برنامه نویسی ژنتیکی بلاکی (BGP)[1] و نقش آن در کشف قوانین و استخراج دانش

کاظم نیک فرجام
دانشگاه آزاد اسلامی – واحد بیرجند
گروه کامپیوتر
Farjam@zohatel.com

چکیده: امروزه از برنامه نویسی ژنتیک به طور موفقیت آمیزی جهت استخراج دانش به شکل قوانین IF-THEN استفاده می شود. در روش های مبتنی بر برنامه نویسی ژنتیک ، جهت کشف دانش از روی داده ها ، افراد را در قالب درختان تصمیم نمایش می دهند در فرآیند تکامل هدف اصلی تکامل بهترین درخت تصمیم (از بین مجموعه درختان تصمیم) می باشند که به وسیله ی آن بتوان داده ها را کلاسه بندی وتوصیف مناسب کرد . بعد از همگرایی برنامه ژنتیک ، از روی بهترین فرد، قوانین استخراج می شود . در روش های جاری برنامه نویسی ژنتیک ، ابتدا افراد بصورت درختان تصمیم کامل در نظر گرفته می شوند.
در این مقاله روش جدیدی براساس برنامه نویسی ژنتیک برای استخراج قوانین ارائه می شود که در آن افراد در ابتدا ساده بوده و فقط شامل یک بلاک می باشند . بلاک های جدید در طی فرآیند تکامل، در صورت نیاز به افراد (درختان تصمیم) اضافه خواهند شد همچنین نتایج حاصل از اجرای این روش روی چند پایگاه داده مختلف با نتایج الگوریتم های C۴.۵ و CN۲ مقایسه شده و مزایا و معایب هر کدام بیان می گردد .

کلمات کلیدی : برنامه نویسی ژنتیک ، استخراج قوانین ، درخت تصمیم ، برنامه نویسی ژنتیکی بلاکی (BGP) – کشف دانش

۱- مقدمه :

امروزه ، ابزارهای استخراج دانش اکثراً ریشه در مباحث هوش مصنوعی دارند این ابزارها ترکیبی از روش هایی همچون شبکه های عصبی ، الگوریتم های ژنتیک ، برنامه نویسی ژنتیک ، منطق فازی ، خوشه بندی و آمار دارد در این مقاله روی برنامه نویسی ژنتیک ، جهت توسعه یک ابزار جدید استخراج دانش تمرکز خواهیم داشت .
در برنامه نویسی ژنتیک جهت استخراج دانش ، ابتدا افراد بصورت درختان تصمیم پیچیده در نظر گرفته می شوند که این باعث استفاده کردن از ساختارهای بزرگ ، پیچیده در ابتدای فرآیند تکامل می شود .
در این مقاله یک روش جدید به نام ساختن بلاک ها [2] معرفی می شود که شرایط [3] و زیر درختان در صورت نیاز به درخت اضافه می شوند . شروع تکامل با جمعیتی از ساده ترین افراد آغاز می شود هر فرد شامل فقط یک شرط (ریشه درخت) یک عملگر باینری و دو قانون ساده است . این افراد ساده مشابه Gp استاندارد تکامل می یابند هنگامیکه سادگی افراد ، پیچیدگی داده ها را نتواند نشان دهد یک بلاک جدید (شرط) به تمام افراد در جمعیت افزوده می گردد .
ادامه مقاله بدین شکل سازماندهی شده ، در بخش بعدی ابزارهای مختلف استخراج دانش معرفی می گردد و انگیزه های روش BGP بیان می شود سپس مقدمه ای بر روش Gp استاندارد داشته سپس روش جدید BGP کامل توضیح داده می شود و نتایج اجرای آن با روش های دیگر استخراج دانش مقایسه و نتیجه گیری انجام می گیرد .

