به نام خدا

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

سیستم های هوشمند

تمرین شماره2

حمیدرضا علی اکبری خویی

810196514

پاییز 99

فهرست

چکیده

این تمرین از ۳ بخش تشکیل شده است که باید در ابتدا با نوجه دسته بندهی و انتخاب داده ها با دو معیار مختلف آشنا بشویم و بعد در قسمت دوم با استفاده از یکی از معیار های بحث شده در سوال اوم درخت و بعد جنگلی درست بکنیم تا بتوانیم مدل ررا بر حس درخت تصمیم آموزش داده و به نتیجه برسانیم. در سوال سوم نیز با استفاده از الگوریتم های ابتکاری (الگوریتم ژنتیک) باید رمز یک دسته از کلمات کد گذری شده را بشکنیم.

سوال 1

بخش اول:

در این سوال با استفاده از فرمول زیر باید مقدار information Gain هر یک از ویژگی بعد هارا پیدا کرد و با توجه به آن تصمیم گرفت که کدام یک از ویژگی ها برای این کار بهتر است، طبیعی است که هر ویژگی که این مقدار در آن بیشتر باشد، باید انتخاب بشود:

$$Gain(S,A) = Entropy(S) - \sum_{v \in values(A)} \frac{S_V}{|s|} Entropy(S_V)$$

حال برای ویژگی های متفاوت این مقدار را حساب میکنیم:

:TOEFL

$$Entropy(TOEFL) = Entropy([4-,5+]) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} = 0.99107$$

$$Entropy(TOEFL, low) = Entropy([2-,1+]) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.91829$$

$$Entropy(TOEFL, med) = Entropy([2-,1+]) = -\frac{2}{3}\log_2\frac{2}{3} - \frac{1}{3}\log_2\frac{1}{3} = 0.91829$$

$$Entropy(TOEFL, med) = Entropy([0-,3+]) = 0$$

$$Gain(TOEFL) = 0.99107 - 2 \times \frac{3}{9} \times 0.91829 = 0.37888$$

:SOP

$$Entropy(SOP) = Entropy([4-,5+]) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} = 0.99107$$

$$Entropy(SOP, Yes) = Entropy([2-,3+]) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97095$$

$$Entropy(SOP, No) = Entropy([2-,2+]) = 1$$

$$Gain(SOP) = 0.99107 - \frac{5}{9} \times 0.97095 - \frac{4}{9} = 0.05165$$

:GPA

$$Entropy(GPA) = Entropy([4-,5+]) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} = 0.99107$$

$$Entropy(GPA, > 8) = Entropy([1-,4+]) = -\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5} = 0.72192$$

$$Entropy(GPA, < 8) = Entropy([3-,1+]) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.81127$$

$$Gain(GPA) = 0.99107 - \frac{5}{9} \times 0.72192 - \frac{4}{9} \times 0.81127 = 0.22943$$

:Research

$$Entropy(Research) = Entropy([4-, 5+]) = -\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} = 0.99107$$

$$Entropy(Research, No) = Entropy([1-, 3+]) = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} = 0.81127$$

$$Entropy(Research, Yes) = Entropy([3-, 2+]) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.97095$$

$$Gain(GPA) = 0.99107 - \frac{5}{9} \times 0.97095 - \frac{4}{9} \times 0.81127 = 0.091088$$

پس چون مقدار Information Gain مربوط به ویژگی TOEFL از همه بیشتر است پس این ویژگی مناسب تر است.

