استفاده از الگوریتم‌های ژنتیک برای بهینه‌سازی داده‌کاوی در یک سیستم آموزشی مبتنی بر وب

نویسندگان: بهروز مینایی بیدگلی، ویلیام اف. پانچ III [1]

مترجم: محسن مؤمنی

چکیده

این مقاله راه‌بردی را برای دسته‌بندی[2] دانشجویان، به منظور پیش‌بینی نمرات نهایی آن‌ها، برپایه‌ی ویژگی‌های استخراج‌شده از داده‌های ثبت‌شده، در یک سیستم مبتنی بر وب آموزشی، ارائه‌می‌کند. ترکیبی از دسته‌بند[3]های چندگانه، راه را برای بهبود کارائیِ دسته‌بندی، می‌گشاید. به‌وسیله‌ی وزن‌دهی به بردار ویژگی‌ها، با استفاده از یک الگوریتم ژنتیک، توانستیم دقّت پیش‌بینی را بهینه سازیم و بهبود قابل ملاحظه‌ای را نسبت به دسته‌بندی عادیِ تنها[4] به‌دست‌آوریم. بیش‌ازاین، این تحقیق نشان‌می‌دهد، هنگامی‌که تعداد ویژگی‌ها اندک است، وزن‌دهی به ویژگی‌ها، به‌تر از انتخاب ویژگی‌ها به تنهایی نتیجه‌می‌دهد.

---

[1] Genetic Algorithms Research and Applications Group (GARAGe)
Department of Computer Science&Engineering
Michigan State University
2340 Engineering Building
East Lansing, MI 48824
{minaeibi, punch}@cse.msu.edu
http://garage.cse.msu.edu

[2] classification

[3] classifier

[4] raw classification

بیان مسئله

بسیاری از مؤسّسه[1]های آموزشی در برپایی یک پیشگاه[2] تعلیم‌وتعلّم اینترنتی[3]، تلاش‌می‌کنند. چندین سیستم با قابلیت‌ها و راهبردهای مختلف برای به‌وجوداوردن امکان تحصیل اینترنتی[4] در محیطی آکادمیک توسعه‌یافته‌اند. دانشگاه ایالت میشیگان[5] نیز، در ایجاد قسمتی از این سیستم‌ها را برای فراهم‌آوردن یک فراساختار برای آموزش انلاین[6] پیشگام‌شده‌است. تحقیقی که پیش روی شماست، در بخشی از آخرین سیستم توسعه‌داده‌شده در MSU، با نام LON-CAPA[7] اجراگردیده است.

در LON-CAPA[8]، ما با دو مجموعه‌ی داده‌ای بزرگ سروکار داریم:

۱) منابع آموزشی، از قبیل صفحات وب، نمایش‌ها[9]، شبیه‌سازی‌ها، و مسائل فردی‌ای که برای استفاده به‌عنوان تکالیف منزل، کوئیزها و امتحان‌ها طرّاحی‌شده‌اند.

۲) اطّلاعاتی درباره‌ی کاربران که ایجاد، تغییر، ارزیابی[10]، این منابع را انجام‌می‌دهند و یا از آن‌ها استفاده‌می‌کنند. به‌عبارت دیگر، ما در این‌جا دو مخزن همواره‌درحال رشد[11] از داده داریم.

ما به مطالعه‌ی روش‌های داده‌کاوی‌ای، که در استخراج دانش مفید، از این مجموعه‌های داده‌ای بزرگ (درباره‌ی دانشجویانی که از منابع آموزشی انلاین استفاده‌می‌کنند و مسیر ثبت‌شده‌ی آن‌ها در منابع آموزشی وب) به‌کارمی‌آمدند، پرداختیم.

در این مطالعه، هدف این بوده‌است که، به سؤالات تحقیقی زیر پاسخ‌دهیم:

آیا می‌توانیم دسته‌هایی را برای دانشجویان بیابیم؟ به‌عبارت دیگر، آیا گروه‌هایی از دانشجویان وجوددارند، که به‌طرز مشابهی این منابع را مورد استفاده قراردهند؟ اگر بله، آیا می‌توانیم کلاسی را برای هر دانشجو تشخیص‌دهیم؟ با این اطّلاعات، آیا می‌توانیم به دانشجویان در بکارگیری بهتر منابع، برپایه ی طرز استفاده‌ی آن‌ها به‌وسیله ی دیگر دانشجویانی که در آن گروه بوده‌اند، راهنمایی لازم را انجام‌دهیم؟

امید داریم بتوانیم الگوهای استفاده‌ی مشابهی، در داده‌ی جمع‌شده از طریق lon-capa بیابیم، و نهایتاً قادر باشیم، سودمندترین روش را برای مطالعه‌ی هر آموزنده برپایه‌ی استفاده‌ی کنونی‌اش، پیش‌بینی‌کنیم. از آن پس سیستم می‌تواند، پشنهادهایی را برای هر آموزنده، در زمینه‌ی این‌که چگونه بیش‌ترین پیشرفت را داشته باشد، به او ارائه‌دهد.

۲ تصویرکردن مسئله بر الگوریتم‌های ژنتیک

---

1 institution
2 presence
3 online
4 Online education
5 Mishigan State university (MSU)
6 Online instruction
7 Learning Online Network with Computer-Assisted Personalized Approach
8 See http://www.lon-capa.org
9 demonstrations
10 assess
11 ever-growing pools

الگوریتم‌های ژنتیک نشان‌داده‌اند که ابزارهای مؤثری برای استفاده در داده‌کاوی و تشخیص الگو هستند. [7][10][6][16][5][13][7]. یک جنبه‌ی مهم از الگوریتم‌های ژنتیک در یک بستر آموزشی، کارکرد آن‌ها در تشخیص الگوست. دو راهبرد متفاوت برای به‌کارگیری الگوریتم ژنتیک در تشخیص الگو وجوددارد:

۱) اعمال مستقیم الگوریتم ژنتیک به عنوان یک دسته‌بند[1]. بندیوپدهایای و مورتی در [3] الگوریتم ژنتیک را برای یافتن محدوده‌ی تصمیم در یک فضای ویژگی n-بُعدی به‌کارگرفتند.

