

Milad Rostamian

April 27, 2014

بهبود عملکرد طبقه بندی ماشین بردار پشتیبان با بهینه سازی ژنتیکی شکل هسته و پارامترها

چکیده

ماشین های بردار پشتیبان (SVM ها) حالتی از هنر عملکرد در برنامه های کاربردی در جهان واقعی را ارائه کردند و در حال حاضر به عنوان یکی از ابزارهای استاندارد برای یادگیری ماشین و داده کاوی بکار برده می شود. مشکل اصلی این روش چگونگی انتخاب هسته بهینه و چگونگی بهینه سازی پارامترهای آن است. برنامه های کاربردی دنیای واقعی نیز به در نظر گرفتن ترکیبی از هسته های چندگانه به منظور افزایش دقت طبقه بندی با تغییراتی اندک در هسته تاکید کرده اند. این ترکیب می تواند خطی یا غیر خطی، وزن دار یا بدون وزن باشد. در حال حاضر روش های متعددی برای پیدا کردن یک ترکیب هسته وزن دار خطی و بهینه سازی پارامترهای آن همراه با پارامترهای SVM پیشنهاد شده اند، اما هیچ روشی برای بهینه سازی ترکیب غیر خطی وزن دار پیشنهاد نشده است. بنابراین، هدف ما تولید خودکار و انطباق ترکیب هسته (خطی و غیر خطی، وزن دار و یا بدون وزن، با توجه به داده) به منظور بهینه سازی هر دو پارامتر هسته و پارامترهای SVM با استفاده از الگوریتم تکاملی در یک چارچوب واحد و یکپارچه است.

کلمات کلیدی: مشکلات طبقه بندی · هسته · هسته های متعدد · SVM · بهینه سازی بیش از حد پارامترهای ماشین های بردار پشتیبان · مدل ترکیبی · برنامه نویسی ژنتیک · MK.

۱ مقدمه

به عنوان یک شاخه گسترده ای از هوش مصنوعی، یادگیری ماشین در رابطه با طراحی و توسعه الگوریتم و تکنیک به رایانه اجازه می دهد که "یاد" بگیرد. هربرت سایمون بسیار ساده، اما فصیح یک تعریف ارائه کرده است: "آموزش فرایندی است که در آن یک سیستم عملکردش از طریق تجربه بهبود می یابد" [49]. به طور کلی مشکل یادگیری ماشین برای جستجو، معمولاً فضای بسیار بزرگ فرضیه های بالقوه برای تعیین یکی از بهترین وجه های مناسب داده و هر گونه دانش قبلی است. امروزه بسیاری از الگوریتم های یادگیری وجود دارند که عملکرد آنها (تخمین زده شده با اقدامات مختلف، دقت و صحت به عنوان مثال طبقه بندی، صحت راه حل، کیفیت راه حل و یا سرعت عملکرد) نه تنها به مساله حل شونده مربوط است، بلکه به پارامترهای موجود نیز وابسته است. بنابراین، بهترین نتایج را می توان تنها با شناسایی ارزش های مطلوب از این پارامترها به دست آورد. اگر چه این کار بسیار پیچیده است، روش های بهینه سازی های مختلف به منظور بهینه سازی پارامترهای الگوریتم های یادگیری ماشین توسعه یافته اند.

در این مقاله، محاسبات تکاملی از لحاظ نظری و تجربی ثابت شده است که برای جستجوی راه حل در فضاها پیچیده و به طور گسترده در بهینه سازی، آموزش شبکه های عصبی، برآورد پارامترها در

شناسایی سیستم و یا برنامه های کاربردی کنترل تطبیقی [11، 38] قوی می باشد. الگوریتم های تکاملی تشکیل زیر مجموعه ای از محاسبات تکاملی می دهند که در آنها به طور کلی فقط تکنیک های پیاده سازی مکانیزم های الهام گرفته از تکامل بیولوژیکی مانند تولید مثل، جهش، نوترکیبی، انتخاب طبیعی و بقای اصلح می باشد. راه حل های نامزد برای مسئله بهینه سازی نقش افراد در جامعه را بازی می کنند، و تابع هزینه محیطی را که در آن راه حل "وجود دارد" تعیین می کند. تکامل از جمعیت پس از آن صورت می گیرد که از اپراتورها استفاده مکرر کنید.

در سال 1995، ماشین بردار پشتیبان ها (SVM ها) در یک دوره جدید الگوی یادگیری از مثال مشخص آغاز شدند. ریشه نظریه یادگیری آماری و اصل کاهش ریسک عملیاتی توسط ولادیمیر وپنیک در AT & T در سال 1963 [56] توسعه یافته، SVM ها به سرعت توجه را از جامعه یادگیری ماشین با توجه به تعدادی از شایستگی های نظری و محاسباتی به دست آورد. ایده اصلی استفاده از جداکننده ی خطی برای طبقه بندی مجموعه ای از اقلام در فضای غیر خطی است. در سال 1995 کورتس و وپنیک [13] ایده حداکثر حاشیه اصلاح شده که برای سیستم های اطلاعاتی نمونه هایی با برچسب پیشنهاد می دهد را ارایه کردند. BOSE و همکاران (1992) [5] کرنل های مربوط به تابع SVM برای داده های غیر خطی ساختند. توابع هسته بردار ورودی را به فضایی با ابعاد بالا انتقال می دهند، احتمالاً به ابعاد بی نهایت، که در آن جدایی خطی بین اقلام بیشتر احتمال دارد. از این رو، پیچیدگی از مرز به دست آمده بستگی به طبیعت و خواص هسته استفاده شده دارد.

