برنامه‌سازی پیشرفته

تمرین کامپیوتری شماره ۳

استاد: **دکتر رامتین خسروی**

طراحان: **محمّد مریدی**، **غزل کلهر**

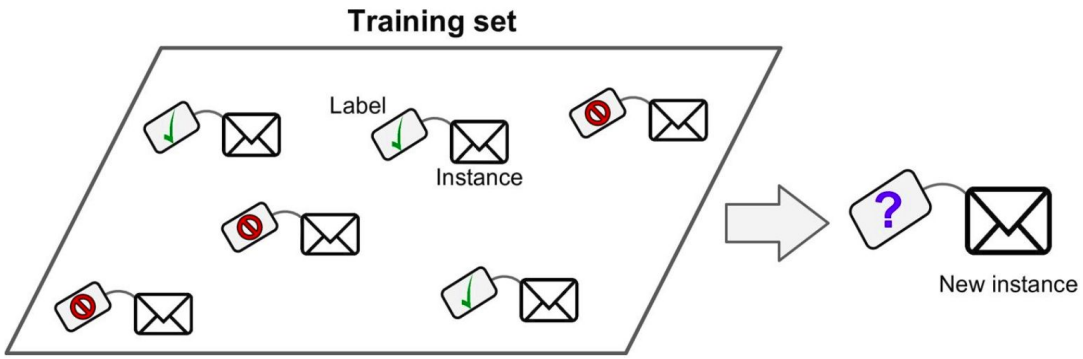
مهلت تحویل: شنبه ۱۱ آبان ۱۳۹۸، ساعت ۲۳:۵۵

مقدمه

هدف از این تمرین، آشنایی شما با طراحی بالا به پایین[1] یک مسأله است. در این تمرین به شبیه‌سازی یکی از روش‌های رایج در یادگیری ماشین[2] پرداخته می‌شود. به عنوان یکی از شاخه‌های وسیع و پرکاربرد هوش مصنوعی، یادگیری ماشین به تنظیم و اکتشاف شیوه‌ها و الگوریتم‌هایی می‌پردازد که بر اساس آن‌ها رایانه‌ها و سامانه‌ها توانایی یادگیری و پیش‌بینی پیدا می‌کنند.

دسته‌بندی[3]

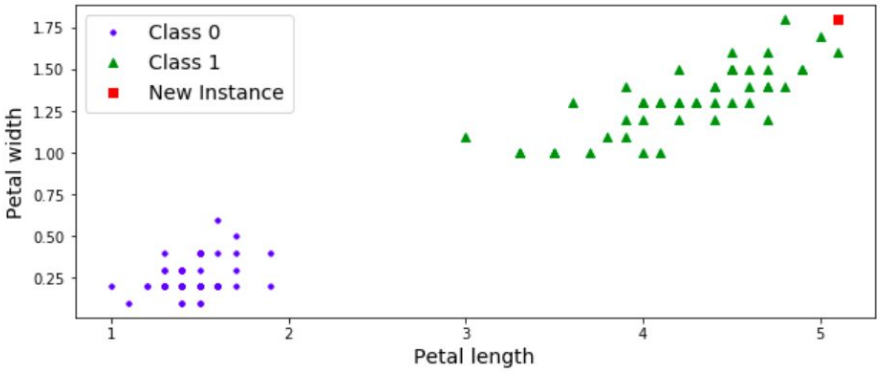
در یادگیری ماشینی، دسته‌بندی مسئله شناسایی تعلق مشاهده جدید، به یکی از دسته‌ها بر اساس مجموعه‌ای از مشاهدات است که عضویت در دسته‌هایشان مشخص است.



برای مثال تصور کنید که می‌خواهید نام یک گل را بر اساس طول و عرض گلبرگ‌های آن تشخیص دهید. بدین منظور لازم است که یک دسته‌بند[4] برای این منظور آموزش ببیند (توانایی تشخیص نوع گل را پیدا کند) و پس از آن بر اساس ویژگی‌هایی که یک گل را

[1] Top-Down Design
[2] Machine Learning
[3] Classification
[4] Classifier

توصیف می‌کند (طول و عرض گلبرگ در این مثال) به دسته‌بند داده شود. این دسته‌بند براساس مشاهداتی که در گذشته داشته است (در مرحله آموزش) تعلق این گل را به یکی از دسته‌ها تشخیص می‌دهد.



