کاربرد الگوریتم‌های تکاملی در داده‌کاوی

تحقیق درس: سیستم‌های خبره و مهندسی دانش
استاد گرامی: دکتر ناصر قاسم‌آقایی
نویسنده: محسن مؤمنی(mohsenmomeni@yahoo.com)

۱) چکیده

گسترش روز افزون کاربردهای داده‌کاوی، و به‌کارگیری آن در مورد داده‌هایی که از حجم نمونه‌ی بالایی برخوردارند، باعث‌گردیده، مشکلات کار با ابزارهای متداول داده‌کاوی بیش از پیش نمایان‌گردد. الگوریتم‌های تکاملی راهبرد پرقدرتی است، که توانایی چیرگی بر این مشکلات را در ترکیب با ابزارهای دیگر، در مراحل مختلف داده‌کاوی، به ما می‌دهد. مقاله‌ی حاضر سعی دارد، به مروری کلی بر چگونگی به کارگیری الگوریتم‌های تکاملی در مراحل گوناگون داده‌کاوی بپردازد.

۲) مقدمه

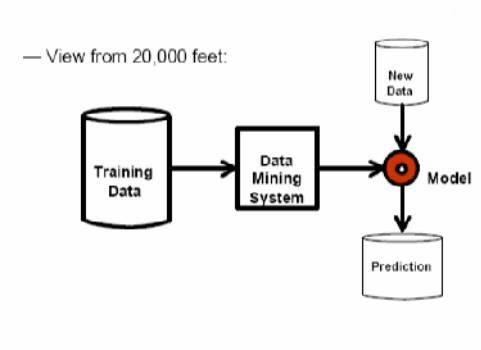
امروزه با وجود حجم بالا و روبه‌رشد پایگاه‌های داده و پدیدارشدن مجموعه‌های داده‌ای ترابایتی و حتا گاهی پتابایتی، یافتن اطلاعات پنهان (اطلاعاتی که بعضاً بسیار مهم و استراتژیک نیز هستند)، در میان داده‌های چنین مجموعه‌هایی، دیگر به آسانی امکان‌پذیر نیست.

داده‌کاوی مجموعه‌ای از ابزارها را در اختیار ما می‌گذارد، تا قادر باشیم، به کشف این اطلاعات پنهان بپردازیم. با گسترش روزافزون به‌کارگیری داده کاوی در کاربردهای مختلف، مشکلات آن نیز به مرور بیش‌تر جلوه‌می‌کند. ابزارهای مختلفی را می‌توان برای غلبه بر این مشکلات جدید به‌کارگرفت. الگوریتم‌های تکاملی ابزارهای پرقدرتی هستند که می‌توانند در حل این مشکلات به‌کارآیند و باعث بهبود کارایی ابزارهایی دیگر گردند.

در این مقاله سعی بر ان داریم که ابتدا شناختی کلّی از داده کاوی به‌دست‌دهیم و بر آن مروری داشته‌باشیم و سپس در ادامه به بررسی کارکرد الگوریتم‌های تکاملی در مراحل مختلف داده‌کاوی بپردازیم.
امّا باید دانست که بررسی داده کاوی، بی نظرکردن به فرآیند بزرگ‌تری که داده‌کاوی جزئی از آن است، میسر نخواهدشد و ما بدون دانستن کلیات داده‌کاوی در درک مفهوم آن مشکل خواهیم‌داشت. لذا این مقاله پس از مرور داده‌کاوی بررسی گذرایی نیز به کلیات فرآیند استخراج دانش دارد و به بررسی آن می‌پردازد.

۲-۱) مروری کوتاه بر داده‌کاوی

داده‌کاوری فرآیندی است که از ابزارهای مختلف تحلیل داده بهره‌می‌گیرد و به طور خودکار، به کشف الگوها و روابط میان داده‌ها می‌پردازد. سپس با استفاده از این تحلیل‌ها، یک پیش‌بینی معتبر انجام‌می‌گیرد.
در داده‌کاوی ابتدا به ساختن یک مدل پیش‌بینی‌کننده (Predictive Model) مبادرت‌می‌کنیم، که برپایه‌ی الگوهای مشخص‌شده از نتایج شناخته‌شده به‌دست‌آمده‌است. سپس، به‌وسیله‌ی نمونه‌هایی که از مجموعه‌ی منبع داده‌های نمونه، کنارگذاشته‌شده‌بودند، مدل خود را آزمایش نموده و در مرحله‌ی نهایی به بازبینی (verify) عملکرد مدل می‌پردازیم.



شکل۱- شمای کلی آنچه در داده‌کاوی صورت‌می‌گیرد. بدون در نظر گرفتن جزئیات واقعی، و به عبارتی از ارتفاع ۲۰۰۰۰پایی

۲-۲) ۴ تعریف شاخص برای داده‌کاوی

در زیر برخی تعاریف متداول داده‌کاوی را مرورمی‌کنیم:

۱) «کلیفتون»، داده‌کاوی را، استخراج الگوهای مفید یا دانش (و نه جزئی، ضمنی و ناآشکار، ناشناخته و بالقوّه مفید) از مبالغ عظیم داده، می‌داند.
۲) فرآیندی است برای کشف دانش، از پایگاه داده‌ی یکپارچه‌ی بزرگ، در مورد تصمیم‌گیری‌های استراتژیک و تاکتیکی، به وسیله‌ی استخراج اطّلاعاتی که تاکنون ناشناخته بوده و قابل تعقیب هستند.
۳) داده‌کاوی یک استخراج داده‌ی خودکار اطلاهات پیش‌بینی‌کننده‌ی پنهان از پایگاه داده است. «تیرلینگ»
۴) داده‌کاوی از الگوریتم‌هایی با قابلّیت مطالعه‌ی کمیت‌های عظیم داده‌ی پیچیده و رسیدن به تفاسیر و مقایسه‌های شرطی است.

در رابطه با تعاریف بالا روی دو چیز باید تأکیدگردد: نخست قدرت پیش‌بینی‌کنندگی که در داه‌کاوری در پی دست‌یابی به آن هستیم و دوم، انجام این عمل به‌طور خودکار.

امّا در این میان دو نکته حائز اهمیت است: نخست آن‌که داده‌کاوی شما را از متخصصان فن و مدیران مجرب بی‌نیاز نمی‌کند، تنها ابزاری برای سهولت و بهبود کار آن‌ها فراهم‌می‌کند. و نکته‌ی دوم این است که برای قوانین و روابطی که میان داده‌ها در داده‌کاوی کشف‌می شود، لزوماً دلیل طبیعی و عقلانی وجودندارد.

۲-۳) چه چیزهایی داده‌کاوی نیست؟

داده‌کاوی را نباید با بعضی فرآیندها وروش‌های دیگر اشتباه‌گرفت. اغلب مفهوم داده‌کاوی به علّت نزدیکی با بعضی مفاهیم دیگر، همانند انبار داده، OLAP، بصری‌سازی‌داده و عامل‌های نرم‌افزاری، با آن‌ها یکی گرفته‌می‌شود. حال آن‌که ابداً این‌گونه نیست. در این‌جا به بیان تفاوت‌های داده‌کاوی با دو مفهوم دیگر می‌پردازیم: انبار داده و OLAP.

۲-۳-۱) تفاوت‌های داده‌کاوی و انبارداده

غالباً آنچه می‌خواهد مورد داده‌کاوی قرارگیرد، از انبار داده استخراج‌شده و به یک مرکز داده‌ی داده‌کاوی منتقل‌می‌گردد. من انبار داده را قبرستان داده‌های عملیاتی می‌نامم. در انبار داده معمولاً داده‌های عملیاتی مرتّباً ذخیره‌می‌گردد و از این داده‌های مجتمع در پیش‌بینی‌ها و تخمین‌ها استفاده‌می‌گردد. ویژگی خوب انبار داده آن است که داده‌های آن سازگارشده‌اند و جامعیت دارد. فلذا یکی از شرایط لازم برای اطمینان از نتایج داده‌کاوی یعنی جامعیت داده در داده‌های استخراج‌شده از انبارداده، فراهم است. حُسن دیگر انبارداده آن است که داده‌ها در آن تمیزشده ذخیره‌می‌شوند. یکی از مراحل پیش‌پردازشی در داده‌کاوی تمیزنمودن داده‌ها است. این عمل برای داده‌هایی که از انبار داده استخراج‌می‌گردند، مورد نیاز نیست، زیرا قبلاً داده‌ها در ورود به انبارداده تمیزشده‌اند.
تفاوت دیگر این است که لزومی به وجود پایگاه‌داده‌ی فیزیکی جداگانه برای داده‌کاوی نیست. پایگاه‌داده‌ی داده‌کاوی می‌تواند به‌صورت منطقی روی یک پایگاه‌داده‌ی دیگر وجودداشته‌باشد. این درحالی است که انبارداده لزوماً به انبارفیزیکی داده‌ها نیاز دارد. البته چنانچه امکان حضور پایگاه‌داده‌ی داده‌کاوی بر روی پایگاه‌داده‌ی دیگر نباشد، باید یک پایگاه‌داده‌ی جداگانه داشته‌باشیم.
ضمناً باید دانست که برای انجام داده‌کاوی لزوماً به وجود انبار داده نیاز نیست. امّا وجود آن به انجام به‌تر داده‌کاوی کمک‌می‌کند.

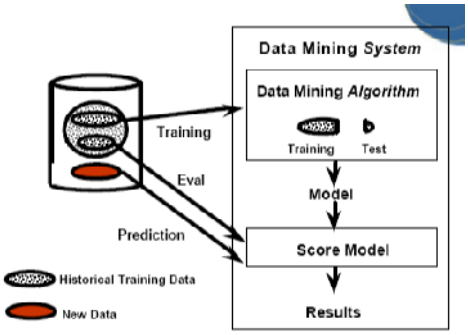
۲-۳-۲) تفاوت‌های داده‌کاوی و OLAP

Olap در واقع به طیفی گسترده از ابزارهای پشتیبانی داده گفته‌می‌شود.در olap یک سری فرضیه مطرح‌می‌گردد، و سپس با استفاده از یک سری پرس‌وجو روی داده‌ها به آن فرضیه‌ها پاسخ‌داده‌می‌شود. یک تفاوت مهم میان olap و داده‌کاوی آن است که olap یک فرآیند استنتاجی است و داده‌کاوی یک عمل استقرائی. امّا باید دانست که olap و داده کاوی یک وجه افتراق اساسی با یک‌دیگر دارند؛ و آن این است که در olap الگویی فرضی و با توجه به اطّلاعاتِ پیشینِ خبره‌ای که با داده‌ها کارمی‌کند، مورد ارزیابی قرارمی‌گیرد و این در حالی است که در داده‌کاوی الگوها در خود داده‌ها جست‌وجو می‌شوند.