در [17، 39، 45، 50] نشان داده شد که برای مسائل طبقه بندی پیچیده، MK عملکرد طبقه بندی کننده SVM را با توجه به ویژگی های داده ها بهبود می بخشد. در این زمینه، چند سوال مربوط به MK بوجود می آید:

- خطی و غیر خطی: کدام ترکیب کارآمدترین هسته است؟
- یک ترکیب وزنی یا غیر وزن هسته لازم است؟
- کدام هسته ها باید برای ترکیب کارآمدتر در نظر گرفته شود: هسته های مختلف کلاسیک و / یا نمونه های چند هسته مشابه، اما با پارامترهای مختلف؟
- چگونگی بهینه سازی برای پارامترهای الگوریتم SVM بر اساس MK؟

مقاله ما، از طریق مدرن ترین روش ها و راه حل پیشنهادی، سعی می کند تا به این پرسش ها پاسخ دهد. بنابراین، از چارچوب تکاملی به منظور انطباق با بیان ترکیب هسته و پارامترهای مربوطه برای چند مسائل طبقه بندی استفاده کنیم.

الگوریتم های تکاملی یک کلاس از الگوریتم های جستجوی احتمالاتی می باشند که شبیه سازی فرآیند تکاملی طبیعی هستند. در این مقاله ما یک چارچوب برای طراحی یک هسته تکاملی پیشنهاد می دهیم که می تواند خطی یا غیر خطی، ترکیبی وزن دار و یا بدون وزن از هسته باشد (بسته به نوع و اقتباس از / به مساله که باید حل شود). ما ترکیب هسته را به عنوان یک هسته (KOK) تعیین خواهیم کرد. بهترین (اقتباس) KOK توسط خود الگوریتم با استفاده از داده های یک مسئله خاص بدست می آید. توجه داشته باشید که مدل ما یکی از اولین روش ها می باشد که به هر دو بیان MK غیر خطی و بهینه سازی ضرایب وزنی برای یک KOK می پردازد. برای این هدف برنامه نویسی ژنتیک (GP) روش [37] که با یک الگوریتم

[56 ، 48] SVM در یک مدل دو سطح هیبریدی ترکیب می شود ، چون به نظر می رسد روش هیبریدی به بهبود عملکرد حل مسائل طبقه بندی کمک می کند [58] . مدل مورد نظر به طور همزمان به دو مشکل می پردازد : پیدا کردن بیان کارآمد ترین تابع KOK و برای بهینه سازی پارامترهای SVM. این دو هدف به طور همزمان به دست می آیند چرا که هر کروموزوم GP شامل هر دو KOK و پارامترهای آن می باشد.

۲ ماشین بردار پشتیبان

یکی از روش های یادگیری بانظارت [۱] است که از آن برای طبقه بندی [۲] و رگرسیون [۳] استفاده می کنند.

این روش از جمله روش های نسبتاً جدیدی است که در سال های اخیر کارایی خوبی نسبت به روش های قدیمی تر برای طبقه بندی از جمله شبکه های عصبی پرسپترون نشان داده است. مبنای کاری دسته بندی کننده SVM دسته بندی خطی داده ها است و در تقسیم خطی داده ها سعی می کنیم خطی را انتخاب کنیم که حاشیه اطمینان بیشتری داشته باشد. حل معادله پیدا کردن خط بهینه برای داده ها به وسیله روش های QP که روش های شناخته شده ای در حل مسائل محدودیت دار هستند صورت می گیرد. قبل از تقسیم خطی برای اینکه ماشین بتواند داده های با پیچیدگی بالا را دسته بندی کند داده ها را به وسیله تابع phi به فضای با ابعاد خیلی بالاتر [۴] می بریم. برای اینکه بتوانیم مساله ابعاد خیلی بالا را با استفاده از این روش ها حل کنیم از قضیه دوگانی لاگرانژ [۵] برای تبدیل مساله مینیم سازی مورد نظر به فرم دوگانی آن که در آن به جای تابع پیچیده phi که ما را به فضایی با ابعاد بالا می برد، تابع ساده تری به نام تابع هسته که ضرب برداری تابع phi است ظاهر می شود استفاده می کنیم. از توابع هسته مختلفی از جمله هسته های نمایی، چندجمله ای و سیگموید می توان استفاده نمود. یکی از معروفترین خودآموزها مربوط به [۶] است.

