

استخراج قواعد طبقه بندی با استفاده از الگوریتم بهینه سازی کلونی مورچه‌ها

محمد رضا کنگاوری^۱، بابک فخار^۲

چکیده

هدف از این مقاله ارائه یک الگوریتم برای داده‌کاوی است. این الگوریتم **Ant-Miner** (داده‌کاوی براساس کلونی مورچه‌ها) نام دارد. هدف از **Ant-Miner** استخراج قواعد طبقه‌بندی از پایگاه‌داده‌ها است. **Ant-Miner** از ترکیب مفاهیم کلونی مورچه‌ها و اصول داده‌کاوی حاصل شده است. تحقیقات اخیر نشان داده است که الگوریتم‌های بهینه‌سازی به روش کلونی مورچه‌ها بطور موفقیت آمیزی در حل مسائل بهینه‌سازی ترکیبی قابل استفاده هستند. هدف ما در این مقاله ارزیابی و تحلیل نسخه‌های مختلف **Ant-Miner** و بررسی نحوه طراحی و تولید نسخه‌ای جدید از **Ant-Miner** است که نسبت به نسخه‌های قبل از عملکرد بهتری برخوردار است. ما **Ant-Miner** ارائه شده را با نسخه قبل بر روی دو مجموعه داده مقایسه کرده‌ایم، نتیجه این مقایسه نشان می‌دهد که دقت قواعد کشف شده در نسخه جدید بیشتر از دقت قواعد تولید شده در نسخه قدیم است.

کلمات کلیدی

کلونی مورچه‌ها، قواعد طبقه بندی، داده‌کاوی، استخراج دانش، پایگاه داده‌ها.

Classification Rule Discovery with Ant Colony Optimization

Mohamadreza Kangavari, Babak Fakhar

Abstract

This paper proposes an algorithm for data mining called Ant-Miner (ant-colony-based data miner). The goal of Ant-Miner is to extract classification rules from data. The algorithm is inspired by both researches on the behavior of real ant colonies and some data mining concepts as well as principles. Recently research shows that ant colony optimization algorithm have been applied successfully to combinatorial optimization problems. In this paper we present an improvement to Ant-Miner. We compare the performance of new algorithm with before algorithm in two public domain data sets.

Keyword

Ant Colony, Classification Rules, Data mining, Extract knowledge, Database.

۱-دکتر محمد رضا کنگاوری عضو هیأت علمی دانشگاه علم و صنعت ایران، دانشکده مهندسی کامپیوتر. تلفن ۰۲۱-۷۷۴۵۱۵۰۳، kangavari@iust.ac.ir.

۲-مهندس بابک فخار عضو هیأت علمی دانشگاه آزاد اسلامی واحد ماهشهر. تلفن: ۰۲۱-۹۱۶۱۵۲۲۵۱، fakharbabak@yahoo.com.

۱- مقدمه

کردن غذا جیسو می‌کنند. نکته اساسی در رفتار عاملها یا مورچه‌ها همکاری و انطباق است. با استفاده از این سیستم می‌توان یک روش فرا اکتشافی برای حل مسائل بهینه سازی NP ارائه کرد.[4] لازمه حل یک مسئله به روش کلونی مورچه‌ها تعریف ساختار آن مسئله در قالب یک گراف است، پس از ارائه این تعریف می‌توان از الگوریتم ACO برای یافتن کوتاه‌ترین مسیر که همانا پاسخ مسئله است استفاده کرد. عملکرد کلی مورچه‌ها در جهت انتخاب گره‌ها در مسیر، وابسته به یک تابع احتمالی براساس معادله (۱) است. این معادله در بخش (۳) مورد بررسی قرار خواهد گرفت.

