دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر و فناوری اطلاعات

پایان‌نامه کارشناسی ارشد
گرایش مهندسی کامپیوتر-هوش مصنوعی

عنوان
دسته‌بندی کارا مبتنی بر رگرسیون تنک

نگارش
پردیس نورزاد

استاد راهنما
محمد رحمتی

اساتید مشاور
نصرالله مقدم چارکاری
محمد مهدی عبادزاده

تیرماه ۱۳۹۱

صفحه فرم ارزیابی و تصویب پایان‌نامه- فرم تأیید اعضاء کمیته دفاع

در این صفحه فرم دفاع یا تایید و تصویب پایان‌نامه موسوم به فرم کمیته دفاع- موجود در پرونده آموزشی- **را قرار دهید.**

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

به نام خدا

تعهدنامه اصالت اثر

تاریخ:

اینجانب پردیس نورزاد متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتی امیرکبیر بوده و به دستاوردهای دیگران که در این پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذکر گردیده است. این پایان‌نامه قبلاً برای احراز هیچ مدرک هم‌سطح یا بالاتر ارائه نگردیده است.

در صورت اثبات تخلف در هر زمان، مدرک تحصیلی صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پیگیری قانونی خواهد داشت.

کلیه نتایج و حقوق حاصل از این پایان‌نامه متعلق به دانشگاه صنعتی امیرکبیر می‌باشد. هرگونه استفاده از نتایج علمی و عملی، واگذاری اطلاعات به دیگران یا چاپ و تکثیر، نسخه‌برداری، ترجمه و اقتباس از این پایان‌نامه بدون موافقت کتبی دانشگاه صنعتی امیرکبیر ممنوع است. نقل مطالب با ذکر مآخذ بلامانع است.

نام و نام خانوادگی دانشجو

امضا

تقدیر و تشکر

از استاد راهنمای عزیزم، پروفسور محمد رحمتی، برای پذیرش بنده به عنوان دانشجوی خود در آزمایشگاه پردازش سیگنال و تشخیص الگو و فراهم سازی مکانی مناسب و مطلوب برای کارهای اینجانب، کمال تشکر و قدردانی را دارم. همچنین از وی به دلیل اختصاص زمان و بردباری در پرداختن به ایده های فراوانم در پایان نامه و ارائه جهت گیری های مناسب و اطمینان بخشی از قرار داشتن در مسیر صحیح در طول مطالعاتم در دانشگاه صنعتی امیرکبیر، سپاسگزارم.

خود را حقیقتا مدیون پروفسور باب. ال. استورم از دانشگاه آلبورگ کپنهاگ می دانم. وی مرا با الگوریتم های تقریبی تنک آشنا نمود، که انگیزه‌ای برای این پایان نامه شد. از ایشان به پاس الهام بخشی قابل توجهشان برای کمک به رسیدن به اطمینان و مهارت لازم جهت دستیابی به آینده در تحقیقات، نهایت تشکر را دارم.

و اما قدردان برادرم پرهام هستم، به جهت مشاوره فنی در فضای هیلبرت توابع، ضرب های داخلی و نرم ها، و برای همه ساعات مشاوره ای که به من اختصاص داد.

بزرگ ترین قدردانی من از پدر و مادرم، معلمان فوق العاده ام، برای حمایت، آرامش و مراقبت در سراسر لحظات زندگیم است.

چکیده

اخیراً مساله کمینه‌سازی زیان مربعی ℓ_1 –منظم، توجه بسیاری را به خود جلب کرده است. بویژه، صحبت زیادی از لسو می‌شود که اصل بهینه‌سازی به کار رفته برای حل مساله معکوس خطی $y = Xa$ تحت شرط تنک بودن بردار ضریب a است. این اصل بهینه‌سازی دارای دو کاربرد اصلی در یادگیری ماشین می‌باشد. کاربرد اول در رگرسیون است که در آن $\mathbf{y}$ بردار خروجی یا برچسب داده‌ی آموزش می‌باشد. زمانی که لسو برای کدگذاری تنک و یادگیری ویژگی، یا در زمینه دسته‌بندی نمایش تنک به کار گرفته شود، $\mathbf{y}$ بردار ویژگی یا خود سیگنال است.

قسمت اول این پایان‌نامه به نحوه‌ی انتخاب الگوریتم مناسب برای مسائل دسته‌بندی عظیم می‌پردازد. در این مسائل نه تنها سرعت آموزش مدل مهم می‌باشد، بلکه سرعت ارزیابی دسته‌بندی‌کننده و حافظه‌ی اشغال شده توسط آن نیز دارای اهمیت می‌باشد. در این قسمت به اهمیت استفاده از دسته‌بندی‌کننده‌ی خطی با زیان مربعی و تنظیم ℓ_1 وزن‌ها، که همان اصل بهینه‌سازی لسو می‌باشد، برای مسائل دسته‌بندی عظیم می‌پردازیم. لازم به ذکر است که استفاده از لسو برای رگرسیون رایج است ولی ما فواید استفاده از این اصل بهینه‌سازی را در مسائل دسته‌بندی ارائه می‌دهیم. برای این کار دلایل مبتنی بر نظریه‌های آمار ارائه داده و همچنین با چند آزمایش صحت این ادعا را به صورت شهودی نیز اثبات می‌کنیم.

در قسمت دوم پایان‌نامه اهمیت استفاده از تنظیم ℓ_1 بردار وزن را برای بازسازی سیگنال (بردار ویژگی) می‌پردازیم و دلایل آماری برای موفقیت این روش در این کاربرد ارائه می‌دهیم. به دنبال موفقیت تنظیم ℓ_1 برای مسئله‌ی بازسازی سیگنال، روشی برای دسته‌بندی مبتنی بر این کاربرد تنظیم ℓ_1 توسط محققان ارائه شده (بخش ۳,۲). ما این

روش را به مسئله‌ی رگرسیون تعمیم می‌دهیم. علاوه بر این، با ارائه آزمایش، کارکرد این روش را (هم برای دسته‌بندی و هم برای رگرسیون) با الگوریتم متداول k نزدیک‌ترین همسایه مقایسه می‌کنیم. نشان می‌دهیم که این روش با آنکه گران‌تر از روش ساده و متداول نزدیک‌ترین همسایه می‌باشد، کارکرد بیشتری فراهم نمی‌کند.

در قسمت سوم بر اساس ساده‌سازی اثباتهای قبلی انجام شده‌ی قبلی نشان می‌دهیم که مسئله‌ی رگرسیون بردار پشتیبان (با زیان محوری) هم ارز مسئله‌ی نویززدایی تعاقب پایه (با زیان مربعی) است. دلیل اهمیت این اثبات این است که راهی برای توضیح موفقیت این دو روش و عملکرد مشابهشان در مسئله‌ی رگرسیون فراهم می‌کند.

واژه‌های کلیدی:

کمینه‌سازی زیان مربعی، ℓ_1 –منظم، دسته‌بندی دودویی، بازنمایی تنک

صفحه

فهرست عناوین

فهرست اشکال

صفحه

فهرست جداول

صفحه

فهرست علائم

علائم لاتین

$\mathbf{w}, \mathbf{a}$	بردار ضریب یا وزن
b	عرض از مبدا
C	فراپارامتر ضریب هزینه
K	کرنل
n	اندازه‌ی مجموعه‌ی داده‌ی آموزشی
$\mathbf{x}$	بردار ویژگی
$\mathbf{X}, \mathbf{D}$	فرهنگ لغات یا ماتریس داده یا ماتریس ویژگی
y	برجسب متناظر با یک بردار ویژگی

علائم یونانی

ξ	حاشیه خطا در دسته‌بندی‌کننده‌ی ماشین بردار پشتیبان
λ	فراپارامتر ضریب زیان

۱

فصل اول مقدمه

این پایان‌نامه دو کاربرد اصلی کمینه‌سازی زیان مربعی ℓ_1–منظم را بررسی می‌کند. نخستین کاربرد، که در فصل ۲ پوشش داده شده است، بکارگیری مساله بهینه‌سازی مطرح شده برای دسته‌بندی دودویی خطی است. کاربرد دوم، که در فصل ۳ پوشش داده شده است، اعمال کمینه‌سازی زیان مربعی ℓ_1– منظم برای نمایش سیگنال و ویژگی‌ها است. در فصل ۴، یک هم‌ارزی بین کمینه‌سازی زیان مربعی ℓ_1– منظم و رگرسیون برداری ϵ- پشتیبان، تحت تعدادی از شرایط محدودسازی توضیح داده می‌شود. در فصل آخر، نتایج و بعضی زمینه‌ها برای کارهای آینده ارائه می‌شوند.

مسئله‌ی بهینه‌سازی لسو که در ۱٫۲ معرفی می‌کنیم دارای سه خصیصه‌ی مهم است: اولاً خطی است، ثانیاً دارای زیان مربعی است و نهایتاً دارای تنظیم ℓ_1 روی وزن‌ها می‌باشد. در ادامه این فصل این سه موضوع را بررسی کرده و با اهمیت هر یک آشنا می‌شویم. در بخش ۱٫۱ راجع به دسته‌بندی با خطای محوری (که یک حالت آن دسته‌بندی‌کننده‌ی بردار پشتیبان می‌باشد) صحبت می‌کنیم. بخش ۱٫۲ به توضیح مسائل معکوس خطی، رگرسیون خطی (با خطای مربعی) و مفهوم تنظیم ℓ_1 و ℓ_2 می‌پردازیم. در بخش ۱٫۲٫۱ رگرسیون لجیستیکی را توضیح داده و با نحوه‌ی به‌دست‌آوردن احتمال واپسین توسط این روش آشنا می‌شویم. در نهایت در بخش ۱٫۳ راجع به قسمت دوم پایان‌نامه که تقریب تنک سیگنال می‌باشد توضیح می‌دهیم.

۱.۱ دسته‌بندی خطی و SVM

دسته‌بندی ماشین بردار پشتیبان (SVM)[1][۴]، [۵۳] دارای دو ویژگی شناخته شده است: کارایی تعمیمی بسیارخوب نسبت به یک جریمه ℓ_2 روی وزن‌ها؛ و یک بردار ضریب پراکندگی نسبت به زیان محوری به عنوان جمله برازش داده‌ها. پس از آن، نشان می‌دهیم که به محض محاسبه خروجی برای یک نمونه آزمون مفروض، دسته‌بندی SVM، تنها زیرمجموعه ای از نمونه‌های آزمایشی، شناخته شده به عنوان بردارهای پشتیبان است که منجر به ارزیابی سریع‌تر می‌شود. در عمل، با این‌حال، تعداد بردارهای پشتیبان با تعداد نمونه‌های آزمایشی قابل مقایسه است: بردار ضریب پراکنده نیست. بنابراین، دسته‌بندی

[1] Support Vector Machine

حاصل هزینه بر است و بسرعت با افزایش نمونه‌های آزمایشی، رشد می‌کند. مساله بهینه‌سازی SVM بشکل زیر است

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \qquad (۱.۱)$$

$$\text{Subject to } \begin{cases} y_i(\,\mathrm{w}.\mathrm{x}_i - b) & \geq 1-\xi_i \\ \xi_i & \geq 0 \end{cases} \quad \text{for } i = 1,\ldots,n,$$

این می‌تواند به‌صورت زیر جایگزین شود

$$\text{minimize } \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \qquad (۲.۱)$$

$$\text{Subject to } \quad \xi \;\geq \max\left(0{,}1 - y_i\left(\mathrm{w}.\mathrm{x}_i - b\right)\right) \qquad \text{for } i = 1,\ldots,n.$$

که در آن

$$\xi_i \geq \left[1 - y_i(w.x_i - b)\right]_+, \text{ where } [x]_+ = \max(0, x) \qquad (۳.۱)$$

و منجر به فرمول زیر برای مساله بهینه‌سازی SVM می‌شود

$$\min \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{n}\left[1 - y_i(\mathrm{w}.\mathrm{x}_i - b)\right]_+ \qquad (۴.۱)$$

که شامل دو جمله به نام های زیان محوری و ℓ_2- منظم از وزن‌ها است.

تنک بودن دسته‌بندی حاصل برای کاربردهای زمان واقعی نظیر تشخیص گفتار، مهم است. همچنین، برای کاربردهایی با تعداد زیادی مشاهده برای ارزیابی؛ برای مثال، در سیستم‌هایی برای خواندن چک‌ها و کدهای پستی، بسیار حیاتی است. یک دسته‌بندی‌کننده‌ی تنک، حافظه کمتری می‌گیرد. مسایل معکوس خطی، رگرسیون و منظم‌سازی

با T_n داده شده، هدف ما، ارزیابی ضرایب رگرسیون حقیقی $a_0, a_1, \ldots, a_p$ از مدل خطی

$$Y = \sum_{j=1}^{p} a_j X_j + a_0 \qquad (۹.۱)$$

بگونه‌ای است که، یک ارزیابی تجربی از مقدار مورد انتظار یک معیار زیـان خـاص، کمینـه شـود. توجـه داشته باشید که، متغیر تصادفی Y متغیر پاسخ (واکنش) و متغیر تصادفی X شامل متغیرهای پیشگو یـا متغیرهای توصیفی $X_1,..., X_p$ است. در غیاب نویز و با استفاده از ماتریس برداری، مساله رگرسـیون بـه یافتن جواب برای دستگاه معادلات زیر تقلیل پیدا می‌کند

$$\mathrm{y} = \mathrm{Xa} \qquad (۱۰.۱)$$

که در آن $y = [y_1,..., y_n]^T$ بردار پاسـخ‌هـای مشـاهده شـده، X یـک مـاتریس $n \times p$ تعریـف شـده به‌صورت $X = [x_1 | x_2 | ... | x_n]$ است که در آن، $x_i = [x_{i1},..., x_{ip}]^T$ و $a = [a_1,...,a_p]^T$ بردارهای هدف ضرایب رگرسیون هستند. در ادامه بحث، فرض می‌کنیم که جمله انحراف a_0 در نمایش ما صریح اسـت، یعنی

$$x_i^T = \left[x_i^T, 1\right] \text{ and } \mathrm{a}^{\mathrm{T}} = \left[\mathrm{a}^{\mathrm{T}}, a_0\right]. \qquad (۱۱.۱)$$

می‌خواهیم مستقیما معادله (۱۰,۱) را حل کنیم، اما X در مساله ما، تقریبا معکوس پـذیر نیسـت. اگـر $n = p$ یا $n < p$ باشد، دستگاه معادلات مشخص شده با (۱۰,۱) فرامعین[۲] است. در این حالـت، جـوابی موجود نیست. اگر $n < p$ باشد، دستگاه فرو معین[۳] است و تعداد نامتناهی جواب وجود دارد. بجای حـل (۱۰,۱)، می‌توان یک اندازه مشخص از باقی‌مانده بین طرفین معادلـه را کمینـه کـرد. یـک معیـار زیـان مشترک برای کاربردهای رگرسیون، تابع زیان کمترین مربعات است. خطای کمترین مربعـات، مجموعـه باقی‌مانده های مربعی است، آن هم در جایی‌که، باقی‌مانده ها به‌صورت

$$r = y - \sum_{j=1}^{p} a_j x_j - a_0. \qquad (۱۲.۱)$$

تعریف می‌شوند، با قرار دادن همه آنها با هم، خطای کمترین مربعات به‌صورت

$$\left\| y - \mathrm{Xa} \right\|_2^2. \qquad (۱۳.۱)$$

در می‌آید، به عبارت دیگر، ما مجاز به مقداری نویز در مـدل هسـتیم، و هـدف کمینـه‌سـازی یـک تـابع مشخص از آن نویز است. راهی برای ارزیابی ضرایب رگرسیون a، کمینه‌سازی (۱۳,۱) است. کمینه‌سازی

[2] Overdetermined

[3] Underdetermined

(۱,۱۳)، یک مساله بهینه‌سازی محدب نامشروط با یک هدف دیفرانسیل پذیر است، که دارای یک کمینه کلی یکتاست. این هدف در (۱,۱۳)، زمانی کمینه می‌شود، که گرادیان آن نسبت به a صفر باشد. این مطلب منجر به دستگاه معادلات زیر به‌نام معادلات نرمال می‌شود

$$(X^T X)\hat{a} = X^T y. \tag{۱۴.۱}$$

معادلات نرمال می‌توانند به‌گونه‌ای موثر حل شوند، اگر $X^T X$ رتبه کامل باشد، در این حالت، X باید دارای رتبه ستونی کاملی باشد. با این‌حال، این شرط در بسیاری از کاربردهای منجر به ناپایداری عددی، برقرار نیست. این امر می‌تواند منجر به ارزیابی کمترین مربعات با بالاترین بی دقتی شود. جوابی بر این مساله، منظم‌سازی تیخونف[۴] است. در منظم‌سازی تیخونف، مربع ℓ_2 نرم ضرایب به هدف افزوده می‌شود، و در نتیجه اولویت را به جواب‌های با ℓ_2 نرم‌های کوچک‌تر می‌دهد. ایده، قرار دادن فرض‌های پیشین در جواب است. منظم‌سازی تیخونف منجر به مساله بهینه‌سازی

$$\|y - Xa\|_2^2 + \lambda\|a\|_2^2 \tag{۱۵.۱}$$

می‌شود، که در آن، λ پارامتر منظم‌سازی است که، تعامل بین کمینه‌سازی جمله کمترین مربعات و کمینه‌سازی جمله جریمه ℓ_2 را کنترل می‌نماید. هدف در (۱۵,۱)، با یک کمینه کلی یکتای داده شده به عنوان جواب دستگاه معادلات زیر، محدب و دیفرانسیل پذیر باقی می‌ماند

$$(X^T X + \lambda I)\hat{a} = X^T y. \tag{۱۶.۱}$$

توجه داشته باشید که، $X^T X + \lambda I$ حتی اگر $X^T X$ تکین باشد، ناتکین است. کمینه‌سازی (۱۵,۱) یک رگرسیون مرزی[۵] نامیده می‌شود.

جریمه ℓ_2 بخشی از خانواده توان جریمه‌ها[۶] با اندیس‌های $\gamma \geq 0$ است

$$\sum_{j=1}^{p} |a_j|^{\gamma}. \tag{۱۷.۱}$$

---

[4] Tikhonov regularization

[5] Ridge regression

[6] Power family of penalties

این یک ℓ_γ نرم از پارامتر برخاسته از توان γ است [۲۰]. برای $\gamma=2$ منظم‌سازی تیخونف یا رگرسیون مرزی را انتخاب می‌کنیم. مشخص شده است زمانی‌که $\gamma=2$ باشد، منظم‌سازی منجر به یک بردار ضریب چگالی می‌شود، یعنی تقریبا همه مقادیر ناصفر هستند. منظم‌سازی با ℓ_2 تنها دارای ویژگی انقباضی[۷] است- انقباض مقادیر مطلق ضریب. در سوی دیگر طیف، $\gamma=0$ یک بردار ضریب تنک را تولید می‌کند. بنابراین، می‌گوییم دارای ویژگی انتخاب متغیر است. رگرسیون کمترین مربعات با $\gamma=0$، رگرسیون همه زیرمجموعه‌ها[۸] نامیده می‌شود. جریمه با ℓ_0 نرم بسیاری از مقادیر بردار ضریب را صفر می‌سازد، اما هیچ مقدار ناصفری منقبض نمی‌شود. در این بین $\gamma=1$ وجود دارد. بهینه‌سازی با جریمه ℓ_1 بهترین نتایج انقباض و انتخاب را به‌دنبال دارد. بهینه‌سازی کمترین مربعات با منظم‌سازی ℓ_1 توسط [۱۰] به عنوان نویز زدایی تعاقبی پایه ای[۹] (BPDN) و توسط Tibshirani [۴۹] به عنوان کمترین انتخاب مطلق و عملگر انقباض[۱۰] (LASSO) معرفی شد و به شهرت رسید. نگاهی دقیق‌تر به مساله کمترین مربعات ℓ_1 منظم شده، می‌اندازیم

$$\min_{a} \frac{1}{2}\|y - Xa\|_2^2 + \lambda\|a\|_1. \qquad (۱۸.۱)$$

هدف در (۱۸,۱)، محدب است، اما به‌دلیل ℓ_1 نرم، دیگر دیفرانسیل‌پذیر نیست. پس، هیچ جواب بسته‌ای مشابه با (۱۳,۱) و (۱۵,۱) موجود نیست. در فصل ۲، به سراغ چندین روش برای حل (۱۸,۱) می‌رویم.

همچنین به جریمه شبکه کشسانی[۱۱] علاقه‌مند هستیم. علاوه بر این، بسراغ ۳ شکل هم‌ارز منظم‌سازی تیخونف نیز می‌رویم.

---

[7] Shrinkage

[8] All-subset regression

[9] Basis Pursuit Denoising

[10] Least Absolute Selection and Shrinkage Operator

[11] Elastic-net penalty

۱.۱.۱ رگرسیون لجستیکی

در رگرسیون لجستیکی، تابع رگرسیون دارای یک رابطه غیرخطی با ترکیب خطی متغیرهای توصیفی است. این رابطه با تابع پروبیت[12] مدل‌سازی می‌شود.