اغلب استفاده از olap در مراحل اوّلیه‌ی استخراج دانش صورت‌می‌گیرد. انجام عمل olap گرچه برای استخراج دانش الزامی نیست، ولی در مواردی که فهم داده‌ها مشکل است، ضرورت می‌یابد. در کل olap در فهم داده که نقش به‌سزایی در فرآیند کشف دانش دارد، بسیار اهمّیت‌دارد.

۲-۴) داده‌کاوی چگونه انجام‌می‌شود؟

در شکل ۲ می‌توان شمایی کلّی از آنچه در داده‌کاوی انجام‌می‌شود، را دید. ان‌گونه که می‌بینید، در داده‌کاوی ما داده را به دو قسمت training data و test data تقسیم می‌کنیم. میزان داده‌ی اختصاص‌یافته برای قسمت آزمون بسته به مسائل گوناگون متفاوت است، امّا معمولاً ۱/۳ داده‌ها برای این کار اختصاص‌می‌یابد. مدل طرّاحی شده به‌وسیله‌ی داده‌های آموزشی با داده‌های قسمت آزمون مورد ارزیابی قرارمی‌گیرد. در صورت کامیابی مدل ساخته‌شده برای استفاده و پیش‌بینی در مورد داده‌های جدید به‌کارگرفته‌می‌شود و در صورت عدم موفّقیت، برای آموزش دوباره و ایجاد تغییرات به قسمت یادگیری بازمی‌گردد.



شکل ۲ - تقسیم‌بندی داده‌های آموزشی و چگونگی به‌کارگیری هر یک در فرآیند داده‌کاوی

۲-۶) داده‌های آموزشی

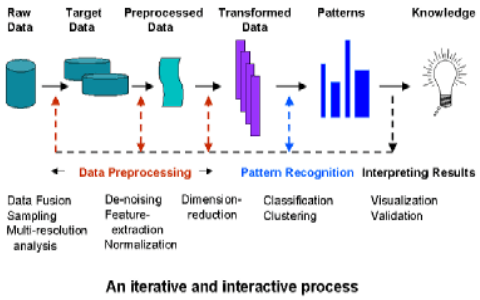
آن‌چنان که پیش از این توضیح‌دادیم، داده‌کاوی از دسته‌ای داده به عنوان داده‌ی آموزشی استفاده‌می‌کند. این داده‌ها ویژگی خاصی ندارند و در واقع همان داده‌های خام پایگاه‌داده هستند، که برای ارزیابی به داده‌کاو محوّل‌شده‌اند. امّا نکته‌ی لازم به ذکر این است که پیش از انجام داده‌کاوی باید عملیاتی روی این داده‌ها صورت‌گیرد. یک‌سری صفات به‌عنوان صفات ورودی داده‌کاوی مشخص‌گردند و در صورتی که یادگیری بانظارت است، یعنی از ابتدا ویژگی داده‌های هر دسته را می‌دانیم، باید صفاتی را به‌عنوان برچسب کلاس و صفات خروجی مشخص سازیم. داده‌ها برحسب این صفات خروجی در دسته‌های مختلف قرارمی‌گیرند. در یادگیری بدون نظارت هیچ صفتی به‌عنوان خروجی نداریم ولذا تعداد دسته‌ها از قبل نامشخص است.

به‌طور کلی داده‌ها را می‌توان به دو نوع categorical و numerical(real-value) تقسیم‌کرد. باید دانست که صفات خروجی در یادگیری نظارت‌شده حتماً بایستی از میان داده‌های categorical انتخاب‌شوند.

Row-id	A	B	C	D	Class label
1	a_1	b_1	c_1	d_1	A
2	a_1	b_2	c_1	d_2	B
3	a_2	b_3	c_2	d_3	A
4	a_1	b_2	c_3	d_3	C
5	a_1	b_2	c_1	d_3	C

Table 1. A training data set.

شکل ۳ – یک نمونه از داده‌های آموزشی



شکل ۴ – مراحل مختلف داده‌کاوی از منظر کاربرد الگوریتم‌های تکاملی

۲-۷) مروری بر چند روش شاخص داده‌کاوی

پیش از آن که به مرور چند روش داده‌کاوی بپردازیم، لازم است توجه‌کنیم، که داده‌کاوی شامل چندین مرحله‌است. در بین مراحل مختلف آن، استخراج ویژگی‌ها از میان داده‌ها، انتخاب چند ویژگی برای داده‌کاوی و بالأخره ایجاد مدل پیش‌بینی‌کننده، با استفاده از الگوریتم‌های دسته‌بندی یا خوشه‌بندی، شاخص‌ترین قسمت‌ها هستند. در این مقاله، پس از بررسی کوتاه الگوریتم‌های داده‌کاوی و مروری بر مراحل استخراج دانش، به بررسی عمیق‌تر کارکرد الگوریتم‌های تکاملی در این مراحل شاخص (که در شکل ۴ دیده‌می‌شوند) خواهیم‌پرداخت.

۲-۷-۱) دسته‌بندی با درخت‌های تصمیم‌گیری

در میان ابزارهای مختلف داده‌کاوی نخست نگاهی گذرا به درخت‌های تصمیم‌گیری خواهیم‌داشت. درخت‌های تصمیم‌گیری شماتیکی ساده از چگونگی ساخت یک مدل شرطی برای طبقه‌بندی داده‌ها هستند. هر گره در درخت تصمیم‌گیری بیان‌گر شرطی است، که روی یک صفت داده اعمال شده و شاخه‌ی مناسب را برای هر نمونه تعیین‌می‌کند. در هر سطح یکی از صفات که مناسب‌ترین تقسیم‌کننده‌ی ممکن است (یعنی به بهترین وجه بتواند نمونه‌ها را از یک‌دیگر جدا کند و شاخه‌های کم‌تری نیز داشته‌باشد.) انتخاب‌شده و بر اساس تقسیم‌بندی آن شاخه‌های زیرین‌اش مشخص‌می‌گردند. چنانچه تمام نمونه‌هی آموزشی یک شاخه، در یک دسته (از میان دسته‌هایی که از پیش مشخص‌گردیده) قرارگیرند، کار در این شاخه به‌پایان‌می‌رسد. در همه‌ی شاخه‌ها کار تا این سطح به‌پیش‌می‌رود و در نهایت مدلی خواهیم‌داشت که داده‌های آموزشی را تماماً به درستی دسته‌بندی می‌کند. حال این درخت به‌دست‌آمده را با داده‌های آزمون، می‌آزماییم و در صورت کسب درجه‌ی تعیین‌شده (که معمولاً با ماتریس

هم‌جوشی تعیین می‌شود.) برای پیش‌بینی به‌کارمی‌رود. البته در میان همه‌ی درختان حائز شرایط، درختی که کم‌ترین سطوح و در هر سطح کم‌ترین شاخه‌ها را داشته‌باشد برگزیده‌می‌شود.

Figure 7. A simple classification tree.

شکل ۵ – شمای یک درخت تصمیم‌گیری ساده

۲-۷-۲) قوانین هم‌باش

قوانین هم‌باش یکی از زیرگروه‌های، نوعی داده‌کاوی به نام تحلیل‌لینک هستند. تحلیل لینک در کل به‌معنی تحلیل روابط موجود بین داده‌هایی است که به‌همراه‌یک‌دیگر می‌آیند. در کل دو نوع تحلیل لینک داریم:

۱) کشف هم‌باشی (تطابق)

که به کشف قانون در مورد چیزهایی که با هم به‌نظرمی‌رسند، می‌پردازد.

۲) کشف روند

که همان کشف تطابق در طول زمان و با توجه به عنصر زمان است.

یک تطابق (هم‌باشی) به صورت کلی $A \rightarrow B$ نشان‌داده‌می‌شود. بدین مفهوم که وجود A، لزوم وجود B را تعیین‌می‌کند. به‌بیان دیگر، چنین قانونی نشان‌می‌دهد که هرگاه A در تاپلی از پایگاه‌داده دیده‌شد، یحتمل B نیز در آن تاپل وجودخواهدداشت. این‌چنین قانونی قدرت پیش‌بینی را در مورد نمونه‌هایی که بعضی ویژگی‌هاشان برای ما مبهم است، فراهم‌می‌کند.

در رابطه با یک هم‌باشی دو چیز مطرح‌می‌شود:

۱) پشتیبانی

که به معنای میزان اتکایی است که در کل پایگاه‌داده بر این قانون وجوددارد و از فرمولی ساده و مشخّص به‌دست‌می‌آید:

تعداد تاپل‌هایی که در طرف چپ قانون‌آمده\ تعدادکل تاپل‌های پایگاه داده

Rule-id	Rule	Support	Confidence
1	$abc \rightarrow A$	80	80%
2	$abcd \rightarrow A$	63	90%
3	$abe \rightarrow B$	36	60%
4	$bcd \rightarrow C$	210	70%

Table 2. Rules found in a training data set.

شکل۶- نمونه‌ای از قوانین هم‌باش

۲) قابلیت اعتماد

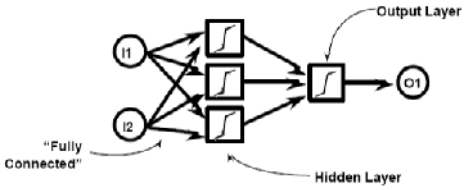
میزان قابلیت اعتمادی که به این قانون می‌توان‌داشت را نشان‌می‌دهد. به‌عبارت دیگر نشان می‌دهد که این قانون در مورد چنددرصد از داده‌ها درست کارمی‌کند. این ویژگی نیز فرمولی ساده دارد:

تعداد تکرار طرف چپ و راست قانون باهم‌دیگر\ تعداد تکرار طرف چپ قانون در کل (به‌همراه و بدون طرف راست)

۲-۷-۳) شبکه‌های عصبی

شبکه‌های عصبی مصنوعی با بهره‌گیری از طبیعت کار شبکه‌ی عصبی حیوانی، ایجادگردیده‌اند. آن‌گونه که می‌دانید، کل ساختار عصبی، از نورون‌هایی تشکیل‌شده که سه قسمت مهم دارند، شاخه‌های ورودی، هسته و شاخه‌ی خروجی. با به‌هم‌پیوستن این عصب‌های ساده، شبکه‌ی پیچیده‌ای به‌وجودمی‌آید که به‌وسیله‌ی آموزش، قادر به یادگیری و فهم‌می‌گردد. شبکه‌های عصبی مصنوعی با الهام از این ایده، سعی بر آن دارند، تا ساختار اعصاب را روی ماشین پیاده‌سازی کرده و آن‌را قادر به فهم سازند.

