بسمه تعالیٰ

تمرین ششم درس داده کاوی

نیم‌سال دوم تحصیلی ۰۰-۱۳۹۹

نام: امیرحسین کارگران خوزانی شماره دانشجویی: ۹۹۲۰۱۱۱۹

دانشکدهٔ مهندسی کامپیوتر
مدرس: دکتر فضلی

سؤال ۱.

حساب و کتاب

راه حل.

برای حل این سوال من از jupyter-notebook تنها به منظور ماشین حساب محاسبات ماتریسی استفاده کردم و عملا عملیات‌ها را به صورت دستی و مرحله به مرحله انجام داده‌ام که نام فایل نوتبوک question1 است که در پوشه question1 گذاشته شده است.

قسمت ۱ بخش آ: ابتدا ماتریس M که همان ماتریس مجاورت گراف است را تشکیل دادم و سپس به محاسبه $M.T$، $M \cdot M.T$، $M.T \cdot M$ پرداختم. سپس بردار h و a را برابر بردار عمودی تماما ۱ با اندازه تعداد راس‌ها که همان ۵ است انتخاب کردم. سپس با توجه به فرمول محاسبه a و h جدید را از ضرب ماتریس‌هایی که در مرحله قبل ساختیم با این بردارها بدست آوردم. تا ۳ راند این عملیات را انجام دادم و هر دفعه خروجی را تقسیم بر بزرگترین مقدار کردم. در راند ۲ و ۳ جواب‌ها یکسان شد که نشان دهنده همگرا شدن است.

قسمت ۱ بخش ب: در این بخش نیز پیج رنک را با استفاده از فرمول مطرح شده در اسلاید و تمرین قبل پیاده کردم و برای ۳ راند آن را دستی محاسبه کردم و برای راند ۴ ام تا راند ۵۲ ام با حلقه آن را محاسبه و چاپ کردم. در نهایت دوباره آن را نرمال کردم. مرحله ۵۲ نیز به این دلیل انتخاب شد که در این مرحله نرم تفاضل دو بردار از یک حد خیلی ریز کوچکتر می‌شد. در این مرحله من از انجا که نود آخر deadnode بود و یالی از آن بیرون نمی‌رفت از مالیات ۲۰ درصد استفاده کردم. در ادامه نیز trustrank را محاسبه کردم که برای این کار کافی است تنها نود D که تنها راس مورد اطمینان است برای بردار e مقدار ۰٫۴ که همان ۰٫۲ مالیات ضربدر $\frac{۱}{n=۵}$ که n تعداد رئوس و برابر ۵ است را داشته باشد و عملیات پیج رنک با این فرض ادامه داده شود. درنهایت این مقدار نیز پساز ۵۳ راند بدست آمد. حال با توجه به فرمول spam mass که به شکل زیر محاسبه می‌شود محاسبه spam mass نیز انجام شده است.

$$spammass = \frac{pagerank - trustrank}{pagerank}$$

قسمت ۲: گراف هدف را به صوت پارامتری تعریف کردم که می‌توان با تغییر d عمق ان را تغییر داد و گام به گام به محاسبه a و h همانند قسمت بخش آ تا مرحله ۴ پرداختم و سپس با یک حلقه برای مرحله ۹۹ جواب آن را نیز بدست اوردم. همانگونه که مشاهده می‌شود مقدار h و a به صورت زیر بدست می‌آید.

$$h = [۱, ۰, ۰, ..., ۰].T, a = [۱, ۱, ۱, ۰, ۰, ..., ۰].T$$

دلیل این امر آن است که اگر برای مرحله i ام بردار h و بردار a را نگاه کنید خواهیم داشت:

$$h_i = [۱, (\frac{۲}{۳})^{i-۱}, (\frac{۲}{۳})^{i-۱}, (\frac{۲}{۳})^{i-۱} \cdots, ۰, \cdots, ۰].T$$
$$a_i = [۱, ۱, ۱, (\frac{۲}{۳})^{i-۱}, (\frac{۲}{۳})^{i-۱}, (\frac{۲}{۳})^{i-۱} \cdots, (\frac{۲}{۳})^{i-۱}, \cdots, (\frac{۲}{۳})^{i-۱}].T$$

از آنجا که ضریب $\frac{۲}{۳}$ کوچکتر از ۱ است به مرور بعد از چندین گام به سمت ۰ میل می‌کند و تنها عددهای ۱ باقی می‌ماند. می‌توان بر اساس d که می‌تواند ۱ یا ۲ یا بزرگتر از ۲ باشد به ۳ دسته مختلف نوع گراف تبدیل کرد.

