یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

راهنمای تحویل

قبل از پاسخ دادن به پرسش‌ها، موارد زیر را با دقت مطالعه نمایید:

- **لازم است برای کسب نمره کامل تمرین، یکی از دو ترکیب سوال‌ها (دسته A یا دسته B) پاسخ دهید. (تنها به یکی از دو ترکیب پاسخ دهید)**
- از پاسخ‌های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام **REPORTS_TEMPLATE.docx** قرار داده شده تهیه نمایید.
- **کیفیت گزارش شما در فرآیند تصحیح از اهمیت ویژه‌ای برخوردار است**؛ بنابراین، لطفا تمامی نکات و فرض‌هایی را که در پیاده‌سازی‌ها و محاسبات خود در نظر می‌گیرید در گزارش ذکر کنید.
- در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
- الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
- **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.**
- **کدهای ارسالی می‌بایست قابلیت اجرای دوباره داشته باشند، با این حال، دستیاران آموزشی ملزم به اجرای کدهای شما نیستند**؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می‌شود.
- **در صورت استفاده از Jupyter لازم است تا تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد در غیر اینصورت ورودی‌ها و خروجی‌ها متناظر می‌بایست در گزارش آورده شوند.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کدها وجود داشته باشد.
- **با این که بحث در مورد تمرین‌ها منعی ندارد اما راه‌حل شما می‌بایست توسط شما (و فقط شما) باشد. همچنین، تمامی مطالب جانبی در گزارش باید رفرنس داده شود. یادآوری می‌شود که عدم صداقت علمی[1] عواقب شدیدی را به همراه دارد.**
- **استفاده از کدهای آماده برای تمرین‌ها به هیچ وجه مجاز نیست.**
- **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکت‌کننده در آن، به میزان بارم سوال نمره منفی لحاظ می‌شود.**
- لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

HW[Number]_[Lastname]_[StudentNumber].zip

در صورت وجود سوال، ابهام و یا درخواست راهنمایی با دستیاران آموزشی مرتبط با هر پرسش از طریق ایمیل‌های آورده شده در سربرگ در ارتباط باشید.

[1] Academic dishonesty

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

فهرست

شکل‌ها

جدول‌ها

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

درخت و مقادیر مفقودشده

در جدول زیر، ۲۰ داده آموزشی به همراه خروجی مورد نظر هر یک از آنها داده شده است. هر یک از سطرهای این جدول مشخص کننده یکی از نمونه‌های آموزشی با ۴ مشخصه F_1، F_2، F_3 و F_4 به همراه خروجی y است. هر یک از چهار مشخصه می‌توانند یکی از مقادیر عددی ۰، ۱ یا ۲ را اختیار کنند. خروجی نیز با توجه به مقدار مشخصه‌ها می‌تواند یکی از مقادیر عددی ۰، ۱ یا ۲ را داشته باشد (مقادیر مفقود شده با ? مشخص شده‌اند).

جدول ۱: دادگان ارائه شده برای ساخت درخت تصمیم

F_1	F_2	F_3	F_4	Label
2	2	1	0	**2**
2	0	1	1	**0**
0	0	2	0	**0**
1	2	0	1	**2**
0	2	?	0	**0**
1	2	1	0	**1**
1	1	1	2	**0**
1	1	0	1	**1**
0	1	0	1	**2**
0	?	1	1	**0**
0	0	0	1	**0**
2	2	2	0	**2**
1	2	1	2	**1**
2	2	2	2	**0**
0	1	2	2	**2**
2	1	1	2	**1**
2	2	0	1	**2**
1	1	2	1	**1**
1	0	0	1	**1**
0	2	2	1	**1**

فرض کنید قصد داریم یک درخت تصمیم برای این داده‌ها تولید نماییم. درخت تصمیم مورد نظر را بدست آورده و ترسیم نمایید. مراحل بدست آوردن درخت به همراه محاسبات مرتبط با آن را به طور دقیق تشریح نمایید. معیار انتخاب مشخصه را هر یک از ۳ حالت زیر در نظر بگیرید و برای هر حالت درخت را به طور جداگانه ترسیم نمایید.

۱. Information Gain
۲. Gini Index
۳. Gain Ratio

ساخت درخت تصمیم با ID3

در این سوال می‌خواهیم با ساختن درخت تصمیم با الگوریتم ID3، تحلیل الگوریتمی و پیاده‌سازی آن آشنا شویم.

٤. آشنایی با الگوریتم ID3: شبه‌کدی برای پیاده‌سازی الگوریتم ID3 برای ساخت درخت تصمیم پیشنهاد دهید. در این مرحله فرض کنید تمام ویژگی‌ها[1] رسته‌ای[2] باشند. بررسی کنید آیا الگوریتم پیشنهاد شده همواره بهترین درخت ممکن را پیدا می‌کند؟

[1] Feature
[2] Categorical

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

۵. **تحلیل زمانی**: قطعه کد نوشته شده را برای بدترین و بهترین حالت برحسب تعداد داده، تعداد ویژگی و طول درخت تحلیل پیچیدگی‌زمانی و حافظه‌ای[۱] کنید. توجه کنید که این تحلیل برای هر دو حالت آموزش و تست انجام شود.[۲]

۶. **تغییرات نسبت به تعداد داده**: در هر کدام از حالت‌های تحلیل شده در مسئله‌ی ۲ اگر تعداد داده‌ها c برابر شود، زمان اجرای الگوریتم چه تغییری می‌کند؟

۷. **داده‌های عددی**: با توجه به این که روش فوق فقط برای داده‌های رسته‌ای کار می‌کند، الگوریتمی پیشنهاد دهید که از داده‌های عددی نیز پشتیبانی کند.

۸. **تحلیل زمانی**: قسمت ۲ را روی الگوریتم جدید انجام دهید.