۳- ساختار کلی Ant-Miner

Ant-Miner یک روش یادگیری با نظارت جهت استخراج قواعد طبقه‌بندی از یک پایگاه داده است. فرآیند آموزش با نظارت شامل سه مرحله است، ۱. انتخاب یک مجموعه آموزشی^۱، ۲. ارائه یک متد برای داده کاوی و آنالیز (یادگیری)، ۳. سنتز و ترکیب قواعد پیش بینی. انتخاب مجموعه آموزشی یکی از مراحل کلیدی در داده کاوی است. مجموعه آموزشی باید حداقل امکان کوچک، ساده و از جامعیت لازم برخوردار باشد.[5][3] در شکل (۲) می‌توان نمای کلی Ant-Miner را مشاهده کرد. خروجی Ant-Miner مجموعه ای از قواعد بصورت شکل (۱) است.

$IF (term1 \text{ AND } term2 \text{ AND } ...) THEN <class>$

شکل (۱): خروجی Ant-Miner

هر ترم بصورت $A_i = V_{ij} \equiv Term_{ij}$ است.

A_i : متغیر آم

V_i : آمین مقدار از دامنه A_i است.

در Ant-Miner ابتدا یک مورچه حرکت خود را با یک قاعده خالی (if then) آغاز می‌کند، سپس در هر مرحله اقدام به انتخاب یک ترم می‌نماید. به بیان ساده‌تر مورچه با انتخاب یک مسیر، یک ترم جدید به بخش جلویی قاعده اضافه می‌کند. به این ترتیب مورچه‌ها با حرکت در گراف ساختار مسئله و با عبور از گره‌های مختلف ترم‌های جدید را به قاعده جاری اضافه می‌کنند.

پس از تکمیل شدن ساخت قاعده کیفیت آن از نظر قدرت در پیش‌بینی محاسبه می‌شود، و بطور موقت در یک لیست نگهداری می‌شود. حال نوبت به مورچه بعد می‌رسد. این مورچه با توجه به میزان فرومون ذخیره شده در مسیرها توسط مورچه‌های مراحل قبل و با استفاده از تابع (۱) اقدام به ساخت قاعده‌ای جدید می‌کند. تعداد مورچه‌ها در این مرحله یکی از پارامترهایی است که توسط کاربر قابل تنظیم است (حلقه Repeat). در نهایت از میان قواعد تولید شده بهترین قاعده (بالاترین کیفیت) انتخاب و به لیست قواعد کشف شده اضافه می‌شود. قاعده کشف شده نمونه‌های متعلق به یک طبقه را

در اصل هدف از Ant-Miner استخراج دانش از پایگاه داده است. داده کاوی دانشی است که دامنه آن از ترکیب علوم مربوط به یادگیری ماشین، مفاهیم آماری و پایگاه داده‌ها حاصل شده است. تأکید می‌شود که در داده کاوی برخلاف روشهای آماری تنها و تنها بدنبال استخراج دانش هستیم، دانشی که نه تنها درست باشد بلکه جامعیت را نیز برای کاربر به همراه داشته باشد [2] [1]. فرآیند داده کاوی شامل وظایف مختلف است. این وظایف عبارتند از طبقه بندی، رگرسیون، مدل سازی مستقل، خوشه بندی و... [1] عملکرد کلی هر یک از این وظایف براساس فرآیند جستجو در بانک‌های اطلاعات بزرگ استوار شده است، نتیجه این جستجوها کشف الگوهای جدید، معتبر و غیر جزئی است، که به کمک آنها می‌توان دانش نهفته در داده‌ها را استخراج و نمایش داد. فرآیند کشف قواعد یکی از مهمترین وظایف در داده کاوی محسوب می‌شود. طی این فرآیند مجموعه‌ای نمادین از قواعد برای شرح هر طبقه یا خانواده از داده‌ها ارائه می‌شود [3].