■

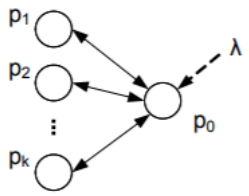
سؤال ۲.

توطئه

راه حل.

برای پاسخگویی به سوالات بخش اول و دوم از مقاله Link Spam Alliances نوشته Gyongyi و همکاران استفاده شده است. در این مقاله ساختار یک شبکه اسپم توضیح داده می‌شود و سپس به روش‌های اتصال آن‌ها با یکدیگر می‌پردازد.

بخش اول: در ساختاری که در شکل ۱ نشان داده شده است سعی شده است که امتیاز صفحه هدف $p_۰$ به حداکثر خودش برسد.



شکل ۱: ساختار بهینه یک شبکه اسپم برای یک صفحه هدف تنها

به منظور ایجاد چنین ساختار بهینه‌ای هیچ‌کدام از صفحات تسریع‌کننده به یکدیگر متصل نیستند و همگی به صفحه هدف متصل‌اند و صفحه هدف تنها به صفحات تسریع‌کننده لینک دارد. همچنین هر یک از صفحات ممکن است از صفحات خارجی دیگری نیز جریان نشتی داشته باشند و نیاز است که تمام نشتی‌ها تنها به صفحه هدف متصل باشد.

در این صورت امتیاز پیج‌رنک چنین ساختاری که در شکل صفحه هدف طبق قضیه ۱ این مقاله به صورت زیر محاسبه می‌شود.

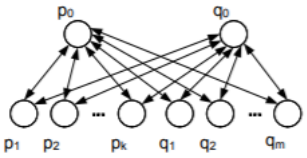
$$p_۰ = \frac{۱}{۱ - c^۲} \left[c\lambda + \frac{(۱ - c)(ck + ۱)}{N} \right]$$

در این رابطه N تعداد صفحات، c درصد مالیات، k تعداد تسریع‌گرها و λ جریان نشتی صفحه هدف است. از آنجا که عبارت $c\lambda$ در این عبارت ثابت است و به ساختار شبکه مرتبط نیست بنابراین در بهینه‌سازی‌های رکیب دو شبکه در ادامه آن‌ها در نظر نمی‌گیریم.

در این مقاله ۳ روش ترکیب شبکه‌های اسپم عنوان شده است. به این منظور نیاز است که دو شبکه اسپم P و Q را در نظر بگیرید. در این صورت اگر به ترتیب هر کدام k و m تسریع‌گر داشته باشند طبق فرمول قضیه ۱، امتیاز پیج‌رنک برای هر کدام برابر خواهد بود با:

$$\bar{p}_۰ = \frac{ck + ۱}{(۱ + c)N} \qquad \bar{q}_۰ = \frac{cm + ۱}{(۱ + c)N}.$$

روش اول: در این روش که بدیهی‌ترین روش است صفحات تسریع‌کننده به خدمت هر دو صفحه هدف در می‌آیند که در شکل ۲ نمایش داده شده است.



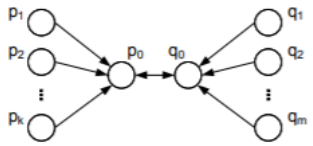
شکل ۲: اشتراک‌گذاری صفحات تسریع‌کننده دو شبکه اسپم

در این صورت طبق قضیه ۴ این مقاله امتیاز پیج‌رنک هر دو صفحه هدف برابر میانگین امتیاز قبل از این عملیات خواهد بود خواهیم داشت:

$$p_۰ = \frac{c(k+m)/۲ + ۱}{(۱ + c)N} = \left(\bar{p}_۰ + \bar{q}_۰\right)/۲ \ .$$

در این صورت هدفی که قبلا امتیاز کمتری داشته امتیاز بهتری کسب کرده و هدفی که قبلا امتیاز بهتری داشته اکنون امتیاز کمتری کسب می‌کند.