۲) استفاده از یک الگوریتم ژنتیک به عنوان یک ابزار بهینه‌سازی برای بازمقداردهی[2] پارامترها در یک دسته‌بند دیگر. بیش‌تر کاربردهای الگوریتم‌های ژنتیک در تشخیص الگوها در بهینه‌سازی برخی پارامترها در فرآیند دسته‌بندی است. تاکنون، بسیاری از محقّقان الگوریتم ژنتیک را در انتخاب ویژگی‌ها به‌کارگرفته‌اند. [18] ,[12] ,[9] ,[2]. الگوریتم‌های ژنتیک در یافتن مجموعه‌ی بهینه از ویژگی‌های وزن‌دار نیز به‌کارگرفته‌شده‌اند، که دقّت دسته‌بندی را فزونی‌می‌بخشد. در این روش، در ابتدا یک روش استخراج ویژگی سنّتی نظیر PCA[3] به‌کارگرفته‌می‌شود و سپس یک ابزار دسته‌بندی نظیر k-NN[4] برای محاسبه‌ی تابع برازندگی برای الگوریتم ژنتیک [19] , [17] استفاده‌می‌شود. ترکیب دسته‌بندها عرصه‌ی دیگری است که از الگوریتم‌های ژنتیک برای بهینه‌سازی آن استفاده‌می‌گردد. کانچیوا و جین در[11] از الگوریتم ژنتیک برای انتخاب ویژگی، همانند انتخاب نوعی از دسته‌بندها در طرّاحی خود از یک سیستم همجوشی دسته‌بندها، بهره‌برده‌اند. الگوریتم‌های ژنتیک هچنین در انتخاب پیش‌گونه[5] در دسته‌بندی مبتنی برمورد[6] استفاده‌شده‌اند[20].

در این مقاله ما بر راهبرد دوم متمرکزمی‌شویم و از الگوریتم ژنتیک برای بهینه‌سازی ترکیبی از دسته‌بندها بهره‌می‌جوییم. مأموریت ما، پیش‌بینی نمرات نهایی دانشجویان، بر اساس ویژگی‌های بهره‌گیری از وب آن‌هاست، که از میان داده‌های تکلیفی استخراج‌می‌گردد. ما مجموعه‌ای از دسته‌بندهای الگو را با پارامترهای مختلف طرّاحی، پیاده سازی و ارزیابی نمودیم، تا کارایی آن‌ها را در مجموعه‌ی داده‌ای بدست‌آمده‌از لن-کاپا، مقایسه‌کنیم. نرخ خطاها برای دسته‌بندهای منفرد، ترکیب‌های آن‌ها و ترکیب بهینه‌شده توسّط الگوریتم ژنتیک ارائه‌شد.

۲-۱ مجموعه‌های داده و برچسب‌های دسته

به عنوان داده‌ی ازمایشی ما داده‌های دانشجویان و دروس، درس فیزیک۱۸۳(فیزیک رشته‌های علوم‌پایه و فنّی‌مهندسیI)، لُن-کاپا را انتخاب‌کردیم، که در ترم بهاره سال ۲۰۰۲ در دانشگاه میشیگان، برقراربوده‌است. در این درس ۱۲ مجموعه تکلیف شامل ۱۸۴ مسئله، جمع‌گردید، که همه‌ی آن‌ها انلاین بودند. حدود ۲۶۱ دانشجو از لُن-کاپا برای این درس استفاده‌کرده‌بودند. بعضی از دانشجویان پس از انجام دومجموعه تمرین این درس را حذف‌کردند، که بنابراین آن‌ها هیچ نمره‌ی پایانی‌ای نداشتند. پس از حذف این دانشجویان، ۲۲۷ نمونه‌ی معتبر باقی‌ماند. توزیع نمرات دانشجویان در شکل ۱ نشان‌داده‌شده‌است.

---

[1] classifier
[2] resetting
[3] Principal Component Analysis
[4] K nearest neighbor
[5] prototype
[6] case-based classification

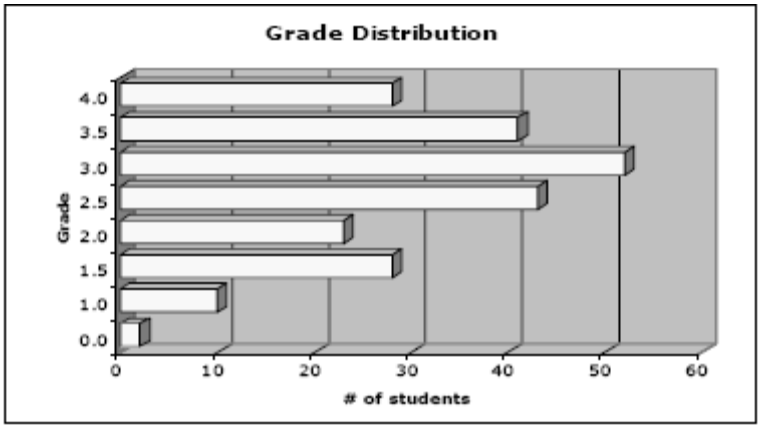

Fig. 1. Graph of distribution of grades in course PHY183 SS02

می‌توانیم دانشجویان را براساس نمرات نهایی آن‌ها به چندین روش گروه‌بندی کنیم، سه تا از این روش‌ها عبارت‌اند از:

۱. اجازه‌دهیم هر ۹ برچسب دسته‌ی ممکن، همان نمرات دانشجویان باشد، آن‌گونه که در جدول۱ نشان‌داده‌شده‌است.