آقای پرپینلی برای اولین بار، سیستمی با نام Ant-Miner1، برای کشف قواعد طبقه بندی از پایگاه داده‌ها طراحی کرد [8]. این سیستم براساس بهینه سازی کلونی مورچه‌ها طراحی شده است. روش کلونی مورچه‌ها، نسبت به سایر روشهای سنتی، از انعطاف و قدرت بیشتری در حل مسائل برخوردار است. در نسخه بعدی Ant-Miner (Ant-Miner2) روشی جدید جهت محاسبه تابع اکتشافی ارائه شد. در این نسخه مقدار تابع اکتشافی بر اساس میزان تکرار یک متغیر-مقدار (ترم) در یک طبقه محاسبه می‌شود. در Ant-Miner3 استراتژی جدیدی جهت بهنگام سازی فرومون مسیرها ارائه شد [11]. در این مقاله ما قصد داریم ضمن مطرح کردن معایب Ant-Miner3 نسخه ای جدید از Ant-Miner ارائه دهیم.

۲- سیستم کلونی مورچه‌ها (ACS)

سیستم کلونی مورچه‌ها شاخه‌ای از علم هوش مصنوعی است که به گروه هوشمند معروف است. مورچه‌ها حشرات مستقلی هستند که فعالیت اشتراکی دارند [4]. مورچه‌ها بدون هر گونه کنترل مرکزی، نظارت متمرکز و اطلاعات تصویری از مسیر قادرند کوتاه‌ترین مسیر از لانه به غذا را جستجو کنند. مورچه‌ها به صورت غیر مستقیم و از طریق ماده‌ای بنام فرومون با هم ارتباط برقرار می‌کنند و اطلاعات مربوط به مسیرها را در اختیار یکدیگر قرار می‌دهند. هر مورچه در زمان حل مسئله با مسیرهای مختلف برخورد می‌کند. هنگامی که یک مورچه از یک مسیر عبور می‌کند میزان فرومون موجود در آن مسیر را متناسب با کیفیت آن مسیر افزایش می‌دهد. به این ترتیب هنگامی که یک مورچه با چند مسیر مختلف مواجه می‌شود، احتمال انتخاب مسیری که میزان فرومون در آن نسبت به سایر مسیرها بیشتر است، افزایش می‌یابد. بنابراین مورچه‌ها همواره کوتاه‌ترین مسیر را برای پیدا

ترم برای اضافه کردن به قاعده در حال ساخت توسط مورچه‌ها طبق معادله (۱) و با توجه به تابع اکتشافی وابسته به مسئله و میزان فرومون موجود در آن مسیر صورت می‌گیرد.

پوشش می‌دهد، سپس همین فرایند برای تولید قاعده‌ای جهت پوشش نمونه‌های متعلق به طبقه دیگر تکرار می‌شود. این حلقه (While) نیز آنقدر تکرار می‌شود تا تمام یا تقریباً تمام نمونه‌ها موجود در مجموعه آموزشی توسط قواعد کشف شده پوشش داده شوند. انتخاب مسیر یا

```

Training set = all training cases;
WHILE (No. of uncovered cases in the Training set > max_uncovered_cases)
    i=0;
    REPEAT
        i=i+1;
        Anti incrementally constructs a classification rule;
        Prune the just constructed rule;
        Update the pheromone of the trail followed by Anti;
    UNTIL (i ≥ No_of_Ants) or (Anti constructed the same rule as the
    previous No_Rules_Converg-1 Ants)
    Select the best rule among all constructed rules;
    Remove the cases correctly covered by the selected rule from the training set;
END WHILE

```

شکل (۲): نمای کلی Ant-Miner [8]

است. زیرا مقدار $InfoT_{ij}$ بین صفر و $\log_2 K$ متغیر است. اگر مقدار V_{ij} برای A_i در مجموعه آموزشی اتفاق نیفتاده باشد آنگاه مقدار $InfoT_{ij}$ برابر بیشترین مقدار یا $\log_2 K$ خواهد شد که به این ترتیب $term_{ij}$ از کمترین قدرت پیش‌بینی برخوردار است. در شرایطی که تمام نمونه‌ها متعلق به یک طبقه یا کلاس مشابه باشند آنگاه مقدار $InfoT_{ij}$ برابر صفر خواهد بود که این موضوع به معنی افزایش احتمال انتخاب $term_{ij}$ است. تابع اکتشافی طراحی شده در Ant-Miner مشابه تابع اکتشافی الگوریتم درخت تصمیم گیری C4.5 است. [7]

