بازیابی پیشرفته اطلاعات

دکتر بیگی

گزارش فاز دوم پروژه

فاطمه هادی‌زاده ۹۵۱۰۵۹۰۲
فائزه پویامهر ۹۵۱۰۵۴۴۳
فاطمه باقری ۹۵۱۰۵۴۱۹

بخش ۱. پیاده‌سازی دسته‌بندها

Naive Bayes .1.1

این الگوریتم تلاش می کند بر اساس رابطه ی bayes احتمال این که یک مستند در یک دسته قرار گیرد را به دست آورد؛ یعنی مقدار $p(c \mid d)$ که در آن c یکی از دسته‌هاست. رابطه ی Bayes بیان می کند که

$$p(c|d) = \frac{p(c)p(d\,|c)}{p(d)}$$

هنگام دسته بندی، از آن جا که مستند ثابت است $p(d)$ ثابت است. پس

$$p(c|d) \propto p(c)p(d\,|c)$$

برای یک مستند که n_d ترم دارد، رابطه ی بالا می شود

$$p(c|d) \propto p(c) \prod_{k=1}^{n_d} p(t_k\,|c)$$

در t_k در آن ترم kام است.
در این پروژه، ما در فایل NaiveBayes.py مقادیر $p(c)$ و $p(t_k\,|c)$ ها را از روی داده های یادگیری با دست می آوریم. $p(c)$ برای هر کلاس به صورت زیر حساب می شود

$$p(c) = \frac{N_c}{N}$$

که برابر است با تعداد مستندات دسته ی c نسبت به تعداد کل مستندات.

برای محاسبه ی $p(t_k\,|c)$ از *smoothing* استفاده می کنیم.

$$p(t\,|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} T_{ct'} + |V|}$$

T_{ct} تعداد دفعات تکرار ترم t در دسته ی c است و V مجموعه ی همه ی کلمات است.
همه ی این ها در تابع *fit* اتفاق می افتد.

در تابع *predict* به ازای مستند d، $p(c=1|d)$ و $p(c=-1|d)$ طبق رابطه ی *Bayes* به دست می آیند و هر کدام که بزرگتر باشد، به عنوان دسته ی *predict* شده انتخاب می شود.

K-NN .1.2

این الگوریتم در فایلknn.py پیاده سازی شده است و حاوی تابع fit و predict است که بر اساس فاصله اقلیدسی در فضای برداری عمل میکند.

پیاده سازی تابع validation: این تابع با train کردن الگوریتم های knn و svm روی 90 درصد از مجموعه آموزش اولیه با گزارش عملکرد روی مجموعه ی validation که 10 درصد از مجموعه آموزش اولیه امان هست پارامتر های متفاوت در هر دو الگوریتم را مقایسه میکند.

مقدار های زیر نتایج اجرای پارامترهای مختلف روی 90 درصد داده های آموزش اولیه و گرفتن گزارش از 10 درصد داده های آموزش (یا validation) را نشان می دهد.

برای این کار روی مقادیر مختلف تعداد feature ها که در یک بازه ای انتخاب شدند به ازای هر کدام الگوریتم را اجرا کردیم و بیشترین مقدار دقت در نهایت درزیر گزارش شده است که مشاهده میشد با افزایش تعداد ویژگی تا حدی میزان دقت افزایش می یافت.همانطور که از مقادیر زیر مشاهده میشود هر 3 پارامتر دارای مقدار های مساوی برای 3 معیار محاسبه شده می باشد بنابراین انتخاب بهترین آن به صورت رندم یکی از آن ها را(پارامتر 5)را انتخاب کردیم.

param 1[{'accuracy': 0.5866666666666667, 'precision': 0.5866666666666667, 'recall': 1.0, 'F1': 0.7394957983193278},

param 5 {'accuracy': 0.5866666666666667, 'precision': 0.5866666666666667, 'recall': 1.0, 'F1': 0.7394957983193278},

param 9{'accuracy': 0.5866666666666667, 'precision': 0.5866666666666667, 'recall': 1.0, 'F1': 0.7394957983193278}]