۱- Building Block Geneeic Programming
۲- Building Block
۳- Condition

۲- ابزار های استخراج دانش

ابزارهای استخراج دانش را می توان به دو دسته تقسیم کرد (این تقسیم بندی بر اساس الگوریتم های یادگیری و کلاس بندی انجام شده است) ۱- روش های درخت تصمیم ۲- روش های استقراء قانون (Induction)

روش های درخت تصمیم مانند الگوریتم های ID۳ ، C۴.۵ و C۵ ، یک درخت تصمیم می سازند . روش های استقرار قانون مانند الگوریتم CN۲ که از استراتژی پرتو افکنی [۱] برای استنتاج قوانین استفاده می کنند . اخیراً توسعه ابزارهای داده کاوی جدید با استفاده از شبکه های عصبی NN و محاسبات تکاملی Ec [۲] انجام می گیرد . استفاده از NN در داده کاوی مستلزم بدست آوردن کلاسه بند در ابتدا است که بدین منظور شبکه آموزش داده می شود . بعد از آموزش ، الگوریتم استخراج قانون اجرا می شود و هر دانش کد شده را به دانش معمولی تبدیل می کند . در روش Gp جهت اکتشاف دانش درخت تصمیم ، تکامل پیدا می کند تا از آن بعنوان کلاسه بند [۳] استفاده شود .

در اینجا می توان از ترکیب Gp و استراتژی آبکاری فولاد برای تکامل درختان تصمیم استفاده کرد در طی تکامل تابع برازندگی [۴] دو مرتبه تعریف می شود .

استراتژی صرفه جویی و امساک [۵] اساس کار روش BGP می باشد که در این مقاله توضیح می دهیم نظریه اصل صرفه جویی بصورت زیر است :

۱- تعدد خواهی غیر ضروری نباید انجام گیرد . ۲- انجام کار با بیشتر بی فایده است وقتی بتوان با کم تر آنرا انجام داد . ۳- موجودیت ها بدون ضرورت تکثیر نمی شوند .

در روش BGP جمعیت افراد از درختان تصمیم است که هر درخت تصمیم شامل چندین شرط است در BGP ابتدا جمعیتی از برنامه های ساده که هر برنامه ساده یک گره (Node) است تشکیل می شود . تصمیم جدید (ساختن بلاک جدید) به تدریج در طول تکامل باعث پیچیدگی نمایش برنامه ها می شود .

۳- Gp و ساختن درختان تصمیم

Gp یک الگوریتم ژنتیکی ویژه است که مشابه GA ، GP نیز روی تکامل نوع ژن ها [۶] تمرکز دارد . اختلاف اصلی در استفاده از طرح نمایش نوع ژن ها آنهاست در GA نمایش افراد به کمک رشته ها (Strings) و در Gp افراد بعنوان برنامه های قابل اجرا (بعنوان درختان) نمایش داده می شوند و هدف Gp تکامل برنامه های کامپیوتر به منظور حل برنامه هایی است مشابه GA ، در هر نسل هر برنامه تکامل یافته (فرد) جهت اندازه گیری کارایی برنامه ، اجرا می شود که بر اساس آن تابع برازندگی هر برنامه تعریف می شود .

به منظور طراحی یک Gp نیازمند به تعریف گرامری که بطور دقیق مسئله و همه شرایط آن را بیان کند ، هستیم در این گرامر مجموعه پایانه [۷] و مجموعه تابع [۸] تعریف می شود . مجموعه پایانه (ترمینال ها) شامل همه ی متغیر ها و ثابت های برنامه است و مجموعه تابع شامل همه ی توابعی که روی اعضای مجموعه عمل می کنند . مانند توابع ریاضی ، توابع منطقی و ... می باشد .

ساختارهای تصمیم مانند IF-THEN-ELSE داخل مجموعه توابع معرفی می شود . در نمایش درختی برنامه عناصر مجموعه پایانه به شکل برگ های درخت تکاملی و عناصر مجموعه تابع به شکل گره های داخلی درخت می باشد . در فرآیند داده کاوی ، یک فرد یک درخت تصمیم را نمایش می دهد . هر گره غیر برگ یک شرط و هر گره برگ یک کلاس را نمایش می دهد و مجموعه پایانه ها همه کلاس ها را مشخص می کند . عملگرهای رابطه ای و

۱- Beam
۲- Evolueisnary Computing
۳- Classifier
۴-Fitness
۵- Parsimony
۶-Genotype
۷-Terminal set
۸-Function set

