پردیس علوم

دانشکده ریاضی، آمار و علوم کامپیوتر

شیوه ماشین بردار پشتیبان برای رده‌بندی چندگانه داده‌ها

نگارنده

نیلیا موسوی

استاد راهنما: دکترمرتضی امینی

پایان‌نامه برای دریافت درجه کارشناسی

در رشته آمار و کاربردها

تیر ماه ۱۳۹۶

چکیده

روش ماشین بردار پشتیبان از جمله روش‌های نسبتاً جدیدی است که در سال‌های اخیر کارایی خوبی نسبت به روش‌های قدیمی‌تر برای طبقه‌بندی از جمله شبکه‌های عصبی پرسپترون نشان داده است. ماشین بردار پشتیبان روش موثری برای مدل‌سازی داده‌هاست که از آن برای طبقه‌بندی و رگرسیون می‌توان بهره برد. ماشین بردار پشتیبان یک طبقه‌بندی‌کننده دودوئی است، این الگوریتم با افزایش ابعاد مسئله و با استفاده از تابع کرنل یک چهارچوب یکپارچه برای اکثر مدل‌ها را فراهم می‌کند.

در این پروژه که هدف اصلی آن استفاده از ماشین بردار پشتیبان در طبقه‌بندی داده‌های بررسی فیلم است، ابتدا بسته‌های نرم‌افزاری و برخی توابع موجود در نرم‌افزار آماری آر در فرآیند طبقه‌بندی به روش ماشین بردار پشتیبان را بررسی کرده و با استفاده از توابع svm() و ksvm() و با به کارگیری فنون متن‌کاوی به طبقه‌بندی نظرات مثبت و منفی پرداخته‌ایم.

کلید واژه‌ها: ماشین بردار پشتیبان، طبقه‌بندی بردار پشتیبان، جداسازی غیرخطی، توابع هسته، متن‌کاوی، طبقه‌بندی داده‌های بررسی فیلم.

پیشگفتار

متن‌کاوی به مثابه تحلیل هوشمند متن، کاوش متن داده یا کشف دانش از متن شناخته شده است. متن کاوی، حوزه ای نو و میان رشته ای است که از رشته های بازیابی اطلاعات، داده کاوی، یادگیری ماشینی، آمار و زبان شناسی محاسباتی مشتق شده است. از آنجا که بسیاری از اطلاعات به شکل متن ذخیره شده اند، متن کاوی، ارزش اقتصادی بسیار بالایی در پی خواهد داشت.

اطلاعات متنی در سال‌های اخیر به شکل چشمگیری گسترش یافته و در زمانی قابل پیش‌بینی میزان این اطلاعات رشد سریع‌تری خواهد یافت. از این رو ضرورت سازماندهی اطلاعات اهمیت بیشتری می‌یابد.

با توجه به افزایش حجم اطلاعات متنی، وجود روش‌های طبقه‌بندی متون ضروری به نظر می‌رسد. روش ماشین بردار پشتیبان یکی از بهترین روش‌ها در دسته‌بندی متون می باشد. این پروژه شامل ۳ فصل می‌باشد:

در فصل اول مختصری از تاریخچه و مفاهیمی که تشکیل دهنده الگوریتم‌های ماشین بردار پشتیبان هستند، ارائه شده است. همچنین در این فصل به محاسبه طبقه‌بندی ماشین بردار پشتیبان در حالتی که نقاط به صورت خطی جداپذیرند و حالتی که طبقات با هم همپوشانی دارند، با استفاده از مسائل بهینه‌سازی پرداخته‌ایم. در ادامه، ماشین بردار پشتیبان بااستفاده از توابع هسته را بررسی نموده‌ایم.

در فصل دوم به معرفی بسته‌های نرم‌افزاری موجود در R و مهم‌ترین توابع موجود در این بسته‌ها در فرایند طبقه‌بندی به روش ماشین بردار پشتیبان پرداخته‌ایم.

در فصل سوم به بیان مفاهیم متن‌کاوی و بررسی و تشخیص داده‌های بررسی فیلم با استفاده از نرم‌افزار آماری آر پرداخته‌ایم. همچنین طبقه‌بندی و مدلسازی این مجموعه داده را توسط توابع ksvm() و svm() انجام داده و نتایج حاصل را تفسیر نموده‌ایم.

نیلیا موسوی خرداد ۹۶

فهرست مطالب

فصل ۱

طبقه‌بندی ماشین بردار پشتیبان

مقدمه

دسته‌بندی به شیوه‌هایی گفته می شود که برای متمایز کردن نقاط نمونه به دسته‌های مختلف استفاده می شود. در صورتی که یک دسته‌بندی از پیش تعیین شده برای نقاط نمونه از پیش معلوم باشد، به آن طبقه‌بندی گفته می‌شود. طبقه‌بندی براساس تعیین منحنی‌هایی به عنوان مرز طبقات انجام می‌شود. اگر این منحنی‌ها به صورت ابرصفحه‌ها (خط، صفحه و صفحات دارای ابعاد بالاتر) باشند، به آن طبقه‌بندی خطی گفته می‌شود. روش طبقه‌بندی ماشین بردار یکی از شیوه‌های طبقه‌بندی خطی است. در این شیوه انتخاب مرز بر اساس نقاطی به نام بردارهای پشتیبان انجام می‌شود و در حالتی که مرز خطی نمی‌تواند طبقه‌بندی مناسبی انجام دهد، به کمک توابع ریاضی آنها را به فضایی دیگر نگاشت می‌دهیم که در آن فضا به صورت خطی تفکیک‌پذیر باشند. الگوریتم ماشین بردار پشتیبان در سال ۱۹۶۳ توسط ولادیمیر واپنیک [۱] ابداع شد و در سال ۱۹۹۵ توسط واپنیک[۲] و کورینا کورتس[۳] برای حالت غیرخطی تعمیم داده‌شد. روش طبقه‌بندی ماشین بردار پشتیبان یک روش طبقه‌بندی دودوئی است برای طبقه‌بندی به بیش از دو طبقه می‌توان این شیوه را به صورت مکرر برای تفکیک هر یک از طبقات از سایر طبقات استفاده کرد و سپس اشتراک این مرزها را به عنوان مرزبندی چند طبقه‌ای استفاده کرد.