بخش دوم :

$$GINI(t) = 1 - \sum_j [p(j|t)]^2$$

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

این بار با استفاده از معیار GINI باید محاسبات را انجام بدهیم:

TOEFL-{Low,Mid} | {High}

$$\{low, mid\} \rightarrow (4-, 2+) \rightarrow GINI_1 = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = \frac{4}{9}$$

$$\{High\} \rightarrow (0,3) \rightarrow GINI_2 = 0$$

$$GINI_{split} = \frac{2}{3} \times \frac{4}{9} = \frac{8}{27} = 0.29629$$

TOEFL-{Low,High} | {Med}

$$\{low, High\} \rightarrow (2-, 4+) \rightarrow GINI_1 = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = \frac{4}{9}$$

$$\{Med\} \rightarrow (2,1) \rightarrow GINI_2 = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$GINI_{split} = \frac{2}{3} \times \frac{4}{9} + \frac{1}{3} \times \frac{4}{9} = \frac{4}{9} = 0.4444$$

TOEFL-{Med,High} | {Low}

$$\{Med, High\} \rightarrow (2-,4+) \rightarrow GINI_1 = 1 - \left(\frac{4}{6}\right)^2 - \left(\frac{2}{6}\right)^2 = \frac{4}{9}$$

$$\{Low\} \rightarrow (2,1) \rightarrow GINI_2 = 1 - \left(\frac{2}{3}\right)^2 - \left(\frac{1}{3}\right)^2 = \frac{4}{9}$$

$$GINI_{split} = \frac{2}{3} \times \frac{4}{9} + \frac{1}{3} \times \frac{4}{9} = \frac{4}{9} = 0.4444$$

در محاسبه GINI آن که از همه کمتر است بهتر است چرا که کمترین خطا را خواهد داشت به همین خاطر {Low,Med}|{High} از همه بهتر است.

سوال 2

بخش اول:

در این قسمت باید یک درخت به عمق 3 را طبق داده های داده شده باید بسازیم و نتایج را ببینیم که با توجه به شکل داده که از 10 ویژگی برای آموزش داده و یک ویژگی که به عنوان ویژگی تست کردن یا ویژگی برگ درخت استفاده شده است. معیار انتخاب ویزگی برتر طبق information gain میباشد که مثلا در ابتدا که کل داده ها حضور دارند میزان Information Gain داده ها به صورت شکل زیر است :

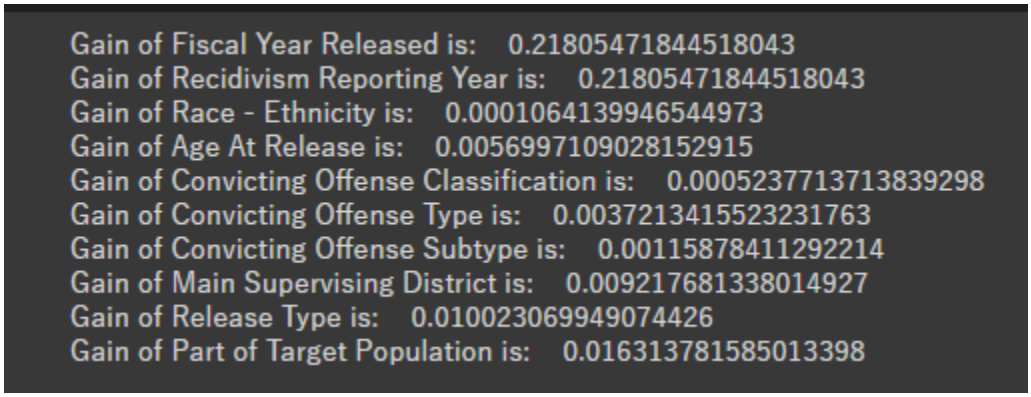


Figure ۱

با توجه به شکل ۱ این معیار برای دو ویژگی بیشینه است که به تزتیب ویژگی های Fiscal Year Released و Recidivism Reporting Year می باشد. که هر دو برابر ۰.۲۱۸ میباشند و درخت یکی را بر حسب رندوم انتخاب خواهد کرد.

