Amirkabir University of Technology
(Tehran Polytechnic)

Department of
Computer Engineering

Course : Machine Learning

Homework 4

Najmeh Mohammadbagheri

99131009

بخش تشریحی

سوال اول

الف)بله. چون در این الگوریتم سعی می‌شود مقادیر متناظر با خط جداکننده پیدا شود. این خط نیز به توزیع داده‌ها بستگی دارد. بنابراین در الگوریتم SVM یادگیری بر توزیع داده‌هاست و پارامترهای خط به توزیع داده‌ها بستگی دارد. این توضیحات مربوط به SVM پایه است. درمورد SVMهای غیرخطی این مطلب صحیح نیست و غیرپارامتری هستند.

ب)بله. حاشیه‌ی هر مدل میزان عملکرد آن مدل در مقابل داده‌های تست را نشان می‌دهد. هرچه این حاشیه بیشتر باشد مدل بهتر است و کمتر دچار اشتباه در تصمیم‌گیری می‌شود.

ج)خیر. در SVM پایه که hard است بردارهای پشتیبان شدیدا به داده‌ها حساس است و دچار بیش برازش میشود ولی در نسخه‌ی soft آن میتوان با ضریب بتا این میزان بیش برازش شدن را کاهش داد.

د)خیر. داده‌های نویزی و پرت در تعیین حاشیه در حالت سخت(hard SVM) تاثیر دارد. ولی در حالت نرم(soft SVM) میتوان تاثیر این داده ها را با کم در نظر گرفتن میزان جریمه، کاهش داد.

ه)خیر. اگر این دسته‌بند ضعیف یک دسته‌بند تصادفی باشد در هر اجرا از داده‌های آموزش برچسب مختلف به داده‌ها میزند و اصلا رفتار قابل پیش‌بینی ای ندارد. ولی اگر یک دسته‌بند معقول و غیرتصادفی باشد این گزاره صحیح است.

و)خیر. اگر خطای یک دسته‌بند بیشتر از ۵۰ درصد باشد باید ضریب منفی به آن بدهیم که که نتیجه را معکوس کند تا عملکرد همچنان خوب باشد. اگر هم دسته بند تصادفی بود ضریب صفر به آن میدهیم که نتیجه اش اصلا درنظر گرفته نشود.

سوال دوم

هرچه سیگما کوچکتر باشد، گاما بزرگتر است و میدانیم که هر چه گاما بزرگتر باشد درجه غیرخطی بودن مرز تصمیم بیشتر میشود بنابراین سیگما در تصویر اول از سمت راست ۰.۲ و در تصویر وسط ۱۰ و در تصویر سمت چپ ۱ است.

سوال سوم

در روش hard تنها تعداد هر کلاس رای گیری میشود. یعنی اگر ۵ مورد کلاس A داشته باشیم و ۴ مورد کلاس B خروجی کلاس A است. اما در روش soft احتمال هر کلاس را داریم و از این احتمال ها میانگین گرفته میشود و در آخر کلاس با بیشترین احتمال به عنوان جواب اعلام میشود.توضیح با مثال:

Classifier 1 predicts class A with probability 99%

Classifier 2 predicts class A with probability 49%

Classifier 3 predicts class A with probability 49%

The average probability of belonging to class A across the classifiers is

(99 + 49 + 49) / 3 = 65.67%. Therefore, class A is the ensemble decision

سوال چهارم

روش hard :

Classifier 1 predicts class 2

Classifier 2 predicts class 1

Classifier 3 predicts class 1

➔ data belongs to class 1

روش Soft:

The average probability of belonging to class 1 across the classifiers is

(0.2 + 0.6 + 0.8) / 5 = 0.32

The average probability of belonging to class 2 across the classifiers is

(1 + 0.3+ 0.6) / 5 = 0.38

The average probability of belonging to class 3 across the classifiers is

(0.8 + 0.1+ 0.6) / 5 = 0.3

➔ data belongs to class 2

بخش پیاده‌سازی

سوال اول

قسمت اول :

بهترین پارامترها با رنگ سبز مشخص شده اند.

