گزارش تمرین شماره ۳ درس داده کاوی

اساتید گرامی :

جناب آقای دکتر فراهانی و جناب آقای دکتر خردپیشه

دستیار آموزشی: جناب آقای شریفی

گردآورنده: هدیه آشوری ۹۹۴۲۲۰۲۲

۱۴۰۰/۰۳/۱۴

۱. درخصوص کرنل های پرکاربرد روش SVM تحقیق کنید. .به صورت کلی چرا ما از ایده کرنل در بحث SVM بهره می بریم آیا می توان درخصوص کرنل ها و استفاده آنها حکم کامل داد. به طور مثال بگوییم از کرنل RBF در این مواقع خاص استفاده می کنیم.

هدف اول در طبقه بندی با ماشین بردار پشتیبان، ایجاد بیشترین فاصله بین بردارهای پشتیبان هر کلاس می باشد. حال گاهی اوقات ممکن است فضای ویژگی به شکلی باشد که با نگاشت داده ها به فضای با ابعاد بالاتر، رسیدن به این هدف به بهترین شکل ممکن محقق شود این عمل با استفاده از توابع کرنل و صرفا با استفاده از توابع کرنل انجام شده و نیازی به نگاشت مختصات داده ها نیست.

- **کرنل خطی**

پایه ای ترین نوع کرنل محسوب می شود که معمولا زمانی بکار می رود که تعداد فیچر بالا داشته باشیم.
معمولا عملکرد آنها در دیتای متن از سایر کرنل ها بهتر است . تابع این کرنل به شکل زیر است :

```
F(x, xj) = sum( x.xj)
```

- **کرنل چند جمله ای**

این کرنل در پردازش تصویر پرکاربرد است. معادله آن به صورت زیر است:

که در آن d درجه چند جمله ای است .

$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j} + 1)^d$$

- **کرنل گاوسی**

این یک کرنل برای اهداف عمومی است. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود ندارد استفاده می شود. معادله آن

به صورت زیر است:

$$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$$

- **تابع پایه شعاعی گاوسی (RBF)**

این کرنلی برای اهداف عمومی کاربرد دارد. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود نداشته باشد، مورد استفاده قرار

می گیرد. معادله آن به صورت زیر است:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

و برای

$$\gamma > 0$$

گاهی اوقات با استفاده از پارامتر زیر استفاده می شود:

$$\gamma = 1/2\sigma^2$$

- **کرنل RBFلاپلاس**

این هم یک کرنل برای اهداف عمومی است. و هنگامی که هیچ دانش پیشینی در مورد داده ها وجود ندارد استفاده می شود. معادله

آن به صورت زیر است:

$$k(x, y) = \exp\left(-\frac{\|x - y\|}{\sigma}\right)$$

- **کرنل تانژانت هیپربولیک(tanh)**

می توانیم از آن در شبکه های عصبی استفاده کنیم. معادله مربوط به آن عبارت است از:

$$k(\mathbf{x_i}, \mathbf{x_j}) = \tanh(\kappa \mathbf{x_i} \cdot \mathbf{x_j} + c)$$

در برخی موارد (نه همیشه) $k > 0$ و $c < 0$.

- **کرنل سیگموئید**

می توان این کرنل را در شبکه های عصبی مورد استفاده قرار داد. معادله مربوط به آن عبارت است از:

$$k(x, y) = \tanh(\alpha x^T y + c)$$

- **کرنل تابع بِسِل (Bessel)از نوع اول**

ما می توانیم از آن برای حذف مقطع عرضی در توابع ریاضی استفاده کنیم. معادله آن عبارت است از:

$$k(x,y) = \frac{J_{v+1}(\sigma\|x-y\|)}{\|x-y\|^{-n(v+1)}}$$

که J تابع بسل از نوع اول است.