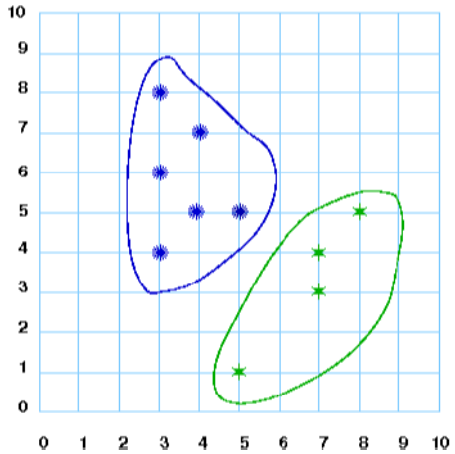
هر شبکه‌ی عصبی یک لایه‌ی ورودی و یک لایه‌ی خروجی و چندین لایه ی پنهان میانی دارد. این لایه‌ها که در هر یک تعداد مشخصی گره‌ی عصبی وجوددارد، به‌وسیله ی اتّصالاتی به یکدیگر متصل‌اند. اتصالات، خروجی یک گره را دریافت کرده، آن را به‌شکل تقویت‌شده و یا تضعیف شده (برمبنای وزنی که هر اتّصال دارد.) به ورودی گره‌ی بعدی انتقال‌می‌دهند. هر گره براساس ورودی‌های دریافت‌شده و تابعی که دارد، خروجی مشخصی خواهدداشت. بنابراین می‌توان صفات ورودی یک دسته را به‌عنوان ورودی شبکه و دسته ی آن را به عنوان خروجی شبکه محسوب‌نمود و شبکه‌ی عصبی را نیز همانند دیگر مدل‌های پیش‌بینی از بیرون دید. شبکه ی عصبی با تغییر وزن اتّصال‌های خود، کاری می‌کند، که میزان اشتباه در تعیین دسته‌ی نمونه‌ها کمینه‌گردد.



شکل۷- شمای کلی یک شبکه‌ی عصبی

۲-۷-۴) خوشه‌بندی

در خوشه‌بندی آیتم‌ها (تاپل‌های) پایگاه‌داده را به گروه‌های متفاوت تقسیم‌می‌کنیم. هدف خوشه بندی یافتن گروه‌هایی است که اعضای انها بسیار مشابه‌یک‌دیگر و ضمناً این گروه با گروه دیگر بسیار متفاوت باشد. برخلاف دسته‌بندی در آغاز عملیات خوشه‌بندی از ویژگی‌های هر خوشه آگاه‌نیستیم. پس از خوشه‌بندی یک فرد آگاه در مورد آن داده‌ها و عرصه‌ی مورد داده‌کاوی، باید به تفسیر نتایج بپردازد. پس از یافتن خوشه‌ها، آن‌ها می‌توانند برای دسته بندی داده‌های جدید مورد استفاده قرارگیرند.



شکل۸- خوشه‌بندی

در این‌جا لازم است، به نکته‌ی مهمی اشاره‌کنیم و آن تفاوت خوشه‌بندی با بخش‌بندی است. بخش‌بندی یک مسئله‌ی کلّی است. مسئله‌ی اصلی در ان تعیین گروه‌هایی است که خصوصیات مشترکی دارند. خواه این خصوصیت مشترک از قبل تعیین‌شده‌باشد و خواه نشده‌باشد.

بخش‌بندی درحالتی‌که خصوصیات گروه‌ها از قبل تعیین‌شده‌است، دسته‌بندی و درحالتی‌که این خصوصیات از قبل تعیین‌نشده‌باشند، خوشه بندی نامیده‌می‌شود.

در پایان به نکات مهمی که در مورد داده‌کاوی بایستی همیشه آن‌ها را به‌خاطرسپرد اشاره‌می‌کنیم:

نخست آن که داده‌کاوی تنها قسمتی کوچک از یک فرایند بسیاربزرگ به نام کشف دانش است. دوم انکه در بسیاری از موارد ضریب قابلیت‌اعتماد مهم‌ترین معیار درستی و کارایی داده‌کاوی نیست. و سوم انکه در داده‌کاوی این خود داده است که اهمیت دارد. بسیاری گمان می‌کنند که در داده‌کاوی الگوریتم‌ها مهم هستند، درحالی که این‌چنین نیست. در داده‌کاوی الگوریتم مناسب با توجه به نوع داده و تشخیص خبرگان فن در مورد خصوصیات آن تعیین‌می‌شود و در مورد داده های مختلف،الگوریتم‌های متفاوتی به‌ترین انتخاب هستند. به‌ترین الگوریتم با توجه به داده‌ای که مورد داده‌کاوی قرارمی‌گیرد، تعیین‌می‌گردد.

در ادامه ابتدا به بررسی فرایند استخراج دانش می‌پردازیم و فرآیندی کلی که داده‌کاوی تنها مرحله‌ای کوچک از آن را دربرمی‌گیرد را برمی‌رسیم. سپس به بیان بخش اصلی مقاله، یعنی کارکردهای الگوریتم‌های تکاملی در مراحل مختلف داده‌کاوی می‌پردازیم.

۳) KDD و فرآیند استخراج دانش

۳-۱) پیشینه‌ی KDD

در سال ۱۶۳۰ سر فرانسیس بیکن، برای نخستین بار روش علم تجربی را در کتاب‌اش «ارغنون جدید» توضیح‌داد. او آن را یک فرآیند چهار مرحله‌ای توصیف‌نمود، مراحلی که آن‌ها را می‌توان به‌صورت زیر خلاصه‌کرد:

۱) تعریف مسئله‌ای که باید حل‌شود.

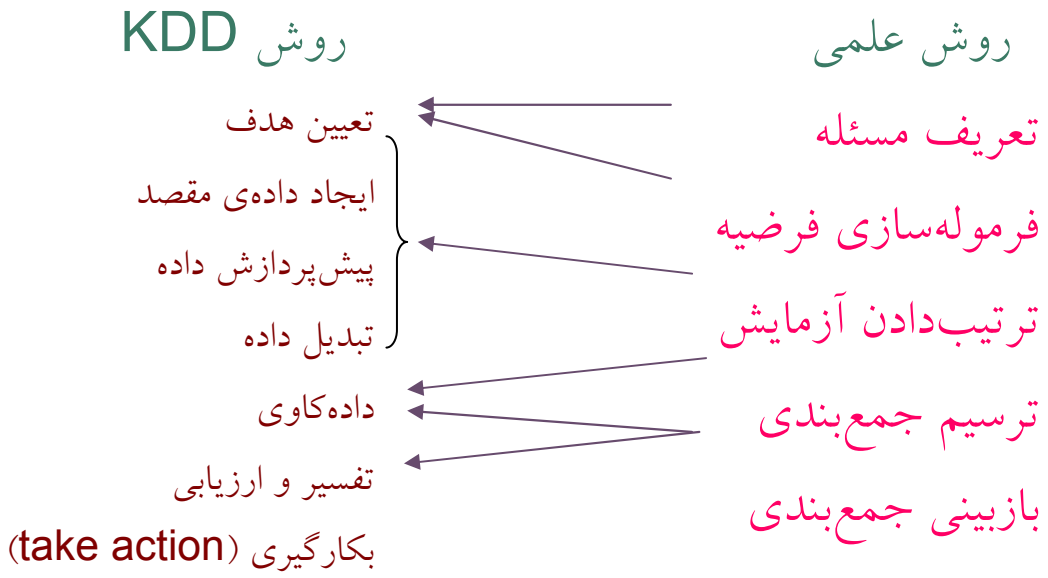
۲) فرموله‌سازی یک فرضیه در مورد آن مسئله

۳) ترتیب‌دادن یک یا چند آزمایش برای تأیید یا تکذیب فرضیه

۴) ترسیم نتایج و بازبینی

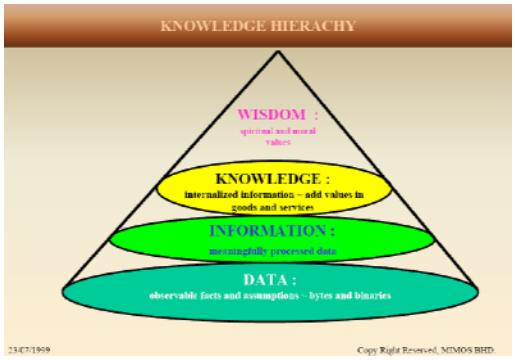
ویژگی خاص این روش آن است که برخلاف روش‌های علمی قدیمی که همه روش‌هایی استنتاجی بودند، یک روش استقرائی است و اجازه‌می‌دهد که دانشی را که از طریق ارتباط با طبیعت و دانسته‌های محدود از یک فضای کلّی به‌دست‌می‌آوریم را در مقوله‌ی علم بگنجانیم و قوانین به دست‌آمده از این روش را معتبربدانیم. در مدل استخراج دانش کامپیوتری، از این مدل معرفی‌شده درحدود ۴۰۰ سال پیش اقتباس‌شده و مدلی جدید برای استخراج چنین دانشی این بار از داده‌هایی که در پایگاه‌های داده هستند، پیشنهادشده‌است.

در روش جدید ۴ مرحله ی روش علم تجربی به ۷ مرحله‌ی استخراج دانش تصویرشده‌است.



شکل۹- چگونگی تصویرشدن علم تجربی بر فرآیند استخراج دانش

این مدل با گردشی bottum-up در سلسله مراتب دانش (که در شکل ۱۰ نشان‌داده‌شده است.) به پی‌جویی دانش و به‌دست‌آوردن دانش جدید می‌پردازد.