برای این درخت ساخته شده با این الگوریتم میزان درستی در تشخیص برابر است با :

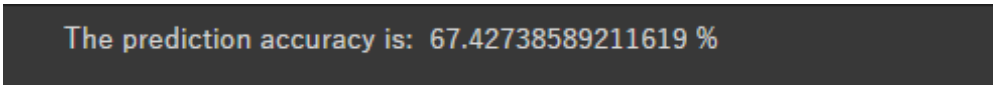


Figure ۲

طبق شکل ۲ این مقدار برابر است با ۶۷.۴۲٪

ماتریس confusion برای این داده ها برابر است با :

$$\begin{bmatrix} 1482 & 214 \\ 1042 & 1118 \end{bmatrix}$$

بخش دوم:

در این بخش با استفاده از الگوریتم ID3 باید ویژگی اصلی را به صورت رندوم به تعداد دلخواه انتخاب بکنیم و یک مجموعه درخت با آن ویژگی های تصادفی ساخت تا در نهایت هر کدام از آن درخت ها در این بار باید لیبل پیش بینی خودشان را بدهند تا در نهایت با قانون اکثریت آن لیبل انتخاب بشود. برای این قسمت با آزمایش های متعدد به نتیجه رسیدم ساختن 15 درخت با دو ویژگی به صورت تصادفی بیشترین مقدار درصد خروجی تطبیق را خواهد داشت که برای درخت های ساخته شده با عمق 3 این عدد به 71.47٪ رسید.

ماتریس Confusion برای این جنگل تصادفی برابر است با :

$$\begin{bmatrix} 1433 & 263 \\ 837 & 1323 \end{bmatrix}$$

بخش سوم :

در ابتدا میزان دقت برای درخت ساخته شده با استفاده از کتابخانه Sklearn را بدست میارم که به صورت زیر است :

میزان دقت : 71.47٪

ماتریس Confusion :

$$\begin{bmatrix} 1433 & 263 \\ 837 & 1323 \end{bmatrix}$$

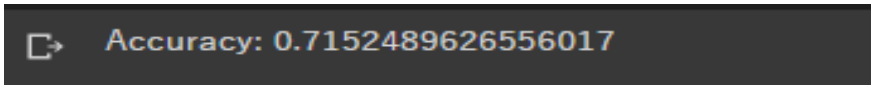
که دقیقا برابر ماتریسی شد که از قسمت قبل برای جنگل تصادفی بدست آوردم ☺

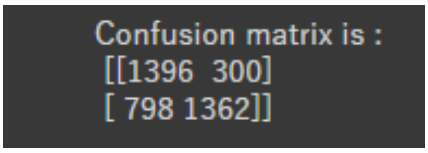
ولی این ماتریس در مقایسه با ماتریس متناظر بدست آمده برای بخش اول سوال دو میتوان گفت که اگر نتیجه ای که من گرفته ام را با این قسمت سوال که با کتابخانه مذکور گرفته شده است مقایسه کنم متوجه خواهیم شد که :

- درخت من در میزان تشخیص درست داده هایی که برچسب 0 داشتند بهتر عمل کرده است
- درخت من در میزان تشخیص درستداده هایی با برچسب 1 ضعیف عملکرده است که در مقایسه با درخت این قسمت اختلاف حدود 200 تایی مشاهده میشود.
- همانطور که میزان دقت در تشخیص درست برچسب های صفر بود اینجا نیز میزان تشخیص برچسب صفر اشتباها به جای برچسب یک خیلی کمتر است ولی در مقال اشتباه در تشخیص

برچسب یک اشتباها به عنوان برچسب صفر خیلی بیشتر است و اختلاف باز هم بسیار استو به عدد 200 میرسد.

برای قسمت جنگل تصادفی مقادیر درصد درستی و ماتریس Confusion به صورت زیر است :





میزان دقت برای جنگ تصادفی در مقایسه با مدل من در حد 0.05٪ است، و البته در مقایسه ماتریس های Confuison دو مدل میتوان به این نتیجه رسید که مدل من در تشخیص درست برچسب های صفر بهتر عمل کرده است و اما مدل این کتابخانه برچسب های یک را کمی بهتر پیشبینی کرده است و در ادامه مدل من در تخیص اشتباه یک به جای صفر بهتر عمل کرده است ولی در تشخیص اشتباه یک بجای صفر بدتر عمل کرده است.