۳-۱- هرس کردن قواعد

در Ant-Miner پس از ساخته شدن هر قاعده رویه هرس کردن فراخوانی می‌شود. هرس کردن قواعد در داده‌کاوی یک امر مرسوم است [8][7]. هدف از هرس کردن قواعد افزایش سادگی و قدرت پیش‌بینی قواعد است. هنگامی که یک مورچه عمل ساخت یک قاعده را به پایان رسانید آنگاه رویه هرس کردن، فراخوانی می‌شود، ایده اصلی در رویه هرس کردن به این ترتیب است که عملیات با یک قاعده که شامل تعدادی ترم است آغاز می‌شود، سپس در هر تکرار یک ترم از قاعده حذف می‌شود و کیفیت قاعده مورد نظر توسط تابع (4) ارزیابی می‌شود. این روند تا زمانی ادامه پیدا می‌کند که با حذف ترم از قاعده کیفیت آن افزایش یابد. باید به این نکته نیز اشاره کرد که این روند ممکن است موجب جابجا شدن طبقه پیش‌بینی شده توسط یک قاعده گردد.

$$Q = \left(\frac{TruePos}{TruePos + FalseNeg} \right) \times \left(\frac{TrueNeg}{FalsePos + TrueNeg} \right) \quad (4)$$

TruePos: تعداد نمونه‌های پوشش داده شده توسط قاعده، بطوری که طبقه آنها توسط قاعده به درستی پیش‌بینی شده است.

$$P_{ij} = \frac{\eta_{ij} \cdot \tau_{ij}}{\sum_{i=1}^a \sum_{j=1}^{b_i} (\eta_{ij} \cdot \tau_{ij}(t))} \quad (1)$$

η_{ij} : مقدار تابع اکتشافی وابسته به مسئله برای $term_{ij}$.

τ_{ij} : مقدار فرومون موجود در مسیر i به j یا $term_{ij}$.

Ant-Miner به ازاء هر $term_{ij}$ قابل استفاده در هر قاعده، مقدار

تابع η_{ij} را محاسبه می‌کند تا مورچه‌ها را در انتخاب ترم‌ها کمک

کند. این تابع براساس ساختار تئوری مسئله طراحی می‌شود [6]

در نسخه اصلی Ant-Miner مقدار تابع اکتشافی از معادله (2)

(اینترپای) بدست می‌آید.

$$\eta_{ij} = \frac{\log(K) - InfoT_{ij}}{\sum_{i=1}^a \sum_{j=1}^{b_i} \log_2^{(k)} - InfoT_{ij}} \quad (2)$$

$$InfoT_{ij} = - \sum_{w=1}^k \left[\frac{freqT_{ij}}{|T_{ij}|} \right] \times \log_2 \left[\frac{freqT_{ij}}{|T_{ij}|} \right] \quad (3)$$

K : تعداد طبقات.

$|T_{ij}|$: تعداد کل نمونه در جزء T_{ij} (یک جزء شامل تعدادی نمونه

است که در آن $V_{ij} = A_{ij}$ وجود داشته باشد).

$freqT_{ij}^w$: تعداد کل تکرار ترم‌ها در جزء T_{ij} که در طبقه W قرار دارند.

a : تعداد کل صفت خاصه (متغیرها)

b_i : تعداد کل مقادیر مربوط به دامنه صفت خاصه i .

در معادلات فوق هر چه مقدار $InfoT_{ij}$ بیشتر باشد مفهومش این است که احتمال انتخاب $term_{ij}$ توسط مورچه در حال ساخت قاعده کمتر

Majority-class T_{ij} : برای محاسبه این مقدار به این ترتیب عمل می‌شود که ابتدا کلیه نمونه‌هایی که در آنها مقدار $A_i = V_{ij}$ ظاهر شده است جدا می‌شوند (T_{ij}) سپس بررسی می‌شود که نمونه‌ها بیشتر متعلق به کدام طبقه هستند. محاسبه تابع اکتشافی با استفاده از معادله (۸) موجب بهبود پیچیدگی زمانی Ant-Miner می‌شود، ولی تغییری در میزان دقت پیش بینی قواعد اعمال نمی‌کند.