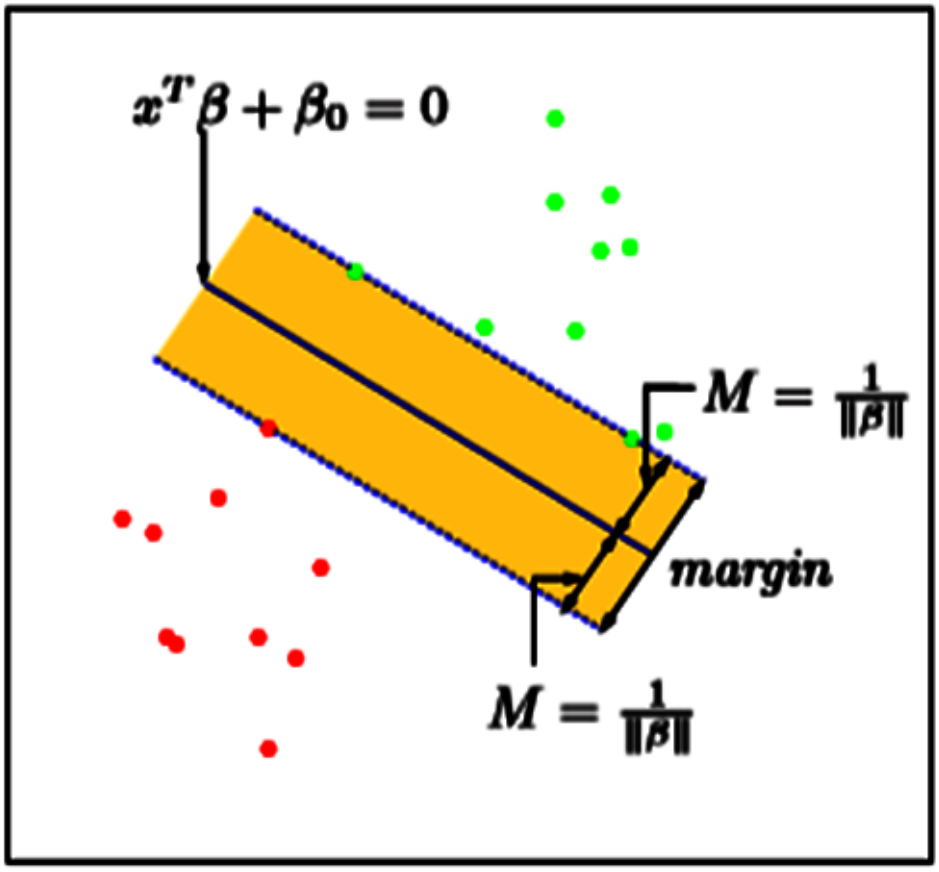
[۱] Vladimir Vapnik

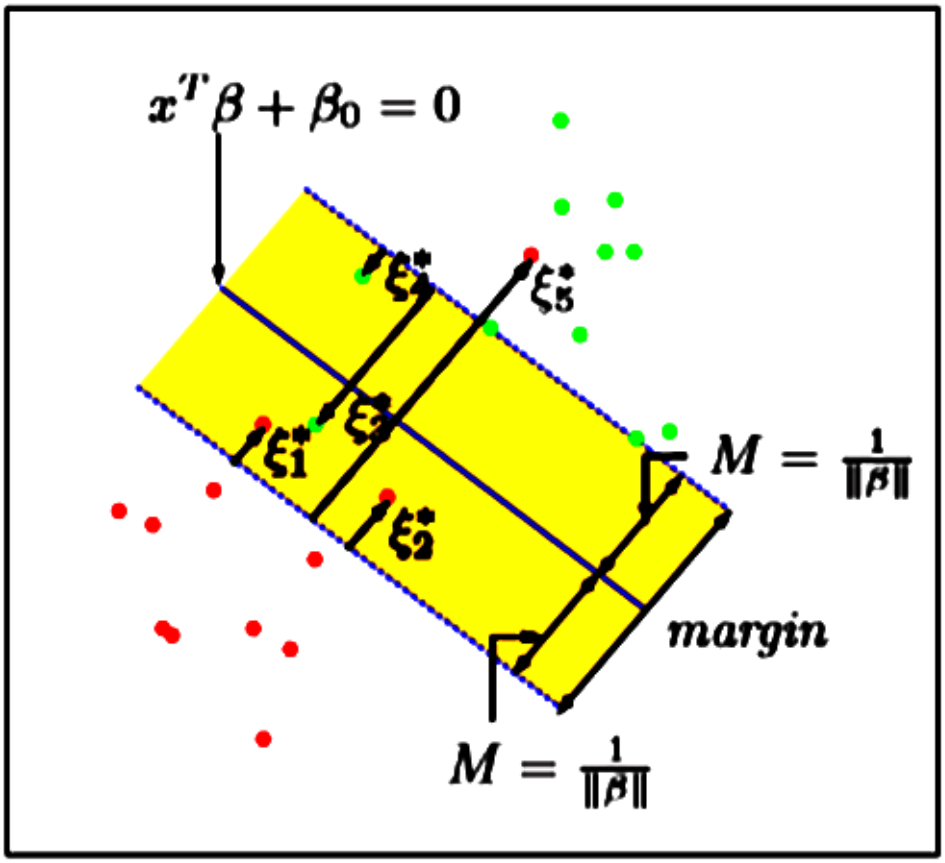
[۲] Vapnik

[۳] Corinna Cortes

۱.۱ طبقه‌بندی ماشین بردار پشتیبان

فرض کنید داده‌های $x_1, ..., x_N$ که x_i ها بردارهای چند بعدی هستند در دو کلاس طبقه‌بندی شده اند این دو کلاس با $y_i = \{-1, 1\}$ برچسب گذاری می‌شوند. برای محاسبه‌ی مرز تصمیم‌گیری ماشین بردار پشتیبان مرز تصمیم گیری باید به گونه‌ای باشدکه فاصله نزدیک‌ترین نمونه‌های آموزشی هر دو کلاس از یکدیگر در راستای عمود بر مرز تصمیم‌گیری، تا جایی که ممکن است حداکثر شود. همچنین این شیوه را می‌توان برای طبقه بندی نقاط جداناپذیری که ممکن است با یک مرز خطی جدا نشوند، تعمیم داد.





شکل ۱.۱: طبقه‌بندی ماشین بردار پشتیبان

در شکل ۱.۱ پنل بالایی یک مجموعه نقاط جداپذیر را نشان می‌دهد. خط توپر مرز تصمیم است.خطوط نقطه چین (مرزی) با حداکثر فاصله عرضی (پهنای حاشیه) ۲M هستند (ابرصفحه ای با حداکثر حاشیه که طبقات را از هم جدا می‌کند، یا به عبارتی فاصله‌ی مرز به دست آمده با بردارهای پشتیبان هر طبقه،یعنی مرزی ترین نقاط هر طبقه، حداکثر است). پنل پایینی یک مجموعه نقاط جداناپذیر را نشان می‌دهد. اگر مقداری خطا در دسته بندی را بپذیریم و (جمله خطا) نشانگر تعداد نمونه‌هایی باشد که توسط ابر صفحه غلط طبقه‌بندی می شوند، بایستی طبقه‌بندی به شکلی انجام شود که از حد خاصی بیشتر نشود.

داده‌های آموزشی به شکل N زوج $(x_1, y_1), ..., (x_N, y_N)$ با $x_i \in \mathbb{R}_p$ و $y_i \in \{-1, 1\}$ را در نظر بگیرید. ابر صفحه جداکننده را به صورت زیر تعریف می‌کنیم

$$\{x : f(x) = x^T\beta + \beta_0 = 0\} \qquad (۱.۱)$$

که در آن یک بردار واحد با طول یک می باشد. طبقه‌بندی استنتاج شده از تابع f به صورت زیر است

$$\begin{cases} \overrightarrow{\beta}\overrightarrow{x_i} + \overrightarrow{\beta_0} > 0, y_i = 1 \text{ اگر} \\ \overrightarrow{\beta}\overrightarrow{x_i} + \overrightarrow{\beta_0} < 0, y_i = -1 \text{ اگر} \end{cases} \Longrightarrow y_i(\overrightarrow{\beta}\overrightarrow{x_i} + \overrightarrow{\beta_0} > 0) \qquad (۲.۱)$$

با در نظر گرفتن حاشیه $M = 1$ برای نقاط جداپذیر داریم

$$\begin{cases} \overrightarrow{\beta}\overrightarrow{x_i} + \overrightarrow{\beta_0} > 1, y_i = 1 \text{ اگر} \\ \overrightarrow{\beta}\overrightarrow{x_i} + \overrightarrow{\beta_0} < -1, y_i = -1 \text{ اگر} \end{cases} \Longrightarrow y_i(\overrightarrow{\beta}\overrightarrow{x_i} + \overrightarrow{\beta_0} > 1) \qquad (۳.۱)$$

در واقع تابع $f(x) = x^T\beta + \beta_0$ تعیین کننده‌ی ابر صفحه‌ای با شرط $\forall i \; y_i f(x_i) > 0$ که در آن مقدار y_i برابر ۱ و ۱- و یا هر x_i یک بردار حقیقیp- بعدی است.

هدف پیدا کردن ابرصفحه جداکننده با بیشترین فاصله از نقاط حاشیه‌ای است که نقاط با $y_i = 1$ را از نقاط با $y_i = -1$ جدا کند.