۹. **نتایج**: درخت خود را بر مجموعه‌دادگان که جدول زیر آورده شده‌اند، آزمایش کرده و نتایج را گزارش کنید (جدول درهم‌ریختگی[۳] و معیارهای فراخوانی[۴] و صحت[۵] و دقت[۶] را به همراه نتایج خود ارائه کنید).[۷]

جدول ۲: نمونه‌های ارائه شده برای بررسی شبه‌کد مدل پیشنهادی

ویژگی یک	ویژگی دو	برچسب
A	1.2	مثبت
B	1.2	مثبت
A	1.2	مثبت
C	2.5	منفی
•	3.7	منفی
B	3.7	مثبت
B	1.2	منفی
C	2.5	منفی
B	2.5	منفی
A	2.5	مثبت

۱۰. **تحلیل**: در مورد Bias - Variance Tradeoff در درختان تصمیم توضیح دهید. با ارائه مثال و توضیح کافی بیان کنید که مدل‌ها بیشتر با خطر زیاد بودن Bias مواجه هستند یا Variance؟

۱۱. **تحلیل**: آیا همواره می‌توان درختی پیدا کرد که روی داده‌های آموزش دقت 100% داشته باشد؟ (با ارائه مثال نشان دهید). آیا این دقت روی دادگان آزمایش هم ممکن است؟

---

[1] Space and Time Complexity

[۲] در صورت نیاز به آشنایی با پیچیدگی زمانی می‌توانید کلیدواژه‌ی Algorithms Time Complexity را جستجو کنید (به عنوان مثال این لینک می‌تواند مفید باشد).

[3] Confusion Matrix

[4] Recall

[5] Precision

[6] Accuracy

[۷] همچنین در سوال می‌توانید از تابع classification_report در scikit-learn استفاده کنید.

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

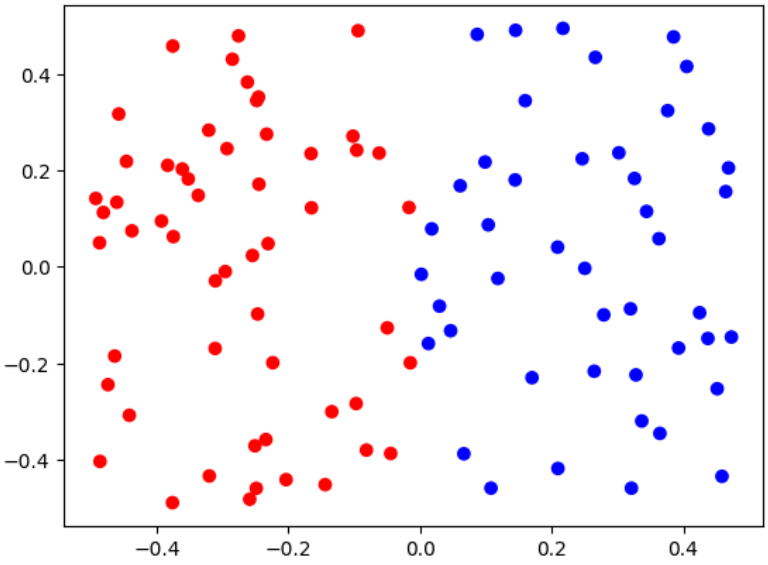
امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

حساس بودن درخت تصمیم به هندسه‌ی داده

در این تمرین می‌خواهیم نشان دهیم که درختان تصمیم نسبت به هندسه‌ی داده حساس هستند. ابتدا داده‌ای تولید می‌کنیم و سپس شکل آن را کمی تغییر می‌دهیم. سپس عملکرد درخت تصمیم در حالت‌های مختلف را می‌سنجیم.

۱. ابتدا با استفاده از کتابخانه‌های پایتون تعدادی داده به شکل زیر در دو کلاس تولید کنید و نمایش دهید.[1]



شکل ۱: دادگان مورد انتظار

۲. حال به کمک تابع کمکی زیر دادگان را دوران داده و نمایش دهید (توجه کنید X آرایه‌ی Numpy است).

```
def rotate_matrix(X, angle=np.pi / 4):
    rotation_matrix = np.array(
        [[np.cos(angle), -np.sin(angle)], [np.sin(angle), np.cos(angle)]]
    )
    return X.dot(rotation_matrix)
```

۳. حال برای هر یک از مجموعه‌دادگان یک درخت تصمیم آموزش دهید (می‌توانید از توابع موجود در scikit-learn استفاده کنید) و مرز تصمیم را رسم کنید.

۴. مشاهده‌های خود را توضیح دهید. مدل آموزش داده شده روی کدام مجموعه‌داده تعمیم‌پذیری[2] بیشتری دارد؟

چگونه از اسپم و حادثه تایتانیک جان سالم به در ببریم؟

هدف از این تمرین، دسته‌بندی «هرزنامه از غیر هرزنامه‌ها» و پیش‌بینی «بازماندگان فاجعه تایتانیک» به کمک پیاده‌سازی درخت تصمیم و جنگل تصادفی است. مجموعه دادگان «Spam» و «Titanic» در اختیار شما قرار گرفته است. پیشنهاد می‌شود با جستجو و مطالعه با تکنیک‌های مختلف درخت تصمیم آشنا شده و تمرین کنید. جهت سهولت در پیاده‌سازی، کدی در اختیار شما قرار می‌گیرد

[1] می‌توانید از np.random.rand() استفاده کنید.

[2] Generalization

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

که در آن پیش‌پردازش و برخی از عملکردهای درخت تصمیم از پیش پیاده‌سازی شده است. از این کد می‌توانید در پیاده‌سازی خود استفاده کرده و یا صرفا پیاده‌سازی خودتان را داشته باشید.

۱. **پیاده‌سازی درخت تصمیم:** از شما انتظار می‌رود که خودتان ساختار داده درخت را پیاده‌سازی کنید (**استفاده از پیاده‌سازی‌های موجود برای درخت تصمیم مجاز نیست!**).
   - ❖ مجموعه‌داده Titanic تمیز نشده است؛ به این معنا که مقادیر گمشده در مجموعه داده موجود است، بنابراین به پیش‌پردازش دادگان پرداخته و درخت را رسم کنید. حذف نمونه‌هایی با ویژگی‌های از دست رفته به دلیل داده‌های ناکافی توصیه نمی‌شود. در طول تمرین، بخش‌های مختلف به توابعی نیاز دارند که باید پیاده‌سازی شوند (برای مثال؛ معیار حداکثر عمق توقف[1]، رسم درخت[2]، ردیابی مسیر نمونه در درخت[3]). در گزارش خود پیاده‌سازی درخت تصمیم خود را آورده و در مورد نحوه و تصمیم‌های گرفته شده بحث کنید.