دسته‌بندی خطی[5]

در حوزه یادگیری ماشین نمونه‌هایی که قصد پیش‌بینی نوع و یا یک ویژگی آن‌ها وجود دارد، با استفاده از تعدادی ویژگی عددی و قابل اندازه‌گیری در قالب بردار ویژگی[6] توصیف می‌شوند.

تعداد زیادی از الگوریتم‌هایی که برای دسته‌بندی وجود دارند، می‌توانند با استفاده از یک تابع خطی[7]، به هر یک از دسته‌ها امتیاز[8]ی اختصاص دهند. این امتیازدهی با استفاده از ضرب داخلی بردار ویژگی با بردار وزن و جمع کردن با $Bias$ (بردار وزنی با وزن برابر با یک) مربوط به هر یک از دسته‌ها صورت می‌گیرد. دسته‌ی پیش‌بینی‌شده، دسته‌ای است که بالاترین امتیاز را بین سایر دسته‌ها به خود اختصاص دهد. این تابع در زیر توصیف شده است:

$$score(X_i,\ k)\ =\ \beta_k.X_i + Bias_k$$

به طوری که X_i بردار ویژگی نمونه iام، β_k بردار وزن دسته kام و $score(X_i,\ k)$ امتیازی است که دسته kام با اختصاص یافتن به نمونه iام بدست می‌آورد.

برای مثال تصور کنید که دسته‌بند توانایی تشخیص دو نوع گل از یکدیگر را دارد. بدین ترتیب این دسته‌بند دارای دو بردار وزن است که هر دسته آن به ویژگی‌های مختلف نمونه، وزن‌های مختلفی اختصاص می‌دهد. نمونه‌ای از بردارهای وزن یک دسته‌بند را در زیر مشاهده می‌کنید:

[5] Linear Classification
[6] Feature Vector
[7] Linear Function
[8] Score

	β_0	β_1	$Bias$
$Class_1$	31.18	-4.74	-8.00
$Class_2$	6.84	-2.79	1.15

حال این دسته‌بند با بردارهای وزن ذکر شده، قصد تشخیص نمونه‌ای که دارای بردار ویژگی زیر است را دارد:

$Length$	$Width$
0.9	0.1

ستون‌های Length و Width همانطور که از نام آن‌ها برمی‌آید معرف طول و عرض گلبرگ مربوط به گل است. برای محاسبه دسته مربوط به نمونه، لازم است که ضرب داخلی بردار ویژگی نمونه در هر یک بردارهای وزن محاسبه شود و سپس با مقدار Bias جمع شود.

$$score(X_i,\ k) = \beta_{k,0} \times Length_i + \beta_{k,1} \times Width_i + Bias_k \Rightarrow$$

$$score(X_i,\ 1) = 31.18 \times 0.9 + (-4.74) \times 0.1 + (-8.00) = 19.588$$

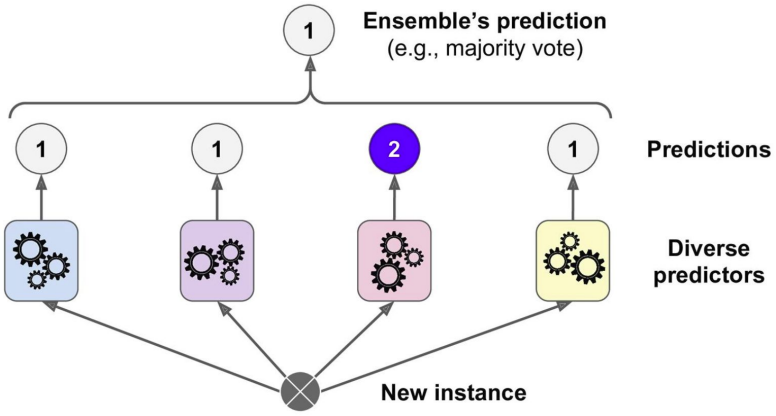
$$score(X_i,\ 2) = 6.84 \times 0.9 + (-2.79) \times 0.1 + 1.15 = 7.027$$

با توجه به این که اولین دسته امتیاز بیشتری را کسب کرد، دسته مربوط به این نمونه دسته شماره یک است.

دسته‌بندی ترکیبی[9]

در حوزه یادگیری ماشین، یکی از روش‌هایی که برای بدست آوردن دقت بیشتر در دسته‌بندی مورد استفاده قرار می‌گیرد، ترکیب کردن نتیجه چندین دسته‌بند و پیش‌بینی دسته، برحسب بیشترین تعداد تکرار برای یک دسته است.