بنابراین مساله بهینه سازی زیر را خواهیم داشت:

$$\begin{array}{l} \max M \\ \beta,\beta_0,||\beta||=1 \\ s.t \quad y_i(x_i^T\beta+\beta_0) \geq M, \quad i=1...N \end{array} \qquad (۴.۱)$$

فرض کنید x_+ و x_- نشان دهنده نقاطی باشند که در دو طرف ابرصفحه $f(x)$ و از بقیه نقاط به آن نزدیکتر باشند. در این صورت بنابر رابطه بالا داریم

$$\begin{cases} f(x_+) = \beta x_+ + \beta_0 = 1 \\ f(x_-) = \beta x_- + \beta_0 = -1 \end{cases} \qquad (۵.۱)$$

در نتیجه

$$M = \frac{f(x_+) - f(x_-)}{2\|\beta\|} = \frac{1}{\|\beta\|} \qquad (۶.۱)$$

۱.۱.۱ محاسبه طبقه‌بندی بردار پشتیبان

مساله بهینه‌سازی (۵.۱) معادل با مساله زیر است

$$\begin{array}{ll} \min\limits_{\beta_0,\beta} ||\beta|| \\ s.t \quad y_i(x_i^T\beta+\beta_0)\geq 1, \quad i=1...N \end{array} \tag{۷.۱}$$

مساله زیر معادل با مساله (۷.۱) است و با استفاده از روش‌های برنامه‌ریزی درجه دوم ساده تر حل می‌شود

$$\begin{array}{ll} \min\limits_{\beta_0,\beta} \frac{1}{2}||\beta||^2 \\ s.t \quad y_i(x_i^T\beta+\beta_0)\geq 1, \quad i=1...N \end{array} \tag{۸.۱}$$

مسئله بهینه‌سازی (۸.۱) محدب و تابع لاگرانژ آن به فرم زیر است

$$L_p = L(\beta, \beta_0, \alpha) = \frac{1}{2}\|\beta\|^2 - \sum_{i=1}^{N} \alpha_i(y_i(x_i^T\beta + \beta_0) - 1) \tag{۹.۱}$$

که در آن α_i بردار ضرایب لاگرانژ است.با مشتق‌گیری نسبت به β و β_0 داریم

$$\begin{cases} \frac{\partial L_p}{\partial \beta_0} = \beta - \sum_i \alpha_i y_i x_i = 0 \\ \frac{\partial L_p}{\partial \beta_0} = \sum_i \alpha_i y_i = 0 \end{cases} \Longrightarrow \begin{cases} \sum_i \alpha_i y_i x_i = \beta \\ \sum_i \alpha_i y_i = 0 \end{cases} \tag{۱۰.۱}$$

که با جایگذاری در رابطه (۹.۱) خواهیم داشت

$$\begin{aligned} L_p &= \frac{1}{2}\|\beta\|^2 - \beta^T\beta + \sum_i \alpha_i y_i \beta_0 + \sum_i \alpha_i \\ &= -\frac{1}{2}\|\beta\|^2 + \sum_i \alpha_i \\ &= -\frac{1}{2}\sum_{i,j} \alpha_i\, y_i\, x_i\, x_j^T\, y_j\, \alpha_j + \sum_i \alpha_i \end{aligned} \tag{۱۱.۱}$$

بنابراین مساله دوگان به صورت زیر است

$$\max\ -\frac{1}{2}\sum_{i,j} \alpha_i\, y_i\, x_i\, x_j^T\, y_j\, \alpha_j + \sum_i \alpha_i \tag{۱۲.۱}$$

$$s.t \quad \sum_i \alpha_i y_i = 0, \quad \alpha_i \geq 0$$

شرایط K.K.T در مساله بالا به صورت زیر هستند

$$\begin{cases} \alpha_i(y_i(x_i^T\beta + \beta_0) - 1) = 0 \\ \\ y_i(x_i^T\beta + \beta_0) - 1 \geq 0, i = 1 \cdots, N \end{cases} \qquad (۱۳.۱)$$

برآورد α از رابطه‌ی (۱۲.۱) به دست می‌آید و با توجه به رابطه‌ی (۱۰.۱) $\hat{\beta} = \sum_i \hat{\alpha}_i y_i x_i$. از رابطه‌ی اول (۱۳.۱) برآورد β_0 بدست می‌آید. با توجه به رابطه‌ی (۱۳.۱) و با توجه به اینکه برای بردارهای پشتیبان $y_i(x_i^T\hat{\beta} + \hat{\beta}_0) - 1 = 0$، برای این بردارها $\hat{\alpha}_i > 0$ و برای سایر نقاط با توجه به اینکه $y_i(x_i^T\hat{\beta} + \hat{\beta}_0) - 1 > 0$ ، $\hat{\alpha}_i = 0$ است. این مسئله در برآوردگر $\hat{\beta} = \sum_i \hat{\alpha}_i y_i$ به معنی این است که تنها بردارهای پشتیبان در برآورد ابرصفحه جداکننده نقش ایفا می‌کنند.

حال فرض کنید طبقات در این فضا با هم همپوشانی داشته باشند و به صورت خطی جداپذیر نباشند، با در نظر گرفتن خطاهای ξ_i در مساله (۷.۱) به مساله بهینه‌سازی زیر می‌رسیم

$$\min_{\beta} \|\beta\|$$

$$s.t \quad \begin{cases} y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \forall i \\ \\ \xi_i \geq 0 \, , \sum \xi_i \leq \rho \end{cases} \qquad (۱۴.۱)$$

که در آن ρ یک مقدار ثابت است. مانند رابطه (۹.۱) تابع لاگرانژ (اولیه) برای مساله (۱۴.۱) را به صورت زیر می‌نویسیم

$$L_p = \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\alpha_i(y_i(x_i^T\beta + \beta_0) - (1 - \xi_i)) - \sum_{i=1}^{N}\mu_i\xi_i \qquad (۱۵.۱)$$

که در آن C و μ_i ها ضرایب لاگرانژ اضافی در این مساله هستند. با صفر قرار دادن گرادیان های L_p نسبت به β و β_0 مشابه حالت قبل به دو رابطه $\beta = \sum_{i=1}^{N} \alpha_i \, x_i \, y_i$ و $\sum_{i=1}^{N} \alpha_i \, y_i$ می رسیم. همچنین با صفر قرار دادن مشتق L_p نسبت به ξ_i ها داریم:

$$\alpha_i = C - \mu_i \quad \forall \, i \qquad (۱۶.۱)$$

بنابراین مشابه رابطه (۱۲.۱) مساله دوگان برابر است با

$$\max \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \, y_i \, x_i x_j^T \, y_j \, \alpha_j \qquad (۱۷.۱)$$

$$s.t \quad \begin{cases} \sum_i y_i \, \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \end{cases}$$

همچنین شرایط K.K.T در این حالت به صورت زیر هستند

$$\begin{cases} \alpha_i(y_i(x_i^T\beta + \beta_0) + \xi_i - 1) = 0 \\ y_i(x_i^T\beta + \beta_0) + \xi_i - 1 \geq 0 \\ \mu_i \, \xi_i = 0 \, ; \mu_i = C - \alpha_i \end{cases} \qquad (۱۸.۱)$$

بنابراین در این حالت داریم

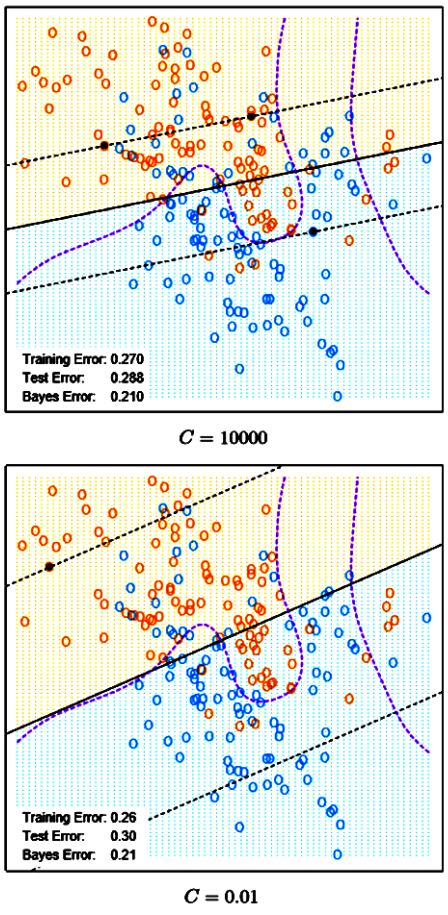
- برای نقاط دورتر از بردارهای پشتیبان $\alpha_i = 0 \, ; \xi_i = 0 \, , y_i\Big(\langle \beta, x_i \rangle + \beta_0\Big) \geq 1$
- برای بردارهای پشتیبان $0 < \alpha_i < C \, ; \xi_i = 0 \, , y_i\Big(\langle \beta, x_i \rangle + \beta_0\Big) = 1$
- برای نقاط نزدیکتر از بردارهای پشتیبان $\xi_i \geq 0 \, ; \alpha_i = C \, , y_i\Big(\langle \beta, x_i \rangle + \beta_0\Big) \leq 1$

درنتیجه تنها بردارهای پشتیبان و نقاط نزدیکتر از بردارهای پشتیبان در برآورد ابرصفحه جداکننده موثر هستند. واضح است که هر چقدر C بزرگتر باشد به نقاط نزدیکتر از بردارهای پشتیبان و همچنین به بردارهای پشتیبان وزن بیشتری در برآورد ابرصفحه داده می شود.

