دانشگاه صنعتی امیرکبیر
دانشکده مهندسی کامپیوتر

گزارش تکلیف سوم درس یادگیری ماشین
آشنائی با دسته‌بند ماشین بردار پشتیبان (SVM)

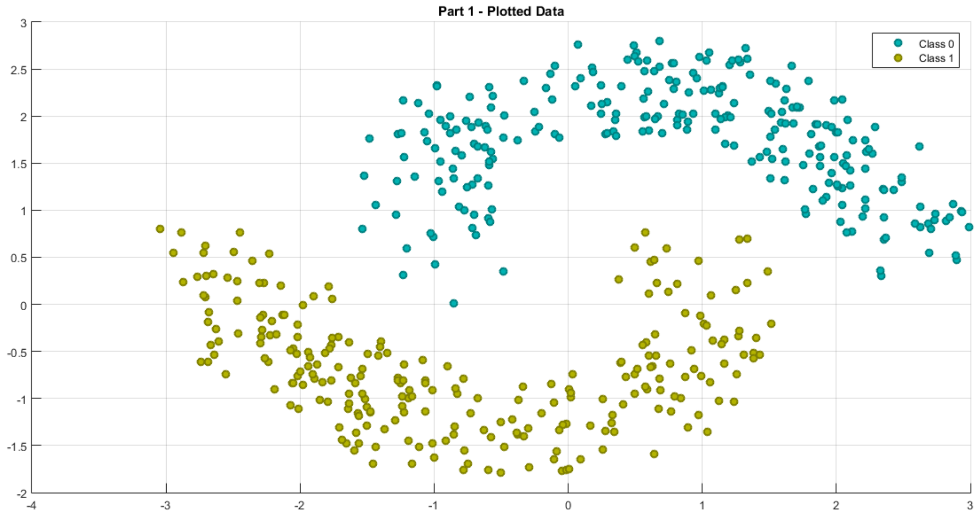
دانشجو:
سید احمد نقوی نوزاد
ش-د: ۹۴۱۳۱۰۶۰

استاد:
دکتر ناظرفرد

بهار ۹۵

نکته: تمامی کدهای اصلی پروژه جدای از زیرتوابع نوشته‌شده در فایل 'main.m' قرار گرفته‌اند.

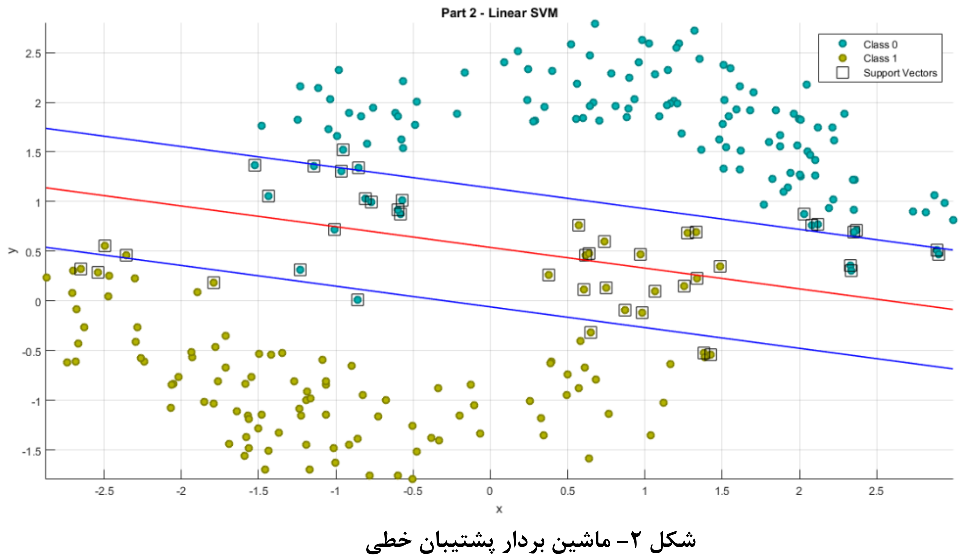
بخش اول:



شکل ۱- مجموعه داده‌ی رسم‌شده

مجموعه‌ی داده‌ی ورودی که شامل ۵۰۰ داده‌ی دوبعدی است و در دو کلاس تنظیم شده‌اند.