در کل جنگل تصادفی در قطعه کد من نسبت به جنگل تصادفی عملکرد خیلی بهتری داشته و در حدود 4 درصد بهبود عملکرد مشاهده میشود.

سوال سوم

استفاده از الگوریتم ژنتیک باید به گونه ای باشد که یک سری مراحل باید طی بشود تا بتوان به نتیجه مطلوب با ایده این الگوریتم رسیدم که این مراحل عبارتند از:

- Selection
- Cross over
- Mutation

در ادامه به توضیح هر یک و نحوه مدل کردن سیستم میپردازم:

Selection:

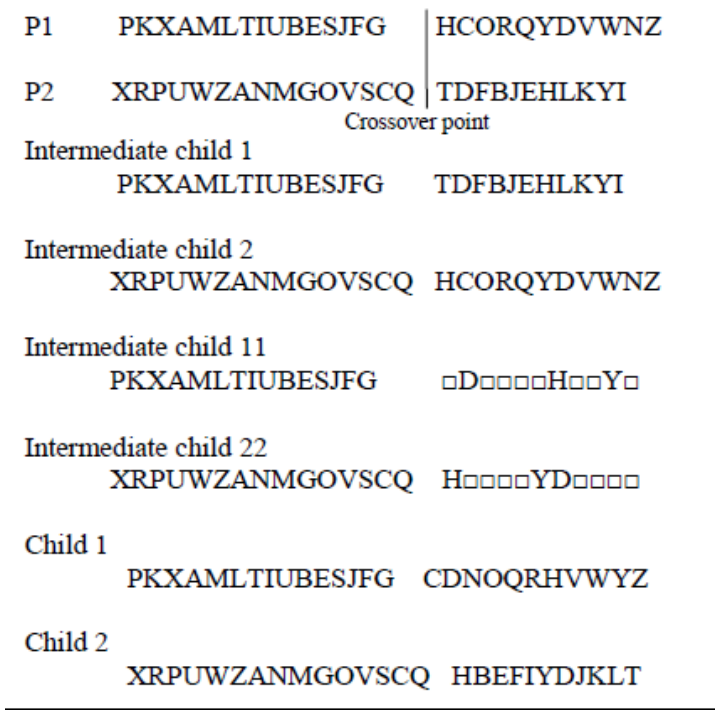
چون الگوریتم در هر حلقه باید به کار گرفته شود تا زمانیکه به جواب مطلوب برسد، حال برای تابع Fitness اینطور تعریف باید بشود که بعد از Decrype کردن کلامت کد شده هر چند کلمه که معادل آن در Dictioany پیدا بشود را به عنوان تابع Fitness در نظر میگیریم.

بعد یک ماتریس ورودی از داده ها خواهیم داشت که باید طبق Fitness طبقه بندی بشوند و بعد طبق رتبه بدست آمده به میزان مشخص شده اندازه size اول از آن را جدا میکنیم تا بتوانیم مراحل را ادامه بدهیم

Cross Over:

برای این قسمت بعد از این که داده ها موفق شدند بقای خود را حفظ کنند، این قسمت باید با استفاده از دو دسته داده که به عنوان والد محسوب میشند، دو Child تولید کنیم. نحوه انتخاب دو والد به ان صورت است که یکبار از 40 درصد از بهترین داده ها و یک بار از 40 درصد از بدترین داده ها و بار دیگر از 20 درصد میانی به صورت تصادفی والدین انتخاب میشوند .