روش دوم: این روش در شکل ۳ نمایش داده شده است و تنها صفحات هدف به یکدیگر متصل می‌شوند و مجموع امتیاز صفحات نیز برابر جمع امتیازهای قبلی آن‌ها خواهد بود. خوبی این روش در آن است که تعداد لینک‌های اضافه شده تنها دو لینک می‌باشد.



شکل ۳: لینک بین دو صفحه هدف بدون لینک دخیل کردن تسریع‌کننده‌ها

روش سوم: این روش نیز همانند روش قبلی است اما از سمت صفحات هدف دیگر به صفحات تسریع کننده لینکی وجود ندارد. در این صورت امتیاز صفحات هدف به شرح زیر خواهد بود.

$$p_۰ = \frac{ck + c^۲m}{(۱ + c)N} + \frac{۱}{N}, \quad q_۰ = \frac{cm + c^۲k}{(۱ + c)N} + \frac{۱}{N}$$

که این امتیاز نسبت به حالت قبل از ایجاد این لینک برای هر دو صفحه بیشتر است و تفاضل آن متناسب با تعداد تسریع‌کنندگان است، همچنین مدیریت لینک بهتری نیز انجام شده است، بنابراین روش بهتری است.

بخش دوم: مجموع امتیازهای پیج رنک در هر ۳ حالت بیان شده برابر $\frac{k+m+۲}{n}$ است و به طور کلی ترکیب شبکه‌های اسپم

با یکدیگر مجموع امتیاز پیج‌رنک راس‌های شبکه‌های اسپم را تغییر نمی‌دهد. در روش اول امتیاز دو صفحه هدف میانگین حالت اولیه آن‌ها شد. روش دوم تنها این مزیت را داشت که به جای $k+m$ لینک جدید تنها ۲ لینک اضافه شد. در روش سوم نیز امتیازها نسبت به حالت اولیه افزایش یافت اما این موضوع به خاطر کاهش امتیاز تسریع‌گرها بود که دیگر لینکی از صفحات هدف به آن‌ها وجود نداشت.

بخش سوم: برای پاسخگویی روش اول بخش سوم از مقاله Link Spam Detection Based on Mass Estimation نوشته Gyongyi و همکاران استفاده شده است.

در روش spam mass الگوریتم پیج‌رنک را یک بار اجرا می‌کنیم و یک بار هم الگوریتم Trustrank که همان الگوریتم پیج‌رنک به همراه یک teleport-set است را اجرا می‌کنیم. راس‌های درون این مجموعه راس‌هایی هستند که ما مطمئن هستیم که اسپم نیستند. در این صورت با محاسبه spam mass می‌تواند به میزان اسپم بودن رئوس پی برد.

درست بودن این روش تمام به set بستگی دارد و اگر صفحاتی غیر اسپم از این مجموعه فاصله زیادی داشته باشند ممکن است به عنوان spam شناخته شوند و false positive case رخ دهد. برای محاسبه spam mass از فرمول زیر استفاده می‌شود.

$$spammass = \frac{pagerank - trustrank}{pagerank}$$

برای پاسخگویی به روش دوم بخش سوم از مقاله Spam, damn spam, and statistics نوشته Fetterly و همکاران استفاده شده است. در این روش توزیع درجات ورودی و خروجی صفحات وب مورد ارزیابی قرار گرفت و با استفاده از برررسی توزیع شبکه‌های اسپم و خاص بودن رفتار آن‌ها مقدار زیادی از آن‌ها را شناسایی کردند، این روش چندان مقیاس پذیر نیست و ممکن است خطا نیز داشته باشد اما تعداد خوبی از آن‌ها را می‌تواند در ابعاد بالا شناسایی کند. ■

سؤال ۳.