- **کرنل پایه شعاعی ANOVA**

ما می توانیم از آن در مسائل رگرسیون استفاده کنیم. معادله مربوط به آن عبارت است از:

$$k(x,y) = \sum_{k=1}^{n} \exp(-\sigma(x^k - y^k)^2)^d$$

- **کرنل splineخطی بصورت یک بعدی**

این کرنل، هنگام کار با بردارهای بزرگ داده پراکنده ، کاربرد زیادی دارد. این کرنل اغلب در دسته بندی متن مورد استفاده قرار می گیرد. کرنل spline همچنین در مسائل رگرسیون عملکرد خوبی دارد. معادله آن عبارت است از:

$$k(x,y) = 1 + xy + xy\ min(x,y) - \frac{x+y}{2}\ min(x,y)^2 + \frac{1}{3}\min(x,y)^3$$

در خصوص استفاده از آنها میتوان حکم کلی در یک موضوع کلی داد نه در یک موضوع جزیی بطور مثال می توان گفت در صورت عدم نبود دانش قبلی از داده کرنل آر بی اف می تواند گزینه خوبی باشد . و این ادعا ها برمبنای خواص موجود در توابع این کرنل ها می باشد.

۲. قبلا با دیتاست کلاس بندی قیمت موبایل در کگل کار کرده ایم . بر روی دیتاست ، روش SVM را اجرا کنید . (استفاده از پکیج ها همانند sklearn مجاز است)

برای اجرای SVM ابتدا داده تست و آموزش را جدا میکنیم و متغیر ها آموزش و برچسب Y را میسازیم .با استفاده از پکیج sklearn روش SVM را اجرا میکنیم .امتیاز 0.9533333333333334 بدست می آید به این معنی که در داده های تست با دقت 95 درصد برچسب ها به درستی تشخیص داده شده است .با استفاده از ماتریس زیر میتوان مشاهده کرد که داده های کدام دسته به درستی تشخیص داده شده و یا خطا داشته است:

```
X = traindf.drop(['price_range'], axis=1)
y = traindf['price_range']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=22, shuffle=True)
```

```
svmrbf = SVC()
svmrbf.fit(X_train, y_train)
svmrbf.score(X_test, y_test)
```

```
0.9533333333333334
```

```
y_pred = svmrbf.predict(X_test)
CM = confusion_matrix(y_test, y_pred)
print(CM)
```

```
[[141   3   0   0]
 [  7 138   0   0]
 [  0   7 143   8]
 [  0   0   3 150]]
```

در این روش ما از کرنل RBF استفده کردیم و حالا در مراحل بعدی نتیجه استفاده از کرنل های بعدی بررسی خواهد شد

۳. برای سوال ۲ حداقل ۵ حالت مختلف از قبیل کرنل ها و پارامترها را بررسی کنید و نتایج آن را گزارش دهید.

برای این کار کرنل های RBF ، linear ، poly ، sigmoid را با مقادیر مختلف برای پارامتر C و GAMMAبررسی کرده ایم.

در هر ستون از جدول زیر میتوانید با توجه به پارامتر ها و نوع کرنل امتیاز Svm روی داده های تست را مشاهده نمایید. ضریب کرنل برای rbf ، poly و sigmoid است.

هرچه مقدار گاما بیشتر باشد، الگوریتم تلاش م یکند برازش را دقیقاً بر اساس مجموعه دادههای تمرینی انجام دهد و این امر موجب تعمیم یافتن خطا و وقوع مشکل بیش برازش (Over-Fitting) می شو د.

```
In [11]: svmlinear = SVC(kernel = 'linear')
         svmlinear.fit(X_train, y_train)
         svmlinear.score(X_test, y_test)
Out[11]: 0.9666666666666667

In [12]: svmpoly = SVC(kernel = 'poly')
         svmpoly.fit(X_train, y_train)
         svmpoly.score(X_test, y_test)
Out[12]: 0.955

In [13]: svmsigmoid = SVC(kernel = 'sigmoid')
         svmsigmoid.fit(X_train, y_train)
         svmsigmoid.score(X_test, y_test)
Out[13]: 0.185

In [14]: svmrbf2 = SVC(C = 0.01)
         svmrbf2.fit(X_train, y_train)
         svmrbf2.score(X_test, y_test)
Out[14]: 0.5816666666666667

In [15]: svmrbf3 = SVC(C = 100)
         svmrbf3.fit(X_train, y_train)
         svmrbf3.score(X_test, y_test)
Out[15]: 0.9683333333333334

In [16]: svmlinear2 = SVC(kernel = 'linear',C = 0.01)
         svmlinear2.fit(X_train, y_train)
         svmlinear2.score(X_test, y_test)
Out[16]: 0.9683333333333334

In [17]: svmlinear3 = SVC(kernel = 'linear',C = 100)
         svmlinear3.fit(X_train, y_train)
         svmlinear3.score(X_test, y_test)
Out[17]: 0.9716666666666667
```