در تعاریف دسته‌بندی، پاسخ یک متغیر دودویی یعنی $y_i \in \{-1,1\}$ است. داده پاسخ برای منظم‌سازی یک متغیر تصادفی برنولی Y با احتمال موفقیت $\eta = P\{Y=1\}$ است. احتمال موفقیت وابسته به پیشگو، یعنی $\eta = \eta(x)$؛ در نظر گرفته می‌شود. برای توزیع برنولی، مشخص شده است که $E\{Y\} = \eta$. اگر پیشگو برای منظم‌سازی یک متغیر تصادفی X در نظر گرفته شود، سپس، $\eta(x)$ انتظار شرطی Y است

$$\mathrm{E}\{Y \mid X\} = \eta(\mathrm{x}). \tag{۱۹.۱}$$

در رگرسیون خطی، انتظار شرطی Y مقدار x از X را به عنوان تابعی آفین از x ارائه می‌کند

$$\mathrm{E}\{Y \mid X\} = \eta(\mathrm{x}). \tag{۲۰.۱}$$

با این‌حال، در دسته‌بندی، یک تابع اتصال[13] یکنوای g انتظار را به ترکیب خطی پیشگوها، منتقل می‌کند

$$g(\mathrm{E}\{Y \mid X\}) = \mathrm{a}^{\mathrm{T}}\mathrm{x}. \tag{۲۱.۱}$$

مدل‌هایی به این شکل، مدل‌های خطی تعمیمی نامیده می‌شوند. در حالت رگرسیون لجستیکی، تابع اتصال به‌صورت لگاریتمی زیر انتخاب می‌شود

$$g(a) = \ln \frac{\mathrm{a}}{1-\mathrm{a}}. \tag{۲۲.۱}$$

معکوس تابع لگاریتمی، تابع لجستیکی است که با $\sigma(z)$ نشان داده می‌شود

$$g^{-1}(z) = \sigma(z) = \frac{1}{1+\exp(-z)} \tag{۲۳.۱}$$

---

[12] Probit function

[13] Link function

بنابراین، انتظار شرطی به شکل زیر در می‌آید

$$\mathrm{E}\{Y \mid X\} = g^{-1}(\mathrm{a}^{\mathrm{T}}\mathrm{x}) = \sigma(\mathrm{a}^{\mathrm{T}}\mathrm{x}). \qquad (۲۴.۱)$$

به‌منظور ارزیابی پارامتر ناشناخته a از مدل، با تشکیل تابع درست نمایی شروع می‌کنیم. فرض می‌کنیم که، مشاهدات بطور مستقل تولید می‌شوند. تابع احتمال از نظر جبری، با تابع چگالی احتمال مشترک مشاهدات، یکسان است، پس

$$\begin{aligned} L(a) = p(\mathrm{y} \mid \mathrm{X}; \mathrm{a}) &= \prod_{i=1}^{n} p(y_i \mid \mathrm{x}_i; a) \\ &= \prod_{i=1}^{n} \sigma(a^T x_i)^{y_i} (1-\sigma(a^T x_i))^{1-y_i}. \end{aligned} \qquad (۲۵.۱)$$

به‌دلایل محاسباتی، بجای لگاریتم تابع درست نمایی، تابع

$$\ell(\mathrm{a}) = \log L(\mathrm{a}) = \sum_{i=1}^{n} \log\, p(y_i \mid \mathrm{x}_i; \mathrm{a}). \qquad (۲۶.۱)$$

را در نظر می‌گیریم، بیشینه‌سازی احتمال معادل با کمینه‌سازی منفی لگاریتم درست نمایی است. مساله بهینه‌سازی به حل مساله زیر تبدیل می‌شود

$$\min_{\mathrm{a}} \sum_{i=1}^{n} -\log\, p(y_i \mid \mathrm{x}_i; \mathrm{a}) = \min_{\mathrm{a}} \sum_{i=1}^{n} -\log\, \sigma(y_i \mathrm{a}^{\mathrm{T}} \mathrm{x}_i) \qquad (۲۷.۱)$$

$$\min_{\mathrm{a}} \sum_{i=1}^{n} (1+\exp(-y_i \mathrm{a}^{\mathrm{T}} \mathrm{x}_i)). \qquad (۲۸.۱)$$

که در آن (۲۷,۱) را دنبال می‌کنیم، زیرا $y_i \in \{-1,1\}$ است. تابع

$$\log(1+\exp(-y_i \mathrm{a}^{\mathrm{T}} \mathrm{x}_i)) \qquad (۲۹.۱)$$

در (۲۸,۱)، اغلب به عنوان زیان لجستیکی[۱۴] مورد اشاره قرار می‌گیرد. همان‌طور که دیده می‌شود، بهینه‌سازی زیان لجستیکی برای تولید یک ارزیابی درست نمایی بیشینه، رخ می‌دهد. بنابراین، رگرسیون

[14] Logistic loss

لجستیکی، برای ارزیابی درست نمایی بیشینه بودن، بررسی می‌شود، البته اگر احتمال پسین $\eta(x)$ بتواند به‌صورت $1/(1+\exp(-f(x)))$ برای $f(x) \in F$ بیان شود.

در عوض، بیایید فرض کنیم که علاقه‌مند به ارزیابی احتمال پسین بیشینه (MAP) از یک پارامتر a هستیم. یک لاپلاسین پیشین روی پارامترها را در نظر می‌گیریم. توزیع احتمال لاپلاسین چند متغیره به شکل

$$p(\mathrm{a}) = (\frac{\lambda}{2})^n \exp(-\lambda\|\mathrm{a}\|_1). \qquad (۳۰.۱)$$

است، مساله بهینه‌سازی، نیازمند حل

$$\max_{\mathrm{a}} \sum_{i=1}^{n} -\log\, p(y_i \mid \mathrm{x}_i;\mathrm{a})\, p(\mathrm{a}) = \max_{\mathrm{a}} \sum_{i=1}^{n} -\log\, p(y_i \mid \mathrm{x}_i;\mathrm{a}) + \log\, p(\mathrm{a}) \qquad (۳۱.۱)$$

$$\min_{\mathrm{a}} \sum_{i=1}^{n} -\log\, p(y_i \mid \mathrm{x}_i;\mathrm{a}) + \lambda\|\mathrm{a}\|_1 \qquad (۳۲.۱)$$

$$\min_{\mathrm{a}} \sum_{i=1}^{n} \log(1+\exp(-y_i \mathrm{a}^{\mathrm{T}} \mathrm{x}_i)) + \lambda\|\mathrm{a}\|_1. \qquad (۳۳.۱)$$

است. بهینه‌سازی در (۳۳,۱)، به عنوان مساله رگرسیون لجستیکی ℓ_1 نامگذاری می‌شود.

۲.۱ تقریب تنک

به تازگی، روش‌هایی برای تقریب و کدگذاری تنک، مورد توجه قرار گرفته است [۶]. بنابراین، نیاز به بهره‌گیری از نتایج جدید و تجدید شده برای ارائه وجود دارد. در ادامه، به‌صورت خلاصه، به مساله تقریب تنک می پردازیم.

ماتریس رتبه کامل $A \in R^{n\times m}$ را با $n < m$ در نظر بگیرید، می‌خواهیم دستگاه معادلات $Ax = b$ را حل کنیم. بوضوح، دستگاه معادلات فرو معین و دارای تعدادی نامتناهی جواب است. مساله را با نیاز به پراکنده بودن x محدود می‌کنیم، یعنی x دارای تعدادی درآیه ناصفر باشد. ℓ_0 شبه نرم را به‌صورت

$$\|x\|_0 = \#\{i : x_i \neq 0\}$$

تعریف می‌کنیم، براساس این تعریف، x پراکنده است، اگر $\|x\|_0 << m$ باشد. مساله بهینه‌سازی که می‌خواهیم حل کنیم، به‌صورت زیر است

$$\text{minimize} \quad \|x\|_0 \quad \text{subject to } Ax = b. \tag{۳۴.۱}$$

با این حال، بدلیل جنبه ترکیبیاتی ℓ_0 نرم، مساله بهینه‌سازی NP-سخت است. در طول سال‌ها، تلاش‌های بسیاری برای حل مساله تقریب تنک (۳۴,۱) صورت گرفته است، برای مثال [۳۶] را ببینید. همچنین، تلاش‌هایی برای برداشتن محدودیت‌ها و مهارسازی تابع هدف (۳۴,۱) صورت پذیرفته است. بسیاری از این روش‌ها، یک نسخه بی قید از این مساله بهینه‌سازی را حل می‌کنند، برای مثال [۱۰]؛ [۴۹] را ببینید. در طول پایان‌نامه در موارد بسیاری به این موضوع برخواهیم گشت.

۱.۲.۱ نمایش تنک سیگنال‌ها و ویژگی‌ها

در سال‌های اخیر، تقریب تنک به عنوان یک روش یادگیری ویژگی بدون نظارت موفق، به ثبوت رسیده است [۱۲]. می‌توان از تقریب تنک (بخش ۳,۱) برای دستیابی به یک نمایش تنک (بعد بالاتر) از یک بردار ویژگی داده شده، استفاده نمود. این روش به عنوان کدگذاری تنک در منابع، شناخته می‌شود. Yang و همکاران [۶۱] نشان دادند که، کدگذاری تنک، زمانی‌که ویژگی‌های یادگیری در یک کار دسته‌بندی تصویری بکار رود، کارایی دسته‌بندی SVM خطی را بهبود می‌بخشد.

اگر فرهنگ لغات بکار رفته در کدگذاری تنک نمونه‌های آزمایشی با برچسب های شناخته شده، ساخته شده باشد، بازنمایی تنک حاصل می‌تواند برای دسته‌بندی بکار رود [۵۵]، نشان می‌دهیم که این آماده سازی می‌تواند برای رگرسیون نیز بکار گرفته شود.

۲

فصل دوم

کمینه‌سازی زیان مربعی ℓ_1-منظم برای دسته‌بندی

همان‌طور که در فصل ۱ توضیح داده شد، ما علاقه‌مند به مساله بهینه‌سازی کمترین مربعات ℓ_1 –منظم شده (۱۸,۱)، هستیم. در این فصل، بـه سـراغ روال هـایی مـوثر بـرای اعمـال ایـن روش‌هـا در مسـایل بهینه‌سازی و مهم‌تر از آن، بکارگیری آنها در تنظیمات دسته‌بندی می‌رویم.

دلایل بسیاری برای استفاده از اصل بهینه‌سازی لسو برای آموزش دادن دسته‌بندی کننده هـا وجـود دارد. در این فصل اهمیت استفاده از قاعده‌ی بهینه‌سازی لسو را برای مسائل دسته‌بنـدی هـم از دیـدگاه نظریه‌های آمار و احتمال و هم از دیدگاه آزمایشی بررسی می‌کنیم. همانطور کـه در ابتـدای فصـل قبـل گفته شد لسو دارای سه خصیصه‌ی مهم است: ۱. خطی بودن با ۲. زیان مربعی، و ۳. منظم‌سازی ℓ_1 روی وزن‌ها. در ادامه اهمیت این خصایص را بررسی می‌کنیم.

در این پایان‌نامه، بر دسته‌بندی خطی تمرکز داریم. دو دلیل اصـلی بـرای ایـن تصـمیم وجـود دارد. دسته‌بندی غیرخطی، مانند SVM غیرخطی، چندان برای آموزش و ارزیابی بکار نمی‌رود. این امر، بویژه، شکستی در حالت چند کلاسه است که، دسته‌بندی‌های بسـیاری بـرای یـک کـار آمـوزش مـی‌بیننـد، و به‌صورت یک در مقابل یک یا یک در مقابل همه، تنظیم می‌شوند. علاوه بر این، برای مسایل با بعد بـالا، دسته‌بندی غیرخطی، مزیت خاصی را ارائه نمی‌نماید. فضاهای داده بـا ابعـاد بـالا –مگـر در کاربردهـایی خاص- معمولا پراکنده هستند، و بنابراین، به احتمال زیاد جدایی‌پذیر خطی می‌شوند.

با حرکت به‌سوی ساخت یک دسته‌بندی کننده[1] سریع‌تر، تنک بودن را از طریق منظم‌سازی توسـط نرم القاکننده‌ی تنک بودن، تضمین می‌کنیم. این کار، امکان کنترل صریح را بر تنک بودن بردار ضریب از طریق پارامتر منظم‌سازی فراهم می‌سازد- برخلاف بهینه‌سازی SVM که چنین کنترلـی در آن چنـدان مشخص نیست (شکل ۱,۲ را ببینید). بویژه، ما از الگوریتم‌های کمینه‌سازی زیان مربعی ℓ_1–منظم شـده از جمله، SPGL1 [۵۲]، برای حل مساله دسته‌بندی استفاده می‌کنیم. این مطلب کـار مـا را از مطالعـه اخیر صورت گرفته توسـط Yuan و همکـاران [۶۲]، [۶۳] متمـایز مـی‌سـازد. آنالیزهـای آنهـا تنهـا بـر الگوریتم‌های بهینه‌سازی لجستیکی و محوری مربعی- که L_2 زیان هم نامیده می‌شود- تاکیـد دارد، ایـن مسایل به‌صورت زیر تعریف می‌شود

$$\max(0,1-ya^{T}x)^2. \qquad (۵.۲)$$

[1] Classifier

زیان لجستیکی برای حالت زیان مربعی، دوبار دیفرانسیل‌پذیر است. با این‌حال، زیان محوری مربعی این گونه نیست.

پس از مرور بر کارهای مرتبط در این زمینه به بیان نتایج آماری پرداخته و سپس نتایج آزمایش‌های انجام شده برای نشان دادن قدرت لسو در دسته‌بندی را بیان می‌کنیم. نشان می‌دهیم که لسو در عین دقت بالا، دسته‌بندی‌کننده‌ی تنک و لذا سریع‌تر به دست می‌دهد.

۱.۱.۲ کارهای مرتبط در کمینه‌سازی زیان مربعی برای دسته‌بندی

Rifkin [۴۲] در رساله دکترای خود، ادعا می‌کند که، زیان محوری راهی برای موفقیت SVM نیست. Rifkin دسته‌بندی کمترین مربعات منظم شده (RLSC)[1]

$$\min_{w} \|y - Xw\|^2 + \lambda \|w\|^2 \qquad (۶.۲)$$

و حالت غیرخطی

$$\min_{c} \|y - Kc\|^2 + \lambda c^T K c \qquad (۷.۲)$$

را ارائه می‌نماید. که در آن، K ماتریس تعریف شده با ارزیابی هسته K روی زوج نمونه‌های آزمایشی است. با استفاده از چندین مطالعه آزمایشی، Rifkin نشان داد که، RLSC به اندازه SVM روی مجموعه داده‌های گوناگون، خوب است. با این‌حال، RLSC منجر به دسته‌بندی کننده تنک نمی‌شود. علاوه بر این، RLSC غیرخطی بیش از SVM برای آزمایش زمان می‌برد.

بهینه‌سازی کمترین مربعات می‌تواند برای دسته‌بندی دودویی بکار رود [۲۶]، [۴۴]، [۶۴]، [۶۵]. آن را همچنین می‌توان در زمینه مدل‌های خطی تعمیمی-برای مثال رگرسیون لجستیکی با بهینه‌سازی کمترین مربعات باز وزنی تکراری [۳۰]، [۴۹] و برای m-ارزیابی [۱۷] بکار برد. ما یک دسته‌بندی کننده

[1] Regularized Least Square Classification

سریع را بدست می آوریم، که هنوز بعضی از قابلیت های تعمیمی را بدلیل ℓ_1 منظم‌سازی، حفظ می‌کند. با این‌حال، بعدها آنالیزهای بیشتری صورت گرفت [۲۲]، [۴۰]، [۶۰].

۲.۱.۲ کارهای مرتبط در ℓ_1- منظم‌سازی برای دسته‌بندی‌کننده تنک

Yuan و همکاران (۲۰۱۰) [۶۲] چندین دسته‌بندی کننده خطی تنک به فرم

$$\min_{\mathrm{w}} \|\mathrm{w}\|_1 + C\sum_{i=1}^{n}\xi(\mathrm{w};\mathrm{x}_i,\mathrm{y}_i) \qquad (۸.۲)$$

را با زیان لجستیکی، محوری و محوری مربعی تعریف شده به‌صورت

- $\xi\log(\mathrm{w};\mathrm{x}_i,\mathrm{y}_i) = \log(1+\exp(-y\mathrm{w}^{\mathrm{T}}\mathrm{x}))$
- $\xi\mathrm{L}_1(\mathrm{w};\mathrm{x}_i,\mathrm{y}_i) = \max(1-y\mathrm{w}^{\mathrm{T}}\mathrm{x},0)$
- $\xi\mathrm{L}_2(\mathrm{w};\mathrm{x}_i,\mathrm{y}_i) = \max(1-y\mathrm{w}^{\mathrm{T}}\mathrm{x},0)^2$

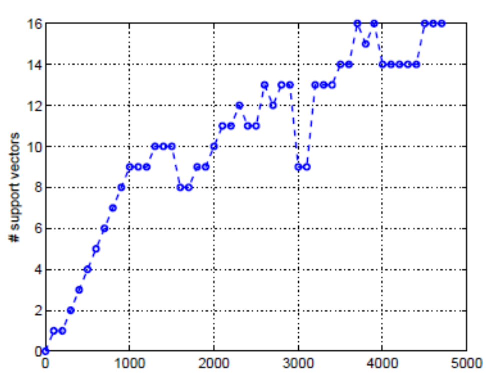
مقایسه می‌کنند، با این‌حال، آنها بهینه‌سازی زیان مربعی برای دسته‌بندی را بررسی نمی‌کنند. همان‌طور که در فصل بعدی نشان می‌دهیم، زیان مربعی دارای ویژگی‌های محاسباتی و آماری مناسبی است.

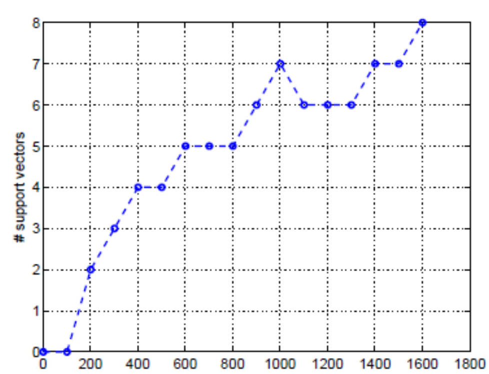
از طریق چندین آزمایش و اثبات ساده نشان می‌دهیم که، مزیت قابل توجهی در ترجیح دادن یک تابع زیان محدب بر دیگران برای کمینه‌سازی در تعاریف دسته‌بندی وجود ندارد. علاوه بر این، مزیت‌هایی برای بهینه‌سازی زیان مربعی بر زیان محوری و زیان لجستیکی موجود است [۲۶]، [۶۴]. زمانی‌که الگوریتم‌های کمینه‌سازی زیان مربعی نه چندان گرانی وجود داشته باشد، استفاده از آنها برای دسته‌بندی، بنظر قابل قبول و موثر است. جدول ۱، شامل اطلاعاتی درباره مجموعه داده‌هایی است که ما برای آزمایش خود در این فصل و فصل بعدی بکار می‌بریم. آنها از منزلگاه (وبسایت) LIBSVM برای مجموعه داده‌های رده دودویی، گرفته شده‌اند.[1] این مجموعه داده‌ها برای نمایش‌های گوناگون امور عملی، انتخاب شده‌اند.

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/dataset/binary.html

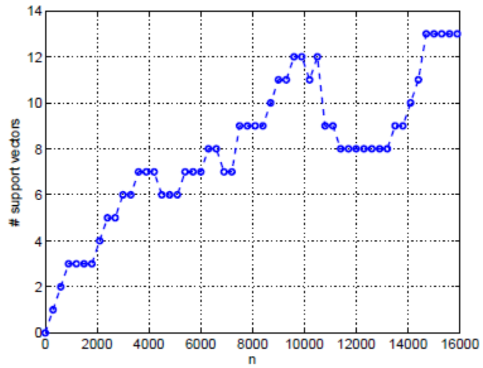
جدول ۱.۱ اطلاعات مجموعه داده‌ها: n نشان‌دهنده تعداد مشاهدات و p نشان‌دهنده تعداد ویژگی‌های هر مشاهده است.

مجموعه داده‌ها	تعداد مشاهدات (n)	تعداد ویژگی‌ها (p)
بزرگسال ۱	۱۶۰۵	۱۲۳
بزرگسال ۴	۴۷۸۱	۱۲۳
بزرگسال ۷	۱۶۱۰۰	۱۲۳
بزرگسال ۹	۳۲۵۶۱	۱۲۳
استرالیا	۶۹۰	۱۴
سرطان کولون	۶۲	۲۰۰۰
نوع پوشش	۵۸۱۰۱۲	۵۴
دیابت	۷۶۸	۸
قلب	۲۷۰	۱۳
یونکره	۳۵۱	۳۴
ال.ای.یو	۳۸	۷۱۲۹
تجویزات کبدی	۳۴۵	۶
قارچ‌ها	۸۱۲۴	۱۱۲

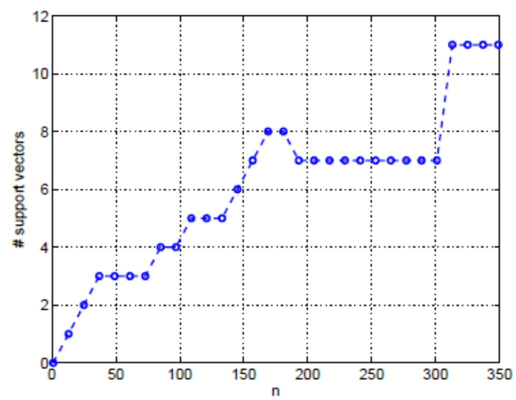


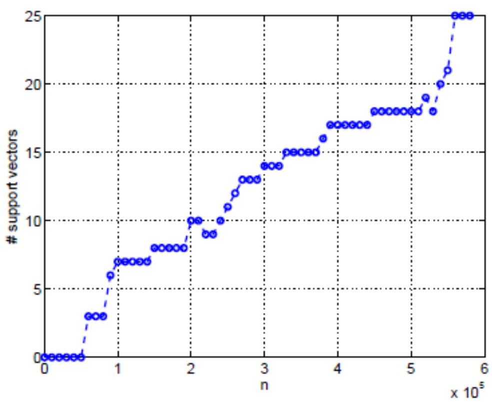


(الف)مجموعه داده بزرگسال ۱ (ب)مجموعه داده بزرگسال ۴



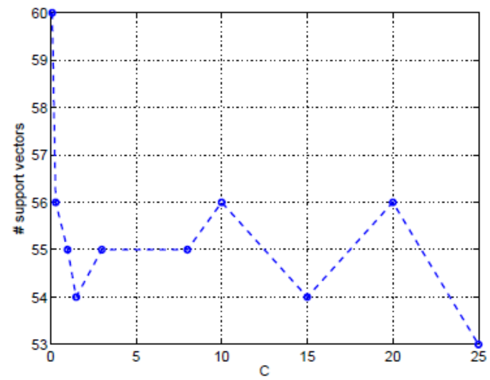
(ج) مجموعه داده بزرگسال ۷



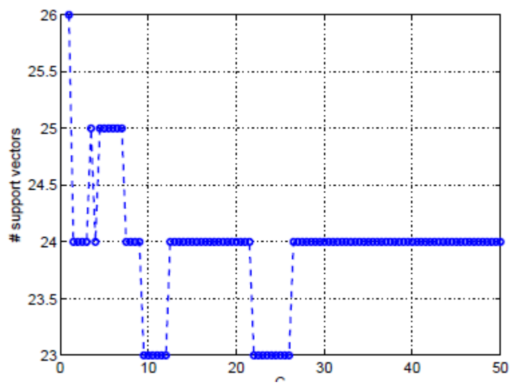


(د) مجموعه داده نوع پوشش (ه) مجموعه داده یونکره

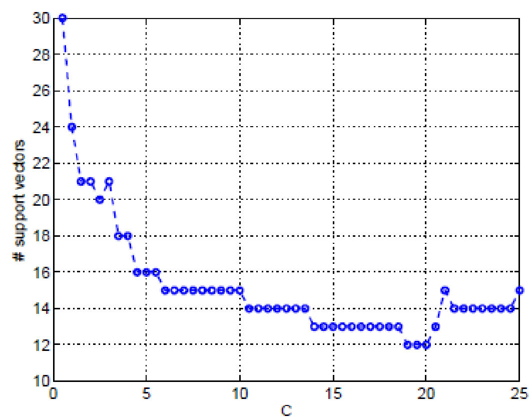
شکل ۱.۱ نمودار، تعداد بردارهای پشتیبان را نشان می‌دهد که با افزایش تعداد نمونه‌های آزمایشی رشد می‌کند.