شکل ۱۰- سلسله‌مراتب دانش

۳-۲) مراحل استخراج دانش

مراحل مختلف KDD به این شرح است:

۱) **تعیین هدف**

که خود شامل مراحل زیر می‌باشد:

- نخست باید یک بیان واضح از مسئله فراهم‌آوریم.
- در مرحله‌ی بعد باید تخمین هزینه‌ی پروژه انجام‌شود.
- سپس نوبت تخمین زمان پروژه می‌رسد.
- در نهایت ارائه‌ی طرحی برای نگه‌داری سیستم را در این مرحله‌داریم.

۲) ایجاد یک مجموعه‌ی داده‌ای نمونه

از میان داده های انبار داده، داده‌های عملیاتی و گاهی فایل‌های ساده دسته‌ای از داده‌ها بـه عنـوان داده‌ی نمونـه برای آموزش و تست مدل برگزیده‌می‌شوند.

۳) پیش‌پردازش داده

۱) متناسب‌ساختن داده‌های مغتشش

این عمل در چند مورد باید انجام گیرد:

۱) جایی که داده‌ی تکرارشده داریم.
۲) جایی که صفات غیرصحیح داریم.
۳) در مورد داده‌های ناهم‌وار
۴) داده‌های جداافتاده

۲) تصحیح داده‌های گم‌شده

چند راهکار در قبال داده‌های گم‌شده وجوددارد:

۱) ازبین‌بردن رکوردی که داده‌ی گم‌شده در آن وجود دارد.
۲) جایگزینی داده‌ی گم‌شده با میانگین آن صفت در تمام نمونه‌ها
۳) استفاده از مقدار آن صفت در نزد شبیه‌ترین نمونه به نمونه‌ای کـه دارای داده‌ی گـم‌شـده است.

۴) تبدیل داده

۱) نرمال‌سازی
۲) تبدیل نوع
۳) انتخاب صفات و نمونه‌ها

۵) داده‌کاوی

۱) انتخاب مجموعه‌ی اموزشی و داده‌ی تست از استخر نمونه‌های دردسترس
۲) طرّاحی یک مجموعه از صفات ورودی
۳) اگر اموزش نظارت‌شده است، انتخاب یک یا چند صفت خروجی
۴) انتخاب مقادیری برای پارامترهای یادگیری
۵) استفاده از ابزار داده‌کاوی برای ساخت یک مدل عمومی در مورد آن داده

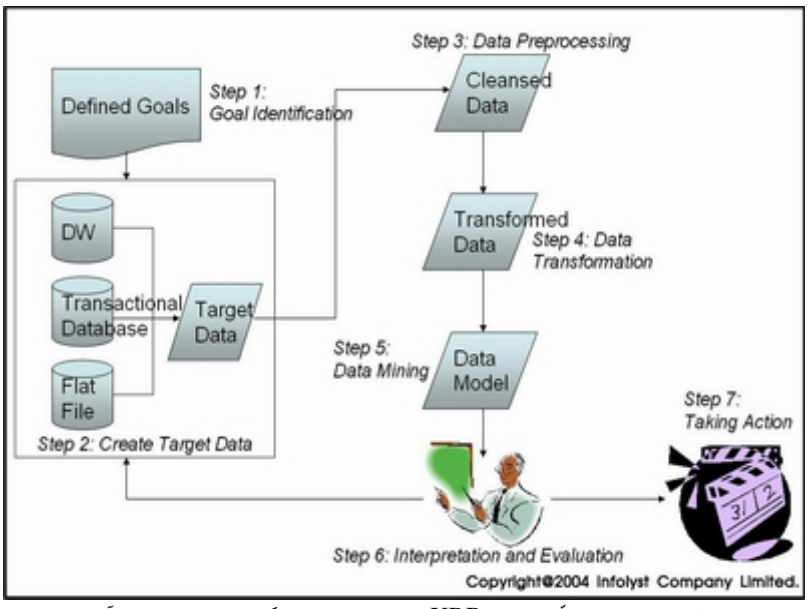
۶) **تفسیر و ارزیابی**

در این مرحله چهار نوع تحلیل روی نتایج به‌دست‌آمده صورت‌می‌گیرد: ۱)تحلیل آماری، ۲)تحلیل اکتشافی، ۳) تحلیل تجربی و ۴)تحلیل انسانی.

۷) **به‌کارگیری**

در مرحله‌ی به‌کارگیری فعالیت‌هایی برای ایجاد امکان استفاده‌ی خوب از مدل به‌دست‌آمده صورت‌می‌گیرد و مثلاً مستنداتی برای استفاده تهیه‌می‌شوند.

آن‌گونه که در شکل ۱۱ می‌بینید، چنانچه در مرحله‌ی ششم، یعنی ارزیابی، مدل موفق نشود کارایی لازم را تذمین‌کند، مراحل داده‌کاوی از مرحله‌ی دوم تا ششم دوباره باید تکرارشوند. برای پرداختن به جزئیات هر یک از مراحل استخراج دانش مجالی دیگر لازم است. در اینجا تنها کلیات مسئله را مورد بررسی قراردادیم.



شکل۱۱ – شمای کلّی مراحل KDD و ترتیب کلی به‌کارگیری مراحل مختلف آن

۳-۳) مدل فرآیند CRISP-DM

کنسرسیومی از چند شرکت تجارتی یک فرآیند را برای ساخت مدل داده‌کاوی استاندارد نمودند. این فرآیند با نام Cross Industry Standard Process for Data Mining خوانده‌می‌شود، که معمولاً به‌شکل مخفف از آن به عنوان CRISP یادمی‌شود. در زیر تنها فازهای کلی آن را نام‌می‌بریم.

شامل ۶ فاز است:

۱) فهم تجارتی

۲) فهم داده‌ای

۳) آماده‌سازی داده

۴) مدل‌سازی

۵) ارزیابی

۶) Deployment و به‌کارگیری

طی مراحلی که crisp معرفی می‌کند و با رعایت جزیات آن‌ها، می‌توانیم داده‌کاوی را در یک فرآیند ازپیش‌مشخص و مهندسی شده به‌انجام‌رسانیم و از نتایج آن اطمینان‌حاصل‌کنیم.

پس از مرور کوتاهی که در این بخش بر کلیات استخراج دانش داشتیم، نوبت آن رسیده که به بحث اصلی این مقاله واردشویم. امّا پیش از آن لازم به نظر می‌رسد که مروری بسیارکوتاه بر چگونگی کار الگوریتم‌های تکاملی و انواع آن‌ها داشته‌باشیم.

۴) مروری بر الگوریتم‌های تکاملی

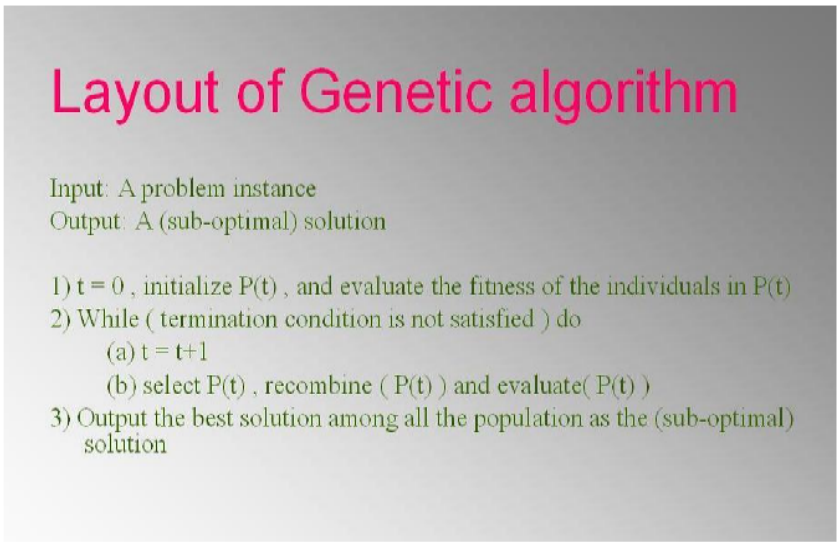
الگوریتم‌های تکاملی رویه‌های جست‌وجوی تصادفی‌ای هستند، که به‌وسیله‌ی مکانیسم ژنتیک و انتخاب طبیعی کار بهینه‌یابی را به‌انجام‌می‌رسانند. الگوریتم‌های تکاملی اغلب در بهینه‌سازی مورد استفاده قرارمی‌گیرند. در کاربردهای داده‌کاوی نیز غالباً کاربرد آن‌ها در همین مورد است.

EAها روی یک جمعیت از افراد کارمی‌کنند. هر فرد حل ممکنی برای یک مسئله است و با یک کروموزوم ارائه‌می شود. جمعیت ابتدایی افراد، می‌تواند تصادفی تولیدگردد و یا آنکه با استفاده از دانش به‌دست‌آمده در حل‌ها یا شناخت‌های قبلی به‌وجودآید. الگوریتم به ارزیابی کروموزوم‌ها می‌پردازد تا مشخص‌کند که هر یک برای حل مسئله چه‌قدر مناسب هستند. این کار با تابع برازندگی انجام‌می‌گیرد. با استفاده از نسل فعلی، افرادی از این نسل برگزیده‌شده و نسل جدید با استفاده از اِعمال عملگرهای ژنتیکی روی این افراد، ایجادمی‌شود. این عمل تا یافتن حل قانع‌کننده دنبال‌می‌شود.

به طور کلی الگوریتم‌های تکاملی را می‌توان در چهار دسته ی کلی قرارداد:

۱) الگوریتم‌های ژنتیکی
۲) برنامه‌سازی ژنتیکی
۳) استراتژی تکاملی
۴) برنامه‌سازی تکاملی

در همه‌ی این‌ها مفاهیم پایه‌ای ثابت است و تنها تفاوت در شکل تصویرکردن مسئله و عملگرهایی است که برای ایجاد نسل بعدی به‌کارمی‌رود.