۲. می‌توانیم برچسب دانشجویان را مرتبط با نمره‌ی آن‌ها و گروهی که آن‌ها در سه کلاس، نمرات «بالا»، که نشان‌دهنده‌ی نمره‌ی بین ۳/۵ تا ۴ است، «متوسط» که ارائه‌دهنده‌ی نمره‌بین ۲/۵ تا ۳ است و «پایین» که نشان‌گر نمرات کمتر از ۲/۵ است، دارند، تعیین‌کنیم. جدول ۲ را ببینید.

۳. هم‌چنین می‌توانیم دانشجویان را با یک یا دو برچسب کلاس طبقه‌بندی‌کنیم. «قبول» برای نمرات بالاتر از ۲ و «رد» برای نمرات کم‌تر یا برابر ۲، آن‌گونه که در جدول ۳ نشان‌داده‌شده‌است.

Table 1. Selecting 9 class labels regarding to students' grades in course PHY183 SS02

Class	Grade	Student #	Percentage
1	0.0	2	0.9%
2	0.5	0	0.0%
3	1.0	10	4.4%
4	1.5	28	12.4%
5	2.0	23	10.1%
6	2.5	43	18.9%
7	3.0	52	22.9%
8	3.5	41	18.0%
9	4.0	28	12.4%

Table 2. Selecting 3 class labels regarding to students' grades in course PHY183 SS02

Class	Grade	Student #	Percentage
High	Grade >= 3.5	69	30.40%
Middle	2.0 < Grade < 3.5	95	41.80%
Low	Grade <= 2.0	63	27.80%

Table 3. Selecting 2 class labels regarding to students' grades in course PHY183 SS02

Class	Grade	Student #	Percentage
Passed	Grade > 2.0	164	72.2%
Failed	Grade <= 2.0	63	27.80%

می‌توانیم پیش‌بینی کنیم که نرخ خطا در گروه‌بندی رده‌ی اوّل، باید بیش از دیگران باشد، زیرا اندازه‌ی نمونه‌ی نمرات بر ۹ کلاس تفاوت نسبتاً زیادی دارد. واضح است که ما داده‌ی کم‌تری برای ۳ کلاس نخست در فاز آموزش خواهیم‌داشت، و بنابراین نرخ‌خطا در فاز ارزیابی یحتمل بالاتر خواهدبود.

۲-۲ استخراج ویژگی‌ها

یکی از مراحل حیاتی در انجام دسته‌بندی، انتخاب ویژگی‌هایی است که برای دسته‌بندی به‌کارمی‌رود. در زیر ما درباره‌ی ویژگی‌های بدست‌امده از لُن-کاپا که مورد استفاده قرارگرفته‌اند، بحث‌خواهیم‌نمود، که چگونه می‌توان آن‌ها را متجسم‌نمود (تا به کار انتخاب کمک‌شود.) و چرا ما داده‌ها را پیش از دسته‌بندی به‌هنجار[1] می‌کنیم. ویژگی هایی که در پی می‌آید، به‌وسیله‌ی سیستم لُن-کاپا ذخیره‌شده‌اند:

۱. نمره‌ی کلّ پاسخ‌های صحیح (نرخ کامیابی)

۲. فهم درست مسئله در تلاش اوّلیه، در برابر آن‌ها که با تعدادی تلاش نمره‌ی بالاکسب‌کرده‌اند. (کامیابی در نخستین تلاش)

۳. تعداد کل تلاش‌ها برای انجام تکلیف. (تعداد تلاش‌ها پیش از به‌دست‌آمدن پاسخ صحیح)

۴. زمان صرف‌شده برای مسئله پیش از آن‌که حل‌شود (به‌عبارت دقیق‌تر، تعداد ساعات تا رسیدن به پاسخ صحیح. تفاوت زمان رخداد کامیابی و نخستین زمانی که مسئله آزمایش‌می‌گردد.). هم‌چنین، زمانی که هر دانشجو به پاسخ صحیح مسئله می‌رسد، با توجّه به روز محاسبه‌می‌گردد. معمولاً دانشجویان به‌تر زودتر تکالیف را تکمیل‌می‌کنند.

۵. کل زمانی که بر مسئله صرف‌می‌شود، جدای از این‌که دانشجو به پاسخ صحیح برسد یا نه. (تفاوت زمان میان آخرین ارائه‌ی پاسخ و شروع زمان ارائه‌ی مسئله برای آزمون).

۶. شرکت در مکانیسم ارتباط، به جای کار تک نفره. لُن-کاپا امکان تعامل انلاین را هم با دانشجویان دیگر و هم با مربّی، فراهم‌می‌سازد. کجا باید استفاده‌گردد؟

۷. مطالعه‌ی مواد پشتیبانی پیش از تلاش برای حل تکلیف، به‌جای تلاش ابتدایی برای حل و سپس بازگشت به خواندن آن.

۸. ارسال تلاش‌های فراوان در زمان کوتاه، بدون توجه به موارد درون آن، به‌جای دادن آن با یک تلاش، بازخوانی، ارسال دوباره و به‌همین‌صورت.

۹. دست‌برداشتن از یک مسئله به جای دانشجویانی که تلاش را بر یک مسئله تا آخرین مهلت صرف‌می‌کنند.