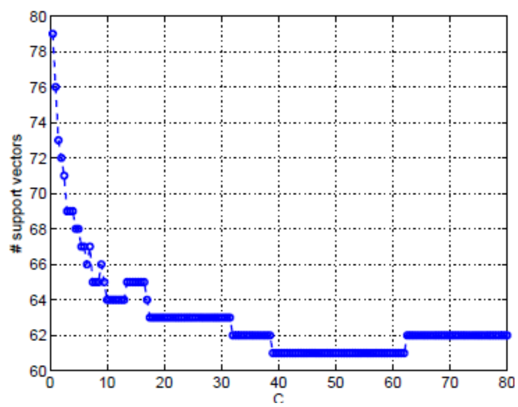
(الف) مجموعه داده‌های استرالیا



(ب) مجموعه داده‌های قلب



(ج) مجموعه داده‌های یونکره



(د) مجموعه داده‌های تجویزهای کبدی

شکل ۲.۲ نمودارها نشان می‌دهد که تعداد بردارهای پشتیبان ارتباط معنی‌داری با فراپارامتر C در بهینه‌سازی SVM ندارند.

۲.۲ دسته‌بندی و کمینه‌سازی زیان محدب

تشخیص الگو، که دسته‌بندی نیز نامیده می‌شود، فرآیند انتساب یک برچسب گسسته به یک مشاهده ناشناخته است [۱۵]. در تشخیص الگو، کار اصلی، یافتن یک تابع $g : R^p \rightarrow \{1,\ldots,M\}$ است، که یک

مشاهده نشان داده شده توسط $x \in R^p$ را گرفته و آن را بـه $y \in \{1,\dots,M\}$ نسـبت مـی‌دهـد: یکـی از کلاس های[1] در دسترس M. تابع g یک دسته‌بندی کننده[2] نامیده می‌شود.

برای آنالیز فعلی، فرض کنید X و Y متغیرهایی تصادفی باشند که مقادیرشان را بـه‌ترتیـب از R^p و $\{1,\dots,M\}$ می‌گیرند. احتمال خطا را برای دسته‌بندی کننده g به‌صورت

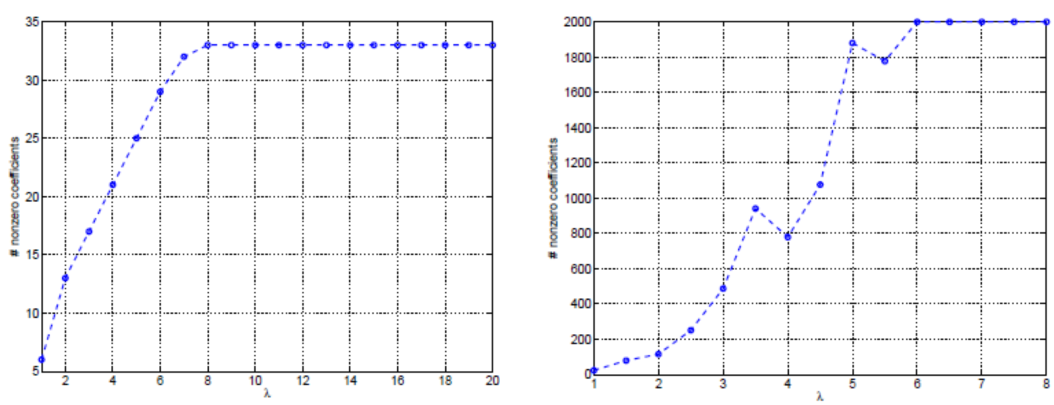
$$L(g) = \mathrm{P}\{g(X) \neq Y\}. \qquad (۲.۲)$$

تعریف می‌کنیم، در نتیجه، دسته‌بندی کننده بهینه g^* عبارت است از

$$g^* = \underset{g:\mathrm{R}^p \to \{1,\dots,M\}}{\arg\min} \ \mathrm{P}\{g(X) \neq Y\} \qquad (۳.۲)$$

و دسته‌بندی کننده بیزی نامیده می‌شود. احتمال متناظر از خطا- احتمال کمینـه خطـا- خطـای بیـزی نامیده شده و با $\mathrm{L}^* = \mathrm{L}(\mathrm{g}^*)$ نشان داده می‌شود.

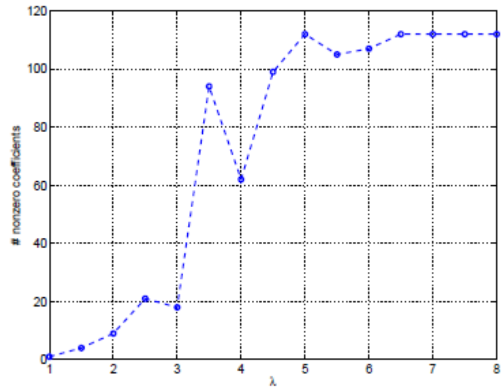
برای محاسبه g^*، نیازمند دانش از توزیع (X,Y) ای هسـتیم کـه ناشـناخته اسـت. بـا ایـن‌حـال، بـا داده‌های کافی، یک دسته‌بندی کننده g را با L(g) کم، می‌یابیم. برای دسته‌بندی، به مجموعه داده‌های $T_n = \{(x_i, y_i) : i = 1,\dots,n\}$ از n مشاهده دسترسی داریم، و فرض می‌کنیم که .i.i.d های نمونـه‌گیـری شده از توزیع (X,Y) باشند.



[1] Class

[2] Classifier

(الف) مجموعه داده‌های سرطان کولون　　　(ب) مجموعه داده‌های یونکره



(ج) مجموعه داده‌های قارچ ها

شکل ۱.۲ در این اشکال، می‌بینیم که، تعداد عناصر ناصفر جواب، زمانی افزایش می‌یابد که، پارامتر منظم‌سازی λ افزایش پیدا کند، بنابراین، ابزاری را برای کنترل تنک بودن جواب در اختیار ما قرار می‌دهد.

در باقی‌مانده این پایان‌نامه، حالت دسته‌بندی دودویی یعنی $M = 2$ را در نظر می‌گیریم. در این حالت، Y مقادیرش را از مجموعه دودویی $\{-1,1\}$ می‌گیرد. دسته‌بندی دودویی می‌تواند به دسته‌بندی چندکلاسی ($M > 2$) در یک تنظیم یک در مقابل همه (OVA)[1] گسترش یابد [۴۳]. قبل از کاوش دقیق موضوع دسته‌بندی، مفهوم مهمی بنام احتمال پسین را در این زمینه معرفی می‌کنیم.

۱.۲.۲ احتمال پسین و دسته‌بندی کننده‌ی جانشین

در آنالیز رگرسیون، هدف ارزیابی Y برای X مفروض با r(X) است، که در آن $r : R^p \to R$ یک تابع می‌باشد. می‌توان نشان داد که، تابعی که خطای مربع میانگین را در این چارچوب کمینه می‌سازد، احتمال پسین η است

$$\eta(\mathrm{x}) = \mathrm{P}\{Y = 1 \mid X = \mathrm{x}\} = \mathrm{E}\{Y \mid X = \mathrm{x}\}. \qquad (۴.۲)$$

برای همه، $r : R^p \to R$ داریم

---

[1] One-Versus-All

$$\mathrm{E}\{(\eta(X)-Y)^2\} \le \mathrm{E}\{(r(X)-Y)^2\} \qquad (۵.۲)$$

اثبات. برای هر $x \in R^p$ داریم

$$\begin{aligned} &\mathrm{E}\{(r(X)-Y)^2 \mid X=\mathrm{x}\} \\ &\quad = \mathrm{E}\{(r(X)-\eta(X)+\eta(X)-Y)^2 \mid X=\mathrm{x}\} \\ &\quad = (r(\mathrm{x})-\eta(\mathrm{x}))^2 + 2(r(\mathrm{x})-\eta(\mathrm{x}))\mathrm{E}\{\eta(X)-Y \mid X=\mathrm{x}\} + \mathrm{E}\{\eta(X)-Y)^2 \mid X=\mathrm{x}\} \\ &\quad = (r(\mathrm{x})-\eta(\mathrm{x}))^2 + \mathrm{E}\{\eta(X)-Y)^2 \mid X=\mathrm{x}\}. \end{aligned} \qquad (۶.۲)$$

توجه داشته باشید که تساوی در (۵,۲) برقرار است، اگر و تنها اگر برای همه $x \in R^p$ داشته باشیم

$$r(x) = \eta(x)$$

اهمیت احتمال پسین در این است که با یک η داده شده، می‌توان یک دسته‌بندی کننده را با احتمال کمینه خطا ساخت. دسته‌بندی کننده $g^*: R^p \to \{-1,1\}$ را با استفاده از تابع رگرسیون η به‌صورت

$$g^*(x) = \begin{cases} -1 & \text{if } \eta(\mathrm{x}) \le \dfrac{1}{2} \\ 1 & \text{otherwise.} \end{cases} \qquad (۷.۲)$$

تعریف می‌کنیم، ادعا این است که، g^* یک دسته‌بندی کننده بیزی است، یعنی احتمال خطا را کمینه می‌سازد. برای اثبات این مطلب، باید نشان دهیم که برای هر دسته‌بندی کننده

$$\mathrm{E}\{g^*(\eta(X) \ne Y)^2\} \le \mathrm{P}\{g(X) \ne Y\}. \qquad (۸.۲)$$

اثبات. برای $X = x$ داده شده، احتمال شرطی خطای دسته‌بندی کننده g عبارت است از

$$\begin{aligned} &\mathrm{P}\{g(X) \ne Y \mid X=\mathrm{x}\} \\ &\quad = 1-\mathrm{P}\{g(X)=Y \mid X=\mathrm{x}\} \\ &\quad = 1-(\mathrm{P}\{g(X)=1,\ Y=1 \mid X=\mathrm{x}\} + \mathrm{P}\{g(X)=-1,\ Y=-1 \mid X=\mathrm{x}\}) \\ &\quad = 1-\left(\chi_{\{g(\mathrm{x})=1\}}\mathrm{P}\{Y=1 \mid X=\mathrm{x}\} + \chi_{\{g(\mathrm{x})=-1\}}\mathrm{P}\{Y=-1 \mid X=\mathrm{x}\}\right) \\ &\quad = 1-\left(\chi_{\{g(\mathrm{x})=1\}}\eta(\mathrm{x}) + \chi_{\{g(\mathrm{x})=-1\}}(1-\eta(\mathrm{x}))\right) \end{aligned} \qquad (۹.۲)$$

که در آن χ_A تابع شاخص مجموعه A است. برای هر $x \in R^p$ می‌توانیم بنویسیم

$$\begin{aligned}&\mathrm{P}\{g(X)\neq Y \mid X=\mathrm{x}\}-\mathrm{P}\{g^*(X)\neq Y \mid X=\mathrm{x}\}\\&=\eta(\mathrm{x})\left(\chi_{\{g^*(\mathrm{x})=1\}}-\chi_{\{g(\mathrm{x})=1\}}\right)+(1-\eta(\mathrm{x}))\left(\chi_{\{g^*(\mathrm{x})=-1\}}-\chi_{\{g(\mathrm{x})=-1\}}\right)\\&=(2\eta(\mathrm{x})-1)\left(\chi_{\{g^*(\mathrm{x})=1\}}-\chi_{\{g(\mathrm{x})=1\}}\right)\geq 0\end{aligned} \quad (۱۰.۲)$$

که آخرین تساوی برقرار است، زیرا

$$\begin{aligned}&\chi_{\{g^*(\mathrm{x})=-1\}}=1-\chi_{\{g^*(\mathrm{x})=1\}}\\&\chi_{\{g(\mathrm{x})=-1\}}=1-\chi_{\{g(\mathrm{x})=1\}}.\end{aligned} \quad (۱۱.۲)$$

با انتگرال‌گیری نسبت به x می‌توانیم به (۸,۲) برسیم.

تابع $\eta(x)$ ناشناخته است. برای تقریب دسته‌بندی کننده بیزی، ما تابع نامنفی $\tilde{\eta}(x)$ را بـه عنـوان تقریبی از $\eta(x)$ بکار گرفته و آن را در شکل دسته‌بندی کننده در معادله (۱۸,۲) جایگذاری می‌کنیم

$$g(x)=\begin{cases}-1 & \text{if } \tilde{\eta}(\mathrm{x})\leq \frac{1}{2}\\ 1 & \text{otherwise.}\end{cases} \quad (۱۲.۲)$$

دسته‌بندی کننده g، دسته‌بندی کننده‌ی جانشین نامیده می‌شود. دسته‌بندی کننده نشانده بخـوبی عمل می‌کند. اگر $\tilde{\eta}(x)$ نزدیک به $\eta(x)$ باشد- که با L_1 نرم مورد انتظار، اندازه‌گیری می‌شـود- سـپس احتمال خطای دسته‌بندی کننده نشانده نزدیک به خطای بیزی است، یعنی

$$\mathrm{P}\{g(X)\neq Y\}-L^*\leq 2\mathrm{E}\{|\eta(X)-\tilde{\eta}(X)|\}. \quad (۱۳.۲)$$

اثبات. توجه داشته باشید که تمایز بین احتمالات خطای شرطی g و g^* زمـانی‌کـه بـرای هـر $x\in R^p$، $g(x)=g^*(x)$ باشد، صفر است. زمانی‌که $g(x)\neq g^*(x)$ باشد، براسـاس (۱۰,۲)، مـی‌تـوانیم تمـایز را به‌صورت

$$\begin{aligned}&\mathrm{P}\{g(X)\neq Y \mid X=x\}-\mathrm{P}\{g^*(X)\neq Y \mid X=x\}\\&=(2\eta(x)-1)\left(\chi_{\{g^*(\mathrm{x})=1\}}-\chi_{\{g(\mathrm{x})=1\}}\right)\\&=|2\eta(x)-1|\chi_{\{g(x)\neq g^*(x)\}}\\&=2|\eta(x)-1/2|\chi_{\{g(x)\neq g^*(x)\}}\\&\leq 2|\eta(x)-\tilde{\eta}(x)|.\end{aligned} \quad (۱۴.۲)$$

بنویسیم، نامساوی اخیر برقرار است، زیرا $g(x)\neq g^*(x)$ ایجاب می‌کند

$$|\eta(x)-\tilde{\eta}(x)| \ge |\eta(x)-1/2|. \qquad (۱۵.۲)$$

برای بررسی درستی این مطلب، هر حالت را به ترتیب، بررسی می‌کنیم

$$\text{if } g^*(x)=-1 \Rightarrow \tilde{\eta}(x)\le 1/2 \text{ and } g(x)=1 \text{ so } \tilde{\eta}(x)>1/2$$
$$|\eta(x)-\tilde{\eta}(x)| = \tilde{\eta}(x)-\eta(x) \le 1/2-\eta(x) = |1/2-\eta(x)|$$
$$\text{if } g^*(x)=1 \Rightarrow \eta(x)>1/2 \text{ and } g(x)=-1 \text{ so } \tilde{\eta}(x)\le 1/2$$
$$|\eta(x)-\tilde{\eta}(x)| = \eta(x)-\tilde{\eta}(x) \le \eta(x)-1/2 = |1/2-\eta(x)|.$$

بر می‌گردیم به اثبات، می‌توانیم با انتگرال‌گیری طرفین نامساوی در (۱۴,۲) نسبت به x به

$$P\{g(X)\ne Y\}-L^* \le 2E\{|\eta(X)-\tilde{\eta}(X)|\}$$

برسیم.

این نتیجه نشان می‌دهد که، یک ارزیاب خوب از η می‌تواند یک دسته‌بندی کننده نشانده خوب را تولید کند. آنچه که روشن است این است که، $\tilde{\eta}(x)$ می‌تواند دور از $\eta(x)$ باشد و دسته‌بندی کننده یکسانی را بدست آورد، همراه با این که هر دو روی طرف یکسانی از ½ باشد. ما اکنون، حاضریم روی آموزش بر راه‌هایی جهت دست یابی به دسته‌بندی کننده خوب ، تمرکز کنیم.

۲.۲.۲ کمینه‌سازی ریسک آزمایشی

کمینه‌سازی احتمال خطا (احتمال دسته‌بندی نادرست)

$$L(g)=\mathrm{E}\{\chi_{\{g(X)\ne Y\}} = \mathrm{P}\{g(X)\ne Y\} \qquad (۱۶.۲)$$

تنها با دانستن توزیع توامان X و Y ممکن است. یک ارزیاب احتمال خطا از یک دسته‌بندی کننده g با T_n داده شده، شمارش خطای متوسط زیر است

$$L_n(g)=\frac{1}{n}\sum_{i=1}^{n}\chi_{\{g(x_i)\ne y_i\}}. \qquad (۱۷.۲)$$

ارزیاب $L_n(g)$ خطای تجربی g است. برای یک بررسی کلی از پیشرفت‌های اخیر [۵] را ببینید.

می‌توانیم به رهیافت کنونی دسته‌بندی با در نظر گرفتن یک کلاس C از دسته‌بندی کننده‌های $g^*: R^p \to \{-1,1\}$ برسیم. با T_n داده شده، دسته‌بندی کننده در C را انتخاب می‌کنیم که، منجر به کمترین خطای تجربی $L_n(g)$ می‌شود. با این‌حال، مساله کمینه‌سازی خطای تجربی، از نظر محاسباتی، بسیار دشوار است. در مواجه با این مساله، نیازمند اصلاح عملکرد برای کمینه‌سازی هستیم. دسته‌بندی کننده‌هایی به شکل زیر را در نظر می‌گیریم

$$g_f(x) = \begin{cases} -1 & \text{if } f(x) < 0 \\ 1 & \text{otherwise} \end{cases} \qquad (۱۸.۲)$$

که در آن $f: R^p \to R$ یک تابع حقیقی مقدار در F است. احتمال خطای g_f به‌صورت

$$\begin{aligned} Lg_f = L(f) &= \mathrm{P}\{\mathrm{sgn}(f(X)) \neq Y\} \\ &= \mathrm{P}\{Y\, f(X) \leq 0\} \\ &= \mathrm{E}\{\chi_{Y\, f(X) \leq 0\}}\}. \end{aligned} \qquad (۱۹.۲)$$

است، کمیت $yf(x)$ حاشیه[1] نامیده می‌شود و زمینه تکراری در باقی این بخش دارد. با T_n داده شده، می‌توان $L(f)$ را با $L_n(f)$ ارزیابی نمود

$$L_n(f) = \frac{1}{n}\sum_{i=1}^{n} \chi_{\{y_i f(x_i) \leq 0\}} \qquad (۲۰.۲)$$

که در آن $\chi_{\{yf(x)\leq 0\}}$ تابع زیان ۰ تا ۱ است. همان‌طور که قبلا توضیح داده شد، کمینه‌سازی خطای تجربی از نظر محاسباتی، دشوار است [۲]. در عوض، به‌دنبال کمینه‌سازی یک کران بالاتر محدب هموار از زیان ۰ تا ۱، یعنی $\chi_{\{yf(x)\leq 0\}}$ هستیم. تابع محدب هموار Φ از حاشیه $v = yf(x)$ تابع هزینه نامیده می‌شود.

[1] Margin

جدول ۱.۲ توابع زیان محدب شناخته شده و تابع کمینه‌سازی متناظرشان.

نام تابع زیان	$\phi(v)$	$f_{\phi}^{*}(\eta)$
زیان مربعی	$(1-v)^2$	$2\eta-1$
زیان محوری	$\max(0,1-v)$	$sign(2\eta-1)$
زیان محوری مربعی	$\max(0,1-v)^2$	$2\eta-1$
زیان لجستیکی	$\ln(1+\exp(-v))$	$\ln\frac{\eta}{1-\eta}$

مثال‌ها شامل تابع زیان نمایی $\phi(v)=\exp(-v)$ بکار رفته در AdaBoost [۱۸] و تابع زیان محوری $\phi(v)=\max(0,1-v)$ استفاده شده در SVM [۴] است. تابعک هزینه عبارت است از

$$A(f)=\mathrm{E}\{\phi(Yf(X))\} \tag{۲۱.۲}$$

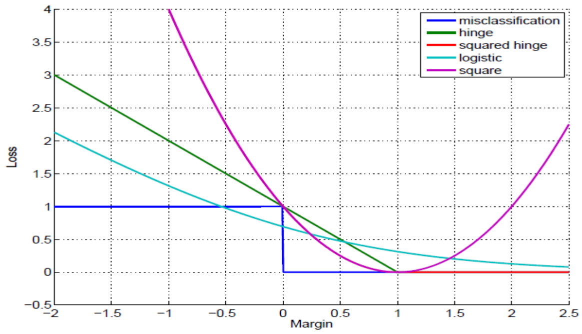
و فرم آزمایشی متناظر نیز به‌صورت

$$A_n(f)=\frac{1}{n}\sum_{i=1}^{n}\phi(y_i f(x_i)). \tag{۲۲.۲}$$

است، توجه داشته باشید که، ϕ یک کران بالاتر روی زیان ۰ تا ۱ است و بنابراین، $L_n(f)\le A_n(f)$ و $L(f)\le A(f)$.