شکل۱۲- شمای کلّی یک الگوریتم ژنتیک

الگوریتم‌های تکاملی با چندین ورودی کنترل‌می‌شوند:

۱) اندازه‌ی جمعیت

۲) نرخ اعمال جهش و آمیزش

عموماً تضمینی وجودندارد که الگوریتم ژنتیک جواب بهینه را در یک مسئله بیابد، امّا راه‌هایی وجوددارد، که شانس موفّقیت را افزایش‌می‌دهد. راه‌های فراوانی برای تصویرکردن یک حل مسئله بر کروموزوم وجوددارد. هر مسئله، انتخاب مناسب خود را دارد و یک انتخاب، که در همه‌ی مسئله‌ها به‌ترین باشد، وجودندارد. در الگوریتم‌های ژنتیک، عموماً کروموزوم‌ها رشته‌ای از صفر و یک را ارائه‌می‌دهند، امّا تصویرهای دیگری نیز از مسئله می‌توان ارائه‌کرد. در برنامه‌سازی ژنتیکی، حل‌ها به‌صورت برنامه‌های کامپیوتری تصویرمی‌شوند. استراتژی تکاملی و برنامه‌سازی تکاملی از اعداد اعشاری استفاده‌می‌کنند، که برای پارامترهای عملگرهای‌شان، مناسب‌تر است. غالباً شیوه‌ی تصویرکردن[1] مسئله، بستگی به ورودی عملگرها دارد. استراتژی تکاملی[2] و برنامه‌سازی تکاملی[3] بیش‌تر از عملگرهای جهش[4] تصادفی بهره‌می‌گیرند.

۵) نقش الگوریتم‌های تکاملی در داده‌کاوی

الگوریتم‌های تکاملی می‌توانند در بهبود دقّت[5] و استحکام[6] تکنیک‌های قدیمی‌ای که در مراحل مختلف داده‌کاوی، یعنی «استخراج ویژگی»، «انتخاب ویژگی»، «دسته‌بندی» و یا «خوشه‌بندی» مفید و مؤثر باشد.

۵-۱) الگوریتم‌های تکاملی و استخراج ویژگی

فرآیند استخراج ویژگی بسیار وابسته به داده و مسئله است. در بعضی انواع داده، نسبتاً به آسانی می‌توان ویژگی‌ها را مشخّص‌کرد. مثلاً در داده‌ی متنی، کلمات ویژگی اصلی هستند و یا در تحلیل سبدبازار، ویژگی مشخص و مهم، اقلام جنسی هستند که در یک دادوستد، خریداری شده‌اند. در هر مورد، پردازش خاصی برای استخراج ویژگی از میان مجموعه ی داده‌ای خام لازم است.

در مورد داده ی تصویری یکی از مشکل‌ترین قسمت‌ها استخراج ویژگی است. و ضمناً با گسترش فراوان به کارگیری تصویر و داده کاوی‌های تصویری، ایجاد یک تکنیک عمومی، مستحکم و دقیق در مورد این مسئله مهم به‌نظر می‌رسد. الگوریتم‌های تکاملی به صورت بسیار مؤثّری، برای حل این مشکل، کارایی دارند. موارد عمده‌ی استفاده‌ی از الگوریتم‌های تکاملی در داده‌کاوی تصویری را می‌توان در بخش‌بندی تصاویر، تصاویر طبّی، تشخیص لبه‌ها در تصویر و ثبت تصویر دانست.

---

[1] encode
[2] ES
[3] EP
[4] Mutation
[5] Accuracy
[6] Robustness

۵-۲) الگوریتم‌های تکاملی در انتخاب ویژگی

چندین دلیل برای دست‌یازیدن به انتخاب ویژگی وجود دارد، که وجـود آن‌را در غالـب مـواردی کـه از داده‌کـاوی استفاده‌می‌شود، امری ضروری می سازد:

۱) در بسیاری موقیت‌ها، دانستن این‌که کدام‌یک از ویژگی‌های استخراج شده با مسئله‌ای که در دست اسـت، بیش‌تر مرتبط‌اند، کاری ممکن و میسّر نیست.

۲) وجود ویژگی‌های نامربوط نه‌تنها پیچیدگی زمانی اجرای بسیاری از الگوریتم‌ها را افزایش مـی‌دهـد، بلکـه زمان مورد نیاز برای استخراج این ویژگی‌ها از میان داده‌های خام نیز تبدیل به معضلی خاص و زمـان‌گیـر می‌شود.

۳) علاوه‌براین، با توجه به این که ابعاد فضای ویژگی انتخاب‌می شود، درهنگام رودررویـی بـا ویژگـی‌هـای کاهش‌یافته، نمونه‌های آموزشی کم‌تری برای یادگیری مورد نیاز هستند.

۴) ضمن آن‌که امکان دارد برخی از ویژگی‌ها با توجه به تطابق زمانی و مکانی، برای استخراج هزینه‌ی تطبیق زمانی و مکانی نیز داشته باشند. نتیجتاً این موارد باید در طول پروسه ی داده‌کاوی وزن‌دهی شوند.

همه‌ی این موارد ما را به سوی یک مسئله می‌کشاند و آن این است کـه، از میـان مجموعـه‌ی خـصوصیّات داده‌هـا، زیرمجموعه‌ای که برای ارائه ی الگوهای متناسب با مسئله، مناسب‌ترند را، از میان مجموعه‌ی بزرگ ویژگی‌هـا، کـه احتمالاً بسیاری از آن‌ها زائد و بی‌مصرف هستند، انتخاب‌کنیم.

ساده‌ترین راه برای حذف خصوصیات نامرتبط و زائد، به‌کارگیری دانش آن عرصه، توسط خبرگان آن فن می‌باشـد. مثلاً در خوشه‌بندی مستندات متنی، ما جزئیاتی مثل "a","an","the" و... را متغیرهای بی‌ربط قلمدادنموده و آن‌ها را حذف‌می‌کنیم. امّا این کار تنها زمانی امکان‌پذیر است که یک خبـره‌ی دامنـه بهآسـانی بتوانـد صـفات نـامرتبط را تشخیص‌دهد، که این امر در موارد نادری امکان‌پذیر است. تکنیک‌های پیچیده‌تر، مثل تحلیل مؤلّفه‌هـای عمـده نیـز می‌توانند در تعیین ترکیبات خطّی ویژگی‌ها به وسیله ی انتخاب آن‌ها از میان موارد مستقیمی که بیش‌ترین واریانس را دارند، به‌کارگرفته‌شوند. الگوریتم تکاملی برای انتخاب خصوصیت، غالبـاً ترکیـب انتخـاب بـا یـادگیری را مـورد استفاده‌قرارمی‌دهد. (چیزی مانند راهبرد wrapper) در این راهبرد، قابلیت زیرمجموعه‌های ویژگی‌ها، که در طـول محاسبات تکاملی مشخص‌می‌گردد، با استفاده از خود الگوریتم یادگیری ارزیابی‌می‌شود.

نخستین تلاش در این زمینه را [اسکلانسکی و سیدلکی، ۸۹] انجام‌دادند. یک کرومـوزوم، بـا مجموعـه‌ای از اوزان صفر ویک، که در آن یک نشان‌دهنده‌ی حضور ویژگی متناظر آن سلول در دسته بندی و صفر نشان‌گر عدم حـضور آن ویژگی بود، از مجموعه‌ی ویژگی‌ها به وجود می‌آمد. سپس از الگوریتم k-NN برای ارزیابی میزان مناسـبت هـر فرد، براساس دقّت دسته‌بندی آن و تعداد ویژگی‌هایی که مورد استفاده قرارداده‌بـودع بـه‌دسـت مـی‌آمـد. در بقیـه‌ی تحقیقاتی که در این زمینه پس از آن انجام شد نیز، از همین روش تصویر مسئله (صفر و یک، برای حـضور و عـدم حضور هر ویژگی) مورد استفاده قرارگرفت. مـثلا [بریـل، بـرون و مـارتین،۹۰] کـه دسـته بنـدی را بـا اسـتفاده از شبکه‌های عصبی انجام‌دادند و یا [برادرتون و سیمپسون،۹۵].

[پانچ و دیگران،۹۳] ایده ی انتخاب ویژگی دودویی را با ارئه‌ی یک کروموزوم کـه بـا سـری اوزان از صـفر تـا ۱۰ برای هر ویژگی، مشخص‌می شد، توسعه‌دادند. در این راه‌برد بعضی از ویژگی‌ها مهم‌تر از دیگر ویژگی‌ها به حساب می‌آمدند و در داده‌کاوی تعیین‌کنندگی بیش‌تری داشتند. در نهایت پـانچ و دیگـران بـه ایـن نتیجـه رسـیدند کـه در

مواردی که داده‌ها مغتشش هستند، روش وزن‌دهی (صفر تا ۱۰) بسیار به‌تر از روش صفر و یک کارمی‌کند. از آن‌جا که در دنیای واقعی غالباً با داده های مغتشش روبه رو هستیم، این روش در بیش‌تر موراد می‌تواند کارایی به‌تری داشته‌باشد.

[وفائی و دژانگ،۹۸] نیز از یک راهبرد مشابه برای اتخاب خصوصیت در استفاده از درخت تصمیم‌گیری برای دسته بندی، سود جستند. در کارهای آن ها به جای آن‌که وزن‌دهی تنها به هر ویژگی صورت‌یرد، اجازه ی وزن‌دهی به ترکیبی از ویژگی‌های موجود نیز داده‌می‌شد. ترکیب ویژگی‌ها که با استفاده از اعمال ساده‌ای چون جمع و ضرب و تقسیم صورت‌می‌گرفت، می‌توانست ویژگی جدیدی را از میان ویژگی‌های موجود فراهم‌آورد. این تبدیل فضای ویژگی متغیر، کاهش قابل ملاحظه‌ای را در تعداد ویژگی‌ها فراهم می‌اورد و دقت دسته بندی را به نحو چشم‌گیری بهبود می‌داد. از کارهای دیگری که در این رابطه انجام‌گرفته می‌توان به [یانگ و هاناوار،۹۷] که برای ایجاد یک سیستم دسته‌بند، از یک شبکه ی عصبی بهره‌می‌جست و استراتژی ساده ی صفر و یک را برای وزن‌دهی به ویژگی‌ها اختیارکرده بود، اشاره‌کرد.

۵-۳) الگوریتم‌های تکاملی و دسته‌بندی

الگوریتم‌های تکاملی را می‌توان برای بهبود دسته بندی انجام‌گرفته، در ترکیب با ابزارهای متداول دسته‌بندی به‌کاربرد. این عمل باعث قابل اعتمادتر شدن مدل دسته‌بندی ساخته‌شده می‌گردد. در زیر کاربرد الگوریتم تکاملی را در سه دسته ی شاخص از ابزارهای دسته‌بندی، یعنی سیستم‌های مبتنی برقانون، شبکه‌های عصبی و درخت‌های تصمیم‌گیری می پردازیم. اوّلی و سومی را به‌طورکلی برمی‌رسیم و دومی را با جزئیات بیش‌تری مورد بررسی قرارمی‌دهیم.