۲. **پیاده‌سازی جنگل تصادفی:** شما به مجاز به استفاده از هیچ پیاده‌سازی از جنگل تصادفی موجود در سطح اینترنت نیستید. اگر پیاده‌سازی تمیزی[4] انجام داده باشید، این بخش یک صرفا یک کپسوله‌سازی[5] ساده از قسمت قبل است. در گزارش خود پیاده‌سازی جنگل تصادفی خود را آورده و در مورد نحوه و تصمیم‌های گرفته شده بحث کنید. جنگل تصادفی یک تکنیک یادگیری ماشین است که با ترکیب چندین درخت تصمیم یک مدل دقیق‌تر و قابل اعتمادتر ایجاد کند. هر درخت تصمیم بر روی یک زیرمجموعه تصادفی از دادگان و ویژگی‌ها آموزش داده می‌شود، که باعث کاهش همبستگی بین درخت‌ها و تنوع تقسیم‌بندی‌ها می‌شود.

۳. **جزئیات خواسته‌شده را در گزارش خود شرح دهید** (مختصر و مفید به هر سوال در حد ۱-۲ جمله پاسخ دهید).
   ۱. چگونه با ویژگی‌های رسته‌ای و مقادیر گمشده برخورد کردید؟
   ۲. معیار توقف شما چه بوده است؟
   ۳. چگونه جنگل تصادفی را پیاده‌سازی کردید؟
   ۴. از چه راهی برای بهبود سرعت آموزش بهره بردید؟
   ۵. آیا پیاده‌سازی خاص یا جالبی به نظر خودتان داشته‌اید؟ چرا فکر می‌کنین که جالب به حساب میاد؟

۴. **ارزیابی عملکرد**[6]: برای هر دو مجموعه دادگان، یک درخت تصمیم و یک جنگل تصادفی آموزش دهید و دقت آموزش و اعتبارسنجی خود را گزارش کنید. به صورت دقیق‌تر شما باید ۸ عدد گزارش کنید (۲ مجموعه داده x ۲ دسته‌بند x آموزش/اعتبارسنجی). علاوه‌براین، برای هر دو مجموعه‌دادگان، بهترین مدل خود را آموزش دهید و پیش‌بینی‌های خود را در Kaggle سابمیت کنید (نامی که در Kaggle با آن نمایش داده می‌شوید[7] و امتیازات عمومی خود را برای هر مجموعه داده، یعنی ۳ امتیاز Kaggle را در گزارش خود ارائه کنید).

۵. **نتایج آزمایش مدل خود بر دادگان تایتانیک که در گزارش لازم است**

---

[1] Maximum Depth Stopping Criterion
[2] Visualizing Tree
[3] Tracing Path of a Sample Through the Tree
[4] Clean Code
[5] Encapsulation
[6] Performance Evaluation
[7] Kaggle Display Name

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

۱. اگر از ویژگی دیگری یا تبدیل ویژگی[1] استفاده کرده‌اید، در گزارش خود شرح دهید. شما ممکن پیاده‌سازی خاصی برای استخراج ویژگی در نظر داشته باشید، می‌توانید آن پیاده‌سازی را در featurize.py در نظر گرفته و ویژگی‌ها را در mat. فایلی ذخیره کنید، برای مثال؛ ممکن است قصد استفاده از کیسه کلمات[2] را داشته باشید.

۲. برای درخت تصمیمی که پیاده‌سازی کردید و نمونه‌هایی از هر کلاس (هرزنامه و غیر هرزنامه)، تقسیم‌بندی‌هایی که درخت شما با آن‌ها نمونه را دسته‌بندی کرده است را ارائه کنید (برای مثال؛ کدام ویژگی و مقدار آن ویژگی تقسیم شود). برای نمونه؛

a) (“budet”) >= 2
b) (“spreadsheet”) >= 1
c) بنابراین این ایمیل هرزنامه نیست!

۳. برای جنگل تصادفی که پیاده‌سازی کردید، متداول‌ترین تقسیم‌بندی‌هایی که در گره ریشه درخت‌ها انجام می‌شود را پیدا کرده و گزارش کنید. برای نمونه؛

a) (“thanks”) < 4 (15 Trees)

که به این معنی است، اگر برای مثال شما ۲۰ درخت تصمیم در جنگل تصادفی داشته باشید، ۱۵ درخت در جنگل تصادفی شما از شرط thanks <4 برای تقسیم دادگان به دو شاخه در گره ریشه می‌کنند.

۴. به صورت تصادفی دادگان را به نسبت ۸۰ به ۲۰ به بخش‌های آموزش و اعتبارسنجی تقسیم کنید. درخت‌های تصمیم را با ثابت در نظر گرفتن تمام پارامترهای دیگر و حداکثر عمق‌های متفاوت آموزش دهید (بازه تغییر عمق را ۱ به ۴۰ در نظر بگیرید). دقت اعتبارسنجی خود را به عنوان تابعی از عمق رسم کنید. کدام عمق بالاترین دقت اعتبارسنجی را داشت؟ (در مورد رفتاری که پلات خود مشاهده می‌کنید، بحث کنید، اگر متوجه شدید که نیاز به رسم اعماق بیشتری است، منعی ندارد)

۶. نتایج آزمایش مدل خود بر دادگان تایتانیک که در گزارش لازم است

۱. یک درخت تصمیم بسیار کم عمق را آموزش دهید (برای مثال؛ درختی با عمق ۳ اگرچه می‌توانید هر عمقی که نتیجه‌ی خوبی دارد را انتخاب کنید) و درخت حاصل را رسم کنید. می‌توانید از هر روشی برای رسم درخت استفاده کنید، چه به صورت چاپ کردن نتایج تا استفاده از کتابخانه‌ای مانند rcviz که در Github در دسترس است. در نهایت برای هر گره غیر برگ، نام ویژگی و قانون تقسیم را درج کنید و برای گره‌های برگ، کلاسی را که درخت تصمیم شما به آن اختصاص می‌دهد را درج کنید. در مورد راه‌حل و چالش‌های خود بحث کنید.