[9] Ensemble Classification



مثال ذکر شده در قسمت قبل را در نظر بگیرید. دسته مربوط به این نمونه با استفاده از بردارهای وزن داده شده برای دسته‌بند مذکور، دسته شماره یک تعیین گردید. حال تصور کنید بردار ویژگی مربوط به این نمونه به دسته‌بندهای دیگری که دارای بردارهای وزن مخصوص به خود می‌باشند داده شده است و این دسته‌بندها به صورت فوق عمل کرده‌اند و هر کدام دسته‌ای را به نمونه اختصاص داده‌اند.

در این مرحله یک رأی‌دهنده[10]، خروجی‌های مربوط به دسته‌بندها را دریافت می‌کند و با توجه به این که کدام دسته بیشتر از سایر دسته‌ها تکرار شده است، دسته نهایی را تعیین می‌کند. برای مثال در شکلی که در بالا آمده است دسته نهایی برای نمونه، دسته شماره یک است.

شرح تمرین

در این تمرین به شبیه‌سازی یک دسته‌بند ترکیبی می‌پردازید. این دسته‌بند شامل چندین دسته‌بند خطی است که این دسته‌بندها آموزش دیده شده‌اند و بردارهای وزن هر یک از آن‌ها در پرونده[11]ای جداگانه در اختیار شما قرار داده شده است. وظیفه‌ای که برنامه شما بر عهده دارد، پیش‌بینی دسته مربوط به نمونه‌هایی است که تحت عنوان مجموعه داده اعتبارسنجی[12] در اختیار شما قرار داده شده است. در ادامه به مراحلی که لازم است برای تمرین صورت گیرد، پرداخته می‌شود. برای انجام این مراحل لازم است داده‌هایی که در اختیارتان قرار داده شده است، تجزیه[13] شوند. به همین منظور پس از شرح مراحل تمرین، ساختار پرونده‌هایی که در اختیارتان قرار داده شده است، تشریح می‌شود.

[10] Voter
[11] File
[12] Validation Dataset
[13] Parse

مراحل تمرین

- دسته‌بندهای خطی، بردارهای وزن مربوطه را از پرونده‌هایی که تهیه شده است استخراج می‌کنند.
- سپس دسته مربوط به هر نمونه‌ای که در پرونده مربوط به مجموعه‌داده‌های اعتبارسنجی است، توسط تمام دسته‌بندهای خطی تعیین می‌شود.
  - این عملیات با محاسبه ضرب داخلی بردارهای وزن هر دسته‌بند با بردار ویژگی مربوط به نمونه صورت می‌گیرد.
- پس از اتمام عملیات دسته‌بندهای خطی بر روی مجموعه داده‌های اعتبارسنجی، دسته مربوط به هر نمونه از طریق محاسبه بیشترین تکرار یک دسته برای نمونه، پیش‌بینی می‌شود.
- پس از اتمام عملیات پیش‌بینی، صحت عملکرد دسته‌بند ترکیبی سنجیده می‌شود.
  - این عملیات از طریق مقایسه اطلاعات بدست آمده از پیش‌بینی با برچسب‌های داده‌های اعتبارسنجی صورت می‌گیرد.

پرونده CSV[14]

مقادیر جداشده با کاما (Comma-separated values) یا CSV، نام یک قالب برای پرونده‌های متنی است که در آن مقادیر با استفاده از نماد کاما (,) از یکدیگر جدا می‌شوند. CSV یکی از روش‌های پرطرفدار برای تبادل اطلاعات بین صفحه‌های گسترده[15] است.

انواع داده‌هایی که در این تمرین در اختیار شما قرار گرفته است، دارای این ساختار می‌باشند که جهت استفاده از این داده‌ها، لازم است آن‌ها را از پرونده‌های مربوط استخراج کرده و سپس تجزیه کنید. در زیر بردارهای وزن مربوط به یک دسته‌بند را در قالب CSV مشاهده می‌کنید.

[14] Comma-separated values

[15] Spreadsheet

```
Betha_0,Betha_1,Bias
-9.720780429252542,-6.002059079071862,36.30322747734118
1.9851588125130801,-2.21734124118296,4.466473721162303
7.735621616584743,8.219400320229264,-40.76970119873617
```

ورودی و خروجی برنامه

پرونده اجرایی برنامه شما، در قالب زیر آدرس مربوط به پوشه[16] بردارهای وزن دسته‌بندها و پوشه مربوط به داده‌های اعتبارسنجی را از طریق آرگومان‌هایی در رابط خط فرمان[17] از کاربر دریافت می‌کند. برنامه شما باید صحت عملکرد سامانه را تا دو رقم اعشار (با گرد کردن عدد اعشاری) نمایش دهد.