شکل (۲.۱) مرز بردار پشتیبان برای داده‌ها با دو کلاسی که با هم همپوشانی دارند را برای دو مقدار متفاوت پارامتر C نشان می دهد. نقاط روی مرز بردارهای پشتیبان هستند.همان طور که می‌بینید حاشیه برای C=۰,۰۱ بزرگتر از حاشیه برای C=۱۰۰۰۰ می باشد. با این حال مقادیر بزرگتر برای C توجه بیشتری را بر نقاط نزدیک مرز تصمیم دارد. در هر صورت به نقاطی که به درستی طبقه‌بندی نشده اند،بدون توجه به اینکه چقدر دورند، وزن داده می‌شود. در این مثال به دلیل ثبات (انعطاف ناپذیری) مرز خطی، فاصله فرایند (طرز عمل) به انتخاب C حساس نیست.

مقدار بهینه برای C را می توان بوسیله اعتبارسنجی متقابل به دست آورد. در این الگوریتم روال کار بدین صورت است که مدل طبقه بندی توسط زیر مجموعه داده آموزشی ساخته شده و بوسیله زیر مجموعه داده آزمایشی مورد ارزیابی قرار می‌گیرد. در روش جامع اعتبار سنجی گانه-k کل مجموعه داده‌ها به k قسمت مساوی تقسیم می‌شوند. از ۱-k قسمت به عنوان مجموعه داده‌های آموزشی استفاده می‌شود و براساس آن مدل ساخته می‌شود و با یک قسمت باقی مانده عملیات ارزیابی انجام می‌شود.

فرآیند مزبور برای هر k قسمت تکرار خواهد شد، به گونه ای که از هر کدام از k قسمت تنها یکبار برای ارزیابی استفاده شده و در هر مرتبه یک دقت برای مدل ساخته شده، محاسبه می‌شود. در این روش ارزیابی دقت نهایی طبقه بندی برابر با میانگین k دقت محاسبه شده خواهد بود. معمول‌ترین مقداری که در متون علمی برای k در نظر گرفته می‌شود برابر با ۱۰ می‌باشد. بدیهی است هر چه مقدار k بزرگتر شود، دقت محاسبه شده برای طبقه بندی قابل اعتمادتر بوده و دانش حاصل شده جامع‌تر خواهد بود و البته افزایش زمان ارزیابی طبقه بندی نیز مهمترین مشکل آن می‌باشد. در حالت k = N کنارگذاشتن یک مشاهده که بردار پشتیبان نیست تغییری در راه حل ایجاد نخواهد کرد. از این رو این مشاهدات که به درستی توسط مرز اصلی طبقه بندی شده اند در فرایند اعتبارسنجی متقابل طبقه‌بندی خواهند شد.



شکل ۲.۱: مرز خطی بردار پشتیبان برای دو مقدارC

در شکل ۲.۱ مرز خطی بردار پشتیبان نمونه داده‌های ترکیبی با دو کلاس که همپوشانی دارند، برای دو مقدار متفاوت C. خطوط شکسته (مرزها) حاشیه ها را نشان می‌دهند. جایی که $f(x) = 1, -1$. نقاط پشتیبان همه در (طرف خطا) حاشیه

قرار گرفته اند. خط چین های سیاه آن نقاط پشتیبانی هستند که دقیقا بر روی حاشیه قرار گرفته اند در پنل بالا ۶۲ درصد مشاهدات، نقاط پشتیبان هستند درحالیکه در پنل پایین شامل ۸۵ درصد نقاط می‌شود. منحنی بنفش که در زمینه با خط چین مشخص شده مرز تصمیم بیزی است.

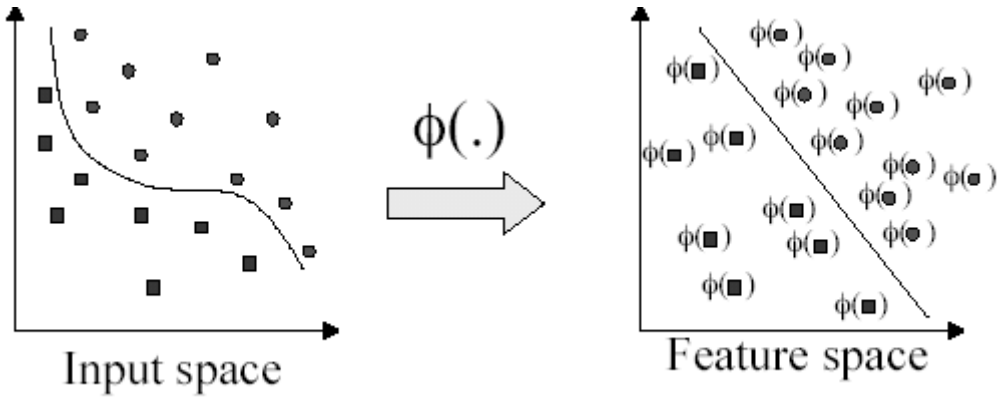
۲.۱ ماشین بردار پشتیبان با استفاده از توابع هسته

مسئله جداسازی غیرخطی در فضای ویژگی

در صورتی که نقاط درفضای داده شده همپوشانی بالایی داشته باشند و به آسانی تفکیک پذیر خطی نباشند، می توان با تبدیل مناسب مانند $\phi(x)$ آن ها را به فضایی با همپوشانی کمتر و تفکیک پذیری خطی بالاتر انتقال داد.

میتوان با نگاشت داده به یک فضای ویژگی آنها را بصورت خطی جداپذیر نمود:

تبدیل داده به فضای ویژگی: $x \to \phi(x)$



شکل ۳.۱: تبدیل داده به فضای ویژگی

شکل(۳.۱) چگونگی تبدیل فضای داده به فضای ویژگی را توسط تبدیل ϕ نمایش می دهد. در واقع با استفاده از این تبدیل ویژگی های اساسی داده ها در فضای تبدیل یافته به طور مشخص و خطی قابل تفکیک خواهند بود.

تابع لاگرانژ دوگان (۱۷.۱) دارای فرم معادل زیر است

$$L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \, \alpha_j \, y_i \, y_j \, \langle \phi(x_i), \phi(x_j) \rangle \qquad (۱۹.۱)$$

همچنین

$$f(\hat{x}) = \phi(x)^T \hat{\beta} + \hat{\beta}_0$$
$$= \sum_{i=1}^{N} \hat{\alpha}_i \, y_i \langle \phi(x), \phi(x_i) \rangle + \hat{\beta}_0$$

(۲۰.۱)

روابط (۱۹.۱) و (۲۰.۱) تنها تابعی از حاصل ضرب‌های داخلی ϕ هستند. درواقع به جای تعیین ϕ کافی است توابع زیر موسوم به توابع هسته را تعیین کنیم.

$$K(x, x') = \langle \phi(x), \phi(x') \rangle \qquad (۲۱.۱)$$

بنابر خاصیت های ضرب داخلی K بایستی متقارن و مثبت باشد و با فرض طول یک داشتن تبدیلات حاصل $K(x, x) = 1$ باشد.

سه انتخاب معروف برای K در ادبیات ماشین بردار پشتیبان به شرح زیر است

- چندجمله ای درجه d $K(x, x') = (1 + \langle x, x' \rangle)^d$
- تابع هسته گاوسی $K(x, x') = exp(-y\|x - x'\|^2)$
- تابع هسته شبکه عصبی $K(x, x') = tanh(k_1 \langle x, x' \rangle + k_2)$

برای مثال یک فضای ویژگی با ورودی‌های X_1 و X_2 و تابع هسته پلینمیال از درجه ۲ را در نظر بگیرید.

$$K(X, X') = (1 + \langle X, X' \rangle)^2$$
$$= (1 + X_1 \, X_1' + X_2 \, X_2')^2$$
$$= 1 + 2 \, X_1 \, X_1' + 2 \, X_2 \, X_2' + (X_1 \, X_1')^2 + (X_2 \, X_2')^2 + 2 \, x_1 \, X_1' \, X_2 \, X_2' \qquad (۲۲.۱)$$

پس $M = 6$ اگر $\varphi_1(X) = 1$ ، $\varphi_2(X) = \sqrt{2}X_1$، $\varphi_3(X) = \sqrt{2}X_2$ ، $\varphi_4(X) = X_1^2$، $\varphi_5(X) = X_2^2$، $\varphi_6(X) = \sqrt{2}X_1 \, X_2$ و $K(X, X') =< \varphi(X), \varphi(X') >$

فصل ۲

معرفی بسته‌های نرم‌افزاری موجود در R

۱.۲ بسته نرم‌افزاری e1071

بسته نرم‌افزاری $e1071$ اولین پیاده سازی الگوریتم ماشین بردار پشتیبان در R بوده است. تابع ()svm یک واسط را به libsvm (کتابخانه‌ای برای ماشین بردار) فراهم می‌کند. در واقع libsvm پیاده سازی سریعی از معروف ترین فرمول ها و معادلات مربوط به رگرسیون و طبقه‌بندی را به وسیله ماشین بردار پشتیبان در دسترس قرار می‌دهد و شامل انواع مختلف توابع هسته (خطی، چند جمله ای، RBF و سیگموید) نیز است.