۱۰. زمانی که نخستین بار کار را شروع‌می‌کنند. (شروع ثبت نام، میان هفته، یا آخرین دقایق!)برمبنای تعداد تلاش‌ها یا تعداد مسائل حل‌شده. یک دانشجو که همه‌ی جواب‌های صحیح را می‌دهد، لزوماً در گروه کامیابان دسته‌بندی نمی‌شود، اگر آن‌ها به‌طور متوسط برای هر مسئله ۵ تلاش داشته‌باشند، امّا این مسئله باید در این تحقیق مورد بازخوانی قرارگیرد.

در این مقاله ما بر ۶ ویژگی نخست در مجموعه‌ی داده‌ای درس فیزیک ۱۸۳، که برای آزمون دسته‌بندی برگزیده‌شده‌است، بهره‌می‌گیریم.

۲-۳ دسته‌بندها

---

[1] normalized

تشخیص الگوها کاربردهای متنوّعی در عرصه‌های مختلف دارد، بدین‌دلیل امکان‌ندارد ابزار دسته‌بندی‌ای تهیه‌کنیم، که در همه‌ی موارد نتایج خوبی را عرضه‌کند. دسته‌بند بهینه در هر مورد به‌شدّت به محدوده‌ی مسئله بستگی‌دارد. در عمل، ممکن است به موردی برخوریم که هیچ ابزار دسته‌بندی‌ای به‌تنهایی نتواند، دسته‌بندی‌ای با دقّتِ قابل قبول ارائه‌دهد. در چنین مواردی شاید به‌تر باشد که از ترکیب نتایج دسته‌بندهای متفاوت برای دست‌یابی به دقّت بهینه استفاده‌کنیم. هر دسته‌بند با جنبه‌های متفاوتی از بردار ویژگی در پرورش و آزمون کارکند. نتیجتاً، به‌فرض وجود شرایط مناسب، ترکیب چندین دسته‌بند، شاید به بهبود کارائی دسته‌بندی، در قیاس با یک دسته‌بند به‌تنهایی، بیانجامد. [4]

تمرکز این مرور، به مقایسه‌ی چند دسته‌بند الگوی غیرپارامتریک مشهور و یک دسته‌بند پارامتریک، بر طبق تخمین خطاها، محدودمی‌شود. ۶ دسته‌بند مختلف بااستفاده از مجموعه‌ی داده‌ای لُن-کاپا، در این مطالعه، مورد مقایسه‌قرارگرفته‌اند. دسته‌بندهایی که در این مطالعه مورد استفاده قرارگرفته‌اند، عبارت‌اند از:

Quadratic Bayesian classifier,1-nearest neighbor (1-NN), k-nearest neighbor (k-NN) Parzen-window, multi-layer perceptron (MLP), Decision Tree.[1]

این دسته‌بندها، برخی از دسته‌بندهای معمول هستند، که در بیش‌تر مسائل کاربردی دسته‌بندی مورد استفاده قرارمی‌گیرند. پس از آن که قدری اعمال پیش‌پردازشی روی مجموعه‌ی داده‌ای اعمال‌شد، نرخ خطا در هر دسته‌بند گزارش‌می‌شود. درنهایت، برای بهبود کارایی، ترکیبی از دسته‌بندها ارائه‌می‌شود.

۲-۴ به‌هنجارسازی[2]

در دسته‌بندهای بیز و پنجره‌ی پارزن، فرض بر آن است که ویژگی‌ها دارای توزیع نرمال هستند. به همین‌دلیل لازم است که همه‌ی ویژگی‌ها به‌هنجارشوند. این امر اطمینان‌می‌دهد که هر ویژگی وزن یکسانی را در فرآیند تصمیم‌گیری خواهدداشت. با فرض این‌که، داده‌ی داده‌شده، توزیع گوسی دارد، این به‌هنجارسازی بااستفاده از میانگین و انحراف‌معیار داده‌ی آموزشی[3] به‌جای‌آورده‌می‌شود. به منظور به‌هنجارسازی داده‌ی آموزشی، نخست لازم است میانگین نمونه محاسبه‌گردد و انحراف معیار هر ویژگی، یا ستون، در مجموعه‌ی داده‌ای، بر مبنای آن تعیین‌شود و سپس داده با استفاده از معادله‌ی (۱) به‌هنجارگردد:

$$x_i = \frac{x_i - \mu}{\sigma} \qquad (1)$$

این امر اطمینان‌می‌دهد که هر ویژگیِ داده‌ی آموزشی توزیع نرمال با میانگین صفر و انحراف‌معیار ۱، داشته‌باشد. به‌علاوه، روش k-نزدیک‌ترین‌همسایگی نیاز به به‌هنجارسازی همه‌ی ویژگی‌ها در یک دامنه‌ی یکسان دارد. به‌هرحال، باید در استفاده از به‌هنجارسازی، پیش از درنظرداشتن تأثیر آن بر کارایی دسته‌بند، برحذرباشیم.

۲-۵ ترکیب چندین دسته‌بند[4] (CMC)

در ترکیب چند دسته‌بند، ما قصدداریم، کارایی دسته‌بند را بهبودبخشیم. راه‌های گوناگونی در ترکیب دسته‌بندها به‌نظرمی‌رسد:

---

[1] The first five classifiers are coded in MATLABTM 6.0, and for the decision tree classifiers we have used some available software packages such as C5.0, CART, QUEST, and CRUISE.