۵- تحلیل و ارزیابی Ant-Miner^۲

در Ant-Miner^۲ مشکل مطرح شده در رابطه با Ant-Miner اصلی حل نشد. (همگرا شدن سریع مورچه‌ها در یک مسیر) در نسخه بعدی Ant-Miner با نام Ant-Miner^۳ روش جدیدی برای بهنگام سازی میزان فرومون به ازاء ترمهای استفاده شده برای ساخت قاعده مورد نظر ارائه شد. در Ant-Miner^۳ عملیات بهنگام سازی فرومون ترمها با استفاده از معادله (۹) انجام می‌شود، و به همین ترتیب میزان فرومون به ازاء ترمهای استفاده نشده نرمال سازی می‌شود.

$$\tau_{ij}(t) = (1-p)\tau_{ij}(t-1) + (1 - \frac{1}{1+Q})\tau_{ij}(t-1), \forall i, j \in R \quad (9)$$

Q: کیفیت قاعده تولید شده.

P: با استفاده از این پارامتر می‌توان سرعت تبخیر فرومون در مسیرها را مشخص کرد.

مقدار Q در این رابطه بین 0 و 1 تغییر می‌کند. بیشترین مقدار Q به معنی بیشترین میزان افزایش فرومون برای ترم استفاده شده و کمترین مقدار Q به معنی کاهش میزان فرومون برای ترمهای استفاده نشده در ساخت قاعده است.

ویژگی اصلی پارامتر P این است که به کمک آن می‌توان میزان تأثیر گزاری گذشت زمان را برای ساخت قاعده جاری کنترل کرد [9]. مقدار این پارامتر نیز بین 0 و 1 است. در Ant-Miner^۳ برای حل مشکل همگرا شدن سریع مورچه‌ها در یک مسیر، انتخاب ترمها برای ساخت یک قاعده با استفاده از الگوریتم 1 صورت می‌گیرد.

```

if  $q \leq \varphi$ 
    Loop
        if  $q_2 \leq \sum_{j \in J_i} P_{ij}$  then
            choose termij
        end-Loop
    else
        choos termij With man  $P_{ij}$ 

```

الگوریتم 1

q_1, q_2 : اعداد تصادفی.

φ : یک پارامتر بین 0 و 1.

J: شماره i امین مقدار برای A_j .

p_{ij} : احتمال محاسبه شده در تابع (1).

FalsePos: تعداد نمونه‌های پوشش داده شده توسط قاعده، بطوری که طبقه آنها با طبقه پیش‌بینی شده توسط قاعده تفاوت دارند.
FalseNeg: تعداد نمونه‌هایی که توسط قاعده پوشش داده نشده‌اند، و به طبقه پیش‌بینی شده توسط قاعده تعلق دارند.
TrueNeg: تعداد نمونه‌هایی که توسط قاعده پوشش داده نشده‌اند، و طبقه آنها با طبقه پیش‌بینی شده توسط قاعده تفاوت دارند.

۳-۲- بهنگام سازی فرومون

در Ant-Miner، در ابتدا میزان فرومون کلیه مسیرها یا ترمها با هم مساوی است و این مقدار با استفاده از رابطه (۵) محاسبه می‌شود. [8]

$$\tau(t=0) = \frac{1}{\sum_{i=1}^a b_i} \quad (5)$$

a: تعداد صفات خاصه (متغیرها).

b_i : تعداد مقادیر ممکن برای a.

بعد از اینکه مورچه‌ای عمل ساخت یک قاعده را به اتمام رسانید میزان فرومون آن مسیر با استفاده از معادله (۶) بهنگام سازی می‌شود.