اگر F شامل توابعی باشد که در پارامترهایشان خطی هستند- بنابراین، آنها را محدب می‌سازیم و سپس کمینه‌سازی هزینه تجربی $A_n(f)$، مساله بهینه‌سازی محدب است. پس، الگوریتم‌هایی موثر برای دستیابی به کمینه‌سازی $A_n(f)$ بر $f\in F$ وجود دارد.

توابع زیان مورد توجه ما در جدول ۱,۲ فهرست شده و در شکل ۲,۲ نشان داده شده‌اند. توابع زیان حقیقی مقدار، به عنوان جانشین هایی برای زیان ۰-۱ عمل می‌کنند. این منجر به یک مساله رگرسیون می‌شود. نتیجه برای دسته‌بندی با آستانه خروجی تابع حاصل از رگرسیون، بدست می‌آید. در ادامه، به‌سراغ این واقعیت می‌رویم که، بهینه‌سازی یک جانشین محدب منجر به دسته‌بندی کننده بیزی می‌شود.



شکل ۲.۲ مقایسه‌ای از توابع زیان محدب. زیان دسته‌بندی نادرست نیز نشان داده شده است.

کمینه‌سازی یک کران بالای محدب از زیان ۰-۱، نه تنها منجر به دسته‌بندی قابل قبولی می‌شود، بلکه کار موفقی است. موفقیت های آزمایشی SVM، و تقویت ها، این نقطه را ایجاب می‌نمایند. می‌توانیم نشان دهیم که کمینه ساز f^* از $A_n(f)$ به‌گونه‌ای است که، دسته‌بندی کننده القایی g_f^* یک دسته‌بندی کننده بیزی است. برای نیل به این هدف، نگاه نزدیک‌تری به تابعک هزینه می‌اندازیم

$$\begin{aligned} A(f) &= \mathrm{E}\{\phi(Yf(X))\} \\ &= \mathrm{E}\{\eta(X)\phi(f(X)) + (1-\eta(X))\phi(-f(X))\} \end{aligned} \tag{۲۳.۲}$$

که در آن، $\eta(x)$ نشان‌دهنده احتمال پسین $P\{Y=1 \mid X=x\}$ است. تعریف زیر را در نظر بگیرید

$$A(f,\eta) = \eta\phi(f) + (1-\eta)\phi(-f) \tag{۲۴.۲}$$

که در آن، $\eta \in [0,1], f \in R$ است. فرض کنید که تابع $f_\phi^* : [0,1] \to R$ کمینه ساز $A(f,\eta)$ باشد

$$f_\phi^*(\eta) = \underset{f \in \mathrm{R}}{\arg\min}\ A(f,\eta). \tag{۲۵.۲}$$

با تعریف f_ϕ^* از بالا، واضح است که $f_\phi^*(\eta(x))$ مقدار $A(f(x))$ را در میان همه توابع $f(x)$ در (۲۱,۲) کمینه می‌سازد. با یک تابع زیان محدب مفروض ϕ، کمینه ساز بهینه f_ϕ^* به آسانی قابل محاسبه است. در ستون سوم از جدول ۲,۱، f_ϕ^* بهینه را برای توابع زیان مطلوب، لیست کرده‌ایم. در این مثال‌ها، به آسانی می‌توان دید که، $f_\phi^*(\eta(x)) > 0$ تنها زمانی‌که $\eta > 1/2$ باشد، برقرار است. اگر فرض کنیم $f(x) = f_\phi^*(\eta(x))$ باشد، با کمینه‌سازی (۲۱,۲)، دسته‌بندی کننده القایی

$$g^*(x) = \begin{cases} -1 & if\ \ f(x)^*(x) < 0 \\ 1 & \quad Otherwise \end{cases}$$

دارای علامتی یکسان با دسته‌بندی کننده بیزی است. این به ما اجازه می‌دهد تا نتیجه بگیریم که، کمینه ساز تابعک هزینه $A(f)$، کران بالایی روی خطای دسته‌بندی درست $L(f)$ است، به‌گونه‌ای که دسته‌بندی کننده القایی g_f یک دسته‌بندی کننده بیزی است، و این سازگاری فیشر توابع هزینه محدب را اثبات می‌کند [۳، ۴، ۵، ۳۲، ۶۵]. توجه داشته باشید که، سازگاری فیشر در این زمینه، به عنوان دسته‌بندی کننده بیزی یعنی $sign(\eta-1/2)$ است.

۳.۲.۲ ارزیابی احتمال پسین

یک مشاهده مهم براساس جدول ۲,۱ این است که، SVM بطور مستقیم دسته‌بندی کننده دودویی را ارزیابی می‌نماید. یعنی، SVM مستقیما $sign(2\eta(x)-1)$ را بجای احتمال پسین $\eta(x)$ ارزیابی می‌کند. این بدان معناست که، SVM نمی‌تواند اطلاعاتی درباره اطمینان از پیش‌بینی به ما بدهد. چنین اطلاعاتی بویژه، برای دسته‌بندی چندکلاسه با استفاده از دسته‌بندی کننده دودویی در تنظیمات یکی در مقابل همه، مفید است. در شرایطی که، احتمالات پسین برای همه کلاس ها زیر ½ است، SVM نمی‌تواند تصمیمات صحیحی بگیرد.

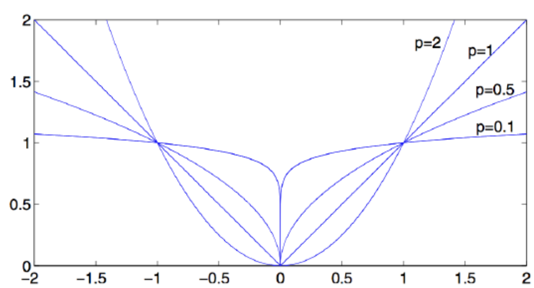
از سوی دیگر، دسته‌بندی کننده کمترین مربعات، احتمال پسین $\eta(x)$ را ارزیابی نموده و بنابراین می‌تواند اطلاعاتی مطمئن را ارائه نماید. برای دسته‌بندی کننده کمترین مربعات، احتمال پسین برابر با $(f(x)+1)/2$ محدود شده به $[0,1]$ است. برای آنالیز عمیق‌تر این موضوع [۶۴] را ببینید.

جدول ۲.۲ مقایسه‌ای از سه دسته‌بندی کننده دیگر با دسته‌بندی لسویی روی ۷ مجموعه داده. فراپارامتر با C یا λ بسته به الگوریتم نشان داده می‌شود. تعداد درآیه‌های ناصفر در بردار جواب نیز با nz# و درصد نمونه‌های آزمون به‌درستی دسته‌بندی شده با Acc نمایش داده می‌شود.

مجموعه داده‌ها	lasso			SVM			ℓ_1-reg L2SVM			ℓ_1-reg logreg		
	λ	دقت	#nz	C	دقت	#nz	C	دقت	#nz	C	دقت	#nz
استرالیا	۱۰	۸۶	۱۴	۲	۸۶	۶۸	۲۰	۸۶	۱۴	۵	۸۷	۱۴
سرطان کولون	۱	۷۷	۱۶	۱	۸۷	۱۱	۱۰	۷۵	۱۱۲	۱۰	۸۳	۹۱
دیابت	۱۰	۸۰	۸	۱	۷۵	۱۰۵	۱۰	۷۷	۸	۱۰	۷۶	۸
قلب	۶	۸۷	۱۳	۱/۵	۸۴	۳۰	۱۰	۸۰	۱۳	۵	۸۳	۱۳
یونکره	۱	۷۷	۱۶	۱	۸۷	۱۱	۱۰	۷۵	۲۸	۲	۸۲	۳۱
تجویزات کبدی	۲	۴۶	۶	۲	۶۲	۷۰	۵	۶۶	۶	۵	۶۷	۶
قارچ ها	۲	۴۸	۱۳	۲	۱۰۰	۹۰	۱۰	۱۰۰	۹۶	۲۰	۱۰۰	۹۵

۳.۲ چرا منظم‌سازی ℓ_1 بردار وزن تنک ایجاد می‌کند؟

شکل ۳,۲، شهود مناسبی را برای این که چرا منظم‌سازی ℓ_1 تنک بودن را در بردار وزن القـا مـی‌کنـد، ارائه می‌دهد.



شکل ۳.۲ همان‌طور که p به صفر می‌رود، $|x|^p$ به تابع شاخص تبدیل می‌شود و تعداد درآیه‌های ناصفر در x را می‌شمارد [۶].

۴.۲ ارزیابی تجربی لسو برای دسته‌بندی

انواع گوناگونی از الگوریتم‌ها برای حل مساله بهینه‌سازی کمترین مربعات ℓ_1 –منظم شده وجود دارد [۳۳]. در این پایان‌نامه، یکی از موفق‌ترین این الگوریتم‌ها را بکار می‌گیریم که، قادر به حل مساله کمینه‌سازی زیان مربعات ℓ_1 –منظم شده، به‌صورت بهینه است. این الگوریتم، روش تصویری طیفی برای ℓ_1 –کمینه‌سازی (SPGL1)[1] [۵۲] است و مساله بهینه‌سازی زیر (که لسو نام دارد) را حل می‌کند

$$\min_{a} \|y - Xa\|_2^2 \ \text{s.t.} \ \|a\|_1 \le \lambda. \qquad (۱.۲)$$

همان‌طور که پیشتر اشاره کردیم، یکی از این دلایل، کنترل صریح روی تنک بودن جواب است. در شکل ۱,۲، ما نشان می‌دهیم که، می‌توان تنک بودن جواب را با استفاده از پارامتر منظم‌سازی λ در فرمول لسوی بالا، کنترل کرد. دلیل مهم دیگر، این است که، زیان مربعی دارای مزایای محاسباتی و آماری فراوانی است. همچنین برتری آماری زیان مربعی بر زیان محوری را بیان می‌کند.

در جدول ۲,۲، مقایسه‌ای از روش‌های گوناگون برای دسته‌بندی دودویی با لسو را می‌بینیم. می‌بینیم که لسو قادر به دستیابی به تنک‌ترین جواب‌ها در مسایل بعد بالاست ،درحالی‌که دقت قابل قبولی را حفظ می‌نماید. در جدول ۳,۲، نتایج را برای رگرسیون مرزی بکار رفته به عنوان دسته‌بندی کننده می‌بینیم.

[1] Spectral Projected Gradient Method for ℓ_1-minimization

جدول ۳.۲ نتایج دقت دسته‌بندهی کننده برای دسته‌بندی براساس رگرسیون با تنظیم‌سازی ℓ_2. در اینجا λ فراپارامتر بهینه است که با اعتبارسنجی متقابل به دست آوردیم. توجه داشته باشید که، تعداد درآیه‌های ناصفر جواب (nz) برابر با تعداد ویژگی‌های مشاهدات است.

مجموعه داده‌ها	فراپارامتر λ	دقت دسته‌بندی (٪)	تعداد nz
استرالیایی	۳۰	۸۶	۱۴
سرطان کولون	۶	۸۷	۲۰۰۰
دیابت	۴۰	۷۶	۸
قلب	۹	۸۶	۱۳
یون کره	۱۰	۷۴	۳۴
تجویزات کبدی	۸	۳۵	۶
قارچ ها	۲۰	۴۹	۱۱۲

۳

فصل سوم

کمینه‌سازی زیان مربعی ℓ_1-منظم برای بازسازی

طرح‌های تقریب تنک برای نمایش سیگنال و ویژگی (یعنی بازسازی)، بسیار مفید شناخته شده‌اند. برای مثال، الگوریتم دسته‌بندی چند کلاسه موفقی برمبنای نمایش‌های پراکنده وجود دارد [۵۵]. در این فصل، به کارایی بازسازی (در تعریف غیر نظارتی) الگوریتم‌های کمینه‌سازی زیان مربعی ℓ_1-منظم شده نگاهی می‌اندازیم.

همان گونه که در فصل ۲ دیدیم، ℓ_1-منظم‌سازی برای تعمیم به داده‌های دیده نشده، بخوبی ℓ_2-منظم‌سازی نیست. با این‌حال، همان‌طور که آزمایش‌های در تقریب تنک نشان می‌دهد [۱۲]،[۵۵]، [۶۱]، ℓ_1-منظم‌سازی برای نمایش ویژگی، کاملا مناسب است. چه چیزی روش‌های تقریب تنک را در یادگیری و نمایش ویژگی، موفق می‌سازد؟ آیا موفقیت برخاسته از تنک نمایش است یا این واقعیت که، آن داده‌ها را بسیار خوب برازش می‌نماید؟ آنالیزهایی روی پایداری الگوریتم‌های ℓ_1-منظم شده صورت پذیرفته است [۴۰]، [۶۰].

در ادامه این فصل، به سراغ کاربرد کمینه‌سازی مربعی ℓ_1-منظم شده برای یادگیری ویژگی در یک کار دسته‌بندی تصویر می‌رویم. بویژه به کار [۱۲] می پردازیم. سپس، الگوریتم دسته‌بندی را با روش جدیدمان برای رگرسیون که توسیعی از SRC به رگرسیون است، بررسی می‌کنیم. درباره این بحث می‌کنیم که، این روش‌ها بهبودی را نسبت به روش‌های موثری نظیر k-میانگین ها و kNN ارائه نمی‌کنند.

۱.۳ کدگذاری تنک و یادگیری فرهنگ لغات برای یادگیری ویژگی

برای بیان اهمیت کمینه‌سازی زیان مربعی ℓ_1-منظم شده برای کدگذاری تنک، به سراغ مقاله [۱۲] می‌رویم. در این مقاله، اهمیت کدگذاری در مقابل یادگیری با کدگذاری تنک و کمی‌سازی برداری، با آموزش، مولفان یادگیری فرهنگ لغت D را معنی کرده و با کدگذاری، آنها منظور از تصویر کردن (نگاشتن) ورودی x به ویژگی f با فرهنگ لغت مفروض D را روشن می‌سازند. مولفان بحث می‌کنند که، اگر ما قادر به شکستن هر روش یادگیری ویژگی به مسیرهایی شامل آموزش و کدگذاری باشیم، می‌توانیم مسیرهای فرعی را از روش‌های یادگیری ویژگی ترکیب کرده و الگوریتم‌های موثرتری را بدون فداکردن کارایی در دسته‌بندی، بدست آوریم. آنها نتایج دسته‌بندی را با استفاده از اعتبار بخشی گذری ۵ بخشی با SVM خطی روی CIFAR-10، NOTB و Caltech 101 (۸۱/۵ ٪، ۹۵ ٪ و ۷۲/۶ ٪،

دوتای اولی نتایج مدرن هستند) گزارش می‌کنند [۱۲]. مولفان، با ۶ روش برای بکارگیری فرهنگ لغت و چندین روش برای کدگذاری با استفاده از فرهنگ لغت، آزمایش می‌کنند. در این جـا، روش‌هـایی بـرای یادگیری/ساکن کردن فرهنگ لغت وجود دارنـد (مـا در جسـتجوی فرهنـگ لغـت $D \in R^{n \times d}$ هسـتیم، بگونه‌ای که هر اتم (ستون) دارای یک ℓ_2-نرم واحد باشد):

۱. کدگذاری تنک[1] (SC) با کاهش مختصاتی:

$$\min_{D,S^{(i)}} \sum \| Ds^{(i)} - x^{(i)} \|_2^2 + \lambda \| s^{(i)} \|_1$$

راهی برای حل این مساله بهینه‌سازی، جایگزین کـردن کمینـه‌سـازی بـین فرهنـگ لغـت D و کدهای تنک $\{s^{(i)}\}$ است (یکی ثابت نگـه داشـته مـی‌شـود، درحـالی‌کـه، تـابع هـدف کمینـه می‌گردد، و به همین ترتیب). مولفان پارامتر λ را با کمینه‌سازی خطای اعتبار گذریش بـر یـک شبکه مقادیر کاندید، بدست می‌آورند.

۲. تعاقب تطابق متعامد[2] (OMP-k): به‌صورت

$$\min_{D,s^{(i)}} \sum_i \left\| Ds^{(i)} - x^{(i)} \right\|_2^2 \qquad (۱.۳)$$

و

$$\text{subject to } \left\| s^{(i)} \right\|_0 \le k, \forall i, \qquad (۲.۳)$$

است، که در آن، k کرانی پایین روی تعداد عناصر ناصفر در $s^{(i)}$ اسـت. بـرای حـل ایـن مسـاله بهینه‌سازی، می‌توان بین D و $\{s^{(i)}\}$ دقیقا مانند بالا، جایگذاری کرد.

- نزول مختصاتی و OMP الگوریتم‌هایی برای دستیابی به کدهای تنـک یـک فرهنـگ لغـت مفروض هستند، فرهنگ لغت از سوی دیگر می‌تواند با استفاده از کـاهش گرادیـان بدسـت آید.
- با بهینه‌سازی (۱.۳) و (۲.۳)، شما کدهای تنک را به عنوان نتیجه فرعی یـادگیری فرهنـگ لغت (فازهای یادگیری و کدگذاری در هم تنیده هستند) بدست می‌آورید. با این‌حال، این ما

[1] Sparse coding

[2] Orthogonal Matching Pursuit

را از برقرار ساختن تنها یک فرهنگ لغت بدست آمده در این گام و محاسبه کدها بـا دیگـر ابزارها باز نمی‌دارد.

۳. ماشین بولتزمن تحدیدی تنک[1] (RBM) و کدگذار خودکار تنک.

۴. زیر نمونه‌گیری تصادفی ماتریس داده X شامل $s^{(i)}$ نرمالیزه شده است.

۵. وزن‌های تصادفی: پرکردن فرهنگ لغت بـا نمونـه‌گیـری سـتونی نرمـال شـده از توزیـع نرمـال استاندارد.

و در این جا، روش‌هایی برای کدگذاری وجود دارد؛ SC: مساله بهینه‌سازی در بالا، با D ثابت، λ متفـاوت و تنظیم کردن عناصر ویژگی f

به‌صورت

$$f_j = \max\{0, s_j\} \tag{۳.۳}$$

و

$$f_{j+d} = \max\{0, -s_j\}. \tag{۴.۳}$$

است، توجه داشته باشید که بجای d بعد، ویژگی f دارای بعد $2d$ است. مولفان این را تقسیم قطبیـت[2] می‌نامند.

۱. OMP-k: تعاریف مانند ۱ است.

۲. آستانه‌سازی نرم: برای آستانه ثابت α، آنها f را به‌صورت زیر نسبت می‌دهند،

$$f_j = \max\{0, D^{(j)T}x - \alpha\} \tag{۵.۳}$$

و

$$f_{j+d} = \max\{0, -D^{(j)T}x - \alpha\}. \tag{۶.۳}$$

[1] Sparse restricted Boltzmann machine

[2] Polarity splitting

۳. کدگذاری طبیعی: اگر فرهنگ لغت با SC فراگرفته شود، سپس، کدهای یادگیری بکار گرفته می‌شوند. این بخش برای OMP یکسان است. برای RBM و کدگذار خودکار، می‌توان فعال‌سازی در گره‌های مخفی را با استفاده از تابع حلقوی لجستیکی g محاسبه کرد:

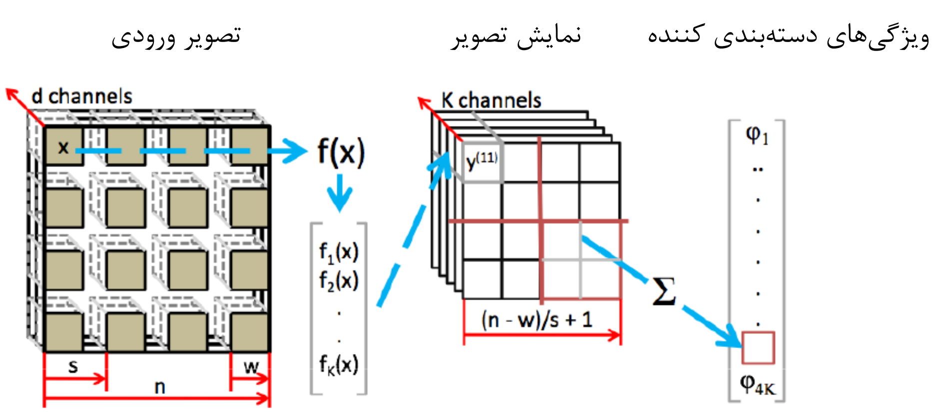
$$f_j = g(W^{(j)}x + b) \qquad (۷.۳)$$

و

$$f_{j+d} = g(-W^{(j)}x + b), \qquad (۸.۳)$$

که در آن $W = D^T$ و W^j ردیف *j*ام از W است. برای (۵.۳) و (۶.۳)، مولفان، فرهنگ لغت را به عنوان نگاشتی خطی بکار می‌برند، یعنی $f = D^T x$ (این مشابه تصویر تصادفی است، بجز این که بجای کاهش بعد، آن را افزایش می‌دهیم، با فرض $d > n$).

مولفان، بهترین نتایج را روی CIFAR-10 با استفاده از OMP-1 برای آموزش و آستانه سازی نرم برای کدگذاری بدست می‌آورند (که نشان‌دهنده فرهنگ لغت وسیع‌تر و بهتری است- *d=600*). آنها بهترین نتیجه را برای NORB با استفاده از وصله‌های تصادفی به عنوان فرهنگ لغت و SC برای کدگذاری، بدست آوردند. نتایج یکسانی برای Caltech 101 بدست می‌آید، هر چند این نتایج به میزان ۳/۱ ٪ حالت جدید را دنبال می‌کند.