توضیحات بیشتر در مورد دادگان و پیاده‌سازی

- **پردازش دادگان تایتانیک:** در این بخش بر فیلدهای مجموعه‌داده تایتانیک مروری خواهیم داشت. در نظر داشته باشید که دادگان ارائه شده را از قبل به فرم قابل استفاده توسط پیاده‌سازی درخت تصمیم پردازش کنید.

survived	**برچسبی که می‌خواهیم پیش‌بینی کنیم. ۱ نشان‌دهنده این که فرد زنده مانده و صفر فرد مرده است.**
pclass	**نشان‌دهنده وضعیت اجتماعی-اقتصادی. ۱: بالا، ۲: متوسط و ۳: پایین**
age	**در صورت کمتر از ۱ بودن، مقدار کسری است.**

[1] Feature Transformation
[2] Bag-of-Words

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

مذکر/مونث	*sex*
تعداد خواهر و برادر/همسرات در کشتی	*sibsp*
تعداد والدین/فرزندان در کشتی	*parch*
شماره بلیط	*ticket*
کرایه	*fare*
شماره کابین	*cabin*
بندر سوار شدن (C=Cherbourg, Q=Queenstown, S=Southampton)	*embarked*

- **دو چالش** «متغیرهای رسته‌ای» و «مقادیر از دست رفته»

متغیر رسته‌ای: اکثر دادگانی که تا کنون با آن‌ها سر و کار داشته‌اید، دارای مقادیر پیوسته بوده‌اند، اما برخی از ویژگی‌ها در این دادگان به صورت نوع/رسته هستند. در ادامه دو روش ممکن در برخورد با مقادیر رسته‌ای بیان می‌شود.

- **روش آسان:** در مرحله استخراج ویژگی، دسته‌بندی‌ها را به متغیرهای دودویی نگاشت کنید. برای مثال فرض کنید که ویژگی ۲ سه مقدار ممکن را به خود می‌گیرد: TA، مدرس و دانشجو. در دادگان این رسته‌ها به سه متغیر دودویی نگاشت می‌شوند، که همان ستون‌های ۲، ۳ و ۴ در دادگان خواهند بود. برای مثال؛ ستون ۲ ویژگی Boolean است (دارای مقادیر ۰ یا ۱) که نشان‌دهنده رسته TA است و به همین صورت برای باقی. به عبارت دیگر، TA با [0,0,1]، مدرس با و دانشجو با – نشان داده می‌شود. توجه داشته باشید که با این کار تعداد ستون‌های دادگان شما را زیاد می‌کند. به این روش «بردارسازی[1]» یا «رمزگذاری تک‌نمود[2]» ویژگی‌های رسته‌ای می‌گویند.
- **روش مشکل‌تر اما قابل تعمیم‌تر:** رسته‌ها را به صورت متنی نگه داشته یا رسته‌ها به شاخص‌هایی نگاشت کنید (برای مثال؛ TA، مدرس و دانشجو به ۰، ۱ و ۲ نگاشت می‌شوند). پس عملکردی را در درخت تصمیم پیاده‌سازی کنید که برای تعیین قوانین تقسیم بر اساس زیر مجموعه‌های متغیرهای رسته‌ای، کسب اطلاعات به حداکثر می‌رسانند. شما نمی‌توانید این مقادیر را به عنوان ویژگی‌های با مقادیر پیوسته در نظر بگیرید، از آنجایی که ترتیب برای این دادگان معنایی ندارد (این واقعیت که $0 < 1 < 2$ هستند، در زمانی که ۰، ۱ و ۲ رسته‌هایی گسسته هستند، اهمیتی ندارد).

مقادیر از دست رفته: برخی از نمونه‌ها فاقد ویژگی هستند. این موارد با مقدار «؟» نشان داده شده‌اند. به طور معمول رویکردهای زیر در نظر گرفته می‌شود:

- **ساده‌ترین روش:** اگر نمونه‌ای برخی از ویژگی‌ها را ندارد، آن را از دادگان حذف کنید (از استفاده از این روش در پیاده‌سازی شما مجاز نیست!).
- **روش معمول:** مقدار ویژگی را از تمام مقادیر دیگر آن ویژگی استنتاج کنید (برای مثال؛ آن را با میانگین، میانه یا مد پر کنید. به این فکر کنید که کدام یک از این موارد می‌تواند بهترین انتخاب است و چرا).

[1] Vectorizing
[2] One-Hot Encoding

- **روش پیچیده‌تر اما قدرتمندتر:** از k-نزدیک‌ترین همسایه برای نسبت‌دادن مقادیر ویژگی بر اساس نزدیک‌ترین همسایگان یک نمونه داده استفاده کنید. در متریک فاصله خود، باید فاصله تا یک مقدار گمشده را تعریف کنید.
- **باز هم روش پیچیده‌تر اما واقعی‌تر:** عملکرد درخت تصمیم خود را برای رسیدگی به مقادیر گمشده بر اساس گره فعلی پیاده‌سازی کنید. راه‌های مختلفی به این منظور وجود دارد. ممکن است مقادیر گمشده را بر اساس میانگین/میانه/مد مقادیر ویژگی نمونه‌ها به صورت مرتب شده تا گره فعلی استنباط کنید. امکان دیگر تخصیص احتمال به هر مقدار از ویژگی گم شده و سپس مرتب‌سازی نمونه‌های وزن‌دار برای هر فرزند است (به عبارت دیگر شما باید به هر نمونه‌ای در درخت وزنی نسبت دهید).

- توصیه می‌شود از کلاس‌های زیر برای نوشتن، خواندن و پردازش داده‌ها استفاده کنید:

```
# csv.DictReader: Reads a CSV file and returns a dictionary of fieldnames and values for each row
# sklearn.feature_extraction.DictVectorizer: Vectorizing Categorical Variables i.e. Transforms a list of feature-value dictionaries into a sparse matrix of one-hot encoded features, There's also sklearn.preprocessingOneHotEncoder, but it's much less clean
# sklearn.preprocessing.OneHotEncoder: Encodes categorical features as a numeric array of binary values
# sklearn.preprocessing.LabelEncoder: If you choose to discretize but not vectorize categorical variables - Encodes labels with values between 0 and n_classes-1
# sklearn.preprocessing.Imputer: For inferring missing feature values in the preprocessing phase - Replaces missing values in a dataset with a specified strategy (mean, median, most frequent, etc.)
```

پیشنهاد: در صورتی که از csv.DictReader استفاده کنید، به صورت خودکار هدر فایل csv را خوانده و مقادیر را به فیلدهای دیکشنری منتسب می‌کند، بدین صورت از DictVectorizer می‌توان برای دودویی کردن مقادیر رسته‌ای استفاده کرد. همچنین برای سرعت بخشیدن در کار خود، ممکن است بخواهید ویژگی‌های تمیز شده را در فایلی ذخیره کنید تا هر بار که پیاده‌سازی خود را اجرا می‌کنید، نیازی به پیش‌پردازش نداشته باشید.