[2] normalization

[3] Training Data

[4] Combination of Multiple Classifiers

- آسان‌ترین راه، یافتن نرخ خطای روی‌هم‌رفته‌ی دسته‌بندها و انتخاب آن ابزار دسته‌بندی‌ای است که حداقل نرخ خطای داده‌شده در مجموعه‌ی داده‌ای را دارد. این راه، یک ترکیب دسته‌بندهای offline نامیده‌می‌شود. این مورد ممکن است فی‌الواقع ترکیبی از دسته‌بندها به‌نظرنرسد؛ ولی درهرحال، عموماً کارائی به‌تری نسبت به یک دسته‌بند به تنهایی دارد.
- روش دوم، که ترکیب دسته‌بندهای انلاین نامیده‌می‌شود، از همه‌ی دسته‌بندها به‌همراه یک رای‌گیری استفاده‌می‌کند. دسته‌ای که بیش‌ترین آراء را، از یک دسته‌بند کسب‌کند، به نمونه‌ی آزمون واگذارمی‌شود. از طریق شهودی، به‌نظرمی‌رسد، که این روش به‌تر از روش قبلی باشد. درهرحال، وقتی بر مواردی از مجموعه‌ی داده‌ای خود این تلاش را انجام‌دادیم، نتایج به‌تر از نتایج روش قبلی نبودند. بنابراین، ما معیار رای اکثریت را از «کسب بیش از ۵۰٪ آراء» به "کسب بیش از ۷۵٪ آراء» تغییر دادیم. این امر منجر به آن شد که بهبود قابل ملاحظه‌ای نسبت به ترکیب‌دسته‌بندهای افلاین به‌دست‌آید.

با استفاده از روش دوم، ما در جدول ۴ نشان‌داده‌ایم که CMC می‌تواند، به بهبود دقّت قابل‌ملاحظه‌ای، در هر سه مورد دسته‌های ۲، ۳ و ۹، دست‌یابد. اینک قصدداریم از الگوریتم ژنتیک، برای کشف این‌که، آیا می‌توانیم کارائی CMC را بیشینه‌کنیم، یا نه؟، استفاده‌کنیم.

۳ بهینه‌سازی CMC با استفاده از الگوریتم ژنتیک

3 Optimizing the CMC Using a GA

We used GAToolBox3 for MATLAB to implement a GA to optimize classification performance. Our goal is to find a population of best weights for every feature vector, which minimize the classification error rate.

The feature vector for our predictors are the set of six variables for every student: Success rate, Success at the first try, Number of attempts before correct answer is derived, the time at which the student got the problem correct relative to the due date, total time spent on the problem, and the number of online interactions of the student both with other students and with the instructor.

We randomly initialized a population of six dimensional weight vectors with values between 0 and 1, corresponding to the feature vector and experimented with different number of population sizes. We found good results using a population with 200 individuals. The GA Toolbox supports binary, integer, real-valued and floatingpoint chromosome representations. Real-valued populations may be initialized using

3 Downloaded from http://www.shef.ac.uk/~gaipp/ga-toolbox/

the Toolbox function *crtrp*. For example, to create a random population of 6 individuals with 200 variables each: we define boundaries on the variables in *FieldD* which is a matrix containing the boundaries of each variable of an individual.

```
FieldD = [ 0 0 0 0 0 0; % lower bound
1 1 1 1 1 1]; % upper bound
```

We create an initial population with `Chrom = crtrp(200, FieldD)`, So we have for example:

```
Chrom = 0.23 0.17 0.95 0.38 0.06 0.26
0.35 0.09 0.43 0.64 0.20 0.54
0.50 0.10 0.09 0.65 0.68 0.46
0.21 0.29 0.89 0.48 0.63 0.89
……
```

We used the simple genetic algorithm (SGA), which is described by Goldberg in [9]. The SGA uses common GA operators to find a population of solutions which optimize the fitness values.

3.1 Recombination

We used "*Stochastic Universal Sampling*" [1] as our selection method. A form of stochastic universal sampling is implemented by obtaining a cumulative sum of the fitness vector, *FitnV*, and generating *N* equally spaced numbers between 0 and sum(FitnV). Thus, only one random number is generated, all the others used being equally spaced from that point. The index of the individuals selected is determined by comparing the generated numbers with the cumulative sum vector. The probability of an individual being selected is then given by

$$F(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N_{ind}} f(x_i)} \qquad \textbf{(2)}$$

where *f(xi)* is the fitness of individual *xi* and *F(xi)* is the probability of that individual being selected.

3.2 Crossover

The crossover operation is not necessarily performed on all strings in the population. Instead, it is applied with a probability *Px* when the pairs are chosen for breeding. We selected *Px* = 0.7. There are several functions to make crossover on real-valued matrices.

One of them is *recint*, which performs intermediate recombination between pairs of individuals in the current population, OldChrom, and returns a new population after mating, NewChrom. Each row of OldChrom corresponds to one individual. *recint* is a function only applicable to populations of real-value variables. Intermediate recombination combines parent values using the following formula [14]:

$$Offspring = parent1 + Alpha \times (parent2 - parent1) \qquad \textbf{(3)}$$

Alpha is a Scaling factor chosen uniformly in the interval [-0.25, 1.25]

3.3 Mutation

A further genetic operator, mutation is applied to the new chromosomes, with a set probability *Pm*. Mutation causes the individual genetic representation to be changed according to some probabilistic rule. Mutation is generally considered to be a background operator that ensures that the probability of searching a particular subspace of the problem space is never zero. This has the effect of tending to inhibit the possibility of converging to a local optimum, rather than the global optimum.

There are several functions to make mutation on real-valued population. We used *mutbga,* which takes the real-valued population, OldChrom, mutates each variable with given probability and returns the population after mutation, *NewChrom = mutbga(OldChrom, FieldD, MutOpt)* takes the current population, stored in the matrix OldChrom and mutates each variable with probability by addition of small random values (size of the mutation step). We considered 1/600 as our mutation rate. The mutation of each variable is calculated as follows:

$$Mutated\ Var = Var + MutMx \times range \times MutOpt(2) \times delta \qquad \textbf{(4)}$$

where delta is an internal matrix which specifies the normalized mutation step size; MutMx is an internal mask table; and MutOpt specifies the mutation rate and its shrinkage during the run. The mutation operator *mutbga* is able to generate most points in the hypercube defined by the variables of the individual and the range of the

mutation. However, it tests more often near the variable, that is, the probability of small step sizes is greater than that of larger step sizes.