خصوصیات یک برنامه را مجموعه تبع مشخص می کند . قوانین از روی درخت تصمیم توسط همه ی مسیر هـای از ریشه تا گره های برگ استخراج می شود که ترکیب عطفی شرایط در هر سطح درخت برای قوانین ضروری است .
برازندگی درخت تصمیم بر اساس صحت و دقت درخت (یعنی تعداد نمونه هایی که به درستی کلاسـه بنـدی شـده اند) بیان می شود . عملگر ترکیب به طور تصادفی دو زیر درخت از درختان پدر را انتخاب و جـای آنهـا را تعـویض می کند .
عملگر جهش با استراتژیهای مختلفی پیاده سازی می شود که عبارتند از :
جهش توسط هرس کردن [1] درخت : یک گره غیر برگ بطور تصادفی انتخاب و با گره برگی که کلاس آن بیشـتر رخ می دهد جایگزین می شود .
جهش رشدی : یک گره بطور تصادفی انتخاب و با یک زیر درخت که تصادفی تولید شده جایگزین می گردد .
جهش گره : محتویات گره ها جهش داده می شود که ممکن است الف) یک صفت را با یکـی از صـفت هـای دیگـر مجموعه صفات تصادفی جابجا کنیم . ب) عملگر رابطه ای تصادفی بـا یـک عملگـر رابطـه ای دیگـر جـایگزین مـی گردد . ج) تغییر دادن مقادیر آستانه برای بعضی از صفات پیوسته
در GP استاندارد ابتدا درختان تصمیم به صورت کامل در نظر گرفته می شـود کـه در نتیجـه انـدازه ایـن درختـان ممکن است متفاوت باشد در BGP این مشکل بروز نمی کند .

فرضیات مورد نیاز BGP :

مقادیر تمام داده ها و صفات کامل بوده و مقدار پوچ در بانک اطلاعاتی وجود ندارد .
صفات مورد بررسی می تواند به صورت : ۱- عددی گسسته ۲- عددی پیوسته ۳- اسمی و ۴- منطقی باشد .
ابزار کشف دانش ارائه شده ، بر اساس مفهوم ساختن بلاک پایه ریزی می شود . هر بلاک نمایش دهنده یـک شـرط است یا یک گره از درخت . هر بلاک شامل سه قسمت است :< **مقدار آستانه** > < **عملگر رابطه ای** > < **صفت** > < **صفت** > هر یک از صفات موجود در بانک اطلاعاتی می تواند باشد .
< **عملگر رابطه ای** > می تواند مجموعه ی $\{\neq,=,\rangle,\geq,\langle,\leq\}$ برای صفات عـددی و مجموعـه ی $\{\neq,=\}$ بـرای صفات اسمی و منطقی باشد .
< **مقدار آستانه** > می تواند یک مقدار و یا یک صفت دیگر باشد .
هر فرد در ابتدا شامل یک گره و دو گره برگ است کلاس گره برگ وابستگی به نحوه ی توزیع نمونه هـای آموزشـی که بین گره ها بخش شده دارد . BGP با مجموعه ای از نمونه های واقعی جهت آموزش و تست شروع به کـار مـی کند . تابع برازندگی که صحت کلاس بندی درخت تصمیم را نشان می دهد بر اساس فرمول زیر محاسبه می گردد :
برای هر قانون R ، C(i) به شکل زیر تعریف می شود :
اگر نمونه i به درستی توسط قانون R ، کلاس بندی شود C(i)=۱ در غیر اینصورت C(i)=۰
اگر هر قانون R تعداد P نمونه از مجموعه آموزشی را بپوشاند دقت قانون برابر است با :

$$Accuracy(R) = \sum_{i=1}^{p} C(i) / P$$

اگر S نشان دهنده مجموعه قوانین باشد برازندگی هر فرد به صورت زیر محاسبه می شود :

$$\text{Fitness (S)} = \text{Min (Accuracy (R)) for all } R \in S$$

جهت انتخاب درختان پدر برای عمل ترکیب از روش انتخاب تورنمنت استفاده می شود . قبل از انجام عمل ترکیـب P_c = احتمال رخ دادن عمل ترکیب توسط کاربر انتخاب و سپس یک عدد تصادفی r بین صفر و یک تولید می شود چنانچه $r\langle P_c$ عملگر ترکیب انجام می پذیرد بدین صورت که ابتدا از والدین کپی برداری شده و روی نسخه کپـی شده بر اساس نقطه ترکیب که تصادفی انتخاب می شود فرزند نهایی تشکیل می شود .