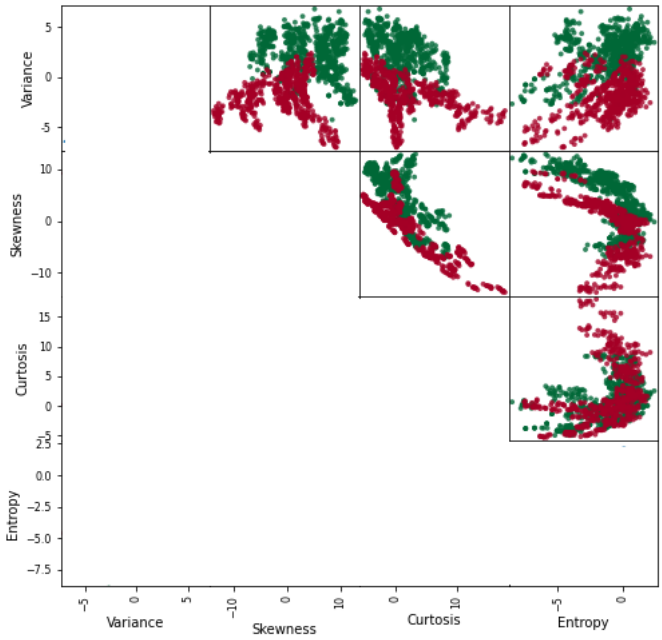
دستگاه تشخیص اصالت اسکناس

راه حل.

نام فایل نوتبوک question3 است که در پوشه question3 گذاشته شده است.

بخش اول: ۶ شکل گفته شده در شکل ۴ کشیده شده است. همانگونه که از این شکل بر می‌اید در شکل Curtosis و Variance دو داده اسکناس واقعی و جعلی نسبت به یکدیگر تا حد خوبی قابل تشخیص هستند. همچنین شکل Variance و Skewness نیز به خوبی این دو را از یکدیگر تمیز داده است. در شکل Variance و Entropy به خوبی دو شکل قبل نیست اما باز هم داده ها از یکدیگر خوب تمیز داده شده است. فصل مشترک این ۳ شکل variance است که نشان می‌دهد vaiance معیار مهمی برای تشخیص است. بدترین شکل برای تشخیص نیز Entropy و Curtosis است که داده ها با یکدیگر کاملا در امیخته شده اند. دو شکل باقی مانده دیگر نیز که Skewness Entropy و Curtosis Skewness هستند نیز به طور متوسط می‌توانند داده‌ها از یکدیگر تمیز دهند.

بخش دوم: با توجه به عملکرد دسته بند درخت می‌توان گفت که هر ۴ مقدار ،recall precision و f-score و accuracy نزدیک مقدار ۰/۹۸ برای هر دو کلاس را بدست آورده است که مقدار قابل توجهی است و ماتریس confusion نیز نشان می‌دهد که تنها از هر کلاس ۲ برچسب اشتباه خورده است.



شکل ۴: انتخاب دو دو فیچرها بر حسب هم دیتا داده شده

بخش سوم: با توجه به عملکرد دسته بند svm می‌توان گفت که هر ۴ مقدار، recall precision و f-score و accuracy نزدیک مقدار ۰/۹۸ برای هر دو کلاس را بدست آورده است که مقدار قابل توجهی است و ماتریس confusion نیز نشان می‌دهد که تنها از هر کلاس ۲ برچسب اشتباه خورده است.

بخش چهارم: من چندین بار با split های مختلف داده (چون رندم است) دادگان را مورد آموزش قرار دادم و آن‌ها را تست کردم. می‌توانم بگویم در هر یک از دفعات هر مدل ممکن بود کمی به اندازه ۱ یا ۲ مورد را نسبت به دیگری بهتر تشخیص دهند و این نشان می‌دهد که هر دو مدل‌ها به اندازه کافی خوب هستند و به یکدیگر در حل این مساله ترجیحی ندارند ولی اگر بخواهم تنها یکی از آن دو را انتخاب کنم مدل tree را انتخاب میکنم که پیچیدگی کمتری از svm دارد و در آموزش و تست نیز با سرعت بیشتری عمل می‌کند.

همچنین برای این که هر مدل از تمامی توانایی‌های خود برای حل این مساله استفاده کنند و پارامترهای پیش‌فرض ان‌ها سدی در نشان دادن خود نشوند، من سعی کردم که ضرایب مختلف را برای tree و svm تنظیم کنم که البته ضرایب svm تنها برای regulariztion هست. بهترین این ضرایب با استفاده از روش grdisearch که خود از k-fold-cross-validation استفاده می‌کند انتخاب می‌شود و این پارامترها بر روی دادگان تست tune نشده اند و با همان مکانیزم تقسیم دادگان آموزش به k داده validation انتخاب شده است و در نهایت با تمامی دادگان اموزش و بهترین آن ضریب آموزش دیده شده است.

■

سؤال ۴.

کرنل

راه حل.