3.4 Fitness Function

During the reproduction phase, each individual is assigned a fitness value derived from its raw performance measure given by the objective function. This value is used in the selection to bias towards more fit individuals. Highly fit individuals, relative to the whole population, have a high probability of being selected for mating whereas less fit individuals have a correspondingly low probability of being selected. The error rate is measured in each round of cross validation by dividing "the total number of misclassified examples" into "total number of test examples". Therefore, our *fitness function* measures the error rate achieved by CMC and our objective would be to maximize this performance (minimize the error rate).

۴ نتایج پژوهش

بدون استفاده از الگوریتم ژنتیک، نتایج کارائی روی‌هم‌رفته‌ی دسته‌بندها روی مجموعه‌ی داده‌ای ما، درخصوص ٤ دسته‌بند درختی، و پنج دسته بند غیردرختی و حالت ترکیبی در جدول ٤ نشان‌داده‌شده‌است. درخصوص دسته بندهای تکی، برای مورد ۲-دسته‌ای، k-NN به‌ترین کارائی را با ۸۲،۳٪ دقّت داشت. در مورد ۳-دسته و ۹-دسته، CART به‌ترین کارایی را، با حدود ۶۰٪ در حالت ۳-دسته‌ای و ۴۳٪ درحالت ۹-دسته‌ای، داشت. درهرحال، با در نظرگرفتن ترکیب دسته بندهای غیرمبتنیبردرخت، CMC به‌ترین کارائی را در هر سه حالت داشت. که، ۸۶،۸٪ دقّت را در حالت ۲-دسته‌ای، ۷۱٪ را در حالت ۳-دسته‌ای و ۵۱٪ را درحالت ۹-دسته ای ارائه‌داد.

در بهینه‌سازی با الگوریتم ژنتیک، ما از ۲۰۰ فرد در جمعیت خود استفاده‌کردیم ، و الگوریتم را در حدوداً ۵۰۰ نسل اجراکردیم. برنامه را ۱۰ بار اجراکردیم و میانگین را به‌دست‌آوردیم، آن‌گونه که در جدول ۵ نشان‌داده‌شده‌است. در هر اجرا ۵۰۰×۲۰۰ بار تابع برازندگی صدازده شد که از تقاطع ۱۰-قاچ ارزیابی برای اندازه‌گیری کارائی میانگین برای CMC استفاده‌کردیم. بنابراین هر دسته بند ۳×۱۰۶ بار برای موارد ۲-دسته‌ای و ۳-دسته‌ای و ۹-دسته ای صدازده‌شده‌است. دراین‌صورت، زمان سربار برای ارزیابی برازندگی، زمان بحرانی تلقّی می‌شود. با استفاده از MLP این فرآیند، حدود ۲ دقیقه به طول‌می‌انجامد و همه‌ی ۴ روش غیردرختی دیگر تنها ۳ ثانیه به‌طول‌می‌انجامد. ما MLP را از گروه دسته بندهای خود، حذف‌کردیم و این نتیجه‌ی معقول از زمان‌های اجرا بود.

Table 5. Comparing the CMC Performance on PHY183 dataset Using GA and without GA in the cases of 2-Classes, 3-Classess, and 9-Classes, 95% confidence interval.

	Performance %		
Classifier	2-Classes	3-Classes	9-Classes
CMC of 4 Classifiers without GA	83.87 ± 1.73	61.86 ± 2.16	49.74 ± 1.86
GA Optimized CMC, Mean individual	94.09 ± 2.84	72.13 ± 0.39	62.25 ± 0.63
Improvement	$\mathbf{10.22 \pm 1.92}$	$\mathbf{10.26 \pm 1.84}$	$\mathbf{12.51 \pm 1.75}$

نتایج در جدول ۵ ارائه شده‌است، میانگین کارائی با آزمون t دوطرفه با ۹۵٪ فاصله‌اطمینان. برای بهبود الگوریتم ژنتیک، در برابر نتایج بدون استفاده از این الگوریتم، یک مقدار P که مشخّص‌کننده‌ی احتمال فرضیه‌های صفر (عدم بهبود کامل) نیز داده‌شده‌است. که بهینه سازی قابل ملاحظه‌ی الگوریتم ژنتیک را نشان‌می‌دهد. در همه‌ی حالت $p<0.000$، مشخص‌کننده‌ی بهبود قابل‌ملاحظه‌است. بنابراین، با استفاده از الگوریتم ژنتیک، در همه‌ی موارد، ما

بیش از ۱۰٪ میانگین بهبود کارائی فردی را داریم و حدود ۱۲ تا ۱۵٪ میانگین بهبود کارائی داریم. شکل ۲، به ترین نتیجه را در ۱۰ اجرا روی مجموعه‌ی داده‌ای ما نشان‌می‌دهد. این نمودارها میانگین جمعیت را ارائه‌می‌کنند، و به‌ترین فرد در هر نسل و به ترین مقدار حاصل‌شده به وسیله‌ی اجرا.