مجموعه داده آموزشی D شامل n عضو را در اختیار داریم که به صورت زیر تعریف می شود:

$$D = \{(x_i, y_i) \mid x_i \in \mathbb{R}^p, y_i \in \{-1, 1\}\}_{i=1}^n$$

جایی که مقدار Y برابر ۱ یا -۱ و هر x_i یک بردار حقیقی p -بعدی است. هدف پیدا کردن ابر صفحه جداکننده با بیشترین فاصله از نقاط حاشیه ای است که نقاط با $y_i = 1$ را از نقاط با $y_i = -1$ جدا کند. هر ابر صفحه می تواند به صورت مجموعه ای از نقاط X که شرط زیر را ارضا می کند نوشت:

$$W \cdot X - b = 0 \text{ جایی که } W \text{ بردار نرمال است، که به ابر صفحه عمود است. ما}$$

می خواهیم W و b را طوری انتخاب کنیم که بیشترین فاصله بین ابر صفحه های موازی که داده ها را از هم جدا می کنند، ایجاد شود. این ابرصفحه ها با استفاده از رابطه زیر توصیف می شوند :

$$W \cdot X - b = 1 \text{ و } W \cdot X - b = -1$$

اگر داده های آموزشی جدایی پذیر خطی باشند، ما می توانیم دو ابر صفحه در حاشیه نقاط به طوری که هیچ نقطه مشترکی نداشته باشند، در نظر بگیریم و سپس سعی کنیم، فاصله آنها را، ماکسیم کنیم. با استفاده از هندسه، فاصله این دو صفحه $\frac{2}{\|w\|}$ است. بنابراین ما باید $\|W\|$ را مینیم کنیم. برای

اینکه از ورود نقاط به حاشیه جلوگیری کنیم، شرایط زیر را اضافه می کنیم: برای هر i

$$w \cdot x_i - b \geq 1 \quad \text{for } x_i \text{ of the first class}$$

یا

$$\mathbf{w} \cdot \mathbf{x}_i - b \leq -1 \quad \text{for } \mathbf{x}_i \text{ of the second class}$$

این می تواند به صورت زیر نوشته شود:

با کنار هم قرار دادن این دو یک مسئله بهینه سازی به دست می آید:

(Minimize w, b) in

$$y_i(w^T x + b) \geq 1, \quad \forall i \in \{1, 2, \dots, m\}, \quad (1) \quad \|\mathbf{w}\| \quad (2)$$

مسئله بهینه سازی مشاهده شده در قسمت قبل،

مسئله سختی، برای حل کردن است، زیرا به $\|\mathbf{w}\|$ وابسته است (نرم w). خوشبختانه می توانیم بدون تغییر در مسئله $\|\mathbf{w}\|$ را با $\frac{1}{2}\|\mathbf{w}\|^2$ جانشین کنیم (عبارت $\frac{1}{2}$ برای آسودگی در محاسبات ریاضی آورده شده). این یک مسئله بهینه سازی (OP) برنامه ریزی غیرخطی (QP) است. به طور واضح تر:

Minimize (in w, b) c

$$\frac{1}{2}\|\mathbf{w}\|^2 \quad (3)$$

می توان عبارت قبل را با استفاده از ضرایب نا

منفی لاگرانژ به صورت زیر نوشت که در آن α_i ضرایب لاگرانژ هستند:

$$\min_{\mathbf{w}, b, \alpha} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\} \quad (4)$$

اما فرمول فوق اشتباه است. فرض کنید ما بتوانیم خانواده ای از ابر صفحات که نقاط را تقسیم می

کنند پیدا کنیم. پس همه $y_i(\mathbf{w} \cdot \mathbf{x}_i - b) - 1 \geq 0$. بنابراین ما می توانیم مینیمم را با فرستادن همه α_i به $+\infty$ پیدا کنیم. با این حال شرط پیش گفته می تواند به صورت پایین بیان شود:

$$\min_{\mathbf{w}, b} \max_{\alpha} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i (\mathbf{w} \cdot \mathbf{x}_i - b) - 1] \right\} \quad (5)$$

ما به دنبال نقاط saddle میگردیم. حالا می توان این مسئله را به کمک برنامه ریزی غیرخطی استاندارد حل کرد. جواب می تواند به صورت ترکیب خطی از بردارهای آموزشی بیان شود:

$$\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i \quad (6)$$

تنها چند α_i بزرگتر از صفر خواهد بود. \mathbf{x}_i متناظر، دقیقاً همان بردار پشتیبان خواهد بود و به شرط را ارضا خواهد کرد. از این می توان نتیجه گرفت که بردارهای پشتیبان شرط زیر را نیز ارضا می کنند: $y_i (\mathbf{w} \cdot \mathbf{x}_i - b) = 1$ که اجازه می دهد مقدار b تعریف شود. در عمل الگوریتم مقاوم تر خواهد بود اگر از تمام N_{SV} بردار پشتیبان میانگین گرفته شود:

$$b = \frac{1}{N_{SV}} \sum_{i=1}^{N_{SV}} (\mathbf{w} \cdot \mathbf{x}_i - y_i) \quad (7)$$

استفاده از این واقعیت که $\|\mathbf{w}\|^2 = \mathbf{w} \cdot \mathbf{w}$ و جانشینی $\mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{x}_i$ می توان نشان داد که دوگان SVM به مسئله بهینه سازی زیر ساده می شود:

Maximize (α_i) in

$$\tilde{L}(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (8)$$

subject to $\sum_{i=1}^m a_i y_i = 0,$

$$0 \leq a_i \leq C, \quad \forall i \in \{1, 2, \dots, m\}.$$

در اینجا هسته به صورت

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$$

تعریف می شود.