برخی توابع بسته نرم‌افزاری $e1071$ در فرآیند طبقه‌بندی به روش ماشین بردار پشتیبان در R بسیار مهم اند و در اینجا نیز به توصیف آن ها می‌پردازیم.

تابع اصلی مدل‌بندی در این بسته، تابع ()svm است از آن برای آموزش ماشین بردار پشتیبان و همچنین رگرسیون و طبقه‌بندی استفاده می گردد، که شکل کلی این تابع به صورت زیر است

```
svm(formula, data = NULL, ...)

svm(x, y = NULL, type = NULL, kernel = "radial", degree = 3

gamma = if (is.vector(x)) 1 else 1 / ncol(x), coef0 = 0, cost = 1, nu = 0.5

class.weights = NULL, cachesize = 40, tolerance = 0.001, epsilon = 0.1

shrinking = TRUE, cross = 0, probability = FALSE, fitted = TRUE, ..., subset

na.action = na.omit)
```

آرگومان formula یک توصیف نمادین ازمدلی است که قرار است به داده‌ها برازش داده شود. به عنوان مثال اگر متغیر پاسخ y و متغیرهای کمکی $x_1, ..., x_n$ باشند، فرمول موردنظر به صورت زیر تعریف می شود

$$Y \backsim x_1 + \ldots + x_N$$

آرگومان data یک dataframe اختیاری شامل همه متغیر های مدل است. به صورت پیش فرض متغیرها از محیطی که svm از آن فراخوانی شده گرفته می‌شوند. همان طور که می دانید شکل کلی یک dataframe شامل ستون هایی از متغیر هاست که می توانند از یک مد نباشند.

آرگومان x یک ماتریس یا بردار و یا ماتریس تنک (شی ای از کلاس matrix که بوسیله بسته نرم افزاری Matrix ، کلاس matrix.csr در بسته نرم‌افزاری SparseM و یا simple-triplet-matrix که با استفاده از بسته نرم‌افزاری slam فراهم می گردد، است) است. همان طور که می دانید ماتریس‌ها آرایه های دو بعدی هستند و ماتریس تنک به ماتریسی گفته می‌شود که عناصر صفر آن زیاد است و نسبتا تعداد کمی عنصر غیر صفر دارد. برای مثال ماتریس‌های قطري و مثلثي نمونه‌هایی از ماتریس‌های تنک هستند.

آرگومان y یک بردار پاسخ با یک برچسب برای هر مولفه ی x می باشد که می تواند یک فاکتور برای طبقه‌بندی و یا یک بردار عددی (در رگرسیون) نیز باشد.

svm به عنوان یک ماشین طبقه بندی و یا رگرسیون می تواند استفاده شود و این بستگی به این دارد که y یک فاکتور هست یا نه و به عبارتی دیگر مقادیر type وابسته به این است که y گسسته است یا پیوسته. مقادیر پیش فرض برای type، به ترتیب C-classification یا eps-regression است. دیگر انتخاب ها شامل nu-classification وone-classification وnu-regression هستند.

در مسائل طبقه‌بندی می‌توان C-classification و nu-classification (برای موارد گسسته) و در رگرسیون nu-regression و eps-regression (برای موارد پیوسته) را به کار برد، همچنین زمانی که فقط یک کلاس از داده‌ها داریم برای تشخیص و کشف نقاط پرت می توان one-classification (در حالت نرمال) را به کار برد.

آرگومان kernel با توجه به انتخاب نوع تابع هسته شامل موارد زیر است (لازم به ذکر است که مقدار پیش فرض برای کرنل radial می باشد) :

$Linear$: تابع هسته خطی به صورت $K(X, X') = u' * v$

$Polynomial$: تابع هسته چند جمله ای به صورت

$$K(X, X') = (gamma * u' * v + coef0)^{degree}$$

$Radial\ basis$: تابع هسته رادیال به صورت $K(X, X') = exp(-gamma* \mid u - v \mid^2)$

$Sigmoid$:تابع هسته سیگموید به صورت $K(X, X') = tanh(gamma * u' * v + coef0)$

آرگومان degree از پارامتر های مربوط به تابع هسته نوع polynomial (با پیش فرض ۳) می باشد، که مرتبه چند جمله ای را معین می کند.

آرگومان gamma پارامتر مربوط به همه توابع هسته گفته شده به جز linear (با پیش فرض بعد داده ها/۱) است. coef0 پارامتر مربوط به عرض از مبدا نوع polynomial و sigmoid است، که به طور پیش فرض صفر در نظر گرفته می شود. پارامتر cost همان ثابت C در تابع لاگرانژ گفته شده در فصل ۱ است.

آرگومان fitted یک مقدار منطقی است که نشان می‌دهد مقادیر برازش داده شده در مدل محاسبه شوند یا خیر و پیش فرض آن TRUE است.

۲.۲ بسته نرم‌افزاری kernlab

این بسته برای کاربران R قابلیت های اساسی تابع هسته مثل محاسبه ماتریس هسته با استفاده از یک تابع هسته خاص را فراهم آورده است. همچنین در این بسته برخی از توابع سودمند که معمولا در روش‌های مبتنی بر تابع هسته استفاده می‌شود، مانند تابع حل کننده برنامه ریزی درجه دوم و الگوریتم های مدرن مبتنی بر تابع هسته نیز فراهم شده است.

این بسته به کاربر امکان تغییر بین توابع هسته را در الگوریتم موجود و حتی ایجاد و استفاده از توابع هسته برای متود های مبتنی بر هسته که در این بسته نرم‌افزاری فراهم شده را نیز به کاربر می‌دهد.

در اینجا به بررسی تابع ksvm() که در این بسته نرم‌افزاری قرار گرفته است، می پردازیم. این تابع امکان استفاده از کلاس بزرگتری از توابع هسته را در روش SVM فراهم می‌آورد. شکل کلی تابع ksvm() به صورت زیر است

```
ksvm(x, y = NULL, scaled = TRUE, type = NULL,
kernel ="rbfdot", kpar = "automatic",
```

```
C = 1, nu = 0.2, epsilon = 0.1, prob.model = FALSE,
class.weights = NULL, cross = 0, fit = TRUE, cache = 40,
tol = 0.001, shrinking = TRUE, ...,
subset, na.action = na.omit)
```

آرگومان x ، y و data در تابع ksvm() مانند این آرگومان ها در دستور svm() هستند.

آرگومان scaled یک متغیر منطقی (می تواند TRUE یا FALSE باشد) است، که بیان گر استاندارد سازی متغیر های x است.

پیش فرض برای type ، در تابع ksvm() برای متغیر y گسسته، C-svc و برای y پیوسته eps-svr است.