SVM .1.3

در این الگوریتم تلاش می شود با پیدا کردن یک تابع در فضای اقلیدسی، نقاط داده ها را از هم جدا کرد. این تابع می تواند خطی باشد. در این صورت، نقاطی که در یک طرف خط قرار می گیرند برچسب *1* و نقاطی که در طرف دیگر قرار می گیرند برچسب *-1* می گیرند. در *soft margin SVM* با پارامتر *C* اجازه داده می شود که هنگام *fit* کردن به تعداد *c* داده به صورت اشتباه دسته بندی شوند. این کار برای جلوگیری از *overfitting* انجام می شود. در این پروژه ما با استفاده از کتابخانه ی *sklearn* یک مدل *SVC* ساخته و آن را با پارامترهای *c= 0.5, 1, 1.5, 2* *fit* کردیم. برای پیش بینی برچسب داده های تست هم از تابع *predict* استفاده کردیم. برای *validate* کردن از بخشی از تابع های یادگیری استفاده کردیم و با استفاده از کتابخانه ی *sklearn.metrics* مقادیر *accuracy*، *recall*، *precision* و *F1* را به دست آوردیم.

مقدار های زیر نتایج اجرای پارامترهای مختلف روی 90 درصد داده های آموزش اولیه و گرفتن گزارش از 10 درصد داده های آموزش (یا validation) را نشان می دهد.

برای این کار روی مقادیر مختلف تعداد feature ها که در یک بازه ای انتخاب شدند به ازای هر کدام الگوریتم را اجرا کردیم و بیشترین مقدار دقت در نهایت درزیر گزارش شده است که مشاهده میشد با افزایش تعداد ویژگی تا حدی میزان دقت افزایش می یافت. همانطور که از مقادیر زیر مشاهده میشود هر 3 پارامتر دارای مقدار های مساوی برای 3 معیار محاسبه شده می باشد بنابراین انتخاب بهترین آن به صورت رندم یکی از آن ها را(پارامتر 1)را انتخاب کردیم.

param 0.5:[{'accuracy': 0.5833333333333334, 'precision': 0.5833333333333334, 'recall': 1.0, 'F1': 0.7368421052631579},

param 1:{'accuracy': 0.5833333333333334, 'precision': 0.5833333333333334, 'recall': 1.0, 'F1': 0.7368421052631579},

param 1.5{'accuracy': 0.5833333333333334, 'precision': 0.5833333333333334, 'recall': 1.0, 'F1': 0.7368421052631579},

param 2: {'accuracy': 0.5833333333333334, 'precision': 0.5833333333333334, 'recall': 1.0, 'F1': 0.7368421052631579}]

1.4. Random Forest

در این قسمت هم با استفاده از *sklearn.ensemble.RandomForestClassifier* مدلمان را ساخته، با استفاده از داده های یادگیری *fit* کرده و سپس *validate* و *predict* کردیم.

بخش ۲. بهبود سیستم بازیابی اطلاعات فاز اول پروژه

در این بخش تغییراتی در کد بخش Console و همچنین کلاس Searcher فاز اول اعمال کردیم. برای جست‌وجو در کلاس خاص دو روش زیر را پیاده‌سازی کردیم.

روش اول: ابتدا تمام مستندات انگلیسی را با استفاده از بهترین دسته‌بندی که در بخش قبلی به دست آوردیم، به دو دسته کلاس ۱، برای پر بازدیدها، و کلاس ۲، برای بقیه، تقسیم می‌کنیم و مستندات دو کلاس را به صورت جداگانه ایندکس می‌کنیم. با توجه کلاسی که می‌خواهیم در آن کوئری وارد شده را جست‌وجو کنیم، جست‌وجو را با استفاده از ایندکس کلاس مربوطه انجام می‌دهیم. با این روش هزینه پیش‌پردازش بیشتر می‌شود اما هزینه‌ی جست‌وجو افزایش نمی‌یابد و حتی نسبت به جست‌وجو در کل داده‌ها که در فاز اول داشتیم، سریع‌تر عمل می‌کند.