۵-۳-۱) الگوریتم‌های تکاملی و کاربرد آن ها در سیستم‌های Rule Based

ارائه‌ی مفاهیم به شکل مجموعه‌ای از قوانین، مدت‌هاست که یکی از محبوب‌ترین روش‌ها در یادگیری ماشین می‌باشد. علت این است که قوانین به آسانی می‌توانند به‌صورت دستورات کامپیوتری ارائه شوند و انسان‌ها به آسانی قادر به تفسیر آن ها هستند.

برای استفاده از الگوریتم‌های تکاملی در سیستم‌های Rule Based دو راه‌برد عمده وجوددارد:

۱) راهبرد میشیگان[holland,75] که در [بوکر، هلند و گلدبرگ،۸۹] مورد برسی قرارگرفته است. در این راهبرد هر فرد جمعیت یک قانون با طول ثابتی را ارائه‌می‌کند. در این‌جا همه ی جمعیت برروی‌هم مفهوم را ارائه‌می‌دهند.

۲) راهبرد پیتزبورگ[smith,80,83] که در [دژانگ، اسپیرس و گوردون، ۹۳] مورد شرح و بررسی قرارگرفته‌است. هر فرد در این راهبرد، ارائه کننده‌ی یک مجموعه‌ی کلی از قوانین است و لذا طولی متغیر دارد.

هر دو راهبرد، محاسن و عایب خود را دارند و هریک با موفقیت در برخی انواع سیستم‌های دسته‌بند، به‌کارگرفته‌شده‌اند.

سیستم‌های دسته‌بند، سیستم‌های rule-basedی هستند که الگوریتم‌های تکاملی را با یادگیری تقویتی ترکیب‌می‌کنند. حلقه‌ی اصلی در یک سیستم دسته‌بند، آن است که سیستم با ورودی‌هایی از محیط، مهیا می‌شود،

ورودی‌ها به پیام تبدیل‌می‌شوند. پیام‌هایی که به یک لیست پیام اضافه‌می‌شوند و قوی‌ترین قانونی که با هر پیام منطبق‌شود، انگیخته‌می‌شود. قوانین یک مقدار برازندگی مبتنی بر پاداش برگشته‌ازمحیط دارند. یک الگوریتم ژنتیک به عنوان مؤلّفه‌ی کشف سیستم به‌کارمی‌رود. که قوانین جدید را بر مبنای به‌ترین قوانین فعلی شکل‌می‌دهد. در اینجا فرصت پرداختن به دسته‌بندها [cs] نیست. خواننده‌ی علاقه‌مند می‌تواند به کتاب [گلدبرگ،۸۹] به عنوان مقدّمه‌ای بر cs، یا مقالات [ویلسون، 95 و 2000a] که توسعه‌های مختلف را شرح‌می‌دهند مراجعه‌کند. [ویلسون و گلدبرگ،۸۹] و [ویلسون،2000b] نیز به‌مرور چشم‌اندازی بر تحقیقات انجام‌گرفته بر روی XCS می‌پردازد. سیستم‌های دسته‌بند، معمولاً به‌عنوان سیستم کنترل در محیط‌های متغیر و غیرمطمئن به‌کارمی‌روند. جایی که یحتمل دانش خبره‌ی روشن و کافی وجود ندارد، تا بتوانیم از روش‌های متداول استفاده‌کنیم.[گلدبرگ،1983] .

در مورد مبحث مورد بحث ما، داده‌کاوی، سیستم‌های دسته‌بند، برای آموختن توابع بولی مورداستفاده‌اند. [ویلسون،۹۵] که استفاده‌ی آن‌ها بسیار با اهمّیت است، زیرا توانایی یادگیری مفاهیم پیچیده‌ی غیرخطی را برآورده‌می‌سازند. کاربردهای دیگری شامل دسته‌بندی نامه‌ها [فری،۹۱] و تشخیص سرطان پستان[ویلسون،۲۰۰۰] نیز برای آن‌ها ارائه‌شده‌است.

طرف چپ قوانین در این سیستم‌ها یک اجتماع از عبارات است. این امر قدرت توصیفی این قوانین را در قیاس با (مثلاً) منطق درجه‌ی اوّل، بسیار بالامی‌برد. مشکل منطق درجه‌ی اوّل چه در راهبرد معین و چه در راهبرد اکتشافی، ماندن در بهینه‌های محلّی است. برای حل این مشکل می‌توان از الگوریتم تکاملی بهره‌گرفت.در اینجا مسئله‌ی عمده پیدا کردن فرمی برای قوانین است، که عملگر تکاملی بتواند روی آن‌ها به صورت مؤثری کارگرافتد. و بدین سان پیش‌رفت قوانین محسوس باشد.

۵-۳-۲) الگوریتم‌های تکاملی و شبکه‌های عصبی

دو روش عمده برای استفاده از الگوریتم‌های تکاملی و شبکه‌های عصبی به‌همراه‌یک‌دیگر وجوددارد:

۱) الگوریتم تکاملی در آموزش شبکه‌ی عصبی مورداستفاده‌قرارگیرد.

به‌خصوص، از الگوریتم‌های تکاملی در یافتن اوزان یال‌های شبکه‌ی عصبی و جست‌وجو برای مقدار مناسب پارامترهای یادگیری استفاده‌می‌شود.

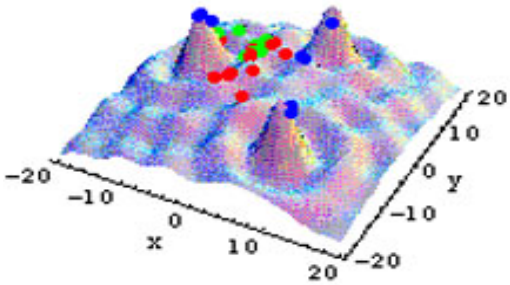
۲) نوع عمده‌ی دیگرِ همکاری الگوریتم‌های تکاملی و شبکه‌های عصبی، استفاده از الگوریتم‌های تکاملی در طرّاحی شبکه‌ی عصبی است.

ساختار شبکه‌ی عصبی تأثیر فراوانی بر کارائی آن و توانایی‌اش در حل مسائل دارد. این امر مشهور است که در مسائل جداسازی غیرخطّی شبکه حداقل به یک لایه‌ی میانی نیاز دارد. امّا تشخیص تعداد و اندازه‌ی لایه‌های پنهان، بیش‌تر یک امر آزمون وخطاست. از الگوریتم تکاملی می‌توان در جست‌وجوی این ساختار و پارامترهای آن استفاده‌کرد.

۵-۳-۲-۱) استفاده از الگوریتم تکاملی در یادگیری شبکه‌ی عصبی

نخست باید بدانیم که یادگیری شبکه ی عصبییک مسئله‌ی بهینه‌سازی است، که هدف در آن یافتن مجموعه‌ای از اوزان است، که اندازه‌ی خطا را کمینه‌می‌کند. فضای جست‌وجو ابعاد فراوانی دارد و دربردارنده‌ی چندین بهینه‌ی محلی است. در روش‌های سنّتی آموزش شبکه، مانند انتشار عقب‌گرد، امکان درجازدن الگوریتم در یک بهینه‌ی

محلی و گیر افتادن در آن وجوددارد. البته این امر را می‌توان با اصلاح پویای پارامتر لاندا بهبود داد، امّا در کل این مشکل برای اغلب الگوریتم‌های انتشار عقب‌گرد وجوددارد. در مقابل الگوریتم ژنتیک با نمونه‌سازی هم‌زمان چندناحیه از فضای مسئله از به‌دام‌افتادن در بهینه‌های محلّی اجتناب‌می‌کند.[شکل۵] این امر بستگی مستقیمی به احتمال جهش در الگوریتم تکاملی دارد. با بالا رفتن احتمال جهش، احتمال گیر افتادن در بهینه ی محلّی کاسته‌می‌شود.



شکل ۱۳- شیوه‌ی کار الگوریتم تکاملی در یافتن جواب بهینه موجب اجتناب از افتادن در بهینه‌ی محلی می‌شود.

یک ترکیب ساده از الگوریتم‌های تکاملی و شبکه‌های عصبی، استفاده از الگوریتم‌های تکاملی در جست‌وجوی اوزانی است که باعث‌می‌شوند، شبکه آن‌گونه که مورد انتظار است، عمل‌نماید. در این‌جا برازندگی هر شبکه با اندازه‌گیری دقّت دسته‌بندی یا رگرسیون روی مجموعه‌ی اموزشی به‌دست‌می‌آید. بنابراین برای هر ارزیابی باید کلّ مجموعه‌ی آموزشی مورد ارزیابی قرارگیرد. اگر مجموعه‌ی آموزشی بزرگ‌باشد، این کار می‌تواند بسیار وقت‌گیر باشد. امّا قابلیت را می‌توان تنها با نمونه‌ی کوچکی از مجموعه‌ی آموزشی نیز تخمین‌زد.

گرچه در کاربردهای مختلف، قابلیت‌ها متفاوت است، الگوریتم‌های تکاملی به‌جست‌وجوی خوب در داده‌های دچار اغتشاش، معروف‌اند.

سه روش مختلف اصلی برای آموزش شبکه ی عصبی به کمک الگوریتم تکاملی وجوددارد:

۱) شروع از یک جمعیت تصادفی و به‌کارگیری اوزانی که توسط الگوریتم تکاملی یافته‌شده‌اند، بدون هیچ‌گونه پالودگی (تقویت) بیش‌تر. از این روش در [کودل و دُلان،۸۹] ، [مانتانا و دیویس،۸۹]، [واتیلی و هانسن،۸۹] استفاده‌شده‌است.