کرنل خطی :

C = 0	C = 3	C=10	C = 60
Accuracy : ۰.۸۷۷۵۵۱۰۲۰۴۰۸۱۶۳۲ Fscore : ۰.۹۲۸۵۷۱۴۲۸۵۷۱۴۲۸۶	Accuracy : ۰.۸۵۷۱۴۲۸۵۷۱۴۲۸۵۷۱ Fscore : ۰.۹۱۷۶۴۷۰۵۸۸۲۳۵۲۹۴	Accuracy : ۰.۸۷۷۵۵۱۰۲۰۴۰۸۱۶۳۲ Fscore : ۰.۹۱۸۹۱۸۹۱۸۹۱۸۹۱۹	Accuracy : ۰.۸۳۶۷۳۴۶۹۳۸۷۷۵۵۱۱ Fscore : ۰.۸۹۴۷۳۶۸۴۲۱۰۵۲۶۳۲

کرنل چندجمله‌ای:

Degree = 3 , c =1	Degree = 5 , c =5	Degree = 10 , c = 1	Degree = 50, c = 1
Accuracy : ۰.۸۹۷۹۵۹۱۸۳۶۷۳۴۶۹۴ Fscore: ۰.۹۴۲۵۲۸۷۳۵۶۳۲۱۸۳۹	Accuracy : ۰.۸۳۶۷۳۴۶۹۳۸۷۷۵۵۱۱ Fscore: ۰.۹	Accuracy : ۰.۸۱۶۳۲۶۵۳۰۶۱۲۲۴۴۹ Fscore: ۰.۸۸۶۰۷۵۹۴۹۳۶۷۰۸۸۶	Accuracy : ۰.۷۷۵۵۱۰۲۰۴۰۸۱۶۳۲۶ Fscore: ۰.۸۷۳۵۶۳۲۱۸۳۹۰۸۰۴۶

کرنل RBF:

Gamma = 1	Gamma =10	Gamma = 0.1	Gamma = 0.001
Accuracy : ۰.۶۹۳۸۷۷۵۵۱۰۲۰۴۰۸۲ Fscore : ۰.۸۱۹۲۷۷۱۰۸۴۳۳۷۳۴۹	Accuracy : ۰.۷۵۵۱۰۲۰۴۰۸۱۶۳۲۶۵ Fscore : ۰.۸۶۰۴۶۵۱۱۶۲۷۹۰۶۹۷	Accuracy : ۰.۸۵۷۱۴۲۸۵۷۱۴۲۸۵۷۱ Fscore : ۰.۹۲۱۳۴۸۳۱۴۶۰۶۷۴۱۶	Accuracy : ۰.۸۹۷۹۵۹۱۸۳۶۷۳۴۶۹۴ Fscore : ۰.۹۳۵۰۶۴۹۳۵۰۶۴۹۳۵

کرنل سیگموئید:

C= 1	C = 10	C = 50	C = 0.1
Accuracy : ۰.۷۳۴۶۹۳۸۷۷۵۵۱۰۲۰۴ Fscore : ۰.۸۴۷۰۵۸۸۲۳۵۲۹۴۱۱۸	Accuracy : ۰.۷۷۵۵۱۰۲۰۴۰۸۱۶۳۲۶ Fscore : ۰.۸۷۳۵۶۳۲۱۸۳۹۰۸۰۴۶	Accuracy : ۰.۷۹۵۹۱۸۳۶۷۳۴۶۹۳۸۸ Fscore : ۰.۸۸۶۳۶۳۶۳۶۳۶۳۶۳۶۴	Accuracy : ۰.۷۷۵۵۱۰۲۰۴۰۸۱۶۳۲۶ Fscore : ۰.۸۷۳۵۶۳۲۱۸۳۹۰۸۰۴۶

قسمت دوم:

کرنل خطی تنها یک پارامتر C دارد که با تغییر دادن آن تنها بایاس مدل بالا پایین میشود و تاثیری در شکل مدل ندارد.