- **عملکرد تقریبی مورد انتظار:** برای دادگان هرزنامه، با یک درخت تصمیم، دقت اعتبارسنجی 79.9% و حدود 80.4% دقت اعتبارسنجی با اجرای جنگل تصادفی بر دادگان تایتانیک.
- **معماری پیشنهادی:** لازم است که در مورد چگونگی ساختارمندی درخت خود فکر کنید. ساختار حدودی پیشنهاد شده، تنها کلیت درخت شما را نشان می‌دهد. نوشتن کد تمیز زندگی شما را آسان‌تر می‌کند و این فقط به نفع خودتان است، ولی نمره بر اساس آن برای شما در نظر گرفته نمی‌شود. درخت تصمیم شما در حالت ایده‌آل باید دارای یک رابط کپسوله‌شده مانند زیر باشد:

```
classifier = DecisionTree(params)
classifier.train(train_data, train_labels)
predictions = classifier.predict(test_data)
```

که در آن train_data و test_data ماتریس های دوبعدی هستند (ردیف‌ها دادگان، ستون‌ها ویژگی‌ها هستند).

DecisionTree یک درخت دودویی است که از Nodeها تشکیل شده است. مقداردهی اولیه شامل تنظیم پارامترهای لازم بسته به تکنیک‌های پیاده‌سازی‌شده است. در زمان آموزش، درخت با انتخاب بهترین ویژگی و آستانه برای تقسیم داده‌ها در هر گره، گره‌هایی را برای دسته‌بندی ایجاد می‌کند. درخت گره‌ها را با اختصاص دادن یک قانون تقسیم، یک فرزند چپ، یک فرزند راست و برچسب (اگر گره یک گره برگ باشد) پیکربندی می‌کند.

فیلدهای Node شامل split_rule (یک دوتایی که ویژگی تقسیم در یک گره و مقدار آستانه برای تقسیم را نشان می‌دهد)، left (فرزند سمت چپ گره فعلی)، right (فرزند سمت راست گره فعلی)، label (اگر تنظیم شود، Node یک گره برگ است و فیلد حاوی برچسبی است برای دسته‌بندی یک نمونه در طول پیمایش درخت استفاده می‌شود).

توابع پیشنهادی درخت تصمیم:

`entropy(labels)`

آنتروپی را برای توزیع برچسب‌ها در یک گره محاسبه می‌کند.

`information_gain(features, labels, threshold)`

به دست آوردن اطلاعات یک تقسیم را با استفاده از یک آستانه مشخص محاسبه می‌کند.

`entropy(label)`

آنتروپی (یا ناخالصی جینی) را بر اساس برچسب‌های داده‌های ذخیره شده در یک گره محاسبه می‌کند.

`fit(data, labels)`

با ساخت گره‌ها درخت تصمیم را تشکیل می‌دهد. در این تابع قوانین تقسیم برای هر گره مشخص و تعیین می‌شود که چه زمانی ساخت درخت ادامه یا متوقف و گره برگ در نظر گرفته شود. بهتر است به صورت بازگشتی پیاده‌سازی شود.

`predict(data)`

برای هر نمونه داده شده، درخت را طی می‌کند تا بهترین برچسب را برای دسته‌بندی نمونه پیدا کند. با شروع از گره ریشه، قوانین تقسیم را در هر گره در طول پیمایش تا رسیدن به یک گره برگ ارزیابی می‌کند. برچسب گره برگ به عنوان برچسب خروجی انتخاب می‌شود.

جنگل تصادفی را می‌توان بدون تکرار کد و به کمک ذخیره گروه‌هایی از درخت‌های تصمیم پیاده‌سازی می‌شوند. هر درخت بر روی زیرمجموعه‌های مختلفی از دادگان (کیسه‌گذاری داده) آموزش داده می‌شود و گره‌های هر درخت بر روی زیرمجموعه‌های مختلفی از ویژگی‌ها (کیسه‌گذاری ویژگی) آموزش داده می‌شوند. بیشتر این قابلیت‌ها باید توسط یک کلاس جنگل تصادفی مدیریت شوند، به جز کیسه‌گذاری که نیاز به پیاده‌یازی در کلاس درخت تصمیم داشته باشد. توضیح‌های ارائه شده نقطه شروعی برای پیاده‌سازی درخت تصمیم است، که اهمیت طراحی دقیق قبل از کدنویسی را تاکید می‌کند.

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

سنجه‌های جداسازی[1]

هدف از این تمرین آشنایی با معیارهای مختلف برای انتخاب بهترین ویژگی جهت جداسازی دادگان است.

۱. **آنتروپی و جینی در دو بعد**: همان طور که می‌دانید آنتروپی و جینی دو تابع برای نمایان کردن میزان پراکندگی دادگان هستند. برای یک مسئله‌ی دو کلاسه، این دو تابع را در یک نمودار رسم کنید و تفاوت‌ها را شرح دهید. (توجه کنید که با دانستن احتمال هر یک از کلاس‌ها احتمال دیگری را می‌توان به دست آورد[2])

۲. **آنتروپی و جینی در سه بعد**: مورد بخش یک را برای مسئله‌ی ۳ کلاسه تکرار کنید (توجه کنید که با دانستن احتمال دو کلاس می‌توان احتمال کلاس سوم را محاسبه کرد).

۳. اگر از معیار کسب اطلاعات[3] برای جداسازی در گره‌ها استفاده کنیم و مساله‌ی ما imbalance باشد یعنی تعداد دادگان کلاس بسیار متفاوت از یکدیگر باشد، چه مشکلی پیش می‌آید؟ راه‌حلی برای آن پیشنهاد دهید (راهنمایی: می‌توانید در مورد معیار gain ratio مطالعه کنید).