نام فایل نوتبوک question4 است که در پوشه question4 گذاشته شده است.

بخش پنجم و بخش ششم: بر خلاف آنچه که صورت سوال گفته دادگان iris خطی تفکیک پذیر هستند و به عنوان مثالی از تحلیل‌های linear discriminant آورده می‌شود. با توجه به عملکرد دسته بند svm توسط کرنل های خطی و rbf و poly

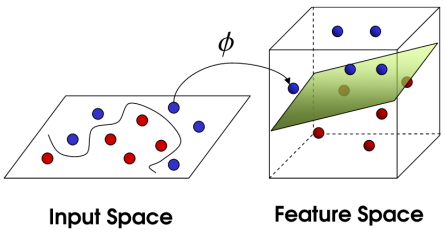
می‌توان گفت که هر ۴ مقدار recall precision، و f-score و accuracy مقدار ۱ را برای هر دو کلاس را بدست آورده است که همه دادگان تست را به درستی پیش‌بینی کرده‌اند و ماتریس confusion نیز این مورد را تصدیق می‌کند.

اما برای کرنل sigmoid مقدار معیارها نزدیک ۰٫۹۷ برای دسته‌بندی‌های مختلف (به غیر از یک کلاس) است که نشان می‌دهد یا همچنان پارامترها توسط من به خوب انتخاب نشده یا در این split از داده‌ها به خوبی بقیه این کرنل عمل نمی‌کند.

برای این که هر کرنل از تمامی توانایی‌های خود برای حل این مساله استفاده کنند و پارامترهای پیش‌فرض انها سدی در نشان دادن خود نشوند، من سعی کردم که پارامترهای مختلف را برای آن‌ها تنظیم کنم. که البته بیشتر پارامترها ضرایب برایregulariztion جلوگیری از overfitting است. بهترین این ضرایب با استفاده از روش grdisearch که خود از k- fold-cross-validation استفاده می‌کند انتخاب می‌شود و این پارامترها بر روی دادگان تست tune نشده اند و با همان مکانیزم تقسیم دادگان آموزش به k داده validation انتخاب شده است و در نهایت با تمامی دادگان اموزش و بهترین آن ضریب آموزش دیده شده است.

ادامه بخش ششم: از انجا که ۳ تا از کرنل‌ها به بهترین دقت رسیده‌اند باید از منظر پیچیدگی آن‌ها را بررسی کرد. پیچیدگی کمتر باعث تعمیم‌پذیری بیشتر و کم شدن over-fitting می‌شود. می‌توان گفت بهترین کرنل همان کرنل خطی است که پیچیدگی ان قطعا نسبت به کرنلی مانند poly با درجه‌ای بزرگتر از ۱ کمتر است و همچنین نسبت به کرنل rbf که یک کرنل گوسی است نیز دوباره کمتر است و پیچیدگی محاسباتی کمتری نیز دارد.

بخش هفتم: کرنل: تکنیکی است که در واقع بر روی بیشتر مدل ها قابل استفاده است، در این تکنیک دادگان به فضای دیگری برده می‌شوند و بر روی آن فضا مساله طبقه‌بندی انجام شده و سپس به فضای قبلی نگاشت می‌شوند. برای مثال در شکل ۵ برای جدا کردن توپ‌های آبی و قرمز باید از یک طبقه‌بند غیرخطی استفاده شود که ممکن است در صورت پیچیده بودن ان نیز overr-fit شود. اما با استفاده از کرنل Φ دادگان به فضای فیچری برده شده اند که در آن فضا با یک طبقه‌بند ساده خطی می‌توان آن‌ها را طبقه‌بندی کرد.



شکل ۵: تابع Φ نگاشت از یک فضا به فضای دیگر را انجام می‌دهد. شکل از link

توابع مختلفی می‌توانند کرنل باشند، به طور کلی هر تابعی که شرط Mercer را بتونه بگذراند می‌تواند کرنل باشد که از جمله آن‌ها می‌توان به این موارد اشاره کرد.