در دسترس دیگر برای type موارد زیر است:

- $C - svc : Cclassication$
- $nu - svc : nuclassication$
- $C - bsvc : bound - constraintsvmclassication$
- $spoc - svc : Crammer, Singernativemulti - class$
- $kbb - svc : Weston, Watkinsnativemulti - class$
- $one - svc : noveltydetection$
- $eps - svr : epsilonregression$
- $nu - svr : nuregression$
- $eps - bsvr : bound - constraintsvmregression$

پارامتر kernel می‌تواند هرتابع هسته باشد که ضرب داخلی در فضای ویژگی را بین آرگومان های دو بردار محاسبه می نماید. بسته نرم افزاری kernlab توابع هسته معروفی که می‌توانند بوسیله پارامتر kernel به صورت یک رشته متن استفاده شوند را فراهم می کند. لازم به ذکر است که پیش فرض این آرگومان تابع rbfdot می باشد. در زیر تعدادی از این توابع آمده است:

- $rbfdot : RadialBasiskernel"Gaussian"$

• $polydot$: $Polynomialkernel$

• $vanilladot$: $Linearkernel$

• $tanhdot$: $Hyperbolictangentkernel$

• $laplacedot$: $Laplaciankernel$

• $besseldot$: $Besselkernel$

• $anovadot$: $ANOVARBFkernel$

• $splinedot$: $Splinekernel$

• $stringdot$: $Stringkernel$

۳.۲ بسته نرم‌افزاری klaR

تابع svmlight() به منظور طبقه‌بندی در این بسته نرم افزاری قرار دارد. طبقه‌بندی چند گروه از طریق یک طبقه از داده‌ها در برابر دیگر داده‌ها انجام می‌شود.

به شکل مختلف استفاده از این تابع به شرح زیر است:

الف) برای حالتی که x یک دیتافریم است، شکل کلی تابع به فرم زیر است

```
svmlight(x, ...)
```

ب) در حالتی که x ماتریس متغیر های پیشگو است، فرم کلی تابع به صورت زیر است

```
svmlight(x, grouping, ...)
```

آرگومان grouping فاکتور کلاس برای هر مشاهده را نمایش می دهد.

ج) برای استفاده از فرمول در تابع svmlight() از دستور زیر استفاده می کنیم

```
svmlight(formula, data = NULL, ...)
```

مقادیر type در این تابع شامل "R" برای رگرسیون یا "C" برای طبقه‌بندی است.

۴.۲ بسته نرم‌افزاری svmpath

تابع svmpath() که در این بسته نرم افزاری قرار دارد، برای برازش دادن کل مسیر تنظیم براساس پارامتر C برای svm دو کلاسه مورد استفاده قرار می‌گیرد. مقدار بهینه C به وسیله اعتبارسنجی متقابل انتخاب می شود.

شکل کلی تابع svmpath() به صورت زیر است

```
svmpath(x, y, kernel.function = poly.kernel, trace,plot.it, ...)
```

آرگومان x یک ماتریس با n سطر (مشاهدات) و p ستون (متغیرها) است.

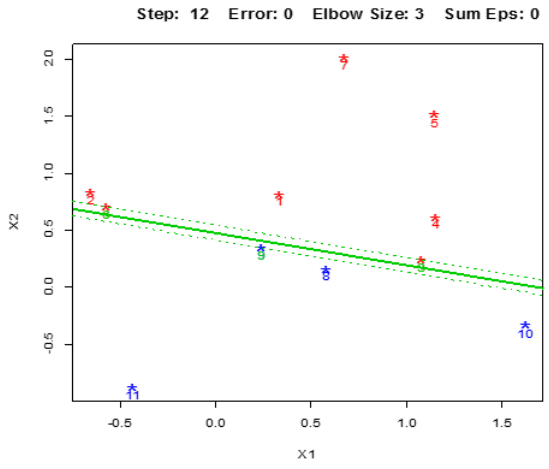
آرگومان y متغیر پاسخی است که مقادیر ۱+ و ۱- را می گیرد.

برای آرگومان kernel.function به طور پیش فرض مقدار poly.kernel در نظرگرفته شده است و همچنین می‌تواند مقدار radial.kernel را نیز اختیار کند.

آرگومان trace یک متغیر منطقی است، که به طور پیش فرض مقدار آن FALSE بوده و گزارشی از الگوریتمی که اجرا می شود را نمایش می‌دهد.

زمانی که آرگومان منطقی plot.it برابر TRUE بوده و بعد x برابر ۲ باشد، نمودار پراکنش نقاط، بردارهای پشتیبان و مرز تصمیم گیری را نمایش می‌دهد.

شکل نمونه نمودار حاصل از آرگومان plot.it برای داده‌هایی فرضی در زیر آمده است



شکل ۱.۲: نمایش بردارهای پشتیبان و مرز تصمیم‌گیری

همان طور که می‌بینید با استفاده از تابع svmpath() و با در نظر گرفتن گزینه plot=TRUE می توان در شکل ۱.۲

بردار های پشتیبان و همچنین مرز تصمیم گیری ماشین بردار پشتیبان را نیز مشاهده نمود. این شکل طبقه بندی دو گروهی که دو گروه به شکل خطی از هم قابل تفکیک هستند، را نمایش می دهد.ابرصفحه جداکننده با خط پر مشخص شده است. حاشیه عبارت است از فاصله از خط تا خط چین در هر دو طرف.

۵.۲ بسته نرم‌افزاری RTextTools

بسته نرم افزاری RTextTools یک کتابخانه جدید و بسیار خوب در R برای تحلیل و واکاوی متن است.
از جمله توابعی که در این پروژه مورد استفاده قرار گرفته و در بسته نرم افزاری RTextTools قرار دارد، تابع()create_matrix است. این تابع یک ماتریس متن-عبارت (document-term) ایجاد می کند. درواقع یک شی از کلاس DocumentTermMatrix، که در بسته tm قرار دارد، ایجاد می کند که می تواند در تابع ()create-container نیز مورد استفاده قرار گیرد. این ماتریس دارای ستون های با نام کلمات موجود در متن ها و سطرهایی برای هر متن است و درایه های آن تعداد دفعات مشاهده هر کلمه در متن متناظر است.
شکل کلی این تابع به فرم زیر است

```
create_matrix(textColumns, language="english", minDocFreq=1
maxDocFreq=Inf,minWordLength=3, maxWordLength=Inf, removeStopwords=TRUE,…)
```

آرگومان textColumns یک بردار کاراکتری است یا یک ()cbind از ستون‌های متنی است. language زبانی ست که برای داده متنی استفاده شده است. آرگومان minDocFreq حداقل تعداد دفعاتی ست که یک کلمه در یک متن باید ظاهر شود که در ماتریس گنجانده می‌شود. همچنین maxDocFreq حداکثر تعداد دفعاتی که یک کلمه در یک متن باید ظاهر شود. آرگومان minWordLength حداقل تعداد حرف های یک کلمه یا کهn-gram باید در ماتریس گنجانده شود. آرگومان removeStopwords یک مقدار منطقی ست که نشان می دهد stopwords یعنی کلمات بی مفهوم و یا کم اهمیت مانند ...،to،which،the در ماتریس در نظر گرفته شوند یا خیر.

۶.۲ بسته نرم‌افزاری rattle

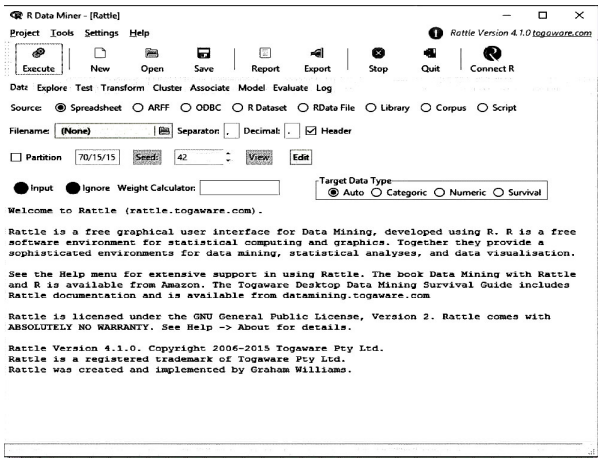
بسته rattle فضای گرافیکی مناسب برای انجام برخی روش ها و الگوریتم های داده کاوی با استفاده از نرم افزار آماری R را، فراهم می کند. در اینجا برخی از بخش های این بسته شرح داده خواهد شد.

واژه rattle مخفف عبارت R Analytical Tool To Learn Easily است. این برنامه برای انجام محاسبات داده کاوی ساده تا تحلیل های پیچیده بر روی داده ها به کمک یک زبان قدرتمند آماری طراحی شده است. rattle شامل انبوهی از بسته ها برای یک داده کاو است.

برای استفاده از محیط rattle باید بسته های $RGtk2$ و rattle را به صورت زیر در R نصب و فراخوانی کرد.

```
install.packages("rattle")
install.packages("RGtk2")
library(rattle)
library(RGtk2)
```

برای باز کردن محیط rattle با وارد کردن دستور ()rattle در R پنجره جدیدی مشابه شکل ۲.۲ باز می شود. برای مطالعه بیشتر در خصوص کاربرد و امکانات این محیط می‌توانید به کتاب R در عمل تالیف (کاباکف روبرت)[۱] مراجعه کنید.