دسته‌بندی گل‌های زنبق

هدف از این تمرین ساخت و بررسی و تحلیل نتایج بدست آمده از درخت تصمیم حاصل از ویژگی‌های متفاوت است. مجموعه دادگان گل‌های زنبق در اختیار شما قرار داده شده است، این دادگان دارای ۴ ویژگی پیوسته و برچسب کلاس است. قصد داریم ۴ درخت تصمیم مختلف بسازیم که هر طبقه‌بند تنها از یکی از ویژگی‌ها برای ساخت مدل استفاده می‌کند. به عبارت دیگر، طبقه‌بند اول از ویژگی ۱، طبقه‌بند دوم از ویژگی ۲ و الی آخر استفاده می‌کند. دادگان خود را به نسبت ۴ به ۱ به دادگان آموزش و آزمایش تقسیم کنید. (**استفاده از کتابخانه مجاز نیست!**)

۱. چهار طبقه‌بند با تنها **یک** ویژگی بسازید و دقت هر کدام را محاسبه کنید.

۲. طبقه‌بند درخت تصمیم با **۴** ویژگی را بسازید و معیارهای طبقه‌بندی را برای آن گزارش کنید.

۳. بهترین طبقه‌بند با ۱ ویژگی را با طبقه‌بند با ۴ ویژگی از نظر دقت مقایسه کنید، در مورد نتایج خود بحث کنید.

بیش‌برازش[4] و کم‌برازش[5]

یکی از مشکل‌های بزرگ مدل‌های یادگیری ماشین بیش‌برازش و کم‌برازش است در این مساله به بررسی دقیق‌تر این پدیده‌ها و نحوه مقابله با آن‌ها در درخت تصمیم می‌پردازیم (می‌توانید از توابع موجود در scikit-learn استفاده کنید).

- با استفاده از کد داده شده نمودار زیر را بازسازی کنید.

```
import numpy as np

np.random.seed(0)
X = np.sort(10 * np.random.rand(500, 1), axis=0)
y = np.sin(X).ravel() + 0.2 * np.random.randn(500)
X_train, X_test, y_train, y_test = X[:400], X[400:], y[:400], y[400:]
```

[1] Spliting Criteria
[2] P, 1-P
[3] Information Gain
[4] Overfitting
[5] Underfitting

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

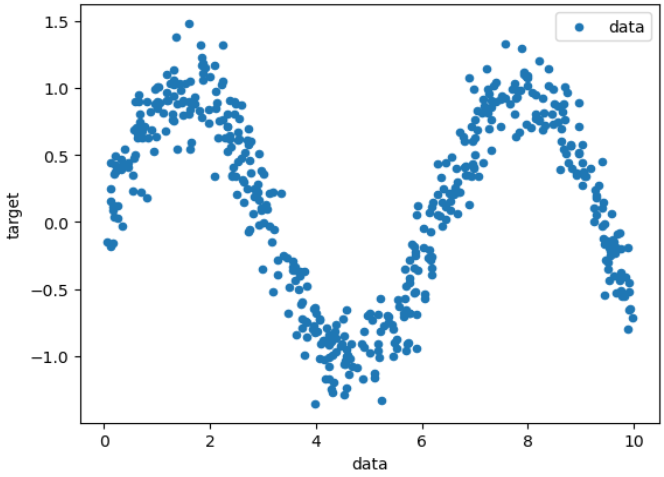
سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵



شکل ۲: نمونه خروجی دادگان تولیدی

۱. دو مدل درخت تصمیم با عمق ۲ و ۱۰ بر روی داده‌ها آموزش دهید، نتایج مدل را برروی کل داده به همراه نقاط واقعی نمایش دهید؟ آیا مدل‌ها خوب عمل می‌کنند؟ چرا؟

۲. برای حل مشکل بیش‌برازش مدل با عمق ۱۰ چه روش‌هایی قابل انجام است؟ این روش‌ها را بر روی مدل اعمال کنید و نتایج را گزارش کنید.

۳. چه هایپرپارامترهایی بر بیش‌برازش و کم‌برازش مدل تأثیر گذارند؟ با تغییر این متغیرها عملکرد مدل را در خطای داده آموزش و تست بررسی کنید و هایپرپارامتر بهینه را گزارش کنید (برای پیدا کردن پارامترها از اعتبارسنجی متقابل[1] استفاده کنید، از داده تست صرفاً در زمانی که پارامترهای بهینه را پیدا کردید جهت تست مدل استفاده نمایید).

هرس پیچیدگی هزینه‌ی مدل[2]

هرس کردن[3] یکی از روش‌های غلبه بر بیش‌برازش است، در این روش بخشی از درخت برای عملکرد بهتر آن حذف می‌شود، به طوری که حساسیت آن به داده‌ی آموزش از بین برود تا بر روی هر نوع داده‌ای خودش را بتواند تطابق دهد. در مدل‌های مبتنی بر درخت‌های تصمیم از این روش برای کاهش اندازه (پیچیدگی) درخت استفاده می‌شود، با این هدف که خطای آموزش کمی افزایش و خطای آزمایش کاهش یابد که نتیجتاً مدلی سازگارتر خواهیم داشت. یکی از روش‌های هرس‌کردن در درخت‌های تصمیم روش هرس پیچیدگی هزینه است. معیار پیچیدگی هزینه به صورت زیر تعریف می‌شود:

$$R_\alpha(T) = R(T) + \alpha|T|$$

به طوری که $R(T)$ مقدار خطای آموزش درخت T است و $|T|$ اندازه درخت T و α یک پارامتر است که مقدار آن براساس تجربه تعیین می‌شود.

---

[1] Cross Validation
[2] Cost Complexity Pruning
[3] Pruning

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

می‌خواهیم به کمک این الگوریتم، بازه‌ای بهینه برای α بدست بیاوریم، به طوری که قبل از این بازه دقت آموزش بالا و دقت آزمایش پایین باشد و پس از این بازه دقت آزمایش افزایش و دقت آموزش کاهش یابد. مراحل زیر را بر دادگان iris.csv به کمک الگوریتم هرس پیچیدگی هزینه انجام دهید (می‌توانید از توابع موجود در scikit-learn استفاده کنید).

۱. دادگان را به صورت تصادفی با نسبت ۴ به ۱ به دو بخش آموزش و آزمایش تقسیم کنید. سپس درخت تصمیم را بر آن آموزش دهید. همچنین دقت آموزش و آزمایش را گزارش کنید.

۲. به کمک هرس پیچیدگی هزینه، مقادیر مختلف آلفا را بدست آورید.