نمونه ورودی و خروجی سامانه (با فرض این که پوشه Assets بارگذاری در سایت درس، در کنار پرونده اجرایی شما قرار گرفته است) در ذیل آمده است:

- نمونه ورودی

```
./EnsembleClassifier.out Assets/validation Assets/weight_vectors
```

- نمونه خروجی

```
Accuracy: 97.20%
```

آرگومان‌های خط فرمان

آرگومان‌های خط فرمان آرگومان‌هایی هستند که سیستم‌عامل در زمان اجرای برنامه آن‌ها را به برنامه انتقال می‌دهد. سپس برنامه می‌تواند آن‌ها را نادیده بگیرد و یا از آن‌ها استفاده کند.

برای استفاده از این آرگومان‌ها، تابع main باید به صورت زیر نوشته شود:

```
int main(int argc, char* argv[])
```

دو آرگومان تابع را می‌توان برای دسترسی به آرگومان‌های خط فرمان استفاده کرد:

- `argc`

  عدد صحیح؛ تعداد آرگومان‌های خط فرمان داده شده به برنامه

[16] Directory

[17] Command Line Interface

این مقدار حداقل برابر با یک است؛ زیرا دستور اجرای برنامه (نام پرونده اجرایی) حتماً در زمان اجرای برنامه مورد استفاده قرار می‌گیرد و همواره به‌عنوان آرگومان‌های خط فرمان شماره صفر به برنامه داده می‌شود.

- `argv`

  آرایه‌ای از رشته‌های مدل زبان C؛ آرگومان‌های خط فرمان داده شده به برنامه

به عنوان یک مثال ساده برنامه زیر را در نظر بگیرید:

```cpp
#include <iostream>

int main(int argc, char *argv[])
{
   std::cout << "There are " << argc << " arguments:" << std::endl;

   // Loop through each argument and print its number and value
   for (int count=0; count < argc; ++count)
      std::cout << count << " " << argv[count] << std::endl;

   return 0;
}
```

اگر برنامه به شکل

```
./a.out Myfile.txt 100
```

اجرا شود، خروجی زیر تولید می‌شود:

```
There are 3 arguments:
0 ./a.out
1 Myfile.txt
2 100
```

برای آشنایی بیشتر با نحوه کار آرگومان‌های خط فرمان به این لینک مراجعه کنید.

نکات تکمیلی

- داده‌های اعتبارسنجی، در پوشه‌ای به‌نام validation قرار داده شده‌اند. در این پوشه پرونده‌ای به‌نام dataset.csv که مجموعه‌داده‌های اعتبارسنجی است و برچسب‌های مربوط به هریک از نمونه‌های موجود در این پرونده، در پرونده‌ای به‌نام labels.csv در اختیار شما قرار داده شده است.
- بردارهای وزن مربوط به هر دسته‌بند در پوشه‌ای به‌نام weight_vectors، تحت عنوان classifier_<number>.csv تهیه شده است.

- درصورتی که تعداد رأی‌های مربوط به چند دسته با یکدیگر **برابر** شد، دسته‌ای بعنوان دسته نهایی انتخاب می‌شود که شماره کوچکتری را داراست.
- **توجه کنید،** تعداد بردارهای وزنی که در پوشه مربوطه وجود دارد متغیر است.
- با توجه به این‌که تجزیه پرونده‌های CSV بخشی از تمرین است، استفاده از کتابخانه‌های موجود جهت تجزیه کردن این پرونده‌ها، **قابل قبول نیست**.

نحوه‌ی تحویل

- تمام فایل‌های خود را در قالب یک **پرونده‌ی زیپ** با نام A3-<SID>.zip در صفحهٔ CECM درس بارگذاری کنید که SID شمارهٔ دانشجویی شماست؛ برای مثال اگر شماره‌ی دانشجویی شما ۸۱۰۱۹۸۹۹۹ است، نام پروندهٔ شما باید A3-810198999.zip باشد.
- برنامهٔ شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- درستی برنامه‌ی شما از طریق **آزمون‌های خودکار** سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید. همچنین **دقت شود** که نام پرونده‌ی مربوط به کد شما باید EnsembleClassifier.cpp باشد.
- طراحی درست، رعایت سبک برنامه نویسی درست و تمیز بودن کد برنامه‌ی شما در نمره‌ی تمرین تأثیر زیادی دارد.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.