---

[1]- Prnue

عمل جهش : با سه روش روی عملگرها ، مقادیر آستانه ، هرس زدن پیشنهاد می شود . قبل از انجام عمل جهش P_m احتمال رخ دادن ان توسط کاربر انتخاب و سپس یک عدد تصادفی r بین صفر و یک تولید می شود و چنانچه $r\langle P_c$ عملگر جهش رخ می دهد .

افزودن یک بلاک جدید به درختان طبق شرایط زیر رخ می دهد :

$IF\ ((ad_t + aw_t) - (ad_{(t-1)} + aw_{(t-1)})\langle L)$ Then add-condition

ad_t میانگین عمق درختان تصمیم در نسل جاری t

aw_t میانگین پهنای درختان تصمیم در نسل جاری t

$aw_{(t-1)}$ میانگین عمق درختان تصمیم در نسل قبلی t-۱

$ad_{(t-1)}$ میانگین پهنای درختان تصمیم در نسل قبلی t-۱

مقدار L=۰ فرض می شود . بنابراین اگر عمق و پهنای درختان نسل جاری کمتر از نسل قبل باشد یک شرط جدید اضافه می گردد . شرط جدید به طور تصادفی انتخاب شده و به تمام درختان اضافه می گردد . این شرط جدید جایگزین یکی از برگ ها به طور تصادفی می شود . چون شرط جدید تصادفی تولید می شود احتمال وجود آن شرط از قبل داخل درخت می باشد .

برای تعیین شرایط رضایت بخش مجموع قوانین مناسب ، BGP از شرایطی شبیه تابع دما در فرآیند آبکاری فولاد[۱] استفاده می گردد . در هر بار اجرای الگوریتم برای تابع برازندگی یک دما معرفی می شود . دمای ابتدایی خیلی بالا است با T_0 نمایش می دهیم . با گذشت زمان دما کم می شود . اگر T(t) دمای نسل t ام باشد آنگاه رابطه $T(t) = T_0 - t$ نشان می دهد دما به مرور کم می شود .

مجموعه قوانین S (نشان دهنده یک فرد یا یک درخت) رضایت بخش است اگر :

$IF\ (Fitness\ \ (S)\rangle e^{(c(\frac{trainsize}{T_0}) - c(\frac{trainsize}{T(i)}))})THEN$ مجموعه قوانین S رضایت بخش است .

پارامتر C=۰/۱ و T_0 توسط کاربر به عنوان پارامتر ورودی مشخص می شود . trainsize تعداد نمونه های آموزشی می باشد . در بخش بعد الگوریتم مورد نظر معرفی می شود .

۴- معرفی الگوریتم BGP :

۱) T=۰

۲)جمعیت اولیه را انتخاب کن

۳)P(T) (برازندگی افراد جمعیت) را ارزیابی کن

۴) تا زمانی که شرایط رضایت بخش رخ نداده مراحل زیر را تکرار کن .

۴-۱) T = T+۱

۴-۲) اگر شرایط اضافه کردن شرط جدید رخ داد شرط جدید اضافه شود .

۴-۳) زیر مجموعه ای از جمعیت انتخاب شود .

۴-۴) عملیات ترکیب سازی مجدد (شامل عملگر ترکیب و جهش) روی افراد زیر مجموعه انجام شود .

۴-۵) برازندگی جمعیت T ارزیابی شود . P(T)

۴-۶) اگر بهترین درخت تصمیم پیدا شد ذخیره شود .

۵- پایان حلقه

۶- پایان الگوریتم

۱- Simulated Annealing

۵- مقایسه و ارزیابی الگوریتم BGP

این الگوریتم را روی سه بانک با خصوصیات زیر اجرا کردیم .
بانک یک شامل ۳۴ صفت پیوسته بین صفر و یک بود که از ۳۵۱ نمونه آن ۳۰۰ نمونه جهت آموزش و ۵۱ نمونه جهت تست می باشد . بانک دو شامل ۴ صفت عددی که از بین ۱۵۰ نمونه آن ۱۰۰ نمونه جهت آموزش و ۵۰ نمونه جهت تست می باشد . بانک سه شامل ۸ صفت عددی که از ۷۶۸ نمونه آن ۵۰۰ نمونه جهت آموزش و ۲۶۸ نمونه جهت تست می باشد .