۲) استفاده از روش انتشار عقب‌گرد یا روش دیگری برای پالایش اوزانِ یافت‌شده به‌وسیله‌ی الگوریتم تکاملی. [کیتانو،۹۰]

انگیزه‌ی استفاده از این روش آن است که الگوریتم تکاملی به سرعت می‌تواند نواحی امیدبخش را جست‌وجو کند، امّا در این کار عمل تنظیم پارامترها را به‌کندی انجام‌می‌دهد. بنابراین برای یافتن یک مجموعه‌ی نواحی امیدبخش اولّیه از یک روش مبتنی بر گرادیان (مثلاً روش انتشار عقب‌گرد) بهره‌می‌برد. بنابراین مثلاً در هر دوره‌ی آموزش، در میان داده‌های آموزشی مورد قبول واقع‌شده، از یک انتشار عقب‌گرد استفاده‌می‌شود. و این امر باعث افزایش زمان پردازش‌می‌گردد.

۳) استفاده از الگوریتم تکاملی در پالودن نتایج یافت‌شده به‌وسیله‌ی الگوریتم‌های یادگیری سنّتیِ شبکه‌ی عصبی [کابادا و نایگارد، ۱۹۹۰]. جمعیت اولّیه در این روش توسط الگوریتم انتشار عقب‌گرد، تعیین‌می‌شود.

امّا به‌راستی چرا چنین تلاش‌هایی در ترکیب الگوریتم تکاملی با دیگر روش‌ها صورت‌می‌گیرد؟ نخست آن که، طول کروموزوم‌ها به سرعت با اندازه‌ی شبکه رشدمی‌کند. از آن‌جا که غالباً لایه‌های هم‌جوار در شبکه‌ی عصبی کاملاً متّصل هستند و همه‌ی اوزان در این روش باید ارائه‌گردند، تعداد ایتم‌هایی که باید در هر کروموزوم ارائه‌شوند، با O(n2) متناسب است. (که در آن n تعداد نورون‌هاست.)

علاوه‌براین، عموماً کروموزوم‌های این چنین نیاز به جمعیت بزرگ‌تری برای آموزش دارند، که هزینه‌ی محاسباتی بدین‌ترتیب فوق‌العاده افزایش‌می‌یابد. بنابراین الگوریتم تکاملی دراین حالت فاقد scalability است. و در شبکه‌های عصبی بزرگ کارائی خود را ازدست‌می‌دهد.

عیب دیگری که وجوددارد و اصطلاحاً مشکل جایگشت نامیده‌می‌شود، آن است که اگر ترتیب گره‌های پنهانی جایگشتی باشد، ارائه‌ی وزن‌ها متفاوت خواهدبود. زیرا ترتیب گره‌ها تناسب‌هایی را به‌وجود می‌آورد و الگوریتم تکاملی می‌تواند این ترتیب‌های مناسب گره‌های پنهانی را درهم‌ریزد. برای حل این مسئله نیز راه‌هایی مانند امکان تبدیل ورودی و خروجی گره‌ها به‌یک‌دیگر، پیشنهادشده‌است. هنکُک، در سال ۹۲ براین اعتقاد بود که مشکل جایگشت‌ها به این سختی که‌نشان‌داده‌می‌شود، نبایدباشد. و تیرِنس در سال ۹۵، شیوه‌ای از ارائه‌ی الگوریتم‌های تکاملی به‌وجودآورد، که از مشکل جایگشت کاملاً اجتناب‌می‌کرد.

۵-۳-۲-۲) استفاده از الگوریتم تکاملی در طرّاحی شبکه‌ی عصبی

دو راهبرد کلی برای استفاده از الگوریتم تکاملی در طرّاحی توپولوژی شبکه‌ی عصبی وجوددارد:

۱) استفاده از ارائه‌ی مستقیم، برای مشخّص‌نمودن اتّصالات شبکه

۲) فراهم‌آوردن یک توصیف غیرمستقیم از اتّصالات گره‌های شبکه

شبکه‌ای که در نتیجه‌ی این طّراحی به وجودمی‌آید، می‌تواند با یکی از راه‌های یادگیری شبکه ی عصبی، مورد آموزش قرارگیرد. می‌توان از الگوریتم تکاملی در یافتن پیکربندی و اوزان، به‌صورت هم‌زمان بهره‌برد.

معمولاً شبکه‌ی عصبی به‌صورت یک گراف درنظرگرفته‌می‌شود. که در آن هر گره، نشان‌گر یک نورون و هر یال نمایاننده‌ی یک اتّصال در شبکه ی عصبی است. روش معمول آن است که با استفاده از یک ماتریس دودوئی اتّصال (که در آن عنصر (i,j)اُم برابر یک است، اگر اتّصالی بین گره i-اُم و j-اُم وجودداشته باشد و درغیراین‌صورت صفر است)، گراف را به‌صورت مستقیم تصویرمی‌کنند. به آسانی می‌توان با concat سطرها یا ستون‌های این ماتریس به یک‌دیگر آن‌را به‌صورت یک رشته‌ی دودوئی درآورد.

چنان‌چه توپولوژی شبکه به‌این‌صورت (یعنی با استفاده از الگوریتم تکاملی) تعیین‌شود، فرآیند یادگیری سریع‌تر از حالت عادی طی‌خواهدشد. امّا در این روش نیز طول کروموزوم O(n2) است و هم‌چنان scalability نخواهیم‌داشت.

یک روش برای بهبود آن است که نیروی خود را بر مسئله‌ی خاصّی که روی آن کارمی‌کنیم متمرکزکنیم و الگوریتم یادگیری خاصی برای آن تدوین‌نمائیم. و سپس از الگوریتم‌های تکاملی برای تعیین پارامترهایی که توصیف شبکه را کامل‌می‌کنند، بهره‌گیریم. مثلاً در توپولوژی fully connected feedforward از الگوریتم تکاملی می‌توانیم برای جست‌وجوی تعداد لایه‌ها و گره‌های هر لایه برای به‌ترین کارائی بهره‌گیریم.

یک راهبرد پیچیده‌تر ارائه‌ی غیرمستقیم، استفاده از گرامر برای کدگذاری قوانینی است، که می‌توانند یک شبکه را به‌وجودآورند. کیتانو در سال ۱۹۹۰ نخستین بار راهبرد مبتنی بر گرامر را معرفی‌نمود. بدین سان در کروموزوم‌ها

قوانین نگه‌داری می‌شوند و در موقع ارزیابی از روی این قوانین ماتریس گراف تشکیل‌می‌شود و سپس ماتریس اتّصال برای ساختن شبکه مورد تفسیر قرارمی‌گیرد.

۵-۳-۳) الگوریتم تکاملی و درخت‌تصمیم‌گیری

درخت تصمیم‌گیری یکی از راه‌های محبوبِ دسته بندی است. زیرا به‌آسانی می‌توان آن‌را ساخت و نخبگان به‌آسانی آن‌را تفسیرمی‌کنند. گره‌های داخلی درخت ارائه‌کننده‌ی تستی برای ویژگی‌ها هستند و برگ ها ارائه‌کننده‌ی برچسب دسته اند. یک مسیر از ریشه ی درخت تا یکی از برگ‌ها ارائه‌کننده ی یک اجتماع از تست‌هاست. از آن جا که برنامه‌سازی ژنتیکی به‌طور سنّتی از درخت‌ها برای نمایش بهره‌می‌گیرد، استفاده از آن برای یافتن درخت‌تصمیم‌گیری مناسب است.

۵-۴) استفاده از الگوریتم‌های تکاملی در خوشه‌بندی

دو راهبرد کلی در این زمینه وجوددارد:

۱) هر جایگاه در کروموزوم، ارائه کننده ی یک آیتم از مجموعه‌ی آموزشی است. (این راهبرد را راهبرد «مبتنی‌بربخش‌بندی» می‌نامیم.)

۲) تنها مراکز خوشه‌ها در هر کروموزوم، مورد بررسی قرارگیرند. (راهبرد مبتنی‌برمرکز)

۵-۴-۱) راهبرد مبتنی‌بربخش‌بندی

در این راهبرد هر جایگاه در کروموزوم ارائه‌کننده‌ی یک آیتم از مجموعه‌ی آموزشی است. در این‌جا وظیفه‌ی الگوریتم تکاملی، یافتن خوشه‌ی مناسب برای هر آیتم است. اگر تعداد خوشه‌ها k (ازپیش‌مشخّص) باشد، مقدار هر جایگاه در بازه‌ی [1,k] خواهدبود. مشکل این روش عدمscalability آن است. مثال بارز آن هم [مورتی و چاوداری،۱۹۹۶] است.

صورت تغییریافته‌ای از این روش تصویر مسئله‌ی خوشه‌بندی بر مسئله‌ی بخش‌بندی گراف (Graph Partitioning) است. [پارک و سونگ،۹۸] این تغییر را در نمایش مستقیم دادند. در این راهبرد آیتم‌های مجموعه‌ی داده گره‌های گراف هستند. اندیس نزدیک‌ترین همسایگان ذخیره‌می‌شود. که در این‌جا تعداد همسایگان تابع ورودی مسئله است. اگر آیتم i، دارای مقدار j باشد، یعنی یک اتّصال در گراف بین iوj وجوددارد. مقدارهای یک آیتم، نشان‌دهنده‌ی اندیس نزدیک‌ترین همسایگان آن است. کار پارک و سونگ روی یک فرهنگ لغت جامع بود و نتایج الگوریتم خود را با الگوریتم‌های خوشه‌بندی دیگر مقایسه‌کردند. آن‌ها به این نتیجه رسیدند که لزومی به تعیین تعداد خوشه‌ها در این روش وجودندارد.

در این مسئله نیز هم‌چنان مشکل scalability وجوددارد. زیرا هم‌چنان طول هر کروموزوم به تعداد آیتم‌های مجموعه‌ی داده بستگی دارد و محاسبه‌ی نزدیک‌ترین همسایگان یک آیتم، برای ابعاد زیاد عملی دشوار است.

۵-۴-۲) راهبرد مبتنی‌برمرکز

در [هال، اوزیوت و بازدک، ۱۹۹۹]، راهبردی تکاملی برای تعدیل مرکز خوشه‌ها معرفی‌گردید. مطالعه ی آن‌ها دربرگیرنده ی هر دو نوع خوشه‌بندی: fuzzy, hard بود. مسئله‌ی کارایی این الگوریتم مشکل‌ساز است و بسیار

زمان‌برتر از الگوریتم‌های معمولی مثلFCM/HCM است. در برابر این عیب، یک حُسن نیز دارد و آن امکان استفاده از آن در جاهایی است، که از هیچ روش بهینه‌سازی متداول دیگری نمی‌توان بهره‌گرفت.