بخش دوم:



شکل ۲- ماشین بردار پشتیبان خطی

در این قسمت ابتدا ترتیب چینش مجموعه داده‌های ورودی را به هم می‌زنیم (به عبارت دیگر داده‌ها را بُر می‌زنیم)، و سپس از نصف آن‌ها برای آموزش یک ماشین بردار پشتیبان خطی (بدون کرنل) استفاده می‌نمائیم. برای رسیدن به فرمولِ مرز تصمیم‌گیری (decision boundary) از فرمول $\bar{w} = \sum_{i=1}^{n} \alpha_i y_i \bar{x}_i$ حاصله از مشتق‌گیری از معادله‌ی لاگرانژ استفاده می‌نمائیم که در آن ضرایب α_i برای داده‌های غیر بردار پشتیبان (support vectors) برابر صفر می‌باشد، و در نهایت از فرمول $x_2 + \frac{1}{w_2}(w_1 x_1 + bias) = 0$ برای رسم مرز تصمیم‌گیری استفاده می‌نمائیم؛ و برای رسم حاشیه‌ها نیز تنها کافیست که در فرمول آخر، مقدار $bias$ را با مقادیر $bias \pm \frac{1}{norm(w)}$ جایگزین نمائیم.

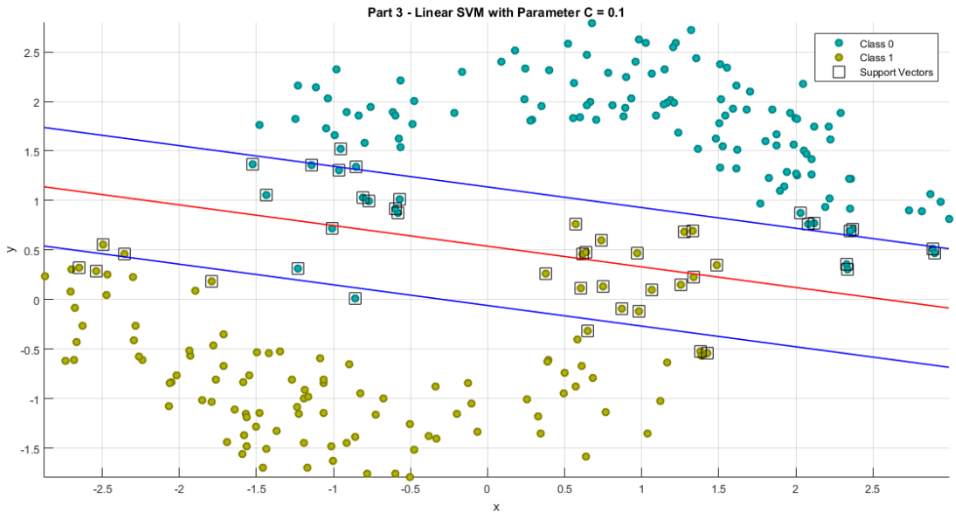
لازم به ذکر است که دقت دسته‌بند بر روی داده‌های آموزشی و داده‌های تست به شرح ذیل می‌باشد:

Linear SVM prediction accuracy	
Training Data	**95.60 %**
Test Data	**98.00 %**

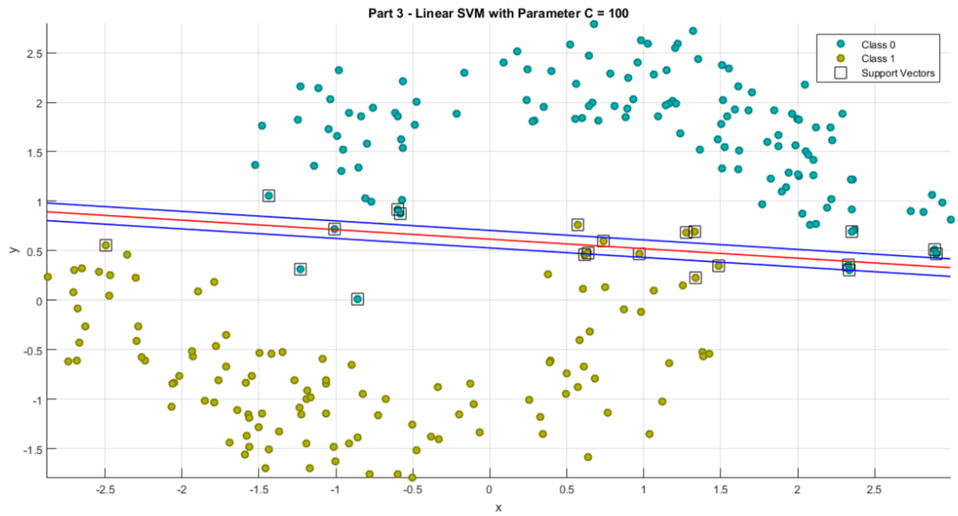
همانطور که مشاهده می‌کنیم، دقت دسته‌بندی برای داده‌های تست بیشتر می‌باشد، که علت آن این است که ما مقدار فعلی پارامتر C -همان BoxConstraint در کد موجود- (ضریب تنظیم‌سازی مجموع خطاهای داده‌های به اشتباه‌دسته‌بندی‌شده ($c\sum_{i=1}^{n}\varepsilon_i$) در معادله‌ی لاگرانژ، که میزان نقض حاشیه را نشان می‌دهد) را برابر 0.1 اختیار نمودیم (Soft Margin)؛ که در نتیجه با این کار به داده‌های آموزشی خود اجازه‌ی خطا را تا حد زیادی می‌دهیم، و به دنبال آن میزان خطا برای داده‌های تست می‌تواند کمتر باشد.

بخش سوم:

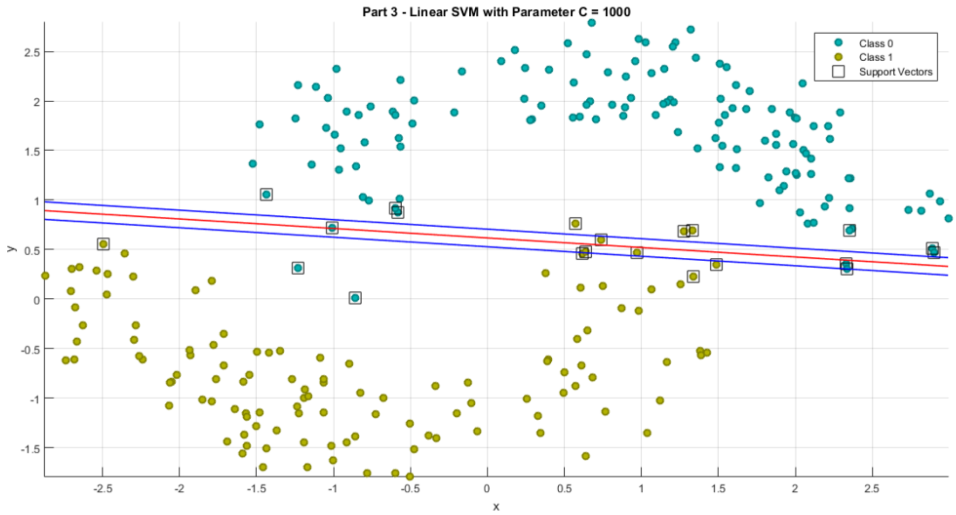
در این‌جا با توجه به پارامتر C در SVM خطی (پارامتر جریمه‌ی مربوط به داده‌های به اشتباه‌دسته‌بندی‌شده)، خطوط حاشیه (margins) و جداساز (decision boundary) را برای سه مقدار متفاوت C رسم می‌نمائیم. داریم:



شکل ۳- SVM خطی با مقدار پارامتر C=0.1



شکل ۴- SVM خطی با مقدار پارامتر C=100



شکل۵- SVM خطی با مقدار پارامتر C=1000

همانطور که در قسمت قبل نیز اشاره شد، پارامتر C میزان نقض حاشیه را نشان می‌دهد و در واقع یک ضریب تنظیم‌سازی (regularization term) در معادله‌ی بهینه‌سازی از نوع کمینه‌سازی –همان معادله‌ی لاگرانژ- برای مجموع خطاهای داده‌های به اشتباه‌دسته‌بندی می‌باشد. حال هرچه این مقدار به سمت صفر میل می‌کند، تمرکز کمینه‌سازی از مجموع خطاها منحرف گشته و به عبارتی ما اجازه‌ی خطا را تا حدی به داده‌های آموزشی می‌دهیم و در نتیجه عرض خیابان افزایش می‌یابد (Soft Margin)؛ و برعکس هرچه این مقدار به سمت بینهایت میل می‌کند، تمرکز کمینه‌سازی در معادله‌ی بهینه‌سازی به سمت مجموع خطاها متمایل گشته و به عبارتی اجازه‌ی خطا از داده‌های آموزشی سلب شده و در نتیجه‌ی آن عرض خیابان کاهش می‌یابد (Hard Margin). همان‌طور که از تصاویر بالا پیداست، به ازای مقادیر نسبتا بزرگ C ، عرض خیابان تقریبا به یک اندازه می‌باشد و علت این امر می‌تواند این باشد که رویه‌ی کمینه‌سازی تا تعداد تکرارهای محدودی ادامه دارد و نیز از آن‌جا که داده‌های آموزشی ما جداپذیر خطی نیستند، لذا خطوط حاشیه و جداساز لزوما بر یکدیگر منطبق نگشته و از یک مرحله به بعد با افزایش مقدار C نتایج یکسانی حاصل می‌گردد.

بخش چهارم:

در این‌جا عملیات بخش دوم را با استفاده از دو تابع کرنل Polynomial و RBF تکرار می‌نمائیم و نتایج دسته‌بندی بر روی داده‌های آموزشی و تست را به شرح ذیل گزارش می‌نمائیم:

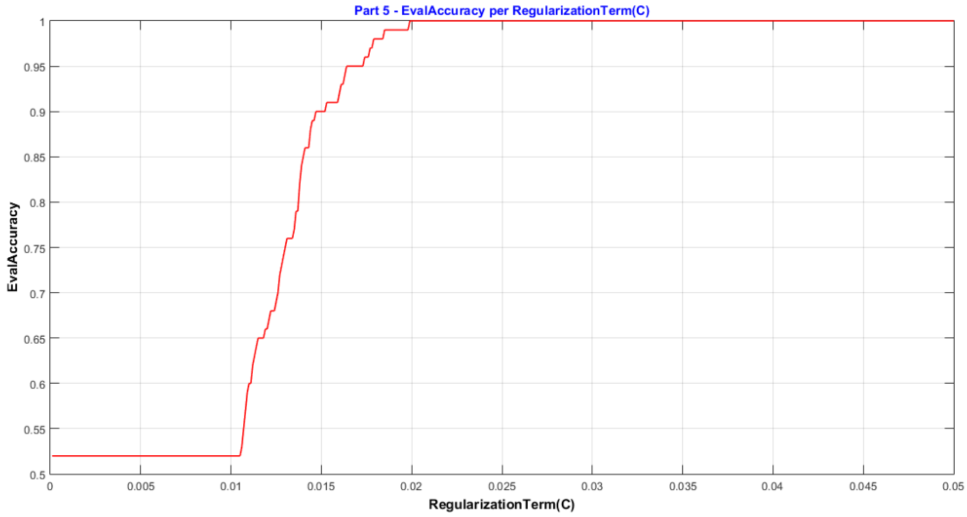
Kernel Function	Train Accuracy	Test Accuracy
Polynomial (order = 2)	98.00	95.60
Polynomial (order = 4)	100.00	100.00
Polynomial (order = 6)	100.00	100.00
Polynomial (order = 8)	100.00	99.60
Polynomial (order = 10)	99.60	98.40
RBF	100	100