عبارت w تشکیل یک دوگان برای بردار وزن ها مجموعه آموزشی می دهد:

$$w = \sum_i \alpha_i y_i x_i.$$

ابرفصحه جداکننده بهینه اولین بار توسط Vapnik در سال ۱۹۶۳ ارائه شد که یک دسته کننده خطی بود. در سال ۱۹۹۲، Isabelle Guyon، Bernhard Boser و Vapnik راهی را برای ایجاد دسته بند غیرخطی، با استفاده قرار دادن هسته برای پیدا کردن ابرفصحه با بیشتر حاشیه، پیشنهاد دادند. [۷] الگوریتم نتیجه شده ظاهراً مشابه است، به جز آنکه تمام ضرب های نقطه ای با یک تابع هسته غیرخطی جایگزین شد. این اجازه میدهد، الگوریتم، برای ابرفصحه با بیشترین حاشیه در یک فضای ویژگی تغییرشکل داده، مناسب باشد. ممکن است، تغییرشکل غیرخطی باشد و فضای تغییر یافته، دارای ابعاد بالاتری باشد. به هر حال دسته کننده، یک ابرفصحه در فضای ویژگی با ابعاد بالا است، که ممکن است در فضای ورودی نیز غیرخطی باشد.

اگر از هسته با تابع گوسیین استفاده شود، فضای ویژگی متناظر، یک فضای هیلبرت نامتناهی است. دسته کننده ی بیشترین حاشیه، خوش ترتیب است، بنابر این ابعاد نامتناهی، نتیجه را خراب نمی کند. هسته های متداول به صورت زیر هستند:

$$k(x_i, x_j) = (x_i \cdot x_j)^d \quad \text{چند جمله ای (همگن) :}$$

$$k(x_i, x_j) = (x_i \cdot x_j + 1)^d \quad \text{چند جمله ای (ناهمگن) :}$$

$$k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \quad \text{گوسیین :}$$

$$\gamma = 1/2\sigma^2 \quad \text{: Radial Basis Function}$$

۳ کار های مرتبط

در حالی که یکی از اولین احساسات در مورد الگوریتم SVM این است که می تواند به طور خودکار مساله یادگیری را حل کند، اما در واقع این مساله به عنوان یک چالش به درخواست SVM ها به شیوه ای کاملاً اتوماتیک باقی می ماند. سوالات در مورد انتخاب تابع کرنل و مقدار پارامترها در برنامه های کاربردی در دنیای واقعی از لحاظ تجربی باقی می ماند. در حالی که تنظیمات پیش فرض پارامترها به طور کلی به عنوان یک نقطه شروع مناسب است، پیشرفت های عمده می تواند از انتخاب دقیق کرنل بهینه منجر شود. در این زمینه، سه روش بهینه سازی می تواند در جهت بهبود اجرای یک الگوریتم SVM شناسایی شود: بهینه سازی عملکرد هسته، بهینه سازی پارامترها و بهینه سازی عملکرد هسته همراه با پارامترها.

در حالی که SVM طبقه بندی کننده ذاتاً توازن بین پیچیدگی مدل و دقت طبقه بندی را حساب می کند، عملکرد هنوز هم به شدت وابسته به انتخاب مناسب پارامترهای جریمه C و هسته است. بنابراین، چندین روش می تواند مورد استفاده برای بهینه سازی پارامترهای یک طبقه بندی کننده SVM قرار بگیرد. در حالت ایده آل، ما دوست داریم که برای انتخاب مقدار پارامترهای کرنل ریسک واقعی از طبقه بندی کننده

SVM به حداقل برسد. متأسفانه، از آنجا که این مقدار در دسترس نیست، یا تخمین زده می شود و یا مرزی برای آن در نظر گرفته می شود.

Cross-validation یک تکنیک معروف برای تخمین خطای تعمیم است و چندین تفسیر [59] وجود دارد. روش Leave-one-out (لو) می تواند به عنوان شکل خاصی از K برابر Cross-validation که در آن K با تعداد نمونه (m) برابر است در نظر گرفته شود. در روش لو (LOO)، یکی از نمونه های برای هر بار آزمایش به سمت چپ می رود، و سپس آموزش و تست ها m بار تکرار می شود. در این مورد از SVM، لازم نیست روش لو در تمام m نمونه اجرا شود. با وجود چندین استراتژی برای سرعت بخشیدن به روش لو [55] و برای بهینه سازی پارامترهای SVM، روش لو هنوز هم بیش از حد پرهزینه است.

برای بهره وری بیشتر، بهتر است که تخمین ساده ای از خطا زده شود که، هر چند خام است، اما از لحاظ محاسبه گران نیست. پس از آن که SVM با مجموعه داده پارامترها آموزش دیده شد، تخمین از مرزهای خطا را می توان با کار اضافی بسیار کمی به دست آورد. در طول چند سال گذشته، چندین جمله تخمین ساده محدود پیشنهاد شده اند، برخی از آنها Xi-Alpha محدود [33]، تقریب cross-validation عمومی [59]، طول تقریبی محدود [57]، VC محدود [57]، شعاع حاشیه محدود [57] و یا کیفیت عملکردی از هسته [44] هستند.