روش دوم: دسته‌بندی را در زمان جست‌وجو انجام می‌دهیم. یعنی در کلاس Searcher هنگام انتخاب مستندات مرتبط، tag مستند را با استفاده از دسته‌بند پیش‌بینی کرده و با کلاس موردنظر مقایسه می‌کنیم و در صورت یکی بودن tag پیش‌بینی شده با کلاس موردنظر، مستند را مرتبط در نظر می‌گیریم. که در این روش هزینه جست‌وجو خیلی زیاد می‌شود اما پیش‌پردازش نداریم.

در مجموع روش اول روش بهتری است. برای اجرای برنامه به روش اول باید بعد از اجرای تابع Console ورودی classify را به برنامه دهیم تا مستندات را دسته‌بندی و به صورت جداگانه ایندکس کند. همچنین برای دیدن tag پیش‌بینی‌شده برای تمام مستندات انگلیسی ورودی p2 را وارد می‌کنیم.

در قسمت search، برای اینکه در کلاس خاصی جست‌وجو انجام شود، عدد کلاس پرسیده می‌شود، که در صورت وارد کردن عدد 1، جست‌وجو در کلاس پربازدیدها، کلاس ۱، انجام می‌گیرد. در صورت وارد کردن عدد -1 جست‌وجو در کلاس ۲ انجام می‌شود. و در صورتی که عددی وارد نکنیم، جست‌وجو در بین کل مستندات انجام می‌شود. می‌توان بعد از مشاهده‌ی لیست مستندات مرتبط، مقدار doc_id مستند را برای نمایش آن مستند وارد کرد. که محتوای خام title و description مستند (قبل از پیش‌پردازش آن) نمایش داده می‌شود. همچنین tag views نیز نمایش داده می‌شود. که با توجه با مقدار آن، اگر بیشتر از میانگین تعداد بازدیدها باشد 1 و در غیر این صورت -1 است.

نمونه جست‌وجو در کلاس‌های مختلف در زیر آمده است. ابتدا جست‌وجو در کل مستندات خواسته شده و در مستندات پر بازدید و در آخر جست‌وجو در مستندات کم‌بازدید انجام شده است.

Output:

```
_____________________
Enter the section number:
    5
Enter question number:
    1
Enter the query:
    stanford university
Enter search field (title/description) or press Enter to skip:

Enter search class (1 for most viewed, -1 for other) or press Enter to skip:

  Scored tf-idf Search result for "stanford univers" is:
  -> [1030, 589, 1210, 889, 213, 2036, 1812, 26, 1821, 311]
  Enter a doc_id to load and show the doc's content(press Enter to skip):



_____________________
Enter the section number:
    5
Enter question number:
    1
Enter the query:
```

```
    stanford university
Enter search field (title/description) or press Enter to skip:

Enter search class (1 for most viewed, -1 for other) or press Enter to skip:
    1
  Scored tf-idf Search result for "stanford univers" is:
  -> [1030, 589, 1210, 213, 2036, 1812, 26, 1821, 311, 600]
  Enter a doc_id to load and show the doc's content(press Enter to skip):
        1030
{'title': 'How to live before you die',
'description': "At his Stanford University commencement speech, Steve Jobs,
CEO and co-founder of Apple and Pixar, urges us to pursue our dreams and see
the opportunities in life's setbacks -- including death itself.",
'views': 1}
------
     doc_id:
        589
{'title': 'The uniqueness of humans',
'description': 'At Stanford University, primatologist Robert Sapolsky offers
a fascinating and funny look at human behaviors which the rest of the animal
kingdom would consider bizarre.',
'views': -1}
------
     doc_id:
        1210
{'title': 'Is our universe the only universe?',
'description': 'Is there more than one universe? In this visually rich,
action-packed talk, Brian Greene shows how the unanswered questions of
physics (starting with a big one: What caused the Big Bang?) have led to the
theory that our own universe is just one of many in the "multiverse."',
'views': 1}
------
     doc_id:
        213
{'title': 'Questioning the universe',
'description': 'In keeping with the theme of TED2008, professor Stephen
Hawking asks some Big Questions about our universe -- How did the universe
begin? How did life begin? Are we alone? -- and discusses how we might go
about answering them.',
'views': 1}
------



_____
Enter the section number:
    5
Enter question number:
    1
Enter the query:
    stanford university
Enter search field (title/description) or press Enter to skip:

Enter search class (1 for most viewed, -1 for other) or press Enter to skip:
    -1
  Scored tf-idf Search result for "stanford univers" is:
  -> [889, 461, 1267, 1473, 77, 918, 2073]
  Enter a doc_id to load and show the doc's content(press Enter to skip):
```

