even better than being globally optimal. *Psychometrika*, *73*(4), 647–664.

Toth, P. (1980). Dynamic programming algorithms for the zero-one knapsack problem. *Computing*, *25*(1), 29–45.

Zhang, X., Wang, W., Norvag, K., & Sebag, M. (2010). K-ap: generating specified k clusters by efficient affinity propagation. In *Data mining (icdm), 2010 icee 10th international conference on* (pp. 1187–1192).

Zhao, J., Huang, T., Pang, F., & Liu, Y. (2009). Genetic algorithm based on greedy strategy in the 0-1 knapsack problem. In *Proc. of the 3rd international conference on genetic and evolutionary computing (WGEC'09)* (pp. 105–107).

Zilio, D. C., Rao, J., Lightstone, S., Lohman, G., Storm, A., Garcia-Arellano, C., & Fadden, S. (2004). DB2 design advisor: Integrated automatic physical database design. In *Proc. of the 30th international conference on very large data bases-volume 30* (pp. 1087–1097). VLDB Endowment.