Gamma=scale	C=1	C=0.01	C=100
RBF	0.953	0.581	0.968
linear	0.966	0.968	0.971
poly	0.955	0.763	0.978
sigmoid	0.185	0.24	0.17

Gamma=0.000005	C=1
RBF	0.946
linear	0.966
poly	0.966
sigmoid	0.24

```
In [18]: svmpoly2 = SVC(kernel = 'poly',C = 0.01)
         svmpoly2.fit(X_train, y_train)
         svmpoly2.score(X_test, y_test)
Out[18]: 0.7633333333333333

In [19]: svmpoly3 = SVC(kernel = 'poly',C = 100)
         svmpoly3.fit(X_train, y_train)
         svmpoly3.score(X_test, y_test)
Out[19]: 0.9783333333333334

In [20]: svmsigmoid2 = SVC(kernel = 'sigmoid',C = 0.01)
         svmsigmoid2.fit(X_train, y_train)
         svmsigmoid2.score(X_test, y_test)
Out[20]: 0.24

In [21]: svmsigmoid3 = SVC(kernel = 'sigmoid',C = 100)
         svmsigmoid3.fit(X_train, y_train)
         svmsigmoid3.score(X_test, y_test)
Out[21]: 0.17

In [22]: svmrbf4 = SVC(gamma=0.000005)
         svmrbf4.fit(X_train, y_train)
         svmrbf4.score(X_test, y_test)
Out[22]: 0.9466666666666667

In [23]: svmlinear4 = SVC(kernel = 'linear',gamma=0.000005)
         svmlinear4.fit(X_train, y_train)
         svmlinear4.score(X_test, y_test)
Out[23]: 0.9666666666666667

In [24]: svmpoly4 = SVC(kernel = 'poly',gamma=0.000005)
         svmpoly4.fit(X_train, y_train)
         svmpoly4.score(X_test, y_test)
Out[24]: 0.9666666666666667
```

۴. برای سوال ۲ سعی کنید مبحث hard margin و soft margin را بررسی کنید و نتایج آن را گزارش دهید .

در واقع هایپرپارامتر C مقدار مجازات مدل برای هر نمونه ای را که اشتباه طبقه بندی میکند را تعیین می کند اگر حاشیه خیلی نرم باشد یعنی سختگیری ما نسبت به میزان خطا کمتر اشد، احتمال Over-Fitting بیشتر میشود اما اگر حاشیه کاملا سخت باشد ممکن است مدل نتواند مرزی تشخیص دهد. میتوانید نتیجه حاصل از تغییر پارامتر C را ببینید.

۵. مهندسی ویژگی یکی از بخش های مهم در فرایندهای علم داده میباشد . بر روی دیتاست موارد زیر را اجرا کنید.

قسمت الف)

یه حالت دسته بندی مختلف با انداره های 3 و 9 و 29 سا خته شد.

قسمت ب)

encoding hot on را برای فیچر های blue ، dual_sim ، touch_screen ، wifi و n_cores اعمال کرده ایم در بخش بعدی نتیجه svm با این تغییرات را بررسی میکنیم.

با اعمال one-hot encoding هر کدام از داده های کتگوریکال به یک بردار به طول n که هرکدام از داده های کتگوریکال از داده ها یک و بقییه صفر هست .