شکل ۱.۳ مراحل دسته‌بندی تصویر [۱۲].

در اینجا، استفاده مولفان از مجموعه داده‌ها در فاز یادگیری بدون نظارت را بررسی می‌کنیم. در حالت CIFAR و NORB، آنها تنظیم کردند که $x^{(i)} \in R^n$ که به‌صورت تصادفی انتخاب شده، نرمال شود و در وصله‌های $3\times(6\times6)$ بردارسازی گردد. بمانند Caltech 101، $s^{(i)}$ یک توصیف گر SIFT با ۱۲۸ بعد استخراج شده از هر وصله 16×16 تصادفی است. قبل از ارسال به الگوریتم آموزش فرهنگ لغت، آنها ZCA-سفیدسازی را روی تمام مجموعه داده $X = [x^{(i)},..., x^{(1600)}]$ ارسال می‌کنند.

با فرض نگاشت ویژگی پارامتری شده با D، به بررسی خطوط لوله‌ای می پردازیم که، مولفان برای اجرای دسته‌بندی راه‌اندازی می‌کنند . نخست، آنها وصله‌های $\{s^{(i)}\}$ را (با اندازه مشخص شده در بالا) با یک جابجایی یک پیکسلی برای CIFAR-10 و NORB و ۸ پیکسل برای Caltech 101 استخراج می‌کنند تا تمام تصویر را پوشش دهند. برای CIFAR-10 و NORB، $s^{(i)}$ها مقادیر پیکسل خام برای وصله هستند، درحالی‌که برای Caltech 101، آنها مقادیر توصیفگر SIFT تکی استخراج شده از وصله می‌باشند. برای هر زوج از روش‌های آزمایشی/کدگذاری، مولفان از فرهنگ لغت D برای رسیدن به ویژگی $f^{(i)}$ برای هر $s^{(i)}$ استفاده می‌کنند. برای مثال، برای هر تصویر 32×32 پیکسلی در مجموعه داده CIFAR-10، ما (در تنظیمات مشخص داده شده) بعد $27\times27\times1600\times2=2332800$ را دریافت می‌کنیم! برای کاهش بعد فضای ویژگی، یک گام تجمیع اجرا می‌شود (که برای هر مجموعه داده متفاوت است):

- CIFAR-10: مولفان مقادیر ویژگی را بر چهار ربع تصویر میانگین می‌گیرند، تا به بردار ویژگی نهایی نشان‌دهنده آن تصویر برسند (با این کار ما به بعد $4\times1600\times2$ می‌رسیم).
- NORB: مولفان دو مرحله از نمونه‌گیری پایین را روی تصاویر 108×108 اصلی قبل از استخراج وصله‌ها، اجرا می‌کنند. همچنین به این نکته اشاره نمی‌کنند که، استراتژی تجمیع آنها پس از نگاشت ویژگی اجرا می‌شود.
- Caltech 101: در اینجا، مولفان تجمیع هرمی مکانی را اجرا می‌کنند، یعنی آنها تجمیع بیشینه را روی ویژگی‌هایی بر شبکه های 4×4، سپس 2×2 و بعد 1×1 در یک نظام سلسله مراتبی اجرا می‌نمایند. آنها نتایج را برای تشکیل بردار ویژگی نهایی نشان‌دهنده تصویر، الحاق می‌کنند.

اکنون، برای در اختیار داشتن بردار ویژگی تکی برای هر مجموعه آزمون و آموزش، مولفان یک SVM خطی را برای دسته‌بندی تمرین می‌دهند.

به‌نظر ما نتایج این مقاله جالب است، اما چندان شگفت انگیز نیست. طرح PCA و تصادفی را در نظر بگیرید (هر چند مانند روش‌های بکار رفته در این مقاله، منجر به فرهنگ لغات وسیعی نمی‌شوند). می‌دانیم که (در پاره ای امور) وزن‌های متعامد یکه تصادفی با بخش های آگاه از داده و یادگرفته شده یعنی مولفه‌های اصلی، قابل مقایسه هستند. نتایج همچنین، توضیح می‌دهند که چرا، الگوریتم k-میانگین در مقاله پیشین آنها [۱۳] -بسیار دورتر از طرح یادگیری اتم‌های فرهنگ لغت- بکار گرفته شده است.

۲.۳ دسته‌بندی بازنمایی تنک

برای ارائه پیش زمینه روشمان برای رگرسیون در بخش بعدی، گذری بر دسته‌بندی بازنمایی تنک خواهیم داشت [۵۵]. دسته‌بندی بازنمایی تنک[1] (SRC) یک روش دسته‌بندی چندکلاسه است. SRC نیازمند این است که هر نمونه آزمونی با تنها چندین نمونه آزمایشی بازسازی شود. این با جستجوی بازنمایی تنک هر نمونه آزمون نسبت به فرهنگ لغت نمونه‌های آزمایشی، میسر می‌شود. در SRC، هر نمونه به یک وزن منتسب می‌شود که، برای درجه مشارکتش در بازسازی یک نمونه آزمون بحساب می‌آید. این اطلاعات امکان تصمیمی مطلع تر روی کلاس یک نمونه آزمون را فراهم می‌سازند.

مجموعه آزمایشی $\{(x_i, y_i): i = 1,\dots,n\}$ را در نظر بگیرید که در آن، $x_i \in R^p$ یک بردار ویژگی آموزشی و $y_i \in \{1,\dots,C\}$ برچسب کلاس متناظر آن است. بردارهای ویژگی $x_i, i = 1,\dots,n$ ستون‌های فرهنگ لغت $D \in R^{p\times n}$ را می‌سازند. فرض می‌کنیم که، فرهنگ لغت بیش از اندازه کامل باشد ($p < n$)، در غیر این صورت، نخست کاهش بعد را روی ویژگی‌ها اجرا می‌کنیم. علاوه بر این، فرض می‌کنیم که هر ستون به نرم واحد، نرمال شده است. برای هر نقطه آزمون x، یک نمایش پراکنده نسبت به فرهنگ لغت نقاط آزمایشی را جستجو می‌کنیم. حل مساله بهینه‌سازی زیر، منجر به نمایش a برای نقطه آزمون x می‌شود،

[1] Sparse Representation Classification

$$\min_{a \in \mathbb{R}^n} \frac{1}{2}\|\mathrm{x} - \mathrm{Da}\|_2^2 + \lambda\|\mathrm{a}\|_1. \tag{۹.۳}$$

براساس مقادیر در بردار a، یک تصمیم می‌تواند درباره برچسب کلاس x ساخته شود. فرض کنید $\mathrm{I}_c \in \{1,\ldots,n\}$ اندیس‌های ستون‌های D باشد که، متناظر با نقاط متعلق به کلاس c هستند. علاوه بر این، فرض کنید بردار $\delta_c(a)$ برابر با a در اندیس‌های I_c و در غیر این صورت صفر باشد. یک نمونه آزمون x به کلاسی یکسان به عنوان نمونه‌هایی که ترکیب خطی آنها بهترین بازسازی از x در معنای کمترین مربعات است، نسبت داده می‌شود، یعنی

$$\hat{c} = \arg\min_{a \in \{1,\ldots,C\}} \|\mathrm{x} - \mathrm{D}\delta_c(\mathrm{a})\|_2. \tag{۱۰.۳}$$

در تعریف رگرسیون، خروجی‌های $y_i, i = 1,\ldots,n$ اعدادی حقیقی هستند. فرض کنید $\{y_1,\ldots,y_n\}$ برداری شامل خروجی‌های متناظر برای همه نمونه‌های آزمایشی در D باشد. برای یک نمونه آزمون مفروض x، پس از حل مساله بهینه‌سازی در (۹,۳)، خروجی $\hat{y}$ را براساس فرمول زیر، پیش‌بینی می‌کنیم

$$\hat{y} = \mathrm{y}^{\mathrm{T}}\mathrm{a}. \tag{۱۱.۳}$$

این مشابه k-نزدیک‌ترین همسایگی رگرسیون با مزیت برخورداری از وزن مشارکت هر نمونه ارائه شده در a است.

۳.۳ SPARROW: رگرسیون وزن‌دار مبتنی بر تقریب تنک

این بخش بر اساس کار منتشر شده در مقاله P.Noorzad و B.L.Strum با عنوان رگرسیون با تقریب تنک داده‌ها، در مجموعه مقالات کنفرانس پردازش سیگنال اروپا، اوت ۲۰۱۲م. است (Noorzad & Sturm, 2012).

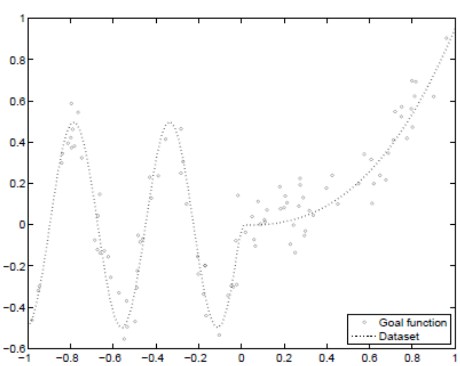
در این بخش، یک روش غیرپارامتری را برای رگرسیون چند متغیره محلی –رگرسیون وزنی تقریب تنک[1]- ارائه و مطالعه می‌کنیم که، تقریب تنک از یک نقطه برحسب متغیرهای پیشگوست. یک روش

[1] SPARse approximation Weighted regression

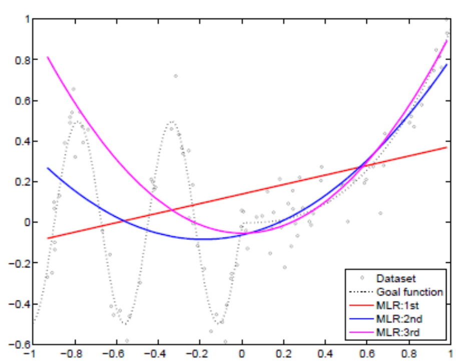
غیرپارامتری مشابه، رگرسیون k-نزدیک‌ترین همسایگی (k-NNR) است [۲۳]، که فرض می‌کند نزدیک‌ترین k متغیر پیشگو به یک نقطه آزمون، متغیرهای پاسخ یکسانی را تولید می‌کنند. هر دو روش می‌تواند به عنوان گونه‌هایی از رگرسیون هسته چندجمله‌ای محلی[1] (LPKR) در نظر گرفته شود [۴۷] که تابع رگرسیون را در یک نقطه با برازش یک چند جمله‌ای در آن نقطه، ارزیابی می‌نماید.

علاوه بر روش‌های محلی نظیر k-NNR و LPKR، پژوهش‌های درخور توجهی با هدف روش‌های غیرپارامتری کلی، مانند مدل‌های جمعی (AMها)[۸] و مدل‌های جمعی تنک (ApAM)[۴۱] صورت پذیرفته است. در AMها، روش‌های تک متغیره برای ارزیابی یک تابع هموار از هر متغیر پیشگو-در یک مدل شامل مجموع چنین توابع مولفه‌ای تک متغیره‌ای- با اجتناب از نیاز به ارتباط مستقیم با ورودی‌های چند بعدی، بکار گرفته می‌شوند. در SpAM، هدف کاهش تعداد توابع مولفه‌ای یک مدل جمعی است [۴۱]. رگرسیون تعاقبی تصویری (PPR) [۱۹] توسیعی به AMهایی است که، قادر به مدل کردن یک کلاس کلی تر از توابع هستند.

هرچند روش‌ها برای رگرسیون پارامتری و غیرپارامتری کلی ممکن است خطای میانگین را در کل مجموعه داده‌ها کمینه سازند، اما امکان ارائه یک برازش محلی خوب را ندارند. **شکل ۲.۳**، توانایی روش‌های محلی در مدل‌سازی داده‌های تولید شده توسط یک توزیع نامعلوم را نشان می‌دهد.

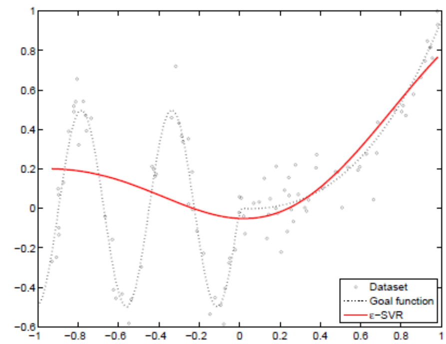


(الف) مجموعه داده تولید شده ما.

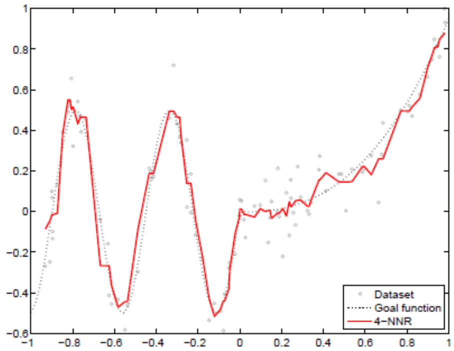


(ب) رگرسیون خطی چندگانه با جملات مرتبه ۱ تا ۳.

[1] Local Polynomial Kernel Regression



(ج) رگرسیون برداری ε-پشتیبان با هسته RBF.



(د) ۴ رگرسیون نزدیک‌ترین همسایگی.

شکل ۲.۳ این اشکال، توانایی روش‌های رگرسیون محلی را برای مدل‌سازی داده‌ها با یک توزیع ناشناخته، نشان می‌دهند. تابع مولد داده‌ها عبارت است از: $y_i = f(x_i)+\varepsilon_i$ $y_i = f(x_i) + \epsilon_i$ که در آن $f(x) = (x^3 + x^2)I(x) + \sin(x)I(-x)$.

روش‌های محلی نظیر LPKR، بر فرض یک مدل پارامتری محلی برای داده‌ها، متکی هستند [۱۱]. در LPKP، می‌توان تابع رگرسیون را در هر نقطه با برازش یک چندجمله‌ای تیلور در آن نقطه، ارزیابی نمود. این می‌تواند مدل‌هایی را بر اساس مرتبه چندجمله‌ای تولید کند که، ثابت محلی، خطی محلی، مربعی محلی و غیره هستند. اصل این روش، کمینه‌سازی یک مجموع وزنی از خطای مربعی است. معمولا، وزن‌ها با یک تابع نزولی از فاصله بین دو نقطه تعریف می‌شوند. SPARROW این وزن‌ها را با استفاده از تقریب تنک داده‌ی آزمون، تعریف می‌کند. این فرض برقرار است که یک نقطه آزمون با ترکیبی از پیشگوها بجای نزدیک به آنها، بهتر مدل‌سازی می‌شود.

مزایای مدل‌سازی داده‌ها با محدودیت‌های تنک بودن بخوبی مستندسازی شده است [۹]، [۱۶]، [۳۴]، به عنوان مثال در عدم پوشش کد فیزیولوژی قشر بصری اصلی پستانداران [۳۸] و در تولید کدهای تنک صداهای طبیعی [۳۱]، تصاویر [۶۱]، آواهای موسیقیایی [۳۹]. در حوزه یادگیری نظارت شده، دسته‌بندی بازنمایی تنک [۵۵]، می‌تواند روش‌های استاندارد را در چارچوب‌های دشوار پیاده‌سازی نماید، برای مثال ترازبندی نادرست و تغییر درخشندگی [۵۵]. شرط تنک بودن همچنین، برای انتخاب متغیر بویژه در لسو بکار می‌رود [۴۹]. در بخش های بعدی، SPARROW را تعریف کرده و نشان

می‌دهیم که چگونه، گونه ای از k-NNR و LPKP است. سپس، نتایج چندین آزمایش را در مقایسه SPARROW با این‌ها و دیگر روش‌های شناخته شده، ارائه می‌کنیم.

۱.۳.۳ رگرسیون وزن‌دار مبتنی بر تقریب تنک

یک مجموعه داده (یا فرهنگ لغت) از N مشاهده را در نظر بگیرید؛ $D := \{(x_i, y_i)\}_{i\in\Omega}$، که ورودی $x_i = [x_{i1},\ldots,x_{Mi}\}^T \in R^M$ متناظر با خروجی $y_i \in R$ است. فرض کنید $\Omega := \{1,2,\ldots,n\}$ اندیس فرهنگ لغت باشد. در رگرسیون غیرپارامتری، می‌توان فرض کرد که $y_i = f(x_i) + \varepsilon$، که در آن $f(x)$ نامعلوم است، اما تابعی است هموار و ε_i خطای مستقل از x_i است. با D و نقطه z مفروض، SPARROW تابع رگرسیون $f(z)$ را با ترکیب خطی خروجی‌های

$$\hat{f}(\mathrm{z}) := \sum_{i\in\Omega} l_j(\mathrm{z}, D)\, y_i \qquad (۱۲.۳)$$

ارزیابی می‌کند که در آن، $l_i(z,D)$، وزن موثر iام است که، SPARROW را به عنوان تابعی از تقریب تنک z در D انتخاب می‌کند.

بجای برازش یک مدل تکی برای همه مجموعه داده‌ها، مانند رگرسیون پارامتری و غیرپارامتری کلی، SPARROW مدل‌های پارامتری درباره هر نقطه آزمون z را با استفاده از مثلا یک بسط تیلور مرتبه صفر، یک یا دو، برازش می‌نماید. اکنون، بحث می‌کنیم که، چگونه SPARROW وزن‌های موثر را در (۱۲،۳) برای ارزیابی تابع رگرسیون در یک نقطه مفروض، تعریف می‌کند.

۲.۳.۳ تعریف وزن‌های موثر

برای دستیابی به ارزیابی مربعی تابع رگرسیون در z، می‌توانیم $f(x)$ را در z با یک چندجمله‌ای تیلور از درجه دو تقریب بزنیم

$$f(\mathrm{x}) \approx f(\mathrm{z}) + (\mathrm{x}-\mathrm{z})^{\mathrm{T}}\theta_{\mathrm{z}} + \frac{1}{2}(\mathrm{x}-\mathrm{z})^{\mathrm{T}}\mathrm{H}_{\mathrm{z}}(\mathrm{x}-\mathrm{z}) \qquad (۱۳.۳)$$

کـه در آن، $\theta_z := \nabla f(z)$ گرادیـان $f(x)$ و $H_z := \nabla^2 f(z)$ هسـی آنسـت کـه هـر دو در z ارزیـابی می‌شوند. اکنون، مساله یافتن $f(z), \theta_z, H_z$ بگونه‌ای است که، خطای مربعی وزنی محلی را در حوالی z برای اندازه‌گیری‌هایی در D کمینه‌سازی نماییم، یعنی

$$\min_{f(z),\theta_z,\mathrm{H}_z} \sum_{i\in\Omega} \alpha_i(\mathrm{z}) \left[y_i - f(\mathrm{z}) - (\mathrm{x}_i - \mathrm{z})^{\mathrm{T}} \theta_{\mathrm{z}} - \frac{1}{2} (\mathrm{x}_i - \mathrm{z})^{\mathrm{T}} \mathrm{H}_{\mathrm{z}} (\mathrm{x}_i - \mathrm{z}) \right]^2 \qquad (۱۴.۳)$$

جایی‌که $\alpha_i(z)$ وزن مشاهده iام است که، می‌تواند به چند طریق مثلا، با یک تابع هسته [۲۳]، [۲۵] یا با تقریب تنک صورت گرفته با SPARROW تعریف شود.

اکنون، ابر بردار پارامتری زیر را تعریف می‌کنیم [۴۷]

$$\Theta_{\mathrm{z}} := \left[f(\mathrm{z}), \theta_{\mathrm{z}}, \mathrm{vech}(\mathrm{H}_{\mathrm{z}}) \right]^{\mathrm{T}} \qquad (۱۵.۳)$$

که در آن $vech(H)$ نشان‌دهنده بردارسازی ماتریس $M \times M$ متقارن اسـت، یعنـی $M \times (M+1)/2$- بردار تشکیل شده با افزودن درآیه‌های قطری و مثلثی پایین H_z. ماتریس قطـری A_z را تعریـف کنیـد که، عنصر قطری iامش، وزن مشاهده $\alpha_i(z)$ است. با تعریف ماتریس

$$\mathrm{X}_{\mathrm{z}} := \begin{bmatrix} 1 & (\mathrm{x}_1 - \mathrm{z})^{\mathrm{T}} & \mathrm{vech}^{\mathrm{T}}\left[(\mathrm{x}_1 - \mathrm{z})(\mathrm{x}_1 - \mathrm{z})^{\mathrm{T}} \right] \\ \vdots & \vdots & \vdots \\ 1 & (\mathrm{x}_N - \mathrm{z})^{\mathrm{T}} & \mathrm{vech}^{\mathrm{T}}\left[(\mathrm{x}_N - \mathrm{z})(\mathrm{x}_N - \mathrm{z})^{\mathrm{T}} \right] \end{bmatrix} \qquad (۱۶.۳)$$

می‌توانیم کمینه‌سازی در (۱۴,۳) را به‌صورت

$$\min_{\Theta_{\mathrm{z}}} \left\| \mathrm{A}_{\mathrm{z}}^{1/2} \left[\mathrm{y} - \mathrm{X}_{\mathrm{z}} \Theta_{\mathrm{z}} \right] \right\|_2^2 \qquad (۱۷.۳)$$

بیان کنیم، که بردار پاسخ $y = [y_1, ..., y_N]^T$ است. پارامترهای تعریف شده بـا جـواب کمتـرین مربعـات عبارتند از [۴۷]

$$\hat{\Theta}_{\mathrm{z}} = (\mathrm{X}_{\mathrm{z}}^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} \mathrm{X}_{\mathrm{z}})^{-1} \mathrm{X}_{\mathrm{z}}^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} \mathrm{y} \qquad (۱۸.۳)$$

که $X_z^T A_z X_z$ معکوس پذیر است. در نهایت، ارزیابی مربعـی محلـی تـابع رگرسـیون در z دقیقـا عنصـر نخست Θ_z، یعنی

$$\hat{f}(\mathrm{z}) = e_1^{\mathrm{T}} (\mathrm{X}_{\mathrm{z}}^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} \mathrm{X}_{\mathrm{z}})^{-1} \mathrm{X}_{\mathrm{z}}^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} \mathrm{y} = \sum_{i\in\Omega} \beta_{\mathrm{i}} y_i \qquad (۱۹.۳)$$

است، که e_1 دارای یک در نخستین سطرش و صفر در جاهای دیگر است. پس، می‌بینیم که iامـین ورن موثر در (۱۲,۳) عبارت است از

$$l_i(\mathrm{z}, D) = e_i^{\mathrm{T}} \mathrm{A}_{\mathrm{z}}^{\mathrm{T}} \mathrm{X}_{\mathrm{z}} (\mathrm{X}_{\mathrm{z}}^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} \mathrm{X}_{\mathrm{z}})^{-1} \mathrm{e}_1 \tag{۲۰.۳}$$

به‌صورت خلاصه، SPARROW تابع رگرسیون را در نقطه z با محاسبه (۱۲,۳) بـا وزن‌هـای مـوثر مفروض (۳,۲۰)، ارزیابی می‌کند. اگر تنها از نخستین ستون X_z در (۲۰,۳) استفاده نماییم، یک ارزیـابی ثابت محلی از $f(z)$ یعنی

$$\hat{f}(\mathrm{z}) = (1^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} 1)^{-1} 1^{\mathrm{T}} \mathrm{A}_{\mathrm{z}} \mathrm{y} = \frac{\sum_{i \in \Omega} \alpha_i(\mathrm{z}) y_i}{\sum_{k \in \Omega} \alpha_k(\mathrm{z})}. \tag{۲۱.۳}$$

تولید می‌شود، با استفاده از $M+1$ ستون از X_z یک ارزیابی خطی محلی را تولیـد مـی‌کنـیم و بـا استفاده از همه X_zها به یک ارزیابی مربعی محلی می‌رسیم. با استفاده از چندجمله‌ای های مرتبه بـالاتر مانند مدل پارامتری محلی، انحراف ارزیابی را کاهش می‌دهیم [۲۵]،[۴۷]، اما این کار به بهای واریانس و افزایش زمان محاسباتی صورت می‌گیرد، زیرا تعداد پارامترهای محلی به‌صورت نمایی افزایش مـی یابـد. علاوه بر این، چندجمله‌ای‌های مرتبه بالاتر، بهبود نه چندان چشمگیری را بر مدل مربعی ارائه می‌کننـد، مگر زمانی‌که به دنبال ارزیابی گرادیان و هسی یعنی θ_z, H_z در (۱۴,۳) هستیم [۴۶].