889
{'title': 'The sound the universe makes',
'description': 'We think of space as a silent place. But physicist Janna Levin says the universe has a soundtrack -- a sonic composition that records some of the most dramatic events in outer space. (Black holes, for instance, bang on spacetime like a drum.) An accessible and mind-expanding soundwalk through the universe.',
'views': -1}
---
doc_id:
461
{'title': 'A university for the coming singularity',
'description': "Ray Kurzweil's latest graphs show that technology's breakneck advances will only accelerate -- recession or not. He unveils his new project, Singularity University, to study oncoming tech and guide it to benefit humanity.",
'views': -1}
---
doc_id:
1267
{'title': 'The 100,000-student classroom',
'description': 'In the fall of 2011 Peter Norvig taught a class with Sebastian Thrun on artificial intelligence at Stanford attended by 175 students in situ -- and over 100,000 via an interactive webcast. He shares what he learned about teaching to a global classroom.',
'views': -1}
---
doc_id:
1473
{'title': '4 pillars of college success in science',
'description': "At age 12, Freeman Hrabowski marched with Martin Luther King. Now he's president of the University of Maryland, Baltimore County (UMBC), where he works to create an environment that helps under-represented students -- specifically African-American, Latino and low-income learners -- get degrees in math and science. He shares the four pillars of UMBC's approach.",
'views': -1}
---
doc_id:

بخش ۳. ارزیابی نهایی

معیارهای خواسته شده بر روی داده‌های آموزش و آزمون برای هر یک از الگوریتم‌های پیاده‌سازی شده در بخش ۱، به صورت زیر است:

```
features num: 1000
-> KNN:
train report:
accuracy: 0.616 precision: 0.9491525423728814 recall: 0.22857142857142856 F1: 0.3684210526315789

test report:
accuracy: 0.5555555555555556 precision: 0 recall: 0.0 F1: 0.0
_____________________

-> Naive Bayes:
train report:
accuracy: 0.676 precision: 1.0 recall: 0.33877551020408164 F1: 0.5060975609756098

test report:
accuracy: 0.6666666666666666 precision: 0.631578947368421 recall: 0.6 F1: 0.6153846153846154
_____________________

-> SVM:
train report:
accuracy: 0.676 precision: 1.0 recall: 0.33877551020408164 F1: 0.5060975609756098

test report:
accuracy: 0.6 precision: 0.5555555555555556 recall: 0.5 F1: 0.5263157894736842
_____________________

-> Random Forest:
train report:
accuracy: 0.534 precision: 1.0 recall: 0.04897959183673469 F1: 0.0933852140077821

test report:
accuracy: 0.5555555555555556 precision: 0 recall: 0.0 F1: 0.0
_____________________
```

برای overfit نشدن دسته‌بند روی مجموعه‌ی داده آموزش، بخشی از کلمات موجود را به عنوان feature برای یادگیری انتخاب کردیم. انتخاب کلمات را بر اساس idf آن‌ها در مجموعه مستندات آموزش انجام دادیم. همچنین برای یادگیری، 1000 مستند را به طور رندوم از بین مستندات آموزش انتخاب می‌کنیم. چون اینکار رندوم انجام می‌شود ممکن است مقادیر معیارهای بالا در اجرای‌های مختلف متفاوت باشد.

با چندین بار اجرای هر یک از الگوریتم‌های دسته‌بندی روی تعداد مستندات مختلف و انتخاب تعداد feature های متفاوت، به این نتیجه رسیدیم که به طور میانگین الگوریتم SVM عملکرد بهتری دارد. بنابراین در بخش ۲ از این دسته‌بند استفاده کردیم.

می‌توان دید که در صورت اجرای الگوریتم‌ها روی تمام مجموعه داده و در نظر گرفتن تمام کلمات آن‌ها، باعث overfit شدن دسته‌بند می‌شود. و دقت روی مجموعه داده‌های آموزش ۱۰۰ درصد شده در حالی که دقت روی مجموعه داده‌های آزمون پایین است.