دلیل استفاده از این انکدینگ از آنجاییکه مدل های یادگیری ماشین صرفا با اعداد سرو کار دارند، باید داده هایی که به شکل کتگوریکال هستند را با استتفاده از یک تکنیک انکدینگ به عدد تبدیل کردن. او از آنجاییکه این انکدینگ نمایش قابل فهم تری از یک داده کتگوریکال ارایه می دهد جزو پرکاربردترین انواع انکدر ها می باشد.

```
In [28]: dfblue= pd.get_dummies(traindf.blue, prefix='blue')
         dfdual_sim = pd.get_dummies(traindf.dual_sim, prefix='dual_sim')
         dftouch_screen= pd.get_dummies(traindf.touch_screen, prefix='touch_screen')
         dfwifi = pd.get_dummies(traindf.wifi, prefix='wifi')
         dfn_cores = pd.get_dummies(traindf.n_cores, prefix='n_cores')

In [29]: df= pd.concat([dfblue, dfdual_sim,dftouch_screen,dfwifi,dfn_cores], axis=1)

In [30]: df2=pd.concat([traindf,df], axis=1)

In [31]: A=['blue','dual_sim','touch_screen','wifi','n_cores']

In [32]: df2.drop(A, axis = 1)
```

Out[32]:

	battery_power	clock_speed	fc	four_g	int_memory	m_dep	mobile_wt	pc	px_height	px_width	...	wifi_0	wifi_1	n_cores_1	n_cores_2	n_cores_3
0	842	2.2	1	0	7	0.6	188	2	20	756	...	0	1	0	1	0
1	1021	0.5	0	1	53	0.7	136	6	905	1988	...	1	0	0	0	1
2	563	0.5	2	1	41	0.9	145	6	1263	1716	...	1	0	0	0	0
3	615	2.5	0	0	10	0.8	131	9	1216	1786	...	1	0	0	0	0
4	1821	1.2	13	1	44	0.6	141	14	1208	1212	...	1	0	0	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1995	794	0.5	0	1	2	0.8	106	14	1222	1890	...	1	0	0	0	0
1996	1965	2.6	0	0	39	0.2	187	3	915	1965	...	0	1	0	0	0
1997	1911	0.9	1	1	36	0.7	108	3	868	1632	...	1	0	0	0	0
1998	1512	0.9	4	1	46	0.1	145	5	336	670	...	0	1	0	0	0
1999	510	2.0	5	1	45	0.9	168	16	483	754	...	0	1	0	0	0

قسمت ج)

گاهی اوقات داده های موجود از یکدیگر پرت هستند و برخی الگوریتم های یادگیری ماشین مثل ماشین بردار پشتیبان به پراکنده بودن داده ها حساس هستند. استفاده از تبدیلاتی مثل تبدیل لگاریتمی یا مین مکس اسکالر موجب نزدیک شدن سمپل ها به یکدیگر از نظر مقدار می شود این تبدیل پیش از طبقه بندی داده ها در سوال دو مورد استفاده قرار دادم و سوال ۶ هم مورد استفاده قراردادم

قسمت د)

فیچر مساحت ساخته شد اما باعث کاهش قدر ت تشخیص اگوریتم شد و به جای آن از روی متغیر طول و عرض تاچ ، قطر آن محاسبه گردید.

۶. برای هریک از حالت های سوال ۵ یک مدل SVM بسازید و بررسی کنید یکبار هم هر ۵ حالت را باهم اعمال کنید و مدل SVM روی آنها اجرا کنید . حاصل این مدل ها را گزارش کنید.

الف)در حالت با سه دسته تغییری در خروجی حاصل نشد و امتیاز الگوریتم 0.953 شد .با حذف فیچر اصلی power batteryامتیاز به 0.798 کاهش پیدا کرد .برای دو حالت دسته های 9 و29 تایی هم تغییری در خروجی حاصل نشد.

(ب)با اعمال encoding hot on روی ویژگی های ذکر شده، امتیاز الگورتم به 0.958 افزایش یافت.

(د)ساخت فچر مساحت باعث کاهش شدید امتیاز الگویتم شد

(ه)در این حالت تمام موارد را با هم اعمال میکنیم و با بهترین کرنل ، امتیاز الگوریتم 0.9716666666666667 بدست آمد.