۳. برای هر مقدار آلفا بدست آمده، درخت تصمیم متناظر آن را بر روی مجموعه‌دادگان آموزش دهید. یک نمودار از دقت آموزش و آزمایش این درخت‌ها نسبت به مقادیر مختلف آلفا رسم کنید، سپس با تحلیل نمودار بازه‌ی بهینه را برای آلفا گزارش دهید.

جنگل تصادفی[1]

در این بخش، قصد داریم به بررسی عملکرد جنگل تصادفی بپردازیم و با پیاده‌سازی آن، مفاهیم اساسی، منطق طراحی و کاربردهای آن در سناریوهای دنیای واقعی را بررسی می‌کنیم. دادگان ارائه شده، دادگان سرطان سینه بوده که در اختیار شما قرار گرفته شده است (می‌توانید از توابع موجود در scikit-learn استفاده کنید).

۱. ترکیب درخت تصمیم با استفاده از تکنیک کیسه‌گذاری[2]

۱. کیسه‌بندی روشی محبوب و ساده برای یادگیری دسته‌جمعی[3] است. در این سوال، ابتدا یک درخت‌تصمیم ساده از صفر را پیاده‌سازی کرده و سپس با ترکیب چندین درخت با استفاده از رای اکثریت[4] عملکرد آن را با یک درخت‌تصمیم به تنهایی مقایسه کنید (جدول درهم‌ریختگی[5] و معیارهای فراخوانی[6] و صحت[7] و دقت[8] را به همراه نتایج خود گزارش کنید).

۲. آیا عملکرد مدل با این روش بهبود پیدا می‌کند؟

۳. مشکل این روش چیست؟

معیار جداسازی شاخه‌ها را gini index در نظر بگیرید، همچنین عمق هر درخت را حداکثر ۵ انتخاب کنید.

۲. **بکارگیری روش Bootstrapping**: وجه تمایز روش جنگل‌تصادفی با متد سوال قبلی انتخاب داده ورودی برای هر درخت به شکل Bootstrapping است، در این روش نمونه‌های ورودی و همچنین ویژگی هر نمونه به شکل تصادفی و با جایگزینی انتخاب شده و به هر درخت به عنوان ورودی داده می‌شود.

۱. روش جنگل تصادفی را پیاده‌سازی کرده و عملکرد آن را با قبلی مقایسه نمایید.

تعداد درخت ها را حداکثر۵۰ انتخاب کنید.

۲. توضیح دهید که دلیل استفاده از روش Bootstrapping چیست؟

---

[1] Random Forest
[2] Bagging
[3] Ensemble Learning
[4] Majority Voting
[5] Confusion Matrix
[6] Recall
[7] Precision
[8] Accuracy

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

۳. در صورت عدم استفاده از روش Bootstraping چه مشکلی در عملکرد مدل به وجود می‌آید؟

۳. مجموعه دادگان را به نحوی تغییر دهید که «تعداد نمونه‌های با برچسب یک ده برابر نمونه‌های با برچسب صفر شود».

۱. مدل Random Forest را دوباره بر روی دادگان جدید آموزش دهید و نتایج را مانند قسمت قبل گزارش کنید. چه تغییر در عملکرد مشاهده می‌شود؟

۲. پس از مطالعه مقاله ضمیمه شده به همراه تمرین مدل BRF[1] را پیاده‌سازی کنید. نتایج را با بخش قبل مقایسه کنید.

AdaBoost

هدف این تمرین آشنایی با Adaboost[2] است. می‌خواهیم نشان دهیم چگونه می‌توان از تعداد زیادی جداساز[3] ضعیف به یک جداساز قدرتمند رسید. فرض کنید در یک مساله‌ی دو کلاسه به تعداد دلخواه جداساز با دقت ٪۵۱ داریم. حال تعداد n تا از این جداسازها را کنار هم قرار می‌دهیم و خروجی داده‌ی آزمایش روی همه‌ی آن‌ها را می‌گیریم. هر کلاسی که توسط جداسازهای بیشتری پیش‌بینی شده‌باشد را به عنوان جواب نهایی انتخاب می‌کنیم، در واقع مطابق رای اکثریت عمل می‌کنیم.

۱. **دقت کم تعداد زیاد!:** ابتدا رابطه‌ی ریاضی دقت n جداساز را محاسبه کرده و دقت جداساز نهایی را برای nهای ۱، ۱۰، ۱۰۰۰ و بی‌نهایت محاسبه کنید.

۲. **افزایش دقت:** حال کمی دقت را افزایش می‌دهیم، فرض کنید دقت هر جداساز ٪۶۰ است. دقت جداساز نهایی را برای n های مشابه بالا محاسبه کنید. چه نتیجه‌ای می‌گیرید؟

۳. **دقت برابر با تعداد متفاوت:** حساب کنید اگر بخواهیم به دقت جداساز ۱ وقتی n برای ۱۰۰۰ است برسیم، به چه تعداد از جداساز ۲ نیاز داریم؟

۴. **بی‌نهایت جداساز:** با توجه به دقت به دست آمده برای بی‌نهایت جداساز، آیا می‌توان همواره به این دقت دست یافت؟ چرا؟

یادگیری جمعی

روش‌های یادگیری جمعی یکی از کاربردی‌ترین متدهای یادگیری ماشین جهت افزایش تعمیم‌پذیری مدل‌ها و کاهش خطای واریانس و بایاس است، روش‌های متفاوتی برای ترکیب مدل‌های پایه وجود دارد که قصد داریم در این سوال با آن‌ها آشنا شویم (جهت پیاده‌سازی این سوال از مجموعه دادگان سرطان سینه استفاده کنید).

۱. در مورد روش تقویت گرادیان[4] مطالعه کنید و آن را با درخت تصمیم پیاده‌سازی کنید. عملکرد مدل را با سه مرتبه افزایش لایه‌ها (به طور مثال ۵، ۱۰ و ۱۵) بررسی کنید (جدول درهم‌ریختگی و معیارهای فراخوانی و صحت و دقت را به همراه نتایج خود گزارش کنید).

---

[1] Balanced Random Forest

[2] برای مطالعه‌ی بیشتر درباره‌ی Adaboost می‌توانید به این یا این لینک مراجعه کنید.

[3] Classifier

[4] Gradient Boosting

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

۲. در مورد روش پشته‌سازی[1] مطالعه کرده و با استفاده از مدل‌های Logistic Regression، SVM، Random Forest و Adaboost یک مدل پشته پیاده کنید و عملکرد آن را بررسی کنید (برای مدل های پایه و درس داده نشده، مجاز به استفاده از کتابخانه هستید).