۳.۳.۳ تعریف وزن‌های شهودی

چون وزن‌های موثر در (۲۰,۳)، تابعی از وزن‌های مشـاهده در (۱۴,۳)، یعنـی $\{\alpha_i(z) : i \in \Omega\}$ هسـتند، مساله باقی‌مانده، تعریف وزن‌های مشاهده است. اگر آنها را به‌صورت محلی در مدل ثابت (۲۱,۳) بـا یـک تابع هسته، تعریف کنیم، ارزیابی رگرسیون نادارایی-واتسون[1] (NWR) را تولید می‌کنیم. در ایـن راسـتا، می‌توانیم وزن‌ها را با

$$\alpha_i(\mathrm{z}) := K\big(S(\mathrm{z}, \mathrm{x}_i)/h\big) \tag{۲۲.۳}$$

تعریف کنیم، که در آن $K : R \to R_+$ یک تابع هسته، $h > 0$ پهنای باند و $S(z, x_i)$ فاصله

[1] Nadaraya-Watson regression (NWR)

$$S(\mathrm{z},\mathrm{x}_i) := (\mathrm{z}-\mathrm{x}_i)^{\mathrm{T}}\mathrm{V}^{-1}(\mathrm{z}-\mathrm{x}_i) \qquad (۲۳.۳)$$

است، که V یا یک ماتریس قطری از ارزیابی های غیر انحرافی واریانس های مشاهده شده در ابعاد پیشگو در D (در این حالت (۲۳,۳) فاصله اقلیدسی مدرج است)، یا ارزیابی غیرانحرافی کوواریانس پیشگو است (در این حالت (۲۳,۳) فاصله ماهالانوبیس[1] مد نظر است).

زمانی‌که وزن‌های مدل ثابت را به‌صورت محلی، تعریف می‌کنیم

$$\alpha_i(\mathrm{z}) := \begin{cases} d(\mathrm{z},\mathrm{x}_i), & i \in N_k(\mathrm{z}) \subset \Omega \\ 0, & \text{else} \end{cases} \qquad (۲۴.۳)$$

که در آن، $N_k(z)$ مجموعه اندیس k نزدیک‌ترین پیشگوی z در D است، سپس (۲۱,۳) k-NNR را تولید می‌کند [۲۳]. اگر $d(z,x_i) := 1$ باشد، سپس پهنای باند هسته ثابت از z حداقل به بزرگی بزرگ ترین فاصله بین زوج های مشاهدات یعنی، $h \geq \max_{i\in N_k(z)} S(z,x_i)$ است. در k-NNR (Wk-NNR)، این وزن را به عنوان معکوس فاصله $d(z,x_i) = 1/S(z,x_i)$ تعریف می‌کنیم.

در تقابل با NWR و k-NNR، در عوض، SPARROW وزن‌های مشاهده را از تقریب تنک z در D تعریف می‌کند. نخست، فرم ماتریسی از پیشگوی نرمال شده فرهنگ لغت را در نظر بگیرید

$$\mathrm{D} := \left[\frac{\mathrm{x}_1}{\|\mathrm{x}_1\|_2}, \frac{\mathrm{x}_2}{\|\mathrm{x}_2\|_2}, \ldots, \frac{\mathrm{x}_N}{\|\mathrm{x}_N\|_2}\right]. \qquad (۲۵.۳)$$

برای یک ورودی z، SPARROW جوابی را بر $z \approx Ds$ بگونه‌ای می یابد که $s = [s_1, s_2, \ldots, s_N]^T$ دارای تعداد زیادی عنصر صفر باشد. راه‌های متنوعی برای تولید تقریب‌های تنک وجود دارد (برای بررسی بیشتر [۶]، [۱۶]، [۵۱] را ببینید). در این کار، ما از اصل نویززدایی تعاقبی پایه ای[2] (BPDN) [۹] استفاده می‌کنیم که دارای مساله

$$\min_{\mathrm{s}\in\mathrm{R}^N} \|\mathrm{s}\|_1 \quad \text{subject to} \quad \frac{\|\mathrm{z} - \mathrm{Ds}\|_2^2}{\|\mathrm{z}\|_2^2} \leq \varepsilon^2 \qquad (۲۶.۳)$$

---

[1] Mahalanobis distance

[2] Basis Pursuit DeNoising

است که در آن، $\varepsilon^2 > 0$ سیگنال را به نسبت خطای تقریب محدود می‌سازد. در نهایـت، SPARROW وزن مشاهده iام را با استفاده از وزن‌های تقریب تنک تعریف می‌کند

$$\alpha_i(\mathrm{z}) := \left[\frac{S(\mathrm{z}, \mathrm{x}_i)}{\min_{j\in\Omega} S(\mathrm{z}, \mathrm{x}_j)} \right]^{-1} \frac{s_i}{\|\mathrm{z}\|_2} \tag{۲۷.۳}$$

که در آن، s_i عنصر iام s است. هدف ضریب نخست، وزن کردن با یک متغیر پاسخ نزدیک‌ترین پیشـگو به z است؛ و هدف تقسیم وزن تقریب تنک با $\|z\|_2$، حذف تاثیر طول آنست. بنابراین، اگرچه مشـابه بـا Wk-NNR، SPARROW بسختی یک مشاهده نزدیک تر به جستجو را وزن می‌کنـد، بـرخلاف Wk-NNR، این کار را زمانی انجام می‌دهد که، دارای ضریب ناصفری در تقریب تنکش با D باشد. زمانی‌کـه وزن‌های $\alpha_i(z)$ را از (۲۷,۳) در (۲۱,۳) جایگزین می‌کنیم، ارزیابی SPARROW (C-SPARROW) را بدست می آوریم، و زمانی‌که از این وزن‌ها را در (۲۰,۳) بـا تنهـا $M+1$ سـتون نخسـت از X_z بکـار می‌بریم، (۱۲,۳) ارزیابی SPARROW خطی (L-SPARROW) را تولید می‌نماید. با استفاده از ستون های X_z ارزیابی SPARROW مربعی (Q-SPARROW) را تولید می‌کنیم.

۴.۳ ارزیابی آزمایشی SPARROW

ما اکنون، کارایی SPARROW را در برابر چندین روش شناخته شـده دیگـر بـرای رگرسـیون محلـی، مقایسه می‌کنیم. در همه حالات، از فاصله اقلیدسی استاندارد شده در (۲۳,۳) استفاده می‌کنـیم. NWR و همتای خطیش، رگرسیون هسته خطی محلی (LLKR) [۲۳]؛ [۲۵] را می آزماییم، که (۱۸,۳) را بـا استفاده از $M+1$ ستون نخست X_z در (۱۶,۳) حل می‌کند. برای هـر دوی NWR و LLKR، هسـته گاوسی را در (۲۲,۳) به‌صورت زیر اتخاذ می‌کنیم

$$K(x) := \frac{1}{\sqrt{2\pi}} e^{-x^2/2}. \tag{۲۸.۳}$$

همچنین k-NNR، Wk-NNR [۲۳] را می آزماییم، که در آن، k را با اعتبارسازی گـذری تودرتـو، تنظیم می‌کنیم. برای یک خط مبنا، روش پارامتری کلی رگرسیون خطـی چندگانـه[1] (MLR) [۲۴]، را آزمایش می‌کنیم، که در آن فرض برآنست که، یک فرم خطی از تابع رگرسیون

$$f(\mathrm{x}) = \left[1, \mathrm{x}^{\mathrm{T}}\right]\mathrm{b} \tag{۲۹.۳}$$

و b برای کمینه‌سازی خطای مربعی میانگین

$$\mathrm{b} = \frac{\arg\min}{\mathrm{b}' \in \mathrm{R}^{M+1}} \left\| \mathrm{y} - [1, \mathrm{X}^{\mathrm{T}}]\mathrm{b} \right\|_2^2 \tag{۳۰.۳}$$

تعریف شده است، که در آن، ستون iام X برابر با x_i است. برای تولید تقریب تنک برای یک نقطه آزمون در (۲۶,۳)، از روش گرادیان تصویری طیفی برای ℓ_1-منظم‌سازی (SPGL1) [۵۲] با حداکثر ۲۰ تکرار و $\varepsilon = 10^{-6}$ استفاده می‌کنیم.

در این جا، ۴ مجموعه داده متمایز را به‌صورت مشترک در رگرسیون بکـار مـی‌بـریم (جـدول ۳,۱ را ببینید).[2]

جدول ۱.۳ اطلاعات مجموعه داده‌ها. آخرین ستون پارامتر k تزار شده در آزمایش‌های مربوط به k-NNR و Wk-NNR را نشان می‌دهد.

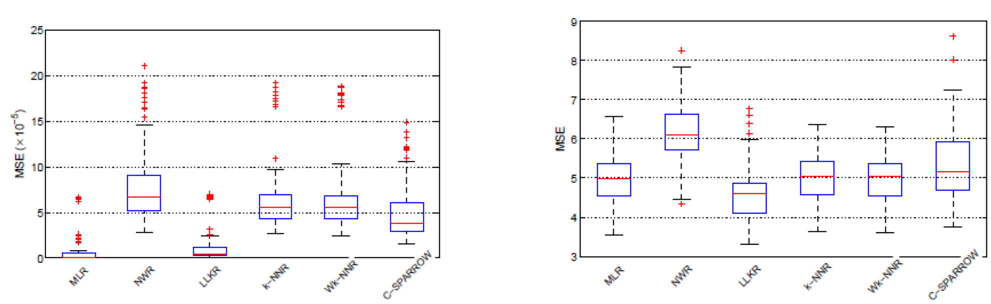
مجموعه داده‌ها	تعداد مشاهدات (N)	تعداد گرایش‌ها (M)	k
abalone	۴۱۷۷	۸	۹
bodyfat	۲۵۲	۱۴	۴
housing	۵۰۶	۱۳	۲
mpg	۳۹۲	۷	۴

[1] Multiple Linear Regression

[2] mpg, abalone and housing are from http://archive.ics.uci.edu/ml/; bodyfat is from http://lib.stat.cmu.edu/dataset/.

بجز برای bodyfat، هر مجموعه داده را بگونه‌ای استانداردسازی کردیم که، بعدشان صفر-میانگین و دارای واریانس یکسان باشد. شکل ۳،۳، ارزیابی های خطای مربع میانگین[1] (MSE) از این الگوریتم‌ها را از ۱۰ دنباله مستقل از اعتبار گذری ۱۰ تایی، نشان می‌دهد. می‌بینیم، در حالی که MLR بخوبی بـرای bodyfat و abalone اجرا می‌شود، برای mpg و housing بسیار ضعیف عمل می‌کند. از سـوی دیگـر، LLKR بخوبی برای همه مجموعه‌های داده، اجرا می‌شود. این مطلـب در کـارایی حاصـل از افـزایش در محاسبات صـورت گرفتـه توسـط LLKR از (۱۸،۳) مشـهود اسـت. بجـز بـرای abalone و housing، می‌بینیم که، C-SPARROW تقریبا مشابه با k-NNR و Wk-NNR اجرا می‌شود. این شگفت انگیـز است، زیرا؛ ۱) C-SPARROW هیچ فرضی از تعداد همسایگی های بکار رفته برای هر نقطه آزمـون را، مطرح نمی‌کند، و ۲) یک ارزیابی ثابت محلی را می‌سازد.

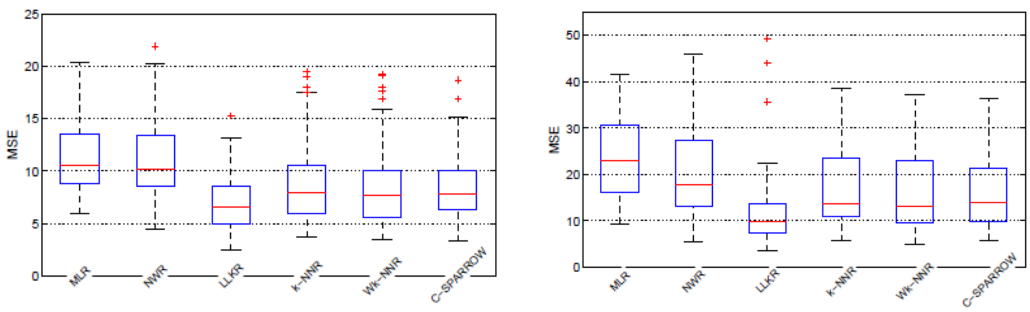
جدول ۲،۳، کارایی L-SPARROW را در مقایسه با C-SPARROW نشان می‌دهـد. مـی‌تـوانیم انتظار داشته باشیم که L-SPARROW بهتر از C-SPARROW عمل کند، زیرا مدل مرتبـه بـالاتری است. با این‌حال، یک مساله با رگرسیون چندجمله‌ای محلی برای چندجمله‌ای های مرتبه بـالاتر (یعنـی مرتبه اول یا دوم) است، آن هم زمانی‌که ورودی به‌صورت محلی، رتبه ناقص است، و جواب بـر (۱۸،۳) را ناپایدار می‌سازد.



(الف) مجموعه داده abalon

(ب) مجموعه داده bodyfat

[1] Mean Squared Error



(ج) مجموعه داده housing (د) مجموعه داده mpg

شکل ۳.۳ ترسیم جعبه ای برای ارزیابی اعتبار سنجی متقابل ۱۰-تایی از خطای مربع میانگین (۱۰۰ بار اجرای مستقل) برای ۴ مجموعه داده گوناگون. هر جعبه ۲۵ تا ۷۵ درصد را معین می‌کند و خط قرمز، میانه را نشان می‌دهد. اکسترمم با علامت + و برون هشته ها با ضربآنها مشخص شده است.

مساله را با حل یک فرم منظم شده از بهینه‌سازی کمترین مربعات وزن شده در (۱۴,۳) حل می‌کنیم. از ℓ_2-نرم پارامترهای محلی به عنوان جمله منظم‌سازی استفاده می‌کنیم، و یک مساله رگرسیون مرزی را حل می‌کنیم [۲۸]، یعنی بجای حل (۱۷,۳)، مساله

$$\min_{\Theta_z,\lambda}\left\|A_z^{1/2}[y-X_z\Theta_z]\right\|_2^2+\lambda\left\|\Theta_z\right\|_2^2 \qquad (۳۱.۳)$$

را حل می‌کنیم، که در آن $\lambda \geq 0$ پارامتر مرزی[۱] است. برای یک λ داده شده، جواب بفرم [۲۴]

$$\hat{\Theta}(z)=(X_z^T A_z X_z+\lambda I)^{-1}X_z^T A_z y. \qquad (۳۲.۳)$$

در می‌آید. ما λ را به همان طریق توصیف شده در بالا برای k تنظیم می‌کنیم. با این وجود، در حالی که می‌بینیم، کارایی L-SpARROW نسبت به استفاده از (۱۸,۳) بهبود یافته است، همچنان تا سطح C-SPARROW پایین باقی می‌ماند.

[1] Ridge parameter

جدول ۲.۳ مقایسه‌ای از ارزیابی های MSE براساس ۴ مجموعه داده با ۱۰ توالی اعتبار گذری ۱۰ تایی از C-SPARROW و L-SPARROW با و بدون منظم‌سازی. آخرین ستون، نشان‌دهنده پارامتر مرزی بکار رفته برای دستیابی به ارزیابی L-SPARROW است.

مجموعه داده‌ها	C-SPAR	L-SPAR. w/R	L-SPAR	λ
abalone	۵	۱۶	۹۸۸	$۱۰^{-۳}$
bodyfat	۳۵×۱۰^{-۵}	۵×۱۰^{-۵}	۹۶۰×۱۰^{-۵}	$۱۰^{-۶}$
housing	۱۰	۴۵	۴۳۰۵	$۱۰^{-۴}$
mpg	۷	۸	۶۳۳۵	$۱۰^{-۳}$

۱.۴.۳ نتیجه‌گیری

در این کار، انواعی سازگار از روش‌های رگرسیون چندجمله‌ای محلی را ارائه نمودیم:NWR، LLKR، k-NNR و Wk-NNR. NWR و LLKR از مجموعه داده، وزن، و متغیر پاسخ هر پیشگو با یک تابع هسته، استفاده می‌کنند. در عوض، k-NNR و Wk-NNR از متغیرهای پاسخ k نزدیک‌ترین پیشگو به نقطه بهره می‌گیرند، تا به‌صورت محلی تابع رگرسیون را ارزیابی نمایند. با SPARROW، ما استفاده از تقریب تنک برای انتخاب سازگاری را ارائه نمودیم که پیشگو را بکار می‌برد، و وزن‌های متغیرهای پاسخ برای ارزیابی تابع رگرسیون در نقطه داده شده، بکار می روند. آزمایش‌های ما، نشان می‌دهد که، SPARROW ثابت، می‌تواند یک الگوریتم رگرسیون رقابتی باشد. کار آینده ما وضعیت‌هایی را آنالیز می‌کند که، موجب توصیف داده‌ها به عنوان ترکیبی خطی (شامل وزن‌های منفی) از داده‌های برچسب گذاری شده می‌شود. علاوه بر این، می‌توانیم، دیگر الگوریتم‌های تقریب تنک از قبیل روش‌های حریصانه[1] را بکار ببریم، که معمولا، از نظر محاسباتی، کم هزینه‌تر از روش‌های بهینه‌سازی محدب از قبیل BPDN هستند.

[1] Greedy approaches

۵.۳ مقایسه kNN با SRC

در جدول ۳,۳، مقایسه‌ای را از الگوریتم دسته‌بندی k نزدیک‌ترین همسایگی و دسته‌بندی بازنمایی تنک روی مجموعه داده‌های دسته‌بندی چند کلاسه، ارائه می‌کنیم. مجموعـه داده‌هـا از منزلگـاه LIBSVM برای مجموعه داده‌های چندکلاسه انتخاب شده است.[1] می‌بینیم که، دسته‌بندی بازنمایی تنک هیچ بهبود قابل توجهی را بر دسته‌بندی k نزدیک‌ترین همسایگی روی اکثریت مجموعه داده‌ها، ارائه نمی‌نماید.

جدول ۳.۳ مقایسه دقت بدست آمده توسط kNN و SRC روی ۵ مجموعه داده دسته‌بندی چندکلاسه.

مجموعه داده‌ها	n	p	تعداد کلاس ها	k	kNN	SRC
دی ان آ	۲۰۰۰	۱۸۰	۳	۱۲۵	۸۶	۸۶
شیشه	۲۱۴	۹	۶	۲	۷۰	۶۵
عنبیه	۱۵۰	۴	۳	۶	۹۵	۷۲
مصوته	۵۲۸	۱۰	۱۱	۲	۹۴	۸۴
شراب	۱۷۸	۱۳	۳	۷	۹۷	۹۹

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html

۴

فصل چهارم

یک هم‌ارزی بین ε-SVR و BPDN

در مقاله ای با عنوان " یک هم‌ارزی بین تقریب تنک و ماشین های بردار پشتیبان"، Girosi مساله برنامه‌ریزی مربعی رگرسیون بردار پشتیبان[1] (هسته) (SVR) را در چارچوب تئوری منظم‌سازی، نتیجه گرفت. این مطلب در تضاد با انحراف اصلی Vapnik از SVR با استفاده از اصل کمینه‌سازی ریسک ساختاری در تئوری یادگیری آماری است. Girosi نشان می‌دهد که، تحت شرط ها و فرضیات فرعی، روش نویززدایی تعاقبی پایه ای (BPDN) منجر به مساله برنامه‌ریزی مربعی مشابهی می‌شود. Chen، Donoho و Saunders، نویززدایی تعاقبی پایه ای را به عنوان یک روش مهارشدنی برای تقریب تنک ارائه نمودند.