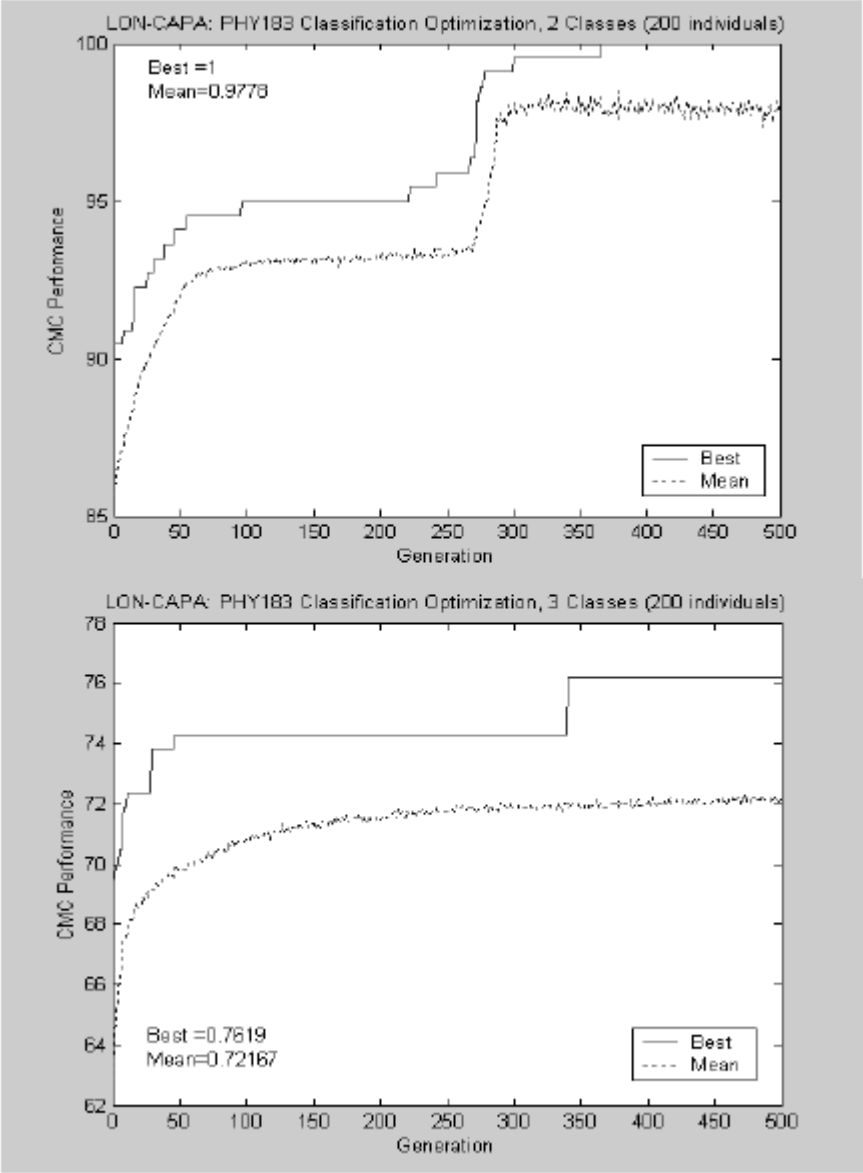

Fig. 2. Graph of GA Optimized CMC performance in the case of 2, and 3-Classes

در نهایت، ما می‌توانیم افراد (اوزان) را برای ویژگی ها به وسیله‌ی به دست آوردن نتایج بهبودیافته، بررسی‌کنیم. این وزن‌دهی به ویژگی‌ها، اهمیت آن‌ها را مشخّص‌می‌سازد، و برای ساختن دسته بندی موردنیاز به‌کارمی‌آید. در بیش‌تر موارد نتایج مشابه رگرسیون خطی چندگانه یا نرم‌افزار مبتنی بر درخت که در روش‌های آماری، برای اندازه‌گیری اهمیت، مورد استفاده‌قرارمی‌گیرد، می‌باشد. جدول ۶ اهمیت ۶ ویژگی در حالت ۳-دسته‌ای، با استفاده از معیار تفکیک آنتروپی، مشخّص‌شده‌است. برمبنای آنتروپی، یک خصوصیت آماری بهره‌ی اطّلاعی[1] نامیده‌می‌شود، که نشان‌می‌دهد، چگونه ویژگی داده‌شده، نمونه‌های آموزشی را در رابطه با کلاس‌های مقصدشان، تفکیک‌می‌کند.

[1] *information gain*

آنتروپی ناخالصی[1] یک مجموعه‌ی اختیاری را از نمونه‌ی S در یک گره N خاص، تعیین‌می‌کند. در [5] ناخالصی گره‌ی N با i(N) مشخص شده‌است، مثلاً به این صورت:

$$Entropy(S) = i(N) = -\sum_j P(\omega_j)\log_2 P(\omega_j) \qquad (5)$$

where $P(\omega_j)$ is the fraction of examples at node N that go to category ω_j.

Table 6. Feature Importance in 3-Classes Using Entropy Criterion

Feature	Importance %
Total Correct Answers	100.00
Total Number of Tries	58.61
First Got Correct	27.70
Time Spent to Solve	24.60
Total Time Spent	24.47
Communication	9.21

نتایج الگوریتم ژنتیک نیز نشان‌می‌دهد که، «تعداد کل پاسخ های صحیح» و "تعداد کل تلاش‌ها» مهم‌ترین ویژگی‌ها برای دسته‌بندی بوده‌اند. ستون دوم در جدول ۶ درصد اهمیت ویژگی‌ها را نشان‌می‌دهد.

۵ نتایج و کوشش‌های پیش رو

چهار دسته بند در تفکیک دانشجویان مورد استفاده‌قرارگرفت. ترکیبی از دسته‌بندهای چندتایی به بهبود قابل‌ملاحظه‌ی دقّت، در هر سه مورد، منجرگردید. وزن‌دهی ویژگی‌ها و استفاده از الگوریتم‌های ژنتیک برای کمینه‌سازی نرخ خطا، دقّت پیش‌بینی را حداقل ۱۰٪، در هر سه حالت ۲، ۳ و ۹ دسته‌ای، بهبودداد. در حالت‌هایی که تعداد ویژگی‌ها کم است، وزن‌دهی ویژگی‌ها بسیار به‌تر از انتخاب ویژگی کارمی‌کند.