۳. عملکرد کدام یک از این سه روش برروی دادگان داده شده بهتر بود؟ به نظر شما دلیل این موضوع چیست؟

بوت‌استرپ: نمونه‌برداری مجدد برای کیسه‌گذاری موفق[2]

هدف از این بخش یادگیری نحوه استفاده از نمونه‌برداری مجدد bootstrap برای ایجاد مدل‌های مختلف و کاهش واریانس پیش‌بینی‌ها است. کیسه‌گذاری یک روش تصادفی برای ایجاد تعداد زیادی یادگیرنده مختلف از یک مجموعه‌داده است.

۱. **انگیزه پشت میانگین‌گیری:** مجموعه‌ای از متغیرهای تصادفی غیر همبسته $\{Y_i\}_{i=1}^n$ با میانگین μ و واریانس σ^2 در نظر بگیرید. امید ریاضی و واریانس میانگین آن‌ها را محاسبه کنید (در بحث روش‌های جمعی[3]، Y_iها را می‌توان به پیش‌بینی‌های که توسط دسته‌بند i انجام می‌شود، تشبیه کرد).

۲. **یادگیری جمعی – کیسه‌گذاری**: کیسه‌گذاری[4] یک روش تصادفی برای ایجاد تعداد زیادی یادگیری مختلف از یک مجموعه‌داده است.

۱. با در دست داشتن مجموعه آموزشی با اندازه n، با نمونه‌گیری با جایگزینی T زیرنمونه تصادفی که هر کدام به اندازه n' هستند، تولید کنید. برخی از نمونه‌ها ممکن است چندین بار انتخاب شوند، در حالی که برخی اصلا انتخاب نشوند. اگر n با n' برابر باشد، حدود 63% از نمونه‌ها انتخاب می‌شوند و باقی 37% نمونه‌های خارج از کیسه[5] نامیده می‌شوند. چرا 63%؟ (راهنمایی: وقتی n بسیار بزرگ است، احتمال اینکه یک نقطه نمونه انتخاب نشود چقدر است؟)

۲. اگر از کیسه‌گذاری برای آموزش مدل خود استفاده کنیم، چه پیشنهادی برای انتخاب هایپرپارامتر T می‌کنید؟ (به خاطر داشته باشید که T تعداد درخت‌های تصمیم در ensemble و تعداد زیرنمونه‌ها است و معمولا بسته اندازه و ماهیت مجموعه آموزشی از ده‌ها تا چندین هزار درخت استفاده می‌شود).

۳. در بخش یک این تمرین دیدیم که میانگین‌گیری واریانس دسته‌بندهای غیر همبسته را کاهش می‌دهد. اگرچه پیش‌بینی در دنیای واقعی کاملا غیر همبسته نخواهد بود، اما کاهش همبستگی بین درخت‌های تصمیم معمولا واریانس نهایی را کاهش می‌دهد. مجموعه‌ای از متغیرهای تصادفی همبسته $\{Z_i\}_{i=1}^n$ که دارای میانگین μ و واریانس σ^2 هستند و در آن هر $Z_i \in R$ یک اسکالر است را در نظر بگیرید. فرض کنید

$$\forall i \neq j, Corr(Z_i, Z_j) = \rho$$

واریانس میانگین آن‌ها را محاسبه کنید.

نقطه جداسازی

هدف از این سوال بررسی مساله‌هایی است که در آن‌ها یک یا تعدادی از ستون‌های ویژگی‌های دادگان دودویی نیستند. در نتیجه نیاز به انتخاب نقطه مرزی است، سپس براساس آن مرز در نظر گرفته شده، ویژگی مدنظر در دادگان را به دو قسمت بزرگ‌تر و کوچک‌تر

---

[1] Stacking
[2] Bootstrap: Resampling for bagging success.
[3] Ensemble Methods
[4] Bootstrap AGGregatING
[5] Out-of-Bag sample points

یادگیری ماشین

تمرین «دو»

دکتر سامان هراتی‌زاده

دانشگاه تهران - دانشکده سامانه‌های هوشمند

نیم‌سال اول ۱۴۰۳-۱۴۰۲

دستیاران آموزشی

سیاوش رزمی

آیدین کیانی

امیر سیف الهی

امیرمحمد کویش‌پور

ددلاین: ساعت ۲۳:۵۹ | ۱۴۰۲/۰۸/۱۵

از نقطه مرزی تقسیم می‌کنیم، سپس مقدار یک را به دادگان بزرگ‌تر و صفر را به دادگان کوچکتر اختصاص می‌دهیم. یکی از روش‌های انتخاب نقطه مرزی به صورت زیر است:

$$m = \frac{x + y}{2}$$

X: کمترین مقدار داده و Y: بیشترین مقدار داده

حال قصد داریم بررسی کنیم به ازای مقادیر مختلف نقطه جداسازی عملکرد درخت تصمیم چگونه خواهد شد (می‌توانید از توابع موجود در scikit-learn استفاده کنید).

۱. **میانگین و میانه:** برای دادگان car_data.csv داده شده به ازای عمق‌های مختلف درخت و بدون محدودیت عمق مدل را آموزش دهید سپس خطا را روی داده‌های آموزش و تست بدست آورید. نمودار خطا به ازای عمق درخت را بدست آورید. قاعدتا عمق بیشتر منجر به بیش‌برازش خواهد شد پس در نتیجه نمودار خطا داده‌های تست بعد از عمق مشخص افزایش خواهد یافت. حال برای نقطه جداسازی میانگین و میانه داده‌ها نیز انجام دهید و برای هر حالت نمودار خطای داده‌های آموزش و آزمایش را رسم کنید.

۲. **بررسی نقطه جداسازی:** حال با توجه نمودار های بدست آمده در قسمت قبل سوال بررسی کنید که کدام نقطه جدا سازی عملکرد مناسب تری دارد؟ فرض کنید داده های شما توزیع نمایی دارند تعیین کنید کدام یک از آماره های میانگین و میانه عملکرد بهتر جداسازی دارد؟ به طور کلی و با توجه به پرسش قبل آیا می توان گفت ارتباطی بین نقطه جدا سازی و نوع تابع توزیع احتمال وجود دارد؟ دلایل خود را شرح دهید.