[بابو و مورتی، ۹۳] راهبردی برای استفاده از الگوریتم‌های تکاملی در جست‌وجوی اوّلیه‌ی مراکز خوشه‌ها و سپس استفاده از الگوریتم‌های خوشه‌بندی متداول، کشف‌کردند.

همانند دیگر عرصه‌ها، در اینجا نیز می‌توانیم از دانش عرصه، برای بهبود کارایی بهره‌گیریم. این استفاده در این‌جا به چندین طریق می‌تواند صورت‌گیرد. یک طریق، طرّاحی عملگرهای خاص برای آن عرصه به شکلی است که نسبت به عملگرهای معمول کاراتر باشند. طریق دیگر ترکیب الگوریتم تکاملی با الگوریتم‌های خوشه‌بندی معمول است. [فرانتی ودیگران،۱۹۹۷] هر دوی این روش‌ها را به‌کارگرفتند. آن‌ها ۵ روش متفاوت آمیزش را استفاده‌نمودند که ۳تای آن‌ها اختراع خودشان بود. پس از هر آمیزش، عضو جدید به تکراری از الگوریتم k-mean ئارد می‌گردید. آن‌ها بعدها الگوریتم‌شان را برای تغییر خودکار پارامترها توسعه‌دادند. [کیویجارْوی،۲۰۰۰] در [فرانتیو دیگران،۹۷] ضرورت استفاده از تکرارهای k-mean بررسی و ثابت‌گردیده‌بود. این امر باعث افت سرعت فوق‌العاده‌ی این الگوریتم می‌گردید. بعدها نشان‌داده‌شد که تنها به تعداد اندکی تکرار k-mean برای کارکرد درست و کارای الگوریتم نیاز است. [کریشنا و مورتی، ۱۹۹۹] این الگوریتم را با یک و تنها یک تکرار k-mean اجرا کردند و نتیجه‌ی مطلوب را به دست‌آوردند. یک راهبرد برای بهبود این الگوریتم استفاده از نمونه‌های خوب در شروع کار است. مثلاً می‌توان از نتایج چند تکرار k-mean روی داده‌ها به‌عنوان مراکز اوّلیه‌ی خوشه‌ها استفاده‌کرد.[مورتی و چاداری،۱۹۹۶] از این راهبرد استفاده‌کردند.

حُسن کلّی این روش در این است که scalable است، زیرا طول کروموزوم، بستگی به ابعاد مسئله و تعداد مراکز خوشه‌ها دارد و نه به تعداد نمونه‌های داده‌ی آموزشی. بنابراین روش مبتنی بر مرکز از این لحاظ بر روش مبتنی بر بخش‌بندی ارجحیت دارد. حُسن دیگر این روش آن است که امکان فُرم‌پذیری بیش‌تری برای خوشه‌ها فراهم‌می‌کند و مثلاً قسمت‌های ناهم‌جوار فضای داده می‌توانند به یک خوشه‌تعلّق‌بگیرند.

۵-۵) کارائی الگوریتم‌های تکاملی

الگوریتم‌های تکاملی کارائی خود را در حل مسائل واقعی داده‌کاوی اثبات‌کرده‌اند. خصوصاً در مواردی که داده دچار اغتشاش است و یا به حلی نیازداریم، که مسئله‌ی بهینه‌سازیِ چندمأموریتی را انجام‌دهد، این الگوریتم‌ها کارائی فراوانی‌دارند. امّا در هر حال خالی از عیب نیز نیستند. مهم‌ترین عیبی که در چندین جا و توسّط چندین نویسنده در عرصه‌های مختلف برای آن‌ها ذکرشده، آن‌است که می‌توانند بسیاروقت‌گیر باشند. و تبدیل به گلوگاه زمانی برنامه شوند.

چندین روش برای رفع این مشکل پیشنهادشده‌است:

۱) استفاده از قسمتی از نمونه‌ی آموزشی به‌جای استفاده از همه‌ی آن

۲) نگه‌داری از یک جمعیت متشکل از global fit individuals و به کارگیری آن‌ها در اکثر موارد. (این مورد تنها در کاربردهای کاوش تصویر مطرح است.)

۳) استفاده از پردازش موازی و موازی‌سازی الگوریتم های ژنتیک

۴) استفاده از ارائه‌ی به تری از مسئله با توجه به ویژگی‌های خاص آن و یا طرّاحی عملگرهای مقتضی همان مسئله.

۶) مرور مطالب ونتیجه‌گیری

- ✓ الگوریتم های تکاملی می‌توانند در تکمیل بسیاری از الگوریتم‌های داده‌کاوی موجود مفید و مؤثر باشند.
- ✓ در استخراج ویژگی، گزینش ویژگی، وزن‌دهی ویژگی، یادگیری شبکه‌ی عصبی، یافتن قوانین دسته‌بندی، ساختن درخت‌های تصمیم‌گیری و خوشه‌بندی به‌کارآیند.
- ✓ الگوریتم‌های تکاملی به‌خصوص در هنگامی‌که مسئله شامل بهینه‌سازی یک تابع ناهموار و چند بُعدی(not smooth and differentiable) یا تابعی که در آن مقدار تابع هدف در طول زمان تغییر می‌کند، بسیار مفید فایده‌اند.
- ✓ با این وجود هزینه‌ای که این الگوریتم‌ها برای اجرا دارند بسیار زیاد است، که البته با توجه به پیش‌رفت‌های پردازشی و توانایی‌های موازی‌سازی، امیدواریم این نقیصه در آینده‌ی نزدیک برطرف شود.
- ✓ در این مقاله به مطالعه‌ی کلّی کاربردهای الگوریتم‌های تکاملی در قسمت‌های مختلف داده‌کاوی پرداختیم. سعی بر آن است که در آینده‌ی کاربردهای جزئی مورد مطالعه قرارگرفته و راهکارهای جدیدی در عرصه‌ی مسائل مختلف داده‌کاوی ارائه‌گردد. هم‌چنین نگارنده قصددارد به بررسی امکان موازی‌سازی این راهبردهای مختلف معرفی‌شده در این‌جا، بپردازد.

سپاس‌گزاری و حدیث آخر

در این‌جا لازم می‌دانم از استاد عزیز جناب‌آقای دکتر ناصر قاسم‌آقائی که مسبّب انجام ایت تحقیق و ارائه‌ی آن در درس سیستم‌های خبره و مهندسی دانش بودند، تشکر نمایم. همچنین از آقای بهروز شاهقلی که به مطالعه‌ی بخشی از مقاله پرداخته و راهنمایی‌های لازم را جهت تصحیح آن به‌عمل آوردند، نهایت سپاس‌گزاری را دارم. این مقاله برای درس سیستم‌های خبره و مهندسی دانش تهیه‌گردیده و بیش‌تر مروری بر مراجع معرفی‌شده در قسمت مرجع داشته‌است. ضمناً به‌عنوان ضمیمه‌ی این مقاله ترجمه‌ی دو مقاله از مقاله‌های مرجع (شماره‌ی ۱ و ۴) انجام‌گرفته‌است.

روزی که این موضوع را به عنوانِ موضوع سمینار خود برای درس سیستم‌های خبره تعیین‌نمودم، با توجّه به سابقه‌ی ذهنی‌ای که از داده کاوی و الگوریتم‌های ژنتیک داشتم، می پنداشتم که کاربردهای الگوریتم‌های ژنتیک تنها به بخشی محدود از داده‌کاوی منحصر است و راه کوتاهی در این بررسی پیش روی دارم. امّا بررسی بیش‌تر نشان داد، که این موضوع دامنه‌ی وسیعی را دربرمی‌گیرد، که برای بررسی کامل آن باید ده‌ها‌هزار application و مقاله‌ی معرفی‌شده در ارتباط با آن‌ها مورد مطالعه قرارگیرد. و ناچار تنها به بررسی مباحث عمده‌ی مطرح‌شده در این موضوع پرداخته و فرصت نظر در جزئیات هر کاربرد را پیدانکردم. امّا گستردگی این بحث حُسنی نیز دارد و آن شاخه‌های مختلف تحقیق‌خیزی است که برای بهبود کارکردها و توسعه‌ی کاربردها، در این موضوع وجوددارد. نگارنده‌ی این سطور امیدوار است که این مقاله ابتدای به‌کارگیری این راهکارهای کلی در جهت بهبود کاربردهای مختلف‌باشد.

Main References:

1) Cantú-Paz, E. and Kamath, C., "ON THE USE OF EVOLUTIONARY ALGORITHMS IN DATA MINING"(2001), Center for Applied Scientific Computing,Lawrence Livermore National Laboratory.

2) Two Crows Corporation, “Introduction to data mining and knowledge discoveri”, third edition(1999), www.twocrows.com.

3) Data Mining Techniques,(2001) by Michael Berry and Gordon Linoff.

4) Minaei-Bidgoli B., Punch III W. F. , Using Genetic Algorithms for Data MiningOptimization in an Educational Web-based System (2003) , GARAGe , Michigan State University.

5) Management Information System for Information Age,Third edition(2002), Haag, Cummings, McCubbery,McGraw-Hill Publishing,P 109,154,418.

6) Roiger. R. J., Geatz M. W. , "Data Mining (A tutorial based primer),(2001), Addison-Wesley published.

Other References:

1) Paradolas, P. M. and Pitsoulis, L. and Marridon, T. and Resede, M. G. C.,”parallel Search for combinational optimization: genetic algorithm, simulated annealing, grasp”,(1996), P.1~5.

2) Cantù-Paz. Eric, "A Survay of Parallel Genetic Algorithms", 1996, Department of computer science and Illinois Genetic Algorithms Laboratory, University of Illonois at Urbana-Champaign.

3) Alex A. Freitas , A Survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery (2002) ,Postgraduate Program in Computer Science, Pontificia Universidade Catolica do Parana , Brazil.

4) Goldberg D. E.,"Genetic Algorithms in Search, Optimization, and Machine Learning", 1989, University of Alabama, 26printing,2004, addison-wesley published.

۵) افسانه فاطمی، ″ایجاد یک درخت جست‌وجوی دودویی نزدیک بهینه با استفاده از الگوریتم ژنتیک، ۱۳۸۴، دانشگاه اصفهان.

6) Hsu, H. W. and Cheng, Y. and Gue, H. and Gustafson, S. M., “Genetic Algorithms for Reformulation of Large-Scale KDD Problems with Many Irrelevant Attribiutes”, (2001), research proposal.