نحوه درست کرد هر دسته child به صورت زیر است:



3Figure

طبق عکس شماره ۳ که از روش ایجاد child استفاده شده در مقاله پیوست 1 آمده است داریم:

- دسته های والد انتخاب میشوند
- بعد یک نقطه crossover point به صورت تصادفی از اعداد 0 تا 25 انتخاب میشود .
- حروف بعد از نقطه انتخابی را در دو والد جابجا میکنیم
- در قسمت های جدید جابجا شده برای هر والد اگر حروف تکراری با قسمت اول(قبل از نقطه) وجود داشته باشد آن ها را خالی میگذاریم و بعد با استفاده از حروف miss شده آن قسمت هارا پر میکنیم.

:Mutation

بعد از تولید child ها باید دو تا از والدین را انتخاب کرد و با یک درصد احتمالی که mutation rate نام دارد و به سیستم داده میشود، دو والد را انتخاب کرده و بعد بچه ای را با استفاده از الگوریتم زیر تولید بکنیم(البته توجه داشته باشید که میزان والد های انتخابی به ازای هر population فرق میکند و امید ریاضی mutation rate در آن population خواهد بود) :

- یک رشته بیت باینری به طول 26 به صورت تصادفی انتخاب میکنیم که 13 بیت یک و 13 بیت صفر داشته باشد

- به ازای بیت های یک بچه اول کروموزوم ها یا حرف های والد اول را به ارث میبرد، همچنین به ازای بیت های صفر بچه دوم کروموزوم های والد دوم یا همان حرف های متناظر را به ارث میبرد
- حروف بجا مانده پر میشوند.

Generating random binary number (26 binary numbers (0, 1))
1 1 0 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 1 0 1 0 0 0 1 0

First Parent (chosen randomly)
P R X A W L T I U G E V J F Q H D O B C Y M S K N Z

Second Parent (chosen randomly)
X K P H M Z A N U B Q V S F G T C O R J E D L W Y I

1st Child
P R X K M L Z I A G U V S F Q H D O B T Y C J E N W

2nd Child
R X P H M T A G U E Q V S F O B C Y K J N D L W Z I

Figure 4

از این قسمت در مدل استفاده میشود تا بتوان از این که سیستم در حین یادگیری به این روش در نقطه مینیمم محلی گیر نکند.

بعد از این که من جمعیتی به اندازه 50 رشته در نظر گرفتم تا بتواند طبق الگوریتم توضیح داده شده در نهایت به کلید مذکور برسد ولی بعد از 2 ساعت ران به کلیدی رسید که دارای 85 درصد موفقیت با توجه به Fitness تولید شده بود که کلید مذکور به صورت زیر است :

```
best_key = 'zyvkosbdixpghtwcarmjlfunqe'
```

```
Best accuracy reached 85.0% for key : zyvkosbdixpghtwcarmjlfunqe
```

طبق شکل میزان درصد 85 برای داده ها بدست آمد.

البته از روش های تحلیل فرکانس استفاده از یک حرف هم برای پیدا کردن کلید استفاده کردم که کد مربوط در قطعه کد موجود است که متاسفانه در صد 10 رسیدم و به اندازه کلید بدست آمده طبق الگوریتم های بالا نبود.

نحوه اجرای برنامه

برای سوال فایل IS_HW2_Q2.ipynb را باید در بستر Jupyter اجرا بکینم و البته من داده های Train,Test ر در Google Drive خود گذاشته بودم تا با سرویس Colab برنامه نویسی بکنم. فایل ها در پوشه IS_HW2 در Google Drive باید باشد تا اجرا بشود.

برای سوال سوم فایل IS_HW2_Q3.ipynb را باید در بستر Jupyter اجرا بکینم و البته من داده های Train,Test ر در Google Drive خود گذاشته بودم تا با سرویس Colab برنامه نویسی بکنم. فایل ها در پوشه IS_HW2 در Google Drive باید باشد تا اجرا بشود.

منابع و مراجع

[1] Using Genetic Algorithm to Break A Mono-Alphabetic Cipher 2010 IEEE Conference on Open Systems (ICOS 2010), December 5-7, 2010, Kuala Lumpur, Malaysia