انتخاب یک تابع هسته مناسب برای SVM ها گام بسیار مهمی برای فرایند یادگیری است. تعداد کمی روش نظام مند برای کمک در این انتخاب وجود دارد. تا به حال، هسته های مختلف برای بردارها پیشنهاد [54] شده است. بسیاری از آنها توسط یک الگوریتم SVM مورد استفاده در قسمت قبل ذکر شده است.

یکی از اولین هسته ها که در الگوریتم SVM مورد استفاده قرار گرفت هسته سیگموئید بود. با توجه به منشاء خود شبکه های عصبی [47] این کاملاً رایج بود. در واقع، یک مدل SVM با استفاده از یک تابع سیگموئید هسته برابر است با شبکه عصبی دو لایه feed-forward [30]. هسته سیگموئید با کسب مثبت ($\sigma > 0$) و آستانه منفی ($r < 0$) همیشه مثبت قطعی است و با موفقیت در عمل استفاده می شود.

امروزه محبوب ترین هسته ها RBF و هسته چند جمله ای می باشد. هسته RBF یکی از متداول ترین هسته ها است که به لطف ظرفیت خود برای تولید توابع طبقه بندی غیر پارامتریک اغلب استفاده می شود. لازم به ذکر است که اکثر این هسته ها به پارامترهای مختلفی بستگی دارند. مشاهده می شود که پارامتر σ کوچکتر (به نام پهنای باند)، بیشتر بردار پشتیبانی ها در اطراف اوج هستند و در نتیجه مرز تصمیم گیری می تواند پیچیده تر باشد. σ بزرگتر مربوط به مرز تصمیم نرم و صاف است. ضریب پارامترها و r می تواند به عنوان افست (یا تغییر) پارامتر در نظر گرفته شوند که آستانه انتقال را کنترل می کند.

روش های بهینه سازی قبلی مبتنی بر گرادیان بسیار کارآمد می باشد. آنها، با این حال، برخی اشکالات و محدودیت های دارند: تابع هدف باید مشتق پذیر باشد. تابع امتیاز، که برای ارزیابی عملکرد پارامترها (یا حداقل یک تقریب دقیق از این تابع استفاده می شود)، نیز با توجه به همه یا برخی از پارامترها باید مشتق پذیر باشد، که مانع اقدامات مناسب مانند تعداد بردار پشتیبانی می شود.

الگوریتم های تکاملی نیز برای بهینه سازی پارامترهای یک طبقه بندی کننده [20، 21، 31] SVM استفاده شده است. در [20]، استراتژی تکامل تک هدف برای پارامترهای SVM تطبیق داده شده تا مساله حل شود. در [21] GA تک هدف برای تنظیم پارامتر C در یک طیف گسسته بهینه سازی شده است. [31] Igel یک رویکرد تکاملی بهبود یافته برای بهینه سازی پارامترها پیشنهاد کرده است. SVM بهینه سازی

پارامترها به عنوان یک مسئله بهینه سازی چند هدفه مشاهده شده، که در آن پیچیدگی مدل و دقت و صحت آموزش به عنوان دو هدف متضاد تعریف شده اند (به عنوان مثال، بایاس در برابر واریانس، ظرفیت در مقابل ریسک عملیاتی). معیارهای بهینه سازی های مختلف ارزیابی شده اند.

۴ هسته تکاملی از هسته ها (eKoK)

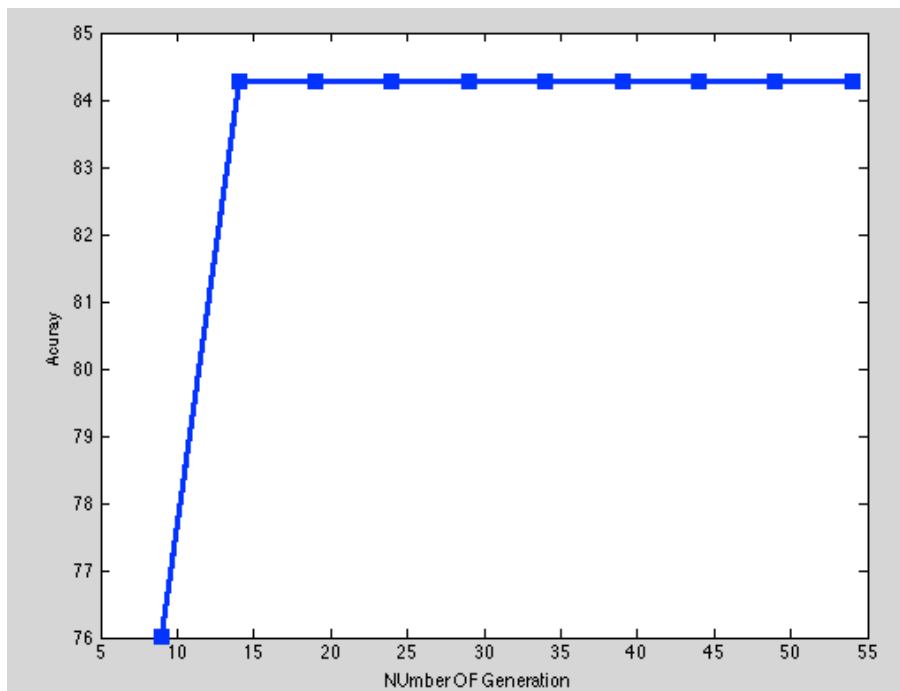
در این بخش رویکرد ما برای طراحی خودکار KoKS توصیف می شود. این مدل یک مدل هیبرید است: که آن را برای ساخت توابع مثبت و متقارن (KoKS)، و بهینه سازی یک تابع fitness با استفاده از یک طبقه بندی کننده SVM به کار می بریم. یک کروموزوم GP بیان تحلیلی مانند KOK فراهم می کند. مدلی که ما پیشنهاد می کنیم در واقع به دنبال جایگزین اطلاعات تخصصی در حوزه مربوط به طراحی تابع کرنل SVM و انتخاب پارامترهای آن، با یک الگوریتم GP می باشد.