- کرنل خطی : $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- کرنل poly : $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
- کرنل rbf :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$

- کرنل sigmoid :

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1\right)$$

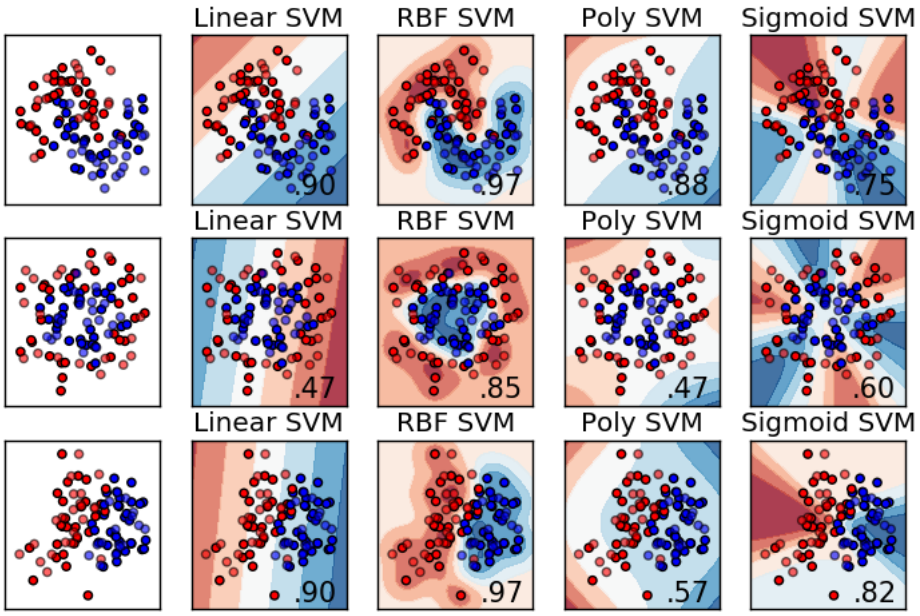
همانگونه که از نام و فرمول محاسبه این کرنل‌ها بر می‌آید کرنل خطی برای دادگان خطی مناسب است و اگر به صورت خطی تفکیک‌پذیر باشند این کرنل به خوبی همانگونه در این سوال هم مشاهده کردیم کار خواهد داد. همانگونه که فرمول آن را مشاهده می‌کنید فرمول بسیار ساده‌ای دارد و سریعتر محاسبه شده و generlization بیشتری را فراهم می‌کند.

کرنل بعدی poly که است که از اسم و فرمول آن مشخص است که همان کرنل چندجمله ای می‌باشد هرچه مقدار p بزرگتر باشد این کرنل پیچیده‌تر است و می‌تواند نواحی غیرخطی پیچیده‌تری را به کمک یک طبقه بند، دسته‌بندی کند. کرنل مناسبی است اگر چندان از p ای بزرگ استفاده نشود و برای داده‌های غیرخطی مناسب است.

کرنل بعدی rbf است که به کرنل گوسی نیز معروف است و تابع آن همان تابع گوسی است. یکی از پیچیده‌ترین و قوی‌ترین کرنل ها می‌باشد که با پارامتر σ می‌توان آن مقدار narrow و wide بودن آن را کنترل کرد. اگر کرنل بسیار narrow باشد (واریانس کم)، بعضی از مناطق خطی را شامل نمی‌شود و نقطه‌هایی را که confidence را در fit بالا می‌برد نادیده می‌گیرد. اگر کرنل بسیار wide باشد (واریانس زیاد) آنگاه شامل نقطه‌هایی می‌شود که fit را کاهش می‌دهد که ممکن است این کرنل شامل مناظق غیرخطی شود. بدلیل قوی بودن این کرنل در مساله‌های غیرخطی بسیار از این کرنل استفاده می‌شود.

کرنل بعدی نیز sgimoid است که بدلیل مشتق‌پذیر بودن tanh در شبکه‌های عصبی نیز زیاد از آن استفاده می‌شود.

تفاوت هر ۴ کرنل را در ۴ نوع داده کشیده شده در شکل ۶ مشاهده می‌کنید. برای مسائل خطی بهترین کرنل خطی و poly، برای مسائل نیمه خطی و غیر خطی کرنل poly و rbf و از کرنل sigmoid هم در مسائل خاص و شبکه‌های عصبی استفاده می‌شود. عددهای زیر هر شکل نشان دهنده دقت این کرنل‌ها در دسته‌بندی هر یک از داده‌هاست.



شکل ۶: مقایسه کرنل‌های مختلف برای ۳ داده متفاوت شکل از link

■