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t).Q \forall i, j \in R \quad (6)$$

برای شبیه سازی مفهوم تبخیر فرومون مسیرها به این ترتیب عمل می‌شود که میزان فرومون به ازاء ترمهایی که در ساخت قاعده استفاده نشده اند کاهش می‌یابد. برای انجام این عمل از معادله (۷) استفاده می‌شود. [6]

$$\tau_{ij} = \frac{\tau_{ij}}{\sum \tau_{ij}} \forall i, j \quad (7)$$

τ_{ij} : میزان فرومون در مسیر i به J یا به ازاء term_{ij}.

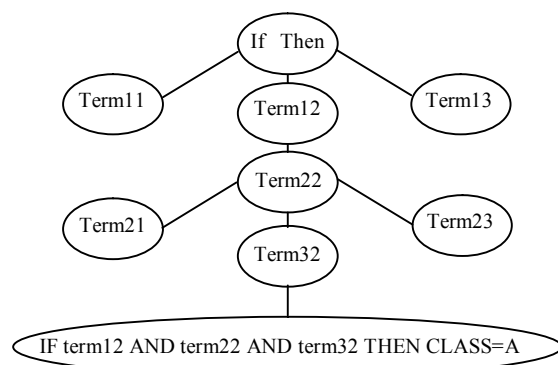
۴- تحلیل و ارزیابی Ant-Miner^۱

این موضوع معقول است که مورچه‌ها انتخاب ترمها را براساس معادله (1) انجام دهند. در این معادله انتخاب یک ترم براساس میزان فرومون و مقدار تابع اکتشافی صورت می‌گیرد. در این معادله میزان فرومون پس از هر مرحله که یک مورچه یک ترم را انتخاب کرد تغییر می‌کند، این میزان برای ترمهایی که در ساخت قاعده استفاده شده‌اند افزایش و برای سایر ترمها کاهش می‌یابد، این در حالی است که میزان تابع اکتشافی در طول ساخت قواعد تغییر نمی‌کند. [9] این شرایط موجب همگرا شدن سریع مورچه‌ها در مسیری خاص می‌شود و خاصیت استخراج دانش جدید در مسیرهای دیگر از بین می‌رود. در Ant-Miner² که به آن Ant-Miner بر پایه غلظت نیز گفته می‌شود. [11]

تابع اکتشافی با استفاده از معادله (۸) محاسبه می‌شود.

$$\eta_{ij} = \frac{\text{majority-class } T_{ij}}{|T_{ij}|} \quad (8)$$

$q1$ و $q2$ بیشتر باشد زمان اجرای الگوریتم نیز بیشتر می‌شود. ویژگی این روش این است که انتخاب ترم‌ها را کاملاً تصادفی انجام نمی‌دهد بلکه ابتدا مسیر اصلی را به کمک تابع (1) تعیین می‌کند سپس از میان ترمهایی که به مقدار این تابع نزدیکتر هستند یکی را بصورت تصادفی انتخاب می‌کند. در این روش گراف فضای حالت مسئله بصورت شکل (4) پیمایش خواهد شد.



شکل(4): گراف فضای جستجو در Ant-Miner جدید

در شکل (۲) فرض کنید مقدار تابع (1) برای $term13$ بیشترین است، در این حالت Ant-Miner با احتمال بالاتر آنرا برای اضافه کردن به قاعده جاری انتخاب می‌کند. ولی در روش ما تنها $term11$ و $term12$ به دلیل نزدیک بودن مقدار تابع (1) و قرار گرفتن بین مقادیر $q1$ و $q2$ در نظر گرفته می‌شوند و از میان این سه ترم یکی بصورت تصادفی انتخاب می‌شود. در نهایت کیفیت قاعده تولید شده محاسبه و عمل هرس کردن بر روی آن انجام می‌شود. با مقایسه‌های صورت گرفته که حاصل آن در جدول (۱) مشاهده می‌شود، این نتیجه بدست می‌آید که کیفیت قواعد تولید شده توسط این نسخه از Ant-Miner در بعضی موارد از Ant-Miner3 بیشتر است. سرعت تولید قواعد در نسخه جدید بیشتر از سرعت تولید قواعد در Ant-Miner3 است. زیرا در نسخه جدید رویه هرس کردن زمان کمتری را به خود اختصاص می‌دهد، دلیل این امر کوچک بودن قواعد تولید شده است.