نام بانک	تعداد صفات بانک	نوع صفات	تعداد کل نمونه ها	تعداد نمونه ها جهت آموزش	تعداد نمونه ها جهت تست	تعداد کلاسها
Bank ۱	۳۴	پیوسته [۱،۰]	۳۵۱	۳۰۰	۵۱	۲
Bank ۲	۴	عددی گسسته	۱۵۰	۱۰۰	۵۰	۳
Bank ۳	۸	عددی گسسته	۷۶۸	۵۰۰	۲۶۸	۲
Bank ۴	۱۰	عددی گسسته	۴۳۲	۳۰۰	۱۳۲	۲

با توجه به نتایج حاصل از اجرای الگوریتم BGP و مقایسه آن با C۴.۵ و CN۲ از مزایای روش BGP می توان به دقت بالای آن در کلاس بندی و استفاده از تعداد قوانین کم اشاره کردیم .
الگوریتم های یادگیری C۴/۵ ، CN۲ , BGP با معیارهای زیر با همدیگر مقایسه شده اند .
الف – صحت کلاس بندی مجموعه قوانین (به کمک نمونه های آموزشی)
ب – توانایی تولید (با صحت کلاس بندی داده های تست)
ج – تعداد قوانین در مجموعه قوانین
د – میانگین تعداد شرایط در هر قانون
در مورد معیار اول الگوریتم BGP روی داده های گسسته موفق تر عمل می کند . از نظر تعداد قوانین به کار رفته و میانگین تعداد شرایط در هر قانون نیز این الگوریتم نتایج بهتری را می دهد .

۶- نتیجه گیری و پیشنهادات

در GP استاندارد ابتدا درختان تصمیم به صورت کامل در نظر گرفته می شود که در نتیجه اندازه این درختان ممکن است متفاوت باشد در BGP این مشکل بروز نمی کند . در روش BGP جمعیت افراد از درختان تصمیم است که هر درخت تصمیم شامل چندین شرط است در BGP ابتدا جمعیتی از برنامه های ساده که هر برنامه ساده یک گره (Node) است تشکیل می شود . تصمیم جدید (ساختن بلاک جدید) به تدریج در طول تکامل باعث پیچیدگی نمایش برنامه ها می شود . از مشکلات این الگوریتم می توان به پیچیدگی زمانی بالای آن و مشکلات زیاد در مقابل صفات پیوسته زیاد اشاره کرد جهت بهبود الگوریتم راهکارهای زیر پیشنهاد می شود :
الف – اضافه کردن فاز جستجوی محلی در بهینه سازی مقدار آستانه شرایط
ب – اضافه کردن قوانین معنایی جهت جلوگیری از مقایسه صفات غیر قابل مقایسه
ج – پیاده سازی تکنیکی که بهترین بلاک را جهت اضافه شدن انتخاب کند .

مراجع :

[۱] Hossein Ali Abbas, Ruhul Amin Saker, Charles S,(۲۰۰۴), "Data Mining A Hurestic Approach" , University of New South Wales, Australia . IGP Press.

[۲] D.E Goldberg, " Genetic and Evolutionary Algorithms Come of Age ",

[۳] Dasgupta, D(۱۹۹۹). Infprmation processing in immune system. Springer- Verlag.

[۴] Goldberg, D.(۱۹۹۰)." Genetic algorithm in search, optimization and machine learning". Addison Wesley.

[۵] Holland,j. (۱۹۹۸)." Adaptation in natural and artifical systems|". MIT Press.

[۶] Cooke, D. and Hunt, j.(۱۹۹۵)." Recognising promoter sequence using an Artifical Immune System". AAAI press.

[۷] Baoding liu (۲۰۰۲), " Theory and practice of uncertain programming",A Springer-Verlag Company.

[۸] Stephanie Forrest, (۲۰۰۲), "Information Immune System"

and Genetic Algorithms". LLC Press. .

[۹] Ansrzej Osyczka , (۲۰۰۲) " Evolutionary Algorithms for single and multicriteria Design Optimization " , A Springer – Verlag Company .

[۱۰] اکبرزاده توتونچی محمد رضا ،یغمایی حسین ،" استفاده از سیستم ایمنی مصنوعی فازی برای امنیت شبکه های کامپیوتری "، پنجمین کنفرانس سراسری سیستم های هوشمند مهر ماه ۱۳۸۲ مشهد تهران.