شکل ۲.۲: فضای گرافیکیRدر rattle

[۱] Kabacoff.Robert

فصل ۳

طبقه بندی و تشخیص داده های بررسی فیلم

۱.۳ آنالیز احساسات و متن کاوی

در دنیای امروز حجم عظیمی از اطلاعات به صورت متن می‌باشد. بنابراین تکنیک های متن کاوی اهمیت یافته اند. کاوش نظرات یا تحلیل احساسات به عنوان شاخه ای از متن کاوی، به معنی یافتن دیدگاه نویسنده متن درباره یک موضوع خاص است.

از یک دیدگاه می توان کاوش نظرات (عقیده کاوی) را متفاوت از آنالیز احساسات و یا شاخه ای از آن دانست، چرا که نظرات نوع ویژه ای از متون هستند که به منظور ارزیابی موضوع، مورد بررسی قرار می گیرند. ولی در حالت کلی در هر دو مورد، هدف کشف دیدگاه نویسنده است و آنالیز احساسات و کاوش نظرات به یک زمینه مطالعاتی مربوط می شوند.

در واقع کاوش نظرات تنها یافتن موضوع متن مورد کاوش نیست بلکه یافتن نگرش ارائه شده در متن است. این کاوش به دو روش یادگیری ماشین و احساسات گرا (مبتنی بر لغت نامه) انجام می‌گیرد. در روش های یادگیری ماشین که با استفاده از تکنیک یادگیری ماشین و داده کاوی سعی می شود رابطه بین متن ها و نظرات مربوط به آن ها پیدا شود.

افزایش اهمیت آنالیز احساسات در متن با رشد رسانه‌های اجتماعی مانند نظرسنجی‌ها، وبلاگ‌ها، توئیتر و شبکه‌های اجتماعی همزمان شده است. سیستم‌های آنالیز احساسات تقریبا در همه زمینه‌های تجاری و اجتماعی مورد استفاده قرار می‌گیرند،زیرا نظرات و عقاید برای همه فعالیت های انسانی مهم بوده و تاثیر کلیدی بر رفتار ما دارند.

داده های بررسی فیلم که در این فصل مورد توجه قرار می گیرد یکی از چنین داده هایی است که طبقه بندی نظرات منتقدین به نظرات مثبت و منفی توسط سیستم های خودکار رایانه ای با توجه به حجم عظیم اطلاعات متنی امری ضروری می نماید.

۲.۳ مجموعه داده بررسی فیلم

مجموعه داده movie-review در بسته نرم افزاری $text2vec$ شامل ۵۰۰۰ مشاهده (سطر) و سه متغیر (ستون)id ، sen-timent و review می باشد.

متغیرهای id و review دارای مد کاراکتری و sentiment دارای مد عددی (۱ برای انتقادات مثبت و ۰ برای انتقادات منفی) است.

در زیر نمونه‌ای از یک متن بررسی فیلم (نمونه ۵ ام) را به عنوان مثال مشاهده می‌کنید:

```
>movie_review$review[5]
```

[1] "Superbly trashy and wondrously unpretentious 80's exploitation, hooray! The pre-credits opening sequences somewhat give the false impression that we're dealing with a serious and harrowing drama, but you need not fear because barely ten minutes later we're up until our necks in nonsensical chainsaw battles, rough fist-fights, lurid dialogs and gratuitous nudity! Bo and Ingrid are two orphaned siblings with an unusually close and even slightly perverted relationship. Can you imagine playfully ripping off the towel that covers your sister's naked body and then stare at her unshaven genitals for several whole minutes? Well, Bo does that to his sister and, judging by her dubbed laughter, she doesn't mind at all.

در اینجا نیاز به نصب و استفاده از بسته نرم‌افزاری، RTextTools برای استفاده از تابع ()create-matrix جهت ساختن ماتریس متن‌ـعبارت و همچنین بسته نرم افزاری text2vec، که داده های movie-review در آن قرار دارد، داریم. از کد های زیر برای نصب و استفاده از این بسته های نرم افزاری استفاده می‌کنیم

```
>install.packages("RTextTools")
>install.packages("text2vec")
>library(RTextTools)
>library(text2vec)
```

داده های موردنظر را با استفاده از دستور ()data فراخوانی و با استفاده از تابع ()str خلاصه ای از ساختار آن ها را مشاهده می‌کنیم

```
 >data(movie_review)
>str(movie_review)
'data.frame':   5000 obs. of  3 variables:
 $ id       : chr  "5814_8" "2381_9" "7759_3" "3630_4" ...
 $ sentiment: int  1 1 0 0 1 1 0 0 0 1 ...
 $ review   : chr
```

ابتدا مجموعه داده‌ای جدید که شامل ستون‌های sentiment و review مجموعه داده اصلی است می‌سازیم و آن را در شی x قرار می‌دهیم

```
>x=data.frame(movie_review$review,movie_review$sentiment)
```

حال برای ساختن ماتریس متن‌ـعبارت از دستور زیر استفاده می‌کنیم

```
>matrix= create_matrix(x[,1], language="english",
removeStopwords=FALSE, removeNumbers=TRUE,stemWords=FALSE)
```

ماتریس متن‌ـعبارت حاصل یک لیست است، با دستور زیر ابتدا آن را تبدیل به یک ماتریس می‌نماییم

```
>mat = as.matrix(matrix)
```

بعد این ماتریس را به صورت زیر می توان مشاهده کرد

```
>dim(mat)
[1]  5000  38504
```

یعنی ماتریس موردنظر شامل ۵۰۰۰ سطر که همان متن ها و ۳۸۵۰۴ ستون که همان عبارت های موردنظر اند، است.

۳.۳ طبقه‌بندی توسط تابع ksvm()

در این بخش از تابع ksvm() در بسته نرم افزاری kernlab استفاده می کنیم. ابتدا این بسته نرم افزاری را نصب و بارگذاری می کنیم

```
>install.packages("kernlab")
>library(kernlab)
```

سپس برای برازش مدل از دستور زیر استفاده می‌کنیم

```
>classifier=ksvm(mat[1:300,],x[1:300,2],data=movie_review,
type="C-bsvc",kernel="vanilladot" ,scaled=F)
```

همان طور که می‌بینید به دلیل حجم زیاد داده‌های موردنظر، نمونه آموزشی را ۳۰۰ متن اول در نظر گرفته ایم. خروجی مدل برازش یافته بالا به صورت زیر است

```
> classifier
Support Vector Machine object of class "ksvm"
SV type: C-bsvc  (classification)
parameter : cost C = 1
Linear (vanilla) kernel function.
Number of Support Vectors : 228
Objective Function Value : -1.1907
Training error : 0
```

برای انتخاب نمونه آزمایشی از بقیه داده ها یک نمونه ۱۰۰ تایی، بدون جایگذاری، انتخاب کرده و با استفاده از تابع predict() به پیش بینی نظرات متن های موردنظر می پردازیم

```
>z=sample(301:5000,100)
> z
```

```
[1] 1971 4671 4839 2906  537  477 2134 4404 4696 4244 2868 3872  508 4633 3897
[16] 1246 4727 2945 4931 3041  335 3610 4698  960 1077 3862 1391 3541 3599 4369
[31] 3891  355 2989 2360 1074 3555  867  954 3119  387 2162 4623 2747 2556 2959
[46] 3313 1002 1053 1338 3845  813 4130 3808  344 2350 1300 3835 1771 3171 1773
[61] 3389 2220 1197  534  584 2104 3015 1380 4578  329 3445 4435  414  402 3433
[76] 3646 3012 2771  747 4731 4497 2243 3915  452 1109 2471 2875 3692 1382 2238
[91] 4725 3884 4472 1075 4828 2923  841 3604 4459 3180
```

برای نمونه آزمایشی z پیش بینی را به صورت زیر انجام می‌دهیم

```
>predicte=predict(classifier,mat[z,])
 m=table(x[z,2], predicte)
```

خروجی این تابع جدولی مشابه جدول۱.۳ است که میزان صحت و نادرستی پیش بینی را نشان می‌دهد. به طور کلی قطر اصلی این جدول پیش‌بینی‌های درست را نشان می‌دهد.