جودل(۱): مقایسه Ant-Miner جدید با Ant-Miner3

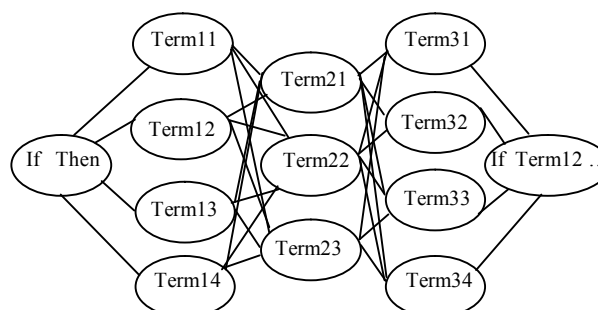
	Breast Cancer		Tic-tac-toe	
Run Num.	new Ant-Miner	Ant-Miner3	new Ant-Miner	Ant-Miner3
1	94.05	94.32	71.84	82.97
2	94.15	93.15	73.12	72.34
3	96.12	91.67	78.81	78.94
4	97.16	97.06	79.88	80
5	96.01	92.75	78.24	72.63
6	94.5	95.65	81.27	80

هدف از اعداد تصادفی $q1$ و $q2$ ایجاد تعادل بین استفاده از دانش استخراج شده توسط مورچه‌های قبل و استخراج دانش جدید است. به این ترتیب که اگر $q1 \leq \varphi$ باشد مفهومی استخراج دانش جدید و در غیر این صورت به معنی استفاده از دانش استخراج شده توسط مورچه‌ها در مراحل قبل است.

۶- ارائه الگوریتم جدید و مقایسه آن با Ant-3

Miner (هدف ما)

استفاده از اعداد تصادفی مطرح شده در Ant-Miner3 شاید موجب استخراج دانش جدید از فضای مسئله گردد ولی از طرفی موجب افزایش زمان اجرای الگوریتم و از طرف دیگر موجب کشف قواعدی با کیفیت پایین می‌گردد. زمان اجرای الگوریتم به این دلیل افزایش می‌یابد که مورچه‌ها با قرار گرفتن در مسیرهای جدید ممکن است قواعدی کشف کنند که از کیفیت پایین برخوردار هستند لذا الگوریتم باید آنها را حذف نماید. این حالت (تولید قاعده و حذف آن) موجب افزایش زمان اجرای الگوریتم می‌گردد. به عنوان مثال اگر در یک مسئله تعداد صفات خاصه سه (متغیرها) ($a=3$) باشد. مورچه‌ها در Ant-Miner در بدترین حالت برای تولید یک قاعده گراف شکل (3) را پیمایش می‌کنند.



شکل(3): گراف فضای جستجو در Ant-Miner3

ما پیشنهاد می‌کنیم برای هدایت مورچه‌ها در جهت انتخاب هر چه بهتر ترمها، بجای استفاده از اعداد تصادفی یا تابع (1) ابتدا کلیه ترمهایی که مقدار تابع (1) برای آنها بین $q1$ و $q2$ است، مشخص شده، سپس از میان آنها یکی بصورت تصادفی انتخاب و به قاعده در حال ساخت اضافه شود. $q1$ و $q2$ پارامترهایی هستند که هم توسط Ant-Miner و هم توسط کاربر قابل تعیین و کنترل هستند. مقدار $q1$ از $q2$ کمتر و P_{ij} بین این دو مقدار قرار دارد. هر چه اختلاف بین $q1$ و $q2$ بیشتر باشد احتمال استخراج دانش جدید و ارزیابی ترمهای دیگر بیشتر و هر چه اختلاف بین این دو پارامتر کمتر و به مقدار تابع (1) نزدیکتر باشد انتخاب ترمهای جدید توسط مورچه جاری بیشتر بر اساس مقدار تابع (1) صورت می‌گیرد. به این ترتیب می‌توان حرکت مورچه‌ها را در کنترل کاربر قرار داد و سرعت و دقت Ant-Miner را افزایش داد. البته باید به این نکته توجه داشت که هر چه اختلاف بین