۱.۴ فضای هیلبرت هسته تکثیری

فضای ضرب داخلی H را روی مجموعه X بر R در نظر بگیرید. اگر H نسبت به متریک القایی توسط ضرب داخلی، کامل باشد، سپس H یک فضای هیلبرت است. ما ضرب داخلی بردارهای $f, g \in \mathrm{H}$ را با $< f, g >$ نشان می‌دهیم.

تعریف ۱,۱,۴. *می‌گوییم* H یک فضای هیلبرت تکثیری است، اگر برای هر $x \in X$، تابعک خطی $F_x : \mathrm{H} \to R$ که در آن $F_x(f) = f(x)$ است، کران‌دار باشد.

چون F_x یک تابعک خطی کران‌دار است، از قضیه نمایش ریس، در می‌یابیم که، برای هر $x \in X$ یک $K_x \in H$ یکتا با ویژگی بازیابی

$$F_x[f] = \langle K_x, f \rangle = f(x) \qquad (۱.۴)$$

وجود دارد، که در آن، f تابعی در H است. برای $t \in X$، یک $K_x \in H$ وجود دارد و با ویژگی بازیابی در معادله (۱,۴)، داریم

$$K_t[x] = \langle K_x, K_t \rangle. \qquad (۲.۴)$$

[1] Support Vector Regression

برای همه $y \in X$ تابع $K : X \times X \to R$ را به‌صورت

$$K(x, y) = K_y(x) \tag{۳.۴}$$

تعریف می‌کنیم و آن را هسته بازیابی[1] (RK) H می نامیم. از یکتایی K_y در می‌یابیم که K کاملا توسط H تعیین می‌شود. قضیه زیر، ارتباطی را بین یک RK و هسته هیلبرت بازیابی متناظرش[2] (RKHS) برقرار می‌سازد:

قضیه ۱،۲،۴. *برای هر* RKHS *یک تابع معین مثبت متقارن یکتا (بنام* RK) *وجود دارد و برعکس برای هر تابع معین مثبت متقارن* K *روی* $X \times X$، یک RKHS یکتا از توابع روی X با K به عنوان خود RK موجود است.

اثبات را می‌توانید در فصل نخست کتاب Wahba [۵۶] درباره اسپلاین‌ها بیابید. ما تنها بسراغ ساختارهای اثبات می‌رویم. در حالت خاص، اگر H یک RKHS باشد، سپس RK عبارت است از

$$K(x, y) = \langle K_x, K_y \rangle \tag{۴.۴}$$

که در آن برای هر $x, y \in X$، توابع نماینده یکتا همان K_x, K_y هستند. برعکس، برای K مفروض، برای همه $x, y \in X$، ما تابع نماینده را به صورت

$$K_x(y) = K(x, y). \tag{۵.۴}$$

تعریف می‌کنیم. RKHS را به عنوان مکمل فضای توابع H_0 تولید شده توسط $\{K_x, x \in X\}$ می‌سازیم، البته با ضرب داخلی تعریف شده به‌صورت

$$\langle f, g \rangle = \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_i \beta_j K(x_i, y_j) \tag{۶.۴}$$

که در آن، f, g توابعی در H_0 هستند،

---

[1] Reproducing Kernel

[2] Reproducing Kernel Hilbert Space

$$f = \sum_{i=1}^{m} \alpha_i K_{x_i} \tag{۷.۴}$$

$$g = \sum_{j=1}^{n} \beta_j K_{y_j}. \tag{۸.۴}$$

از این تعریف نتیجه می‌گیریم

$$K(x, x_i) = \left\langle K_x, K_{x_i} \right\rangle, \tag{۹.۴}$$

و نیز

$$\begin{aligned} \left\langle K_x, f \right\rangle &= \left\langle K_x, \sum_{i=1}^{m} \alpha_i K_{x_i} \right\rangle \\ &= \sum_{i=1}^{m} \alpha_i \left\langle K_x, K_{x_i} \right\rangle \\ &= \sum_{i=1}^{m} \alpha_i K(x, x_i) \\ &= \sum_{i=1}^{m} \alpha_i K_{x_i}(x) \\ &= f(x). \end{aligned} \tag{۱۰.۴}$$

آخرین معادله نشان می‌دهد که هسته، نماینده ارزیابی است.

تا کنون، می‌دانیم که، یک RKHS دارای هسته بازیابی یکتای K است. چون K یک تابع معین مثبت است، دارای یک آنالیز طیفی به شکل

$$K(x, y) = \sum_{n=1}^{\infty} \lambda_n \phi_n(x) \phi_n(y) \tag{۱۱.۴}$$

می‌باشد، که در آن، $\Phi_1, \Phi_2, \ldots$ دنباله متعامد یکه توابع ویژه در $L_2[X]$ و $\lambda_1 \ge \lambda_2 \ge \ldots \ge 0$ مقادیر ویژه متناظر هستند. توابع ویژه از RK، H را به عنوان RKHS تولید می‌کند. پس، هر تابع در H را می‌توان به‌صورت

$$\hat{f}(x) = \sum_{n=1}^{\infty} c_n \phi_n(x) \tag{۱۲.۴}$$

با نرم

$$\|\hat{f}\|_H^2 = \langle \hat{f}, \hat{f} \rangle_H = \sum_{n=1}^{\infty} \frac{c_n^2}{\lambda_n}. \tag{۱۳.۴}$$

نشان داد. تابعک‌هایی بشکل (۱۳,۴)، به عنوان تابعک‌های هموار متناظر با مقادیر کوچک‌تر نسبت به توابع هموارتر، شناخته می‌شوند.

۲.۴ ماشین‌های بردار پشتیبان برای رگرسیون

مجموعه داده تصادفی $D = \{(x_i, y_i) : x_i \in R^d, y_i \in R, i = 1,...,l\}$ با نمونه‌گیری تصادفی (در غیاب نویز) از یک تابع نامعلوم f بدست آمده است. هدف ما، پوشش f یا یک ارزیابی از D است. این مساله رگرسیون، جواب‌های بسیاری دارد، زیرا، توابع فراوانی از مجموعه نقاط داده شده، می گذرند. مساله را با این فرض که، درمیان تمامی توابع درون یابی، جواب مساله ما دارای بیشترین همواری است، محدود می‌سازیم (که در آن، همواری شامل نقاط نزدیکی است که مقادیر نزدیکی دارند). فرض کنید $\Phi[f]$ تابعکی هموار باشد. در $\varepsilon - SVR$، هدف ما یافتن تابع $\hat{f}$ با حداکثر انحراف ε از داده‌های آزمایشی است، یعنی برای $i = 1,...,l$ همواره $|\hat{f}(x_i) - y_i| \le \varepsilon$. مساله بهینه‌سازی عبارت است از

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\Phi[\hat{f}] \\ &\text{subject to} \quad \begin{cases} y_i - \hat{f}(\mathrm{x}_i) \le \varepsilon \\ \hat{f}(\mathrm{x}_i) - y_i \le \varepsilon \end{cases} \quad \text{for } i = 1,\cdots,l. \end{aligned} \tag{۱۴.۴}$$

متغیرهای کمبود ζ_i, ζ_i^* را در ارتباط با امکان غیرعملی بودن مساله بهینه‌سازی محدود شده در (۱۴,۴) معرفی می‌کنیم. مساله بهینه‌سازی به‌صورت

$$\text{minimize} \quad \frac{1}{2}\Phi[\hat{f}] + C\sum_{i=1}^{l}(\varsigma_i + \varsigma_i^*)$$
$$\text{subject to} \begin{cases} y_i - \hat{f}(\mathrm{x}_i) \le \varepsilon + \varsigma_i \\ \hat{f}(\mathrm{x}_i) - y_i \le \varepsilon + \varsigma_i^* \\ \varsigma_i \ge 0 \\ \varsigma_i^* \ge 0 \end{cases} \quad \text{for } i = 1, \cdots, l \qquad (۱۵.۴)$$

در می‌آید، که در آن، پارامتر آزاد C>0 تعامل بین همواری $\hat{f}$ و مقدار انحراف آن را –فراتر از ε- از داده‌های آزمایشی، کنترل می‌کند. مساله در (۱۵,۴) یک مساله برنامه‌ریزی محدب است و بنابراین دارای یک کمینه یکتاست. با معرفی ضرایب لاگرانژ (متغیرهای دوگان) برای افزودن محدودیت‌ها به تابع هدف، لاگرانژین های مساله را در (۱۵,۴) تشکیل می‌دهیم

$$L(f,\varsigma,\varsigma^*;\alpha,\alpha^*,\mathrm{r},\mathrm{r}^*) = \frac{1}{2}\Phi[\hat{f}] + C\sum_{i=1}^{l}(\varsigma_i + \varsigma_i^*) + \sum_{i=1}^{l}\alpha_i^*(y_i - f(\mathrm{x}_i) - \varepsilon - \varsigma_i^* + \sum_{i=1}^{l}\alpha_i(\hat{f}(x_i) - y_i - \varepsilon - \varsigma_i) - \sum_{i=1}^{l}(r_i\varsigma_i + r_i^*\varsigma_i^*) \qquad (۱۶.۴)$$

بگونه‌ای که α,α^*,r,r^* در شرایط نامنفی صدق کنند. لاگرانژین در (۱۶,۴) دارای یک نقطه زینی در جواب بهینه است. بنابراین، بهینه‌سازی با کمینه‌سازی (۱۶,۴) نسبت به متغیرهای اصلی $\hat{f},\zeta,\zeta^*$ و بیشینه‌سازی نسبت به متغیرهای دوگان α,α^*,r,r^* سروکار دارد.

توجه داشته باشید که تا این جا، هیچ ساختاری را برای $\hat{f}$ یا هموار‌ی تابع در نظر نگرفته ایم. اما همان‌طور که Cucker و Smale می گویند "*فرآیند یادگیری در خلا بوقوع نمی پیوندد.*"

بمنظور یافتن تابع $\hat{f}$، نیازمند تعیین فضای فرضیه هستیم که در جستجویمان آن را در نظر می‌گیریم. در این جا، فرض می‌کنیم که، تابع درون یابی $\hat{f}$ متعلق به یک RKHS مانند H باشد. بنابراین، می‌توان آن را به‌صورت (۱۲,۴) با نرمی به شکل (۱۳,۴) نشان داد. با این‌حال، با دنبال کردن انحراف Vapnik از $\varepsilon - SVR$، انحراف صریحی را برای $\hat{f}$ یعنی

$$\hat{f}(\mathrm{x}) = \sum_{n=1}^{\infty} c_n\phi_n(\mathrm{x}) + b. \qquad (۱۷.۴)$$

در نظر می‌گیریم. با جایگذاری (۱۷,۴) برای $\hat{f}$ و (۱۳,۴) برای $\Phi[\hat{f}]$، لاگرانژین به شکل

$$\begin{aligned} L(b,\mathrm{c},\varsigma,\varsigma^*;\alpha,\alpha^*,\mathrm{r},\mathrm{r}^*) = \frac{1}{2}\sum_{n=1}^{\infty}\frac{c_n^2}{\lambda_n} + C\sum_{i=1}^{l}(\varsigma_i+\varsigma_i^*) \\ +\sum_{i=1}^{l}\alpha_i^*\left(y_i-\sum_{n=1}^{\infty}c_n\phi_n(\mathrm{x_i})-b-\varepsilon-\varsigma_i^*\right) \\ +\sum_{i=1}^{l}\alpha_i\left(\sum_{n=1}^{\infty}c_n\phi_n(\mathrm{x_i})+b-y_i-\varepsilon-\varsigma_i\right) \\ -\sum_{i=1}^{l}(r_i\varsigma_i+r_i^*\varsigma_i^*). \end{aligned} \tag{۱۸.۴}$$

در می‌آید. برای رسیدن به تابع هدف دوگان، نیازمند کمینه‌سازی لاگرانژین نسبت به متغیرهای اولیه و حذف آنها با جایگذاری هستیم. ما نیازمند این هستیم که، مشتقات جزیی L نسبت به متغیرهای اولیه $\hat{f}$ (b,c_n)، ζ_i,ζ_i^* صفر باشد

$$\frac{\partial L}{\partial c_n}=0\Rightarrow c_n=\lambda_n\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)\phi_n(\mathrm{x}_i) \tag{۱۹.۴}$$

$$\frac{\partial L}{\partial b}=0\Rightarrow \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)=0 \tag{۲۰.۴}$$

$$\frac{\partial L}{\partial \varsigma_i}=0\Rightarrow r_i=C-\alpha_i \tag{۲۱.۴}$$

$$\frac{\partial L}{\partial \varsigma_i^*}=0\Rightarrow r_i^*=C-\alpha_i^* \tag{۲۲.۴}$$

با جایگذاری (۱۹,۴) در مدل ما برای درون یابی تابع داریم

$$\begin{aligned}\hat{f}(\mathrm{x}) &= \sum_{n=1}^{\infty} c_n \phi_n(\mathrm{x}) + b \\ &= \sum_{n=1}^{\infty}\left(\lambda_n \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\varphi_n(x_i)\right)\phi_n(\mathrm{x}) + b \\ &= \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\sum_{n=1}^{\infty}\lambda_n \varphi_n(\mathrm{x}_i)\,\phi_n(\mathrm{x}) + b \\ &= \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)K(\mathrm{x}, \mathrm{x}_i) + b\end{aligned} \qquad (۲۳.۴)$$

که در آن، آخرین تساوی از (۱۱,۴) می‌آید. به‌صورت مشابه، ما (۱۹,۴) را در معادله یمان برای نرم جایگذاری می‌کنیم

$$\begin{aligned}\left\|\hat{f}\right\|^2 &= \sum_{n=1}^{\infty}\frac{c_n^2}{\lambda_n} \\ &= \sum_{n=1}^{\infty}\lambda_n\left(\sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\phi_n(\mathrm{x}_i)\right)\left(\sum_{j=1}^{l}(\alpha_j^* - \alpha_j)\phi_n(\mathrm{x}_j)\right) \\ &= \sum_{n=1}^{\infty}\lambda_n \sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\phi_n(\mathrm{x}_i)\,\phi_n(\mathrm{x}_j) \\ &= \sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\sum_{n=1}^{\infty}\lambda_n |\phi_n(\mathrm{x}_i)\phi_n(\mathrm{x}_j) \\ &= \sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(\mathrm{x}_i, \mathrm{x}_j).\end{aligned} \qquad (۲۴.۴)$$

با جایگذاری (۱۲,۴)، (۲۲,۴)، (۲۳,۴) و (۲۴,۴) در لاگرانژین، به

$$\begin{aligned} L(\alpha,\alpha^*) &= \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(\mathrm{x}_i,\mathrm{x}_j)+C\sum_{i=1}^{l}(\varsigma_i+\varsigma_i^*) \\ &+\sum_{i=1}^{l}\alpha_i^*\left(y_i-\sum_{j=1}^{l}(\alpha_i^*-\alpha_i)K(\mathrm{x}_i,\mathrm{x}_j)-b-\varepsilon-\varsigma_i^*\right) \\ &+\sum_{i=1}^{l}\alpha_i\left(\sum_{j=1}^{l}(\alpha_i^*-\alpha_i)K(\mathrm{x}_i,\mathrm{x}_j)+b-y_i-\varepsilon-\varsigma_i\right) \\ &-\sum_{i=1}^{l}(C-\alpha_i)\varsigma_i-\sum_{i=1}^{l}(C-\alpha_i^*)\varsigma_i^* \\ &= -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(\mathrm{x}_i,\mathrm{x}_j)-\varepsilon\sum_{i=1}^{l}(\alpha_i^*-\alpha_i)+\sum_{i=1}^{l}y_i(\alpha_i^*-\alpha_i). \end{aligned} \tag{۲۵.۴}$$

می‌رسیم. با داشتن (۲۱,۴) و (۲۲,۴) و این واقعیت که، متغیرهای دوگان در محدودیت‌های نامنفی صدق می‌کنند، به محدودیت $0\le\alpha_i,\alpha_i^*\le C, i=1,..l$ می‌رسیم. بنابراین، مساله دوگان به شکل زیر است

$$\begin{aligned} &\text{minimize} \quad -L(\alpha,\alpha^*) \\ &\text{subject to} \quad \begin{cases} 0\le\alpha_i,\alpha_i^*\le C & \text{for } i=1,\cdots,l \\ \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)=0 & \\ \alpha_i\alpha_i^*=0 & \text{for } i=1,\cdots,l \end{cases} \end{aligned} \tag{۲۶.۴}$$

که در آن بجای کمینه‌سازی L، ما $-L$ را بیشینه می‌کنیم. آخرین محدودیت به‌صورت خودکار برقرارست، در غیر این صورت، وجود کمبودهای ناصفر را در هر دو مسیر، نشان می‌دهد. ما آن را برای مقایسه های بعدی، در نظر می‌گیریم.

بطورکلی، در یک مساله کمینه‌سازی محدب، تابع هدف اولیه، همیشه، بزرگ‌تر یا مساوی با تابع هدف دوگان در مقادیر متغیر اولیه و دوگان است. تفاوت بین آنها، شکاف دوگانگی[1] نامیده می‌شود. برای مساله برنامه‌ریزی مربعی در (۲۶,۴)، نشان داده شده است که این شکاف صفر است. بنابراین، جواب بهینه را با حل مساله دوگان می‌یابیم. این بدان معناست که، برای مساله کمینه‌سازی منظم شده در

[1] Duality Gap

(۱۵,۴)، کمینه‌سازی بر فضای توابع هیلبرت به کمینه‌سازی بر R^l می رسد. این نتیجه با قضیه نمایش Kimeldorf و Wahba سازگار است [۲۹].

۳.۴ $\epsilon - SVR$ و تنک بودن

در یک کمینه محلی مساله بهینه‌سازی محدود شده، شرایط Karush-Kuhn-Tucker برقرار است. در میان این‌ها، شرط فقدان تمامیت، بیان می‌کند که در یک نقطه x_i، حاصل ضرب بین محدودیت‌ها و متغیرهای دوگان، صفر است. برای این مساله در (۲۶,۴)، دو شرط از این دست عبارتند از

$$\begin{aligned} \alpha_i(f(\mathrm{x}_i) - y_i - \varepsilon - \varsigma_i) = 0 \\ \alpha_i(y_i - f(\mathrm{x}_i) - \varepsilon - \varsigma_i) = 0. \end{aligned} \tag{۲۷.۴}$$

براساس (۲۷,۴)، زمانی‌که $|f(x_i) - y_i| < \varepsilon$ باشد، α_i, α_i^* باید صفر باشد. زمانی‌که، $|f(x_i) - y_i| \geq \varepsilon$ باشد، α_i یا α_i^* ممکن است ناصفر باشند. نقاطی با α_i یا α_i^* ناصفر، بردارهای پشتیبان نامیده می‌شوند. با افزایش پارامتر آزاد ε، ما تعداد بردارهای پشتیبان را در حالی که تنک بودن جواب در (۲۳,۴) رخ می‌دهد، کاهش می‌دهیم. تاثیر پارامتر آزاد C روی تنک بودن جواب، ممکن است تنها به‌صورت تجربی نشان داده شود. زمانی‌که، مانند تنظیمات مساله فعلی ما، هیچ نویزی در داده‌ها موجود نباشد، مقدار بهینه C بی نهایت است. بنابراین، ε تنها پارامتر آزاد این فرمول‌بندی است (بدون احتساب پارامترهای هسته).