هر کروموزوم شامل یک سری عدد می باشد که نشان دهنده کرنل های پایه می باشد در ابتدا این اعداد به صورت تصادفی انتخاب می شوند و در مراحل بعدی با توجه به fitness آنها با یک دیگر ترکیب می شوند. پس از آنکه ۲ کروموزوم انتخاب می شوند با یک دیگر ترکیب می شوند که ترکیب کروموزوم ها به صورت single point می باشد، در مورد انتخاب کروموزوم ها از ۳ رویکرد تورنمنت، تصادفی و چرخ رولت استفاده شده است همچنین برای عملگر جهش بعد از انتخاب یک کروموزوم با یک احتمال خاص یکی از ژن های آن عوض می شود. در مدل ارائه شده به دلیل ترکیب هسته ها تعداد بردارهای پشتیبان افزایش می یابد که همین امر به افزایش دقت در اکثر موارد منجر می شود.

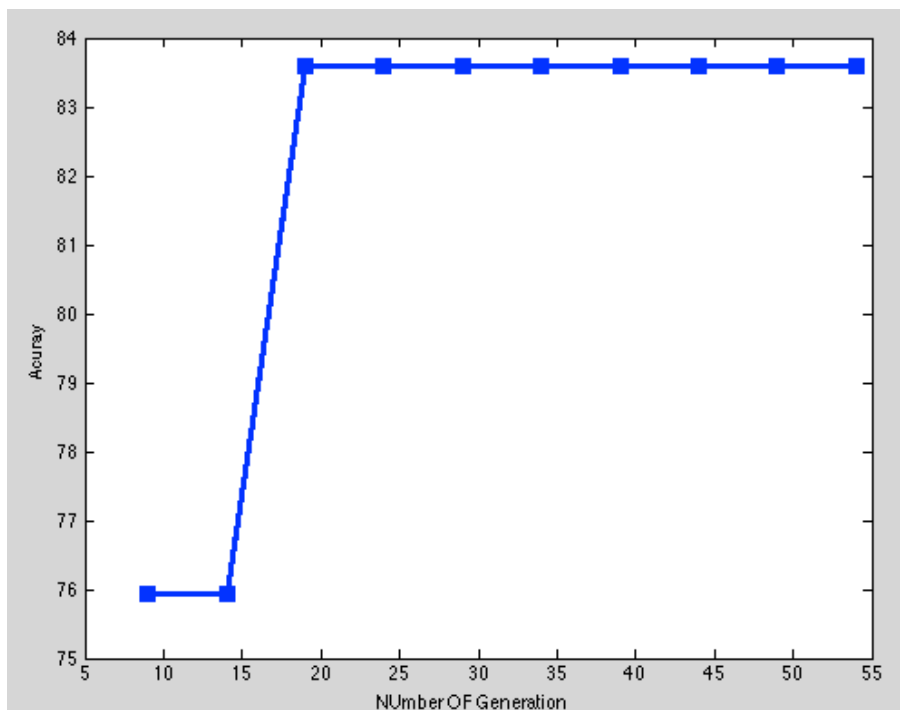
۵ آزمایش ها

گزارش های این بخش در مورد اعتبار آزمایشی eKoK، در یک مجموعه از مسائل استاندارد تست می باشد [19]. مدل ترکیبی ما در چارچوب طبقه بندی کننده ی SVM [8] LIBSVM ارائه شده است. تمام مجموعه داده ها مربوط به مسائل باینری، طبقه بندی با اندازه های مختلف (تعداد اقلام و تعداد آنها) می باشد و متعلق به حوزه های مختلف: رشته های پزشکی، اقتصادی و جغرافیایی هستند. شرح کوتاهی از هر یک از مجموعه داده ها در جدول ۱ آمده است.