نکته‌ی جالب توجه در مورد تابع کرنل Polynomial این است که با افزایش درجه‌ی چندجمله‌ای در ابتدا شاهد افزایش دقت دسته‌بندی در هر دوی مجموعه داده‌های آموزشی و تست می‌باشیم، اما با رسیدن به درجه‌ی ۱۰ شاهدیم که دقت برای هر دو مورد کاهش می‌یابد. افزایش دقت دسته‌بندی برای داده‌های آموزشی و کاهش آن برای داده‌های تست با افزایش درجه‌ی چندجمله‌ای، نشان از بیش‌برازش دسته‌بند بر روی داده‌های آموزشی دارد؛ اما انتظار نداشتیم که با افزایش درجه، شاهد کاهش دقت دسته‌بندی بر روی حداقل داده‌های آموزشی باشیم (مورد order=10)!!؟

در مورد تابع کرنل RBF (Radial Basis Function) نیز شاهدیم که دقت دسته‌بندی برای هر دو مجموعه داده‌های آموزشی و تست برابر ۱۰۰ می‌باشد که این نشان از عملکرد فوق‌العاده‌ی این تابع کرنل در بهبود دسته‌بندی دارد.

بخش پنجم:

در این قسمت نیز از همان ترتیب به‌هم‌ریخته‌ی داده‌های بخش دوم استفاده می‌کنیم و به جای تقسیم داده‌های ورودی به دو بخش آموزشی و تست به صورت ۵۰-۵۰، آن‌ها را به سه قسمت ۶۰ درصد آموزشی، ۲۰ درصد ارزیابی و ۲۰ درصد تست تقسیم می‌نمائیم. حال اگر منظور از یافتن پارامتر بهینه همان پارامتر جریمه‌ی مربوط به داده‌های به اشتباه‌دسته‌بندی‌شده می‌باشد، در یک حلقه‌ی **for** و با استفاده از تابع کرنل **RBF**، به ازای مقادیر افزایشی برای پارامتر **C** در هر بار تکرار حلقه، دقت دسته‌بندی بر روی داده‌های ارزیابی را محاسبه می‌نمائیم که نتایج آن به شکل زیر می‌باشد:



شکل ۶- نمودار دقت دسته‌بندی داده‌های ارزیابی بر مقادیر افزایشی پارامتر C

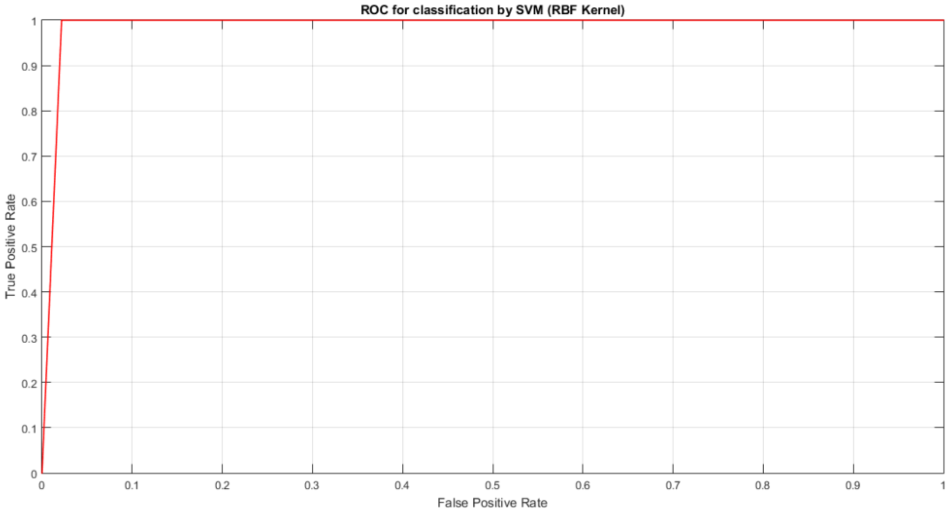
همانطور که مشاهده می‌کنیم، با افزایش مقدار پارامتر **C** در ابتدای کار، میزان دقت دسته‌بندی برای داده‌های ارزیابی ثابت و حدوداً برابر ۵۰ درصد می‌باشد، و این نشان از آن دارد که الگوریتم به صورت تصادفی در مورد داده‌های ارزیابی قضاوت می‌کند؛ اما از یک مرحله به بعد (**C=0.0105**) میزان دقت رو به فزونی گذاشته و این مسئله حاکی از آن است که تمرکز الگوریتم بر روی کمینه‌کردن خطای دسته‌بندی متوجه می‌شود؛ تا این‌که با رسیدن به یک مقدار مشخص برای پارامتر **C** (**C=0.0199**)، شاهدیم که دقت دسته‌بندی برای داده‌های ارزیابی به اوج خود یعنی ۱۰۰ درصد رسیده و بعد از آن با افزایش این پارامتر، دیگر میزان دقت تغییری ندارد. لذا بهترین مقدار برای پارامتر **C** همان حدّ آستانه‌ی ۱۰۰ درصدی می‌باشد، چرا که با اتخاذ مقدار بیشتر برای **C** ، تنها هزینه‌ی زمانی و محاسباتی افزایش یافته و بهبودی در دقت دسته‌بندی حاصل نمی‌گردد.

لازم به ذکر است که دقت دسته‌بندی برای داده‌های تست با استفاده از مقدار بهینه‌ی حاصله برای پارامتر **C** (با توجه به داده‌های ارزیابی)، برابر ۴۶ درصد و با استفاده از مقدار ماکسیمم **C** با توجه با ماکسیمم تکرار حلقه‌ی **for** ، برابر ۱۰۰ درصد می‌باشد؛ و این نشان از آن دارد که دقت دسته‌بندی برای داده‌های ارزیابی و داده‌های تست از یک رویه‌ی یکسان پیروی نکرده و به عبارتی ما در اینجا با دو مجموعه‌ی داده‌ی تست متفاوت کار می‌کنیم که برقراری توازن میان آن‌ها به سادگی ممکن نمی‌باشد.

بخش ششم:

در این بخش، با استفاده از مدل بهینه‌ی بخش قبل و مقدار بهینه‌ی حاصله برای پارمتر **C**، احتمال تعلق داده‌های تست به هر کدام از کلاس‌ها را با استفاده از تابع **svmpredict()** متعلق به کتابخانه‌ی **LIBSVM** ، اندازه می‌گیریم که مقادیر آن در یک ماتریس **m*2** با نام **p6ProbEst** ذخیره شده‌اند، که در آن **m** برابر تعداد داده‌های تست بوده و ۲ نیز بیانگر تعداد کلاس‌ها می‌باشد؛ و البته جمع احتمالات در هر سطر طبعاً برابر یک می‌باشد.

در نهایت با استفاده از تابع perfcurve() ، منحنی ROC دسته‌بند SVM (با استفاده از تابع کرنل RBF) با توجه به پارامترهای بهینه‌ی حاصله از بخش قبل را رسم می‌نمائیم که در شکل زیر قابل مشاهده می‌باشد:



شکل ۷- منحنی ROC مربوط به دسته‌بند SVM با استفاده از تابع کرنل RBF