بهینه‌سازی موفّقِ دسته‌بندی دانشجویان، در هر سه حالت شایستگی استفاده از داده‌های لُن-کاپا برای پیش‌بینی نمرات نهایی دانشجویان را برپایه‌ی ویژگی‌های آن‌ها، که از داده‌های مربوط به تکالیف شان استخراج شده بود، نشان می‌داد. ما درصدد آن‌هستیم که از برنامه‌سازی ژنتیکی برای تولید ترکیب‌های به‌مراتب متفاوت تری از ویژگی‌ها بهره‌گیریم، تا ویژگی‌های جدیدی استخراج کنیم و پیش‌بینی را بهبودبخشیم. ما قصد داریم با استفاده از الگوریتم‌های تکاملی برای دسته‌بندی دانشجویان و مسائل به صورت مستقیم نیز، استفاده‌کنیم.

همچنین قصدداریم، الگوریتم های ژنتیکی را برای یافتن قوانین هم‌باش و وابستگی میان گروه های مسائلِ تکالیف لُن-کاپا به کارببریم.[2]

سپاس‌گزاری

این کوشش تاحدودی توسط بنیاد دانش ملّی[3] مورد حمایت قرارگرفته‌است. تحت ITR ۰۰۸۵۹۲۱ .

منابع

1. Baker, J. E.: Reducing bias and inefficiency in the selection algorithm, Proceeding ICGA 2,Lawrence Erlbuam Associates, Publishers, (1987) 14-21

---

[1] impurity
[2] Mathematical, Optional Response, Numerical,Java Applet, and so forth
[3] National Science Foundation

2. Bala J., De Jong K., Huang J., Vafaie H.: and Wechsler H. Using learning to facilitate the evolution of features for recognizing visual concepts. Evolutionary Computation 4(3) – Special Issue on Evolution, Learning, and Instinct: 100 years of the Baldwin Effect. (1997)

3. Bandyopadhyay, S., and Muthy, C.A.: Pattern Classification Using Genetic Algorithms, Pattern Recognition Letters, Vol. 16, (1995) 801-808

4. De Jong K.A., Spears W.M. and Gordon D.F.: Using genetic algorithms for concept learning. Machine Learning 13, (1993) 161-188

5. Duda, R.O., Hart, P.E., and Stork, D.G.: Pattern Classification. 2nd Edition, John Wiley & Sons, Inc., New York NY. (2001)

6. Falkenauer E.: Genetic Algorithms and Grouping Problems. John Wiley & Sons, (1998).

7. Freitas, A.A.: A survey of Evolutionary Algorithms for Data Mining and Knowledge Discovery, See: www.pgia.pucpr.br/~alex/papers. A chapter of: A. Ghosh and S. Tsutsui. (Eds.) Advances in Evolutionary Computation. Springer-Verlag,(2002)

8. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, MA, Addison-Wesley, (1989)

9. Guerra-Salcedo C. and Whitley D.: Feature Selection mechanisms for ensemble creation: a genetic search perspective. In: Freitas AA (Ed.) Data Mining with Evolutionary Algorithms: Research Directions, Technical Report WS-99-06. AAAI Press, (1999)

10. Jain, A. K.; Zongker, D.: Feature Selection: Evaluation, Application, and Small Sample Performance, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No. 2, February (1997)

11. Kuncheva , L.I., and Jain, L.C.: Designing Classifier Fusion Systems by Genetic Algorithms, IEEE Transaction on Evolutionary Computation, Vol. 33 (2000) 351-373

12. Martin-Bautista MJ and Vila MA. A survey of genetic feature selection in mining issues. Proceeding Congress on Evolutionary Computation (CEC-99),Washington D.C.,July(1999) 1314-1321

13. Michalewicz Z.: Genetic Algorithms + Data Structures = Evolution Programs. 3rd Ed. Springer-Verlag, (1996)

14. Muhlenbein and Schlierkamp-Voosen D.: Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization, Evolutionary Computation, Vol. 1, No. 1, (1993) 25-49

15. Park Y and Song M.: A genetic algorithm for clustering problems. Genetic Programming 1998: Proceeding of 3rd Annual Conference, Morgan Kaufmann, (1998), 568-575.

16. Pei, M., Goodman, E.D. and Punch, W.F.: Pattern Discovery from Data Using Genetic Algorithms, Proceeding of 1st Pacific-Asia Conference Knowledge Discovery & Data Mining (PAKDD-97) Feb. (1997)

17. Pei, M., Punch, W.F., and Goodman, E.D.: Feature Extraction Using Genetic Algorithms, Proceeding of International Symposium on Intelligent Data Engineering and Learning'98 (IDEAL'98), Hong Kong, Oct. (1998)

18. Punch, W.F., Pei, M., Chia-Shun, L., Goodman, E.D.: Hovland, P., and Enbody R. Further research on Feature Selection and Classification Using Genetic Algorithms, In 5th International Conference on Genetic Algorithm, Champaign IL, (1993) 557-564

19. Siedlecki, W., Sklansky J., A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters, Vol. 10, (1989) 335-347

20. Skalak D. B.: Using a Genetic Algorithm to Learn Prototypes for Case Retrieval an Classification. Proceeding of the AAAI-93 Case-Based Reasoning Workshop, Washigton, D.C., American Association for Artificial Intelligence, Menlo Park, CA, (1994) 64-69