کرنل چندجمله‌ای دو پارامتر درجه و C دارد که با افزایش پارامتر d؛یعنی درجه، میزان پیچیدگی مدل افزایش میابد و با زیادشدن هرچه بیشتر آن بیش برازش بیشتری صورت می‌گیرد. تاثیر C نیز مانند حالت قبل است.

کرنل RBF یک پارامتر گاما دارد که هرچه گاما بیشتر باشد پیچیدگی مدل بیشتر است و در نتیجه مدل بر داده‌ها منطبق میشود و بیش برازش رخ میدهد.

کرنل سیگوئید یک پارامتر C دارد که این پارامتر نیز با توجه به آزمایشها بیشتر بودنش بهتر است. ولی از حدی نباید بیشتر شود. چون عملکرد کاهش می یابد.

قسمت سوم :

یک روش ساده که به ذهنم میرسد، این است که در یک بازه‌ی معقول تعداد زیادی مقدار تولید کنیم و به ازای تمام این مقادیر مدل را بسازیم. سپس به ازای هر مدل مساحت زیر نمودار ROC را برایش محاسبه کنیم و این مقادیر را نگه داریم. در نهایت پارامتر متناظر با مدلی که بیشترین مساحت زیر سطح نمودار را داشت، به عنوان بهترین پارامتر انتخاب کنیم.(اگر داده‌ها توازن نداشتند میتوانیم به جای استفاده از نمودار ROC از نمودار precison_recall استفاده کنیم.)

همچنین میتوانیم برای دقت بالاتر، این محاسبات را بر روی داده‌های ارزیابی انجام دهیم.(استفاده از روش‌های cross-validation و k-fold)

سوال دوم

قسمت الف:

N_features = 8 N_estimaitors = 100 Max_depth = 4	N_features = 5 N_estimaitors = 40 Max_depth = 2	N_features = 3 N_estimaitors = 20 Max_depth = 3
accuracy 0.7987012987012987	accuracy 0.7727272727272727	Accuracy 0.7662337662337663

(من با مقادیر مختلف زیادی امتحان کردم، به ازای مقادیر مختلف بازه‌ی تغییرات بسیار کم بود.

قسمت ب:

N_estimaitors تعداد مدل‌هایی که یادگیری میشوند را بیان میکند. در نهایت نیز برای پیش‌بینی داده‌های تست از خروجی تمام این مدل ها میانگین گرفته میشود. در روش جنگل تصادفی باید میزان این پارامتر بالا تنظیم شود تا عملکرد نهایی این روش قابل قبول باشد.

N_features بیانگر تعداد ویژگی‌هایی است که برای هر مدل داده میشود. ایده‌ی جنگل تصادفی این است که علاوه بر اینکه هر مدل بر روی داده‌های مجزایی کار میکند، ویژگی‌های مختلفی نیز به هر مدل داده شود. با مشخص کردن این پارامتر در واقع میگوییم به هر مدل به صورت تصادفی چندتا از ویژگی‌ها داده شود. این کار باعث میشود که میزان تصادفی بودن مدل ها افزایش یافته و نتیجه ی نهایی بایاس نشده باشد.

Max_depth نیز بیانگر عمق درخت در هر مدل است. هدف از این روش جنگل تصادفی این بود که مدل ها همگی عمق خیلی کمی داشته باشند و سعی شود که از بیش برازش شدن جلوگیری شود. پس باید دقت کنیم که عمق درخت خیلی کم درنظر گرفته شود.

در کل ترکیبی از این سه پارامتر باید منطقی باشد که دقت بالا بدست اوریم. مثلا تعداد ویژگی ۳ و عمق درخت ۸ اصلا انتخابی منطقی نیست. چون با ۳ ویژگی نمیتوانیم که به عمق ۸ برسیم!

قسمت سوم:

1. Adaboost with n_estimators = 200

    accuracy Adaboost : 0.8311688311688312

2. SVM with RBF kernel , gamma=0.01

    accuracy SVM RBF : 0.8246753246753247

3. Bagging with base_estimator = LogisticRegression()

    accuracy Bagging : 0.8181818181818182