۷- نتیجه گیری

درخت تصمیم گیری قیاسی یکی از بهترین متدهای شناخته شده در رابطه با پیدا کردن قواعد طبقه بندی در بانکهای اطلاعاتی است. آقای پریپنلی ثابت کرد که به کمک Ant-Miner^۱ می‌توان قواعدی ساده‌تر و با صحت و دقت بالاتری نسبت به درخت تصمیم گیری C_{4.5} تولید کرد. در این مقاله Ant-Miner^۱، Ant-۲، Miner و Ant-Miner^۳ مورد ارزیابی قرار گرفتند و نحوه تولید نسخه ای بهتر بررسی شد. در نسخه جدید، کمک پارامترهای گفته شده می‌توان رفتار مورچه‌ها را به شکل قابل کنترل تری به نمایش درآورد. در این نسخه استخراج دانش با سرعت و دقت بیشتری صورت می‌گیرد. این خاصیت موجب افزایش کاربری Ant-Miner در سیستم‌های بزرگ می‌گردد. Ant-Miner هایی که تاکنون ارائه شده‌اند قادر به کشف لیست مرتبی از قواعد هستند، لذا تحقیقات آینده در زمینه Ant-Miner را می‌توان در زمینه‌ای ادامه داد که Ant-Miner قادر به کشف لیست نامرتبی از قواعد باشد. لیست قواعد نامرتب شرایطی را فراهم می‌کند که می‌توان هریک از قواعد را بصورت مستقل تفسیر کرد. این ویژگی قابلیت درک دانش استخراج شده را آسانتر می‌کند.

مراجع

- [10] Bo Liu, Hussein A. Abbass, and Bob McKay "Classification Rule Discovery with Ant Colony Optimization" IEEE Computational Intelligence Bulletin February 200۳
- [11] Lotti Admane, Karima Benatchba, Mouloud KOUDIL, Habiba "Using ant colonies to solve data-mining problems" DFUAS IEEE International Conference on Systems, Man and Cybernetics Said GHAROUT and Nacer HAMAN1 Institut National de formation en Informatique, BP 68M, 16270, Oued Smar, Alg6rie.2004.
- [1] S. M. Weiss and C. A. Kulikowski, Computer Systems that Learn, San Francisco, CA: Morgan Kaufmann, 1991.
- [2] A. A. Freitas and S. H. Lavington, Mining Very Large Databases with Parallel Processing, London, UK: Kluwer, 1998.
- [3] T. M. Cover and J. A. Thomas, "Elements of Information Theory, New York, NY: John Wiley & Sons, 1991.
- [4] M. Dorigo, G. Di Caro and L. M. Gambardella, "Ant algorithms for discrete optimization," Artificial Life, vol. 5, no. 2, pp. 137-172, 2000.
- [5] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From data mining to knowledge discovery: an overview," In: Advances in Knowledge Discovery & Data Mining, U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy (Eds.) Cambridge, MA: AAAI/MIT, pp. 1-34, 1996.
- [6] Dorigo, M., & Maniezzo, V. "The ant system: optimization by a colony of cooperating agent"s. IEEE Transactions on Systems, Man, and Cybernetics, 26(1), 1-13 1996.
- [7] Ziqiang Wang, Boqin Feng, Classification Rule Mining with an Improved Ant Colony Algorithm, Lecture Notes in Computer Science, Volume 3339, Jan 2004.
- [8] Parepinelli, R. S., Lopes, . "An Ant Colony Algorithm for Classification Rule Discovery". In H. A. a. R. S. a. C. Newton (Ed.), Data Mining:
- [9] M. P. Oakes, "Ant Colony Optimisation for Stylometry: The Federalist Papers." International Conference on Recent Advances in Soft Computing, November 2004.