ID	Name	#items	#characteristics
P_1	ionosphere	351	34
P_2	breast	683	10
P_3	heart	270	13
P_4	a1a	4217	123
P_5	a2a	2591	123
P_6	sonar	208	60
P_7	diabetes	768	8



شکل ۱. دقت بدست آمده با cross over تصادفی بر روی دیتاست a2a



شکل ۲. دقت بدست آمده با cross over تصادفی بر روی دیتاست a1a

۶ نتیجه

مدل جدید هیبریدی که به منظور حل مشکلات طبقه بندی ارائه شده است: یک الگوریتم GP همراه با یک طبقه بندی کننده SVM برای تولید KoKS. چندین آزمایش عددی به منظور مقایسه eKoKS با هسته های دیگر (کلاسیک و یا تکامل یافته، ساده و یا چند گانه) انجام شده است. نتایج عددی نشان داده است که eKoKS از هسته های کلاسیک بهتر است اگرچه مدل ارائه شده دارای هزینه محاسباتی بالاتر در مرحله آموزش است، اما یک بار eKoK ساخته شده و در مرحله طبقه بندی به سرعت مدل های قبلی است.

References

1. BachFR, ThibauxR, JordanMI (2004) Computing regularization paths for learning multiple kernels. In: NIPS, pp 1–10
2. Banzhaf W (1998) Genetic programming: an introduction: on the automatic evolution of computer programs and its applications
3. Bennett K, Hu J, Ji X, Kunapuli G, Pang J-S (2006) Model selection via bilevel optimization. In: IJCNN'06. International joint conference on neural networks. IEEE Computer Society, Los Alamitos, pp 1922–1929
4. BoardmanM, TrappenbergT (2006) A heuristic for free parameter optimization with SVM. In: IJCNN 2006. IEEE, New York, pp 1337–1344
5. Boser BE, Guyon I, Vapnik V (1992) A training algorithm for optimal margin classifiers. In: COLT, pp 144–152
6. Bousquet O, Herrmann DJL (2002) On the complexity of learning the kernel matrix. In: Becker S et al (eds) NIPS. MIT Press, Cambridge, pp 399–406
7. ChangBR, TsaiH-F (2007) Composite of adaptive support vector regression and non-linear conditional heteroscedasticity tuned by quantum minimization for forecasts. Appl Intell 27(3):277–289
8. ChangC-C, LinC-J (2001) LIBSVM library for SVM. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
9. Chapelle O (2004) Support vector machines: induction principle, adaptive tuning and prior knowledge. PhD thesis, UPMC
10. Chapelle O, Vapnik V, Bousquet O, Mukherjee S (2002) Choosing multiple parameters for Support Vector Machines. Mach Learn 46(1/3):131–159

11. Cho S-B, Shimohara K (1998) Evolutionary learning of modular neural networks with genetic programming. *Appl Intell* 9(3):191–
12. Chung K-M, Kao W-C, Sun C-L, Wang L-L, Lin C-J (2003) Radius margin bounds for Support Vector Machines with the RBF kernel. *Neural Comput* 15(11): 2643–2681
13. Cortes C, Vapnik V (1995) Support-vector networks. *Mach Learn* 20:273–297
14. Crammer K, Singer Y (2002) On the algorithmic implementation of multiclass kernel-based vector machines. *J Mach Learn Res* 2:265–292
15. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines. Cambridge University Press, Cambridge
16. Cristianini N, Shawe-Taylor J, Elisseeff A, Kandola JS (2001) On kernel-target alignment. In: Dietterich TG, Becker S, Ghahramani Z (eds) NIPS 2001. MIT Press, Cambridge, pp 367–373
17. Diosan L, Oltean M, Rogozan A, Pécuchet JP (2007) Improving SVM performance using a linear combination of kernels. In: ICANN'07. LNCS, vol 4432, pp 218–227
18. Diosan L, Rogozan A, Pécuchet J-P (2007) Evolving kernel functions for SVMs by genetic programming. In: ICMLA'07, Ohio, USA
19. Frank A, Asuncion A (2010) UCI machine learning repository
20. Friedrichs F, Igel C (2005) Evolutionary tuning of multiple SVM parameters. *Neurocomputing* 64:107–117
21. Fröhlich H, Chapelle O, Schölkopf B (2003) Feature selection for SVM by means of GAs. In: ICTAI. IEEE, New York, pp 142–148
22. Gagne C et al (2006) Genetic programming for kernel-based learning with co-evolving subsets selection. In: Runarsson TP et al (eds) 9th PPSN'06. Springer, Berlin, pp 1008–1017
23. Girolami M, Rogers S (2005) Hierarchical Bayesian models for kernel learning. In: ICML, pp 241–248
24. Gold C, Sollich P (2003) Model selection for Support Vector Machine classification. *Neurocomputing* 55(1–2):221–249
25. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison Wesley, Reading