جدول ۱.۳: بررسی میزان صحت پیش‌بینی

	۰	۱
۰	۴۷	۶
۱	۱۱	۳۶

```
>recall_accuracy(x[z,2], predicte)
[1] 0.83
```

همان طور که مشاهده می‌کنید میزان دقت این مدل ۸۳ درصد است.

۴.۳ طبقه‌بندی توسط تابع svm()

در اینجا برای طبقه‌بندی از تابع svm() استفاده می‌کنیم که در بسته نرم‌افزاری $e1071$ قرار دارد. ابتدا بسته نرم افزاری $e1071$ را نصب و بارگذاری می‌کنیم

```
>install.packages("e1071")
>library(e1071)
```

سپس برای برازش مدل از دستور زیر استفاده می‌کنیم

```
>classifier=svm(mat[1:300,],data=movie_review)
```

خروجی مدل برازش یافته بالا به صورت زیر است

```
> classifier
Call:
svm.default(x = mat[1:300, ], data = movie_review)
Parameters:
SVM-Type:  one-classification
SVM-Kernel:  radial
gamma:  2.597133e-05
nu:  0.5
Number of Support Vectors:  154
```

همان طور که می‌بینید تابع کرنل به طور پیش فرض radial و این مدل شامل ۱۵۴ بردار پشتیبان است.

برای انتخاب نمونه آزمایشی از همان نمونه ۱۰۰ تایی قبل که با استفاده از تابع sample() گرفته شده بود، استفاده می‌کنیم و آن را در شی z قرار می دهیم

```
 z=c(1971, 4671, 4839, 2906,  537,  477, 2134, 4404, 4696, 4244, 2868, 3872,508, 4633,
3897, 1246, 4727, 2945, 4931, 3041,  335, 3610, 4698,  960, 1077, 3862, 1391, 3541,
```

```
3599,4369,3891, 355, 2989, 2360, 1074, 3555, 867, 954, 3119, 387, 2162, 4623, 2747,
2959,2556,2959,3313, 1002, 1053, 1338, 3845, 813, 4130, 3808, 344, 2350, 1300, 3835,
3171,1773,1771,3389,2220,1197,534, 584, 2104, 3015,1380, 4578,329, 3445, 4435,414,
3433,402,3646, 3012, 2771, 747, 4731, 4497, 2243, 3915, 452, 1109, 2471, 2875, 3692,
2238, 1382,4725, 3884, 4472, 1075, 4828, 2923, 841, 3604, 4459, 3180)
>predicte=predict(classifier,mat[z,])
```

تابع table( ) به صورت زیر جدولی تولید می‌کند که میزان صحت و نادرستی پیش بینی را نشان می‌دهد

```
>m=table(x[z,2], predicte)
```

جدول ۲.۳ نشان می‌دهد که در داده آزمایشی ۲۶ نفر نظرشان منفی بوده و مدل نیز منفی بودن نظر را درست پیش بینی کرده است، ۲۷ نفر نظرشان منفی بوده، اما این مدل به اشتباه آن را مثبت پیش بینی کرده است و به همین ترتیب بقیه خانه های جدول تفسیر می شوند. به طور کلی قطر اصلی این جدول پیش بینی های درست را نمایش می‌دهد.

جدول ۲.۳: بررسی میزان صحت پیش‌بینی

	FALSE	TRUE
۰	۲۶	۲۷
۱	۲۰	۲۷

تابع زیر جمع اعداد روی قطر اصلی این جدول را تقسیم بر مجموع اعداد کل جدول بالا می کند و میزان دقت مدل برازش داده شده را مشخص می‌نماید.

```
>recall_accuracy(x[z,2], predicte)
[1] 0.53
```

همان طور که مشاهده می‌کنید برای نمونه آموزشی ۳۰۰ و آزمایشی تصادفی قبلی با مدلی که توسط تابع svm( ) برازش داده شده است به دقت ۵۳ درصد رسیده ایم.

نتیجه‌گیری

در این پروژه روش ماشین بردار پشتیبان برای حالات جداپذیر و جداناپذیر بررسی و به محاسبه طبقه‌بندی‌کننده ماشین بردار پشتیبان با استفاده از مسائل بهینه‌سازی پرداختیم. همچنین مسئله جداسازی غیرخطی در فضای ویژگی با استفاده از توابع کرنل مورد بررسی قرار گرفته است. پس از معرفی بسته‌های نرم‌افزاری موجود در R در فرآیند طبقه‌بندی ماشین بردار پشتیبان، با استفاده از توابع svm() و ksvm() به طبقه‌بندی نظرات مثبت و منفی مجموعه داده بررسی فیلم پرداخته‌ایم. در فرآیند طبقه‌بندی این داده‌ها، به دلیل حجم عظیم اطلاعات متنی نمونه آموزشی را ۳۰۰ متن اول در نظر گرفته و با استفاده از توابع ksvm() و svm() دو مدل برازش می‌دهیم. برای انتخاب نمونه آزمایشی، یک نمونه تصادفی ۱۰۰ تایی، بدون جایگذاری، انتخاب کرده و به پیش‌بینی مدل‌های موردنظر می‌پردازیم. در مدلی که با استفاده از تابع svm() و با انتخاب تابع هسته گاوسی برازش داده شده، به دقت ۵۳ درصد می‌رسیم. این در حالیست که با همین نمونه آزمایشی و با استفاده از مدلی که توسط تابع ksvm() و با انتخاب تابع هسته خطی برازش داده شده، به دقت بالاتری (۸۳ درصد) جهت پیش‌بینی مدل برازش داده شده، رسیده‌ایم.

کتاب‌نامه

[1] Christopher J.C Burges. "A Tutorial on support Vector Machines for pattern Recognition".Data Mining and knowledge Discovery 2:121-167,1998

[2] Hasti, R.Tibshirani and J. Friedman, 2009. The Elements of Statistical Learning. Springer

[3] https://cran.r-project.org

[4] R in action:data analysis and graphicswith R by Kabacoff.Robert

[5] V. gupta and S. lehal, " a survey of text mining technique and applications", journal of emerging technologies in web intelligence, vol.1, no.1, august 2009.

واژه‌نامه‌ی فارسی به انگلیسی

آ

آنالیز احساسات ..... sentiment analysis

الف

اعتبارسنجی متقابل ..... cross-validation

ب

برنامه‌ریزی درجه‌دوم ..... quadratic programming

ت

تابع هدف دوگان ..... dual objective function

تابع هسته پلینمیال ..... polynomial kernel

تابع هسته خطی ..... linear kernel

تابع هسته شبکه‌عصبی ..... neural network kernel

تابع هسته گاوسی ..... radial basis kernel

ح

حداکثر حاشیه ..... maximum margin

ض

ضرایب لاگرانژ ..... lagrange multipliers

ضرب داخلی ..... inner product

ط

طبقه‌بندی ..... classification

ف

فضای ویژگی ..... feature space

م

ماتریس تنک ..... sparse matrix

ماشین بردار پشتیبان ..... support vector machines

متن‌کاوی ..... text mining

مجموعه داده بررسی فیلم ..... movie-review dataset

مرز تصمیم‌گیری ..... decision boundary

مسئله بهینه‌سازی محدب convex optimization problem

ه

همپوشانی overlap

College of Science

School of Mathematics, Statistics, and Computer Science

support vector machines for multiple classification

Nilia Musavi

supervisor:Dr.Morteza Amini

A thesis submitted to Graduate Studies Office

in partial fulfillment of the requirements for the degree of

B.Sc./Master of Science/Doctor of Philosophy in

Pure Mathematics/ Applied Mathematics/ Statistics/ Computer Science

yyyy

Abstract

Support vector machine is one of the rather new methods which in recent years has shown a good performance in comparison with the older methods for the classification such as perceptron neural network. Support vector machines are an effective method for fitting models which a applied for the classification and regression. Support vector machine is a binary classification, this algorithm provides an integrated framework for the most of models by increasing dimensions and kernel function. This project in which its main aim is to use support vector machines for classification of movie-review dataset, investigates software packages and some functions in R for the classification process by the method of support vector machines and this study has classified negative and positive views by using the functions of svm() and ksvm() and text mining techniques.

Keywords: *support vector machines, the support vector classifier, nonlinear separation, kernel functions, text mining, movie-review dataset classifier.*