26. Gunn S, Kandola J (2002) Structural modelling with sparse kernels. *Mach Learn* 48:137–163
27. Hastie T, Rosset S, Tibshirani R, Zhu J (2003/2004) The entire regularization path for the SVM. *J Mach Learn Res* 5:1391–1415
28. Hooke R, Jeeves TA (1961) Direct search solution of numerical and statistical problems. *J ACM* 8:212–229
29. Howley T, Madden MG (2005) The genetic kernel Support Vector Machine: description and evaluation. *Artif Intell Rev* 24(3–4):379–395
30. Huang Y (2009) Advances in artificial neural networks—methodological development and application. *Algorithms* 2(3):973–1007
31. Igel C (2005) Multi-objective model selection for SVM. In: Coello Coello CA et al (eds) EMO 2005. LNCS, vol 3410. Springer, Berlin, pp 534–546
32. Imbault F, Lebart K (2004) Stochastic optimization approach for parameter tuning of SVM. In: ICPR (4), pp 597–600
33. Joachims T (2001) The maximum-margin approach to learning text classifiers. *Künstl Intell* 15(3):63–65
34. Keerthi S, Sindhvani V, Chapelle O (2006) An efficient method for gradient-based adaptation of hyperparameters in SVM models. In: NIPS'06. IEEE Computer Society, Los Alamitos, pp 1–10
35. King RD (1992) *Statlog databases*
36. Kirkpatrick S, Gelatt CD Jr, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220:671–680
37. Koza JR (1992) *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge
38. Lacerda E, Carvalho AC, Braga AP, Ludermir TB (2005) Evolutionary radial basis functions for credit assessment. *Appl Intell* 22(3):167–181
39. Lanckriet GRG et al (2004) Learning the kernel matrix with Semi-definite Programming. *J Mach Learn Res* 5:27–72

40. Mallick BK, Ghosh D, Ghosh M (2005) Bayesian classification of tumours by using gene expression data. *J R Stat Soc Ser B* 67(2):219–234
41. Mercer J (1909) Functions of positive and negative type and their connection with the theory of integral equations. *Philos Trans R Soc* 209:415–446
42. Momma M, Bennett KP (2002) A pattern search method for model selection of SV Regression. In: Grossman RL et al (eds) *SIAM 2002*. SIAM, Philadelphia, pp 2–16
43. Ohn S-Y, Nguyen H-N, Chi S-D (2004) Evolutionary parameter estimation algorithm for combined kernel function in SVM. In: *Content computing, AWCC 2004*. Springer, Berlin, pp 481–486
44. Ong CS, Smola A, Williamson B (2005) Learning the kernel with hyperkernels. *J Mach Learn Res* 6:1043–1071
45. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2007) More efficiency in multiple kernel learning. In: *ICML*, pp 775–782
46. Lessmann RS, Crone S (2005) Genetically constructed kernels for SVM. In: *Proc. of GOR*. Springer, Berlin, pp 257–262
47. Schölkopf B (2000) The kernel trick for distances. In: Leen TK, Dietterich TG, Tresp V (eds) *NIPS*. MIT Press, Cambridge, pp 301–307
48. Schölkopf B, Smola AJ (2002) *Learning with kernels*. MIT Press, Cambridge
49. Simon HA (2001) *The sciences of the artificial*, 3rd edn. MIT Press, Cambridge
50. Sonnenburg S et al (2006) Large scale multiple kernel learning. *J Mach Learn Res* 7:1531–1565
51. Staelin C (2003) Parameter selection for Support Vector Machines. Tech Rep HPL-2002-354R1, Hewlett Packard Laboratories
52. Sullivan K, Luke S (2007) Evolving kernels for SVM classification. In: Lipson H (ed) *GECCO 2007*. ACM, New York, pp 1702–1707
53. Syswerda G (1991) A study of reproduction in generational and steady state Genetic Algorithms. In: Rawlins GJE (ed) *FOGA*.

Morgan Kaufmann, San Mateo, pp 94–101

54. Taylor JS, Cristianini N (2004) Kernel methods for pattern analysis. Cambridge University Press, Cambridge

55. Tsuda K, Rätsch G, Mika S, Müller K-R (2001) Learning to predict the leave-one-out error of kernel based classifiers. In: LNCS, vol 2130, pp 331–338

56. Vapnik V (1995) The nature of statistical learning theory. Springer, Berlin

57. Vapnik V, Chapelle O (2000) Bounds on error expectation for SVM. *Neural Comput* 12(9):2013–2036

58. Verma B, Hassan S (2009) Hybrid ensemble approach for classification. *Appl Intell*, 1–21

59. Wahba G, Lin Y, Zhang H (1999) GACV for support vector machines. In: Smola B, Schölkopf S (eds) *Advances in large margin classifiers*. MIT Press, Cambridge

60. Wang G, Yeung D-Y, Lochoovsky FH (2007) A kernel path algo-

rithm for SVM. In: *ICML 07*. ACM Press, New York, pp 951–958
61. Xiong H, Swamy M, Ahmad M (2005) Optimizing the kernel in the empirical feature space. *IEEE Trans Neural Netw* 16(2):460–

474

62. Zhang Z, Jordan MI (2006) Bayesian multicategory support vector machines. In: *The twenty-second conference on uncertainty in artificial intelligence (UAI), 2006*

63. Zhang Z, Kwok JT, Yeung D-Y (2006) Model-based transductive learning of the kernel matrix. *Mach Learn* 63(1):69–101