۴.۴ ارتباط با تقریب تنک

در تقریب تنک، هدف ما، تقریب یک تابع نامعلوم f با یک ترکیب خطی از مجموعه ثابت Φ از توابع

$$f(\mathrm{x};\mathrm{a}) = \sum_{i=1}^{n} \alpha_i \phi_i(\mathrm{x}) \tag{۲۸.۴}$$

است، که در آن، $\Phi = \{\Phi_i(x) : i = 1,\dots,n\}$ فرهنگ لغت نامیده می‌شود. فرهنگ لغت معمولا بیش از اندازه کامل است، که بر این دلالت دارد که، a در (۲۸,۴) یکتا نیست، زیرا بعضی عناصر Φ معمولا، ترکیباتی خطی از دیگر عناصر هستند. ما مساله را با نیاز به این که، a باید جوابی پراکنده باشد، محدود می‌سازیم؛ جوابی با حداقل تعداد عناصر ناصفر. تابع هزینه زیر، یک فرمول بندی را از این مساله نشان می‌دهد

$$E(a) = \frac{1}{2}\left\| f(\mathrm{x}) - \sum_{i=1}^{n} \alpha_i \phi_i(\mathrm{x}) \right\|_{L_2}^{2} + \lambda \|\mathrm{a}\|_{L_0}^{p} \qquad (۲۹.۴)$$

که در آن، $\|.\|_{L_0}$ تعداد عناصر ناصفر یک بردار را می شمارد، و $\|.\|_{L_2}$ همان L_2-نرم است. با این‌حال، بدلیل L_0-نرم، کمینه‌سازی تابع هزینه در (۲۹,۴) NP-سخت است. برای رسیدگی کردن به مهارنشدنی بودن (۲۹,۴)، [۱۰] کمینه‌سازی تابع هزینه محدب

$$f(\mathrm{x};\mathrm{a}) = \sum_{i=1}^{n} \alpha_i \phi_i(\mathrm{x}) \qquad (۳۰.۴)$$

را ارائه می‌نماید و آن را نویززدایی تعاقبی پایه ای می نامد. مولفان $L1$-نرم را به عنوان تقریبی بر L_0-نرم بکار می‌برند. ما توابع فرهنگ لغت را به کمک هسته بازسازی K از یک RKHS، H تعیین می‌کنیم

$$\phi_i(x) = K(x, x_i), \quad i = 1,\dots,l \qquad (۳۱.۴)$$

که در آن، $\{(x_i, y_i) : i = 1,\dots,l\}$ مجموعه داده حاصل از نمونه‌گیری f در غیاب نویز است. علاوه بر این، ما معیار L_2 را در (۳۰,۴) با H نرم جایگزین کرده و ε را به عنوان پارامتر منظم‌سازی تعریف می‌کنیم. به تابع هزینه زیر می‌رسیم

$$E(\mathrm{a}) = \frac{1}{2} \| f(\mathrm{x}) - \hat{f}(\mathrm{x},\mathrm{a}) \|_{H}^{2} + \varepsilon \| \mathrm{a} \|_{L_1} \qquad (۳۲.۴)$$

که در آن، تابع تقریب عبارت است از

$$\hat{f}(\mathrm{x},\mathrm{a}) = \sum_{i=1}^{l} \alpha_i K(\mathrm{x},\mathrm{x}_i). \tag{۳۳.۴}$$

با فرض این که تابع هزینه f دارای میانگین صفر در H است، یعنی تصویرش روی تابع ثابت، صفر است (فرض بر این نیست که تابعی ثابت در H است)

$$\langle f, 1\rangle_H = 0 \tag{۳۴.۴}$$

نیازمند این هستیم که تابع تقریب نیز دارای میانگین صفر در H باشد

$$\langle \hat{f}, 1\rangle_H = 0. \tag{۳۵.۴}$$

به این منظور، ما K را بگونه‌ای نرمال می‌کنیم که

$$\langle 1, K(\mathrm{x},\mathrm{y})\rangle = 1. \tag{۳۶.۴}$$

با جایگذاری (۳۳,۴) در (۳۵,۴) و با استفاده از (۳۶,۴)، به محدودیت زیر می‌رسیم

$$\begin{aligned} \langle \hat{f}, 1\rangle &= \left\langle \sum_{i=1}^{l} \alpha_i K(\mathrm{x},\mathrm{x}_i), 1\right\rangle \\ &= \sum_{i=1}^{l} \alpha_i \langle K(\mathrm{x},\mathrm{x}_i), 1\rangle \\ &= \sum_{i=1}^{l} \alpha_i = 0. \end{aligned} \tag{۳۷.۴}$$

نرم H را در معادله (۳۲,۴) از تابع هزینه گسترش می‌دهیم، یعنی

$$\begin{aligned} E(\mathrm{a}) = \frac{1}{2}\| f \|_H^2 - \sum_{i=1}^{l} a_i \langle f(\mathrm{x}), K(\mathrm{x},\mathrm{x}_i)\rangle_H \\ + \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j \langle K(\mathrm{x},\mathrm{x}_i), K(\mathrm{x},\mathrm{x}_j)\rangle_H + \varepsilon \|\mathrm{a}\|_{L_1}. \end{aligned} \tag{۳۸.۴}$$

دو ویژگی زیر از یک هسته بازیابی را به یاد بیاورید

$$\langle K(\mathrm{x},\mathrm{x}_i), K(\mathrm{x},\mathrm{x}_j)\rangle_H = K(\mathrm{x}_i,\mathrm{x}_j) \tag{۳۹.۴}$$

$$\left\langle f(\mathrm{x}), K(\mathrm{x}, \mathrm{x}_j)\right\rangle_H = f(\mathrm{x}_i) \qquad (۴۰.۴)$$

جایی‌که در آخرین معادله، $f(x_i) = y_i$ است، زیرا داده‌ها بدون نویز هستند. تابع هزینه به شکل

$$E(\mathrm{a}) = \frac{1}{2}\| f \|_H^2 - \sum_{i=1}^{l} a_i y_i + \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j K(\mathrm{x}_i, \mathrm{x}_j) + \varepsilon \| \mathrm{a} \|_{L_1} . \qquad (۴۱.۴)$$

در می‌آید. بردار a می‌تواند به بخش های مثبت و منفی آنالیز شود. این منجر به این می‌شود که L_1-نرم به‌صورت زیر نوشته شود

$$\| a \|_{L_1} = \sum_{i=1}^{l} | a_i | = \sum_{i=1}^{l} (a_i^+ + a_i^-) \qquad (۴۲.۴)$$

به‌گونه‌ای که برای هر $i = 1,\ldots,l$ داشته باشیم $a_i^+, a_i^- = 0$ و $a_i^+, a_i^- \geq 0$. با استفاده از معادله (۴۲,۴) و این واقعیت که $\| f \|_H^2$ نسبت به a_i^+, a_i^- ثابت است، به مساله برنامه‌ریزی مربعی

$$\begin{aligned}
&\underset{a_i^+, a_i^-}{\text{minimize}} \quad -\sum_{i=1}^{l}(a_i^+ - a_i^-)y_i + \frac{1}{2}\sum_{i=1}^{l}(a_i^+ - a_i^-)(a_j^+ - a_j^-)K(\mathrm{x}_i, \mathrm{x}_j) + \varepsilon \sum_{i=1}^{l}(a_i^+ + a_i^-) \\
&\text{subject to} \quad \begin{cases} a_i^+, a_i^- \geq 0 & \text{for } i = 1,\cdots,l \\ \sum_{i=1}^{l}(a_i^+ - a_i^-) = 0 & \\ a_i^+ a_i^- = 0 & \text{for } i = 1,\cdots,l \end{cases}
\end{aligned} \qquad (۴۳.۴)$$

می‌رسیم؛ که در آن، محدودیت دوم با آنچه که در معادله (۳۷,۴) است، یکسان می‌باشد. با تغییر نام ضریب a_i^+ به α_i^+ و a_i^-, به α_i، واضح بنظر می رسد که، معادله (۴۳,۴) مساله برنامه‌ریزی مربعی مشابه با معادله (۲۶,۴) را مشخص کند. نتیجه می‌گیریم که اگر

- مجموعه داده‌ها بدون نویز باشد، یعنی $y_i = f(x_i)$ ،
- L_2-نرم با H نرم در جمله برازش داده‌های BPDN جایگزین شود،
- تابع $\hat{f}$ دارای میانگین صفر در RKHS، H باشد،
- اتم‌های فرهنگ لغت بکار رفته در BPDN به‌صورت (۳۱,۴) تعریف شوند،
- و پارامتر منظم‌سازی در SVR مایل به صفر باشد، زمانی‌که $C \to \infty$.

سپس، BPDN و $\varepsilon-SVR$ هم‌ارز هستند، زیرا آنها به مساله برنامه‌ریزی مربعی یکسانی می رسند. آخرین شرط، به این واقعیت منتقل می‌شود که، $\varepsilon-SVR$ منجر به تابع درون یابی خواهد شد که، محدود به بیش برازش داده‌هاست (زیرا، تاثیر جمله منظم‌سازی تعدیل می‌شود). این مطلب برای آنالیز آنچه که درباره طرح تقریب تنک می‌گوییم، مهم است، این نسخه بروز شده BPDN است.

۵

فصل پنجم

نتیجه گیری و کارهای آینده

در این پایان‌نامه، نشان داده ایم که، کمینه‌سازی زیان مربعی ℓ_1-منظم شده برای دسته‌بندی، یک موفقیت از هر دو نظر محاسباتی و آماری است. همچنین، نشان دادیم که کمینه‌سازی زیان مربعی دسته‌بندی ℓ_1-منظم شده برای بازسازی، چندان ارزشی ندارد. روش‌های ساده‌تر، نظیر دسته‌بندی kNN و WkNNR حداقل خوب هستند.

چهار حوزه را ارائه نمودیم که، برای پژوهش‌های آینده نیز مناسب هستند. یکی، حالتی که طراحی با نگاشت های غیرخطی متغیرهای مشاهده شده، پر می‌شود. چگونه این بر مساله بهینه‌سازی تاثیر خواهد گذاشت و چگونه بر اجرای دسته‌بندی تاثیر گذار خواهد بود. [۴۴] این مساله را برای رگرسیون مرزی (یا دسته‌بندی کمترین مربعات منظم شده) پاسخ داده است. ما همچنین، به جستجوی نزدیک تر در مساله دوگان SVM علاقه‌مند هستیم. آیا این مساله می‌تواند بگونه‌ای موثر با افزودن شرط تنک بودن روی متغیر دوگان α حل شود؟ آیا این منجر به یک دسته‌بندی کننده سریعتر خواهد شد که، در کارایی با SVM قابل مقایسه باشد؟ یک نقص مهم اصل بهینه‌سازی لسو این است که، جواب‌های لسو، پایدار نیستند. در نهایت، ما علاقه‌مند به دانستن این هستیم که، آیا kNN یا WkNN برای یادگیری ویژگی در مقایسه با روش‌های تقریب تنک در یادگیری بازنمایی تنک به عنوان مثال در دسته‌بندی تصاویر، مناسب هستند. در ادامه به هر یک از این ایده ها، می پردازیم.

۱.۵ رگرسیون غیرخطی

مجموعه $\{\phi_i, i=1,\dots,p\}$ از p تابع از پیش تعیین شده را با حداقل یک عضو غیرخطی در نظر بگیرید. می‌توان این توابع را برای بیان متغیرهای اصلی $\{x_i, i=1,\dots,p\}$ بکار برد و به تابع رگرسیون غیرخطی حاصل برای ترکیب خطی این نگاشت ها، رسید

$$\sum_{i=1}^{p} a_i \phi_i(\mathrm{x}_i) + a_0. \qquad (۱.۵)$$

می‌توان این مدل را با استفاده از الگوریتم‌هایی یکسان برای برازش مدل‌های خطی، برازش نمود، البته با مزیت دستیابی به جوابی که یک ارتباط غیرخطی را بین متغیرهای پاسخ و تفسیر، مدل‌سازی می‌کند [۲۷]. الگوریتم‌هایی یکسان را می‌توان بکار برد، زیرا این مدل هنوز، در پارامترها (ضرایب رگرسیون) خطی است. این مدل تنها در متغیرهای تفسیری غیرخطی است.

مثالی از کاربرد این روش، تعاقب تطابق هسته است [۵۴]. در این چارچوب، $\phi_i(x) = k(x_1, x)$ است، که در آن، k تابعی دومتغیره است که، لزوما شرایط مرسر [۱] را ندارد. مثالی دیگر، فرمول بندی لسوی هسته ای شده است [۵۷].

۲.۵ منظم‌سازی α: متغیر دوگان SVM

بیاد بیاورید که، انگیزه ما برای ℓ_1-منظم‌سازی، رسیدن به یک دسته‌بندی کننده است که، سریع‌تر از SVM باشد؛ البته در این معناکه، ارزیابی های هسته کمتری در زمان آزمون وجود دارد

$$f(x) = \sum a_i k(x_i, x). \tag{۲.۵}$$

همچنین، بیاد بیاورید که، می‌گوییم، کنترل مستقیمی بر تنک بودن α در مساله بهینه‌سازی SVM نداریم. هدف ما از مستقیما پراکنده ساختن چیست؟ بیایید نگاهی دیگر به مساله بهینه‌سازی دوگان SVM بیندازیم.

۳.۵ ناپایداری و غیریکتا بودن جواب‌های لسو

جواب‌های لسو زمانی‌که $rank(X) \neq p$ باشد، یکتا نیستند [۵۰]. این حالت بویژه زمانی اتفاق می‌افتد که $p > n$ باشد. در این حالت، جواب‌های چندگانه بر مساله بهینه‌سازی لسو وجود دارد. جواب مساله بهینه‌سازی لسو باید به‌صورت

$$\mathrm{a} \in \arg\min_{\mathrm{a}} \frac{1}{2} \| \mathrm{y} - \mathrm{Xa} \|^2 + \lambda \| \mathrm{a} \|_1 . \tag{۳.۵}$$

باشد، این امر منجر به دو مساله بزرگ می‌شود؛ نخست این که، یک جواب می‌تواند یک ضریب iام مثبت باشد، در حالی که، دیگری دارای یک ضریب iام منفی است. علاوه بر این، دو جواب متمایز می‌تواند دارای پشتیبان‌های متمایزی باشند.

[1] Mercer ‘s conditions

این مطلب، زمانی‌که کار انتخاب متغیر پیش‌بینی نیست، مساله ساز است. با این‌حال، ما علاقه‌مند به مطالعه تاثیرات ناسازگاری جواب‌های لسو روی کارایی پیش‌بینی و پایداریش هستیم.

۴.۵ kNN یا تعاقب تطابقی برای یادگیری ویژگی

اگر تقریب تنک برای کدگذاری تنک خوب باشد، ترجیح می‌دهیم که بدانیم، آیا تعاقب تطابقی یا kNN می‌تواند موثر باشد. مزیت تعاقب تطابقی و kNN این است که، آنها از نظر محاسباتی کمتر از تقریب تنک با روش‌های آزادسازی محدب نظیر SPGL1، هزینه‌بر هستند.

واژه‌نامه

الف

آستانه‌سازی نرم	soft thresholding
احتمال پسین	posterior probability
اعتبارسنجی متقابل	cross-validation
انقباض	shrinkage

ب

بازنمایی تنک	sparse representation
بردار ضریب	coefficient vector
بردار وزن	weight vector
بردار ویژگی	feature vector
برنامه‌ریزی مربعی	quadratic programming

پ

پیشگو	regressor

ت

تعاقب پایه	basis pursuit

تعاقب تطابقی	matching pursuit
تقریب تنک	sparse approximation
تنک	sparse

د

دسته‌بندی بازنمایی تنک	sparse representation classification
دسته‌بندی بردار پشتیبان	support vector classification
دسته‌بندی‌کننده‌ی جانشین	plug-in classifier
دقت (دسته‌بندی کننده)	accuracy

ر

رگرسیون $\ell 2$-منظم	ridge regression
رگرسیون تنک	sparse regression
رگرسیون وزن‌دار مبتنی بر تقریب تنک	sparse approximation weighted regression (SPARROW)

ز

زیان محوری	hinge loss
زیان مربعی	square loss
زیرنمونه‌گیری	Down sampling

ش

شرط تنک بودن	sparsity constraint

ف

فضای هیلبرت هسته تکثیری	reproducing kernel Hilbert space
فیشر سازگار	Fisher consistent

ق

قابلیت تعمیم (دسته‌بندی کننده)	generalization performance

ک

کارآمد	efficient
کاهش مختصاتی	coordinate descent
کدگذار خودکار	auto-encoder
کدگذاری تنک	sparse coding
کرنل	kernel

م

ماشین بولتزمن تحدیدی	restricted Boltzmann machine (RBM)
مدل‌های جمعی	additive models
مسأله‌ی معکوس خطی	linear inverse problem
مشاهدات آموزشی	training samples
منظم‌سازی	regularization

ن

نظریه‌ی بازنمایی ریس	Riesz representation theorem

و

وزن‌های شهودی	observational weights
ویژگی	feature

ی

یادگیری ویژگی	feature learning

منابع و مراجع

1. Robert Andersen. Robust regression for the linear model. In Modern Methods for Robust Regression, pages 47-70. Sage Publications, 2008.

2. S. Arora, L. Babai, J. Stern, and Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. In Proceedings of the IEEE 34th Annual Foundations of Computer Science, pages 724-733, Washington, DC, USA, 1993. IEEE Computer Society.

3. Peter L. Bartlett, Michael I. Jordan, and Jon D. McAulife. Convexity, classiffication, and risk bounds. Technical Report 638, Department of Statistics, U.C. Berkeley, 2003.

4. Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classiffiers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory (COLT), 1992.

5. Stephane Boucheron, Olivier Bousquet, and Gabor Lugosi. Theory of classiffcation : A survey of some recent advances. ESAIM: Probability and Statistics, 9:323-375, 2005.

6. Alfred M. Bruckstein, David L. Donoho, and Michael Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. SIAM Review, 51(1): 34-81, 2009.

7. Peter Buhlmann and Bin Yu. Boosting with the L2 loss: regression and classi_cation. Journal of the American Statistical Association, 98(462):324-339, 2003.

8. Andreas Buja, Trevor Hastie, and Robert Tibshirani. Linear smoothers and additive models. The Annals of Statistics, 17(2):435-555, 1989.

9. S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. SIAM J. Sci. Comput., 20(1):33-61, Aug. 1998.

10. Scott S. Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. Technical Report 479, Department of Statistics, Stanford University, May 1995.

11. W. S. Cleveland and S. J. Devlin. Locally weighted regression: an approach to regression analysis by local fitting. Journal of the American Statistical Association, 83(403):596- 610, 1988.

12. Adam Coates and Andrew Ng. The importance of encoding versus training with sparse coding and vector quantization. In International Conference on Machine Learning (ICML), pages 921-928, 2011.

13. Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. Journal of Machine Learning Research – Proceedings Track, 15:215-223, 2011.

14. Felipe Cucker and Steve Smale. On the mathematical foundations of learning. Bulletin of the American Mathematical Society, 39:1-49, 2002.

15. L. Devroye, L. Gyorfi, and G. Lugosi. A Probabilistic Theory of Pattern Recognition. Springer, 1996.

16. M. Elad. Sparse and redundant representations: From theory to applications in signal and image processing. Springer, 2010.

17. John Fox. Robust regression: Appendix to an R and S-PLUS companion to applied regression, 2002.

18. Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1, part 2):119-139, 1997.

19. Jerome H. Friedman. Fast sparse regression and classiffication. In Paul Eilers, editor, Proceedings of the 23rd International Workshop on Statistical Modelling, pages 27-57. Statistical Modelling Society, 2008.

20. Jerome H. Friedman and Werner Stuetzle. Projection pursuit regression. Journal of the American Statistical Association, 76(376):817-823, 1981.

21. Jort F. Gemmeke, Tuomas Virtanen, and Antti Hurmalainen. Exemplar-based sparse representations for noise robust automatic speech recognition. IEEE Transactions on Audio, Speech and Language Processing, 19(7):2067-2080, 2011.

22. Frederico Girosi. An equivalence between sparse approximation and support vector machines. Neural Computation, 10:1455-1480, August 1998.

23. W. Hardle and O. Linton. Applied nonparametric methods. Technical Report 1069, Yale University, 1994.

24. T. Hastie, R. Tibshirani, and J. Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2 edition, 2009.

25. T. J. Hastie and C. Loader. Local regression: Automatic kernel carpentry. Statistical Science, 8(2):120-129, 1993.

26. Trevor Hastie and Ji Zhu. Comment. Statistical Science, 21:352-357, 2006.

27. Tim C. Hesterberg, Nam H. Choi, Lukas Meier, and Chris Fraley. Least angle and ℓ_1- penalized regression: A review. Statistics Surveys, 2:61-93, 2008.

28. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12:55-67, 1970.

29. George S. Kimeldorf and GraceWahba. Some results on Tchebycheffian spline functions. Journal of Mathematical Analysis and Applications, 33:82-95, 1971.

30. Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient L_1 regularized logistic regression. In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), 2006.

31. M. S. Lewicki. Effcient coding of natural sounds. Nature Neuroscience, 5(4):356-363, Mar. 2002.

32. Yi Lin. A note on margin-based loss functions in classiffication. Technical report, Department of Statistics, University of Wisconsin, Madison, 2002.

33. Julien Mairal. Sparse Coding for Machine Learning, Image Processing and Computer Vision. PhD thesis, Ecole Normale Superieure de Cachan, 2010.

34. S. Mallat. A Wavelet Tour of Signal Processing: The Sparse Way. Academic Press, Elsevier, Amsterdam, 3rd edition, 2009.

35. J. Marron and M. Todd. Distance weighted discrimination. Technical report, School of Operations Research and Industrial Engineering, Cornell University, 2002.

36. Hosein Mohimani, Massoud Babaie-Zadeh, and Christian Jutten. A fast approach for overcomplete sparse decomposition based on smoothed ℓ_0 norm. Transactions on Signal Processing, 57:289-301, January 2009.

37. E. Nadaraya. On estimating regression. Theory of Probability and its Applications, 9 (1):141-142, 1964.

38. B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381(6583):607-609, 1996.

39. Mark D. Plumbley, Thomas Blumensath, Laurent Daudet, Remi Gribonval, and Mike E. Davies. Sparse Representations in Audio and Music: from Coding to Source Separation. Proceedings of the IEEE, 98(6):995-1005, 2010.

40. Tomaso Poggio, Lorenzo Rosasco, and Andre Wibisono. Sufficient conditions for uniform stability of regularization algorithms. Technical Report CBCL-284, Center for Biological and Computational Learning, MIT, December 2009.

41. Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. Journal of The Royal Statistical Society (Series B), 71(5):1009-1030, 2009.

42. Ryan Rifkin. Everything old is new again: a fresh look at historical approaches in machine learning. PhD thesis, Sloan School of Management, Massachusetts Institute of Technology, 2002.

43. Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classi_cation. Journal of Machine Learning Research, 5:101-141, December 2004.

44. Ryan Rifkin, Gene Yeo, and Tomaso Poggio. Regularized least-squares classi_cation. In Advances in Learning Theory: Methods, Model and Applications, volume 190, pages 131-153. IOS Press, 2003.

45. Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? Neural Computation, 16(5):1063-107, 2004.

46. D. Ruppert. Local polynomial regression and its applications in environmental statistics. Technical report, Cornell University, 1996.

47. D. Ruppert and M. P. Wand. Multivariate locally weighted least squares regression. The Annals of Statistics, 22:1346-1370, 1994.

48. Alex J. Smola and Bernhard Scholkopf. A tutorial on support vector regression. Statistics and Computing, 14:199-222, August 2004.

49. Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society (Series B), 58:267-288, 1996.

50. Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. Journal of the Royal Statistical Society (Series B), 73(3):273-282, 2011.

51. Ryan J. Tibshirani. The lasso problem and uniqueness. arXiv:1206.0313v1, 2012. J. A. Tropp and S. J. Wright. Computational methods for sparse solution of linear inverse problems. Proceedings of the IEEE, 98(6):948-958, June 2010.

52. E. van den Berg and M. P. Friedlander. Probing the Pareto frontier for basis pursuit solutions. SIAM Journal on Scientific Computing, 31(2):890-912, 2008.

53. Vladimir N. Vapnik. The nature of statistical learning theory. Statistics for Engineering and Information Science. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

54. Pascal Vincent and Yoshua Bengio. Kernel matching pursuit. Machine Learning, 48: 165-187, September 2002.

55. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma. Towards a practical face recognition system: Robust alignment and illumination via sparse representation. To appear in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2011.

56. Grace Wahba. Spline models for observational data, volume 59 of CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

57. Gang Wang, Dit-Yan Yeung, and Frederick H. Lochovsky. The kernel path in kernelized lasso. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2007.

58. Sijian Wang, Bin Nan, Saharon Rosset, and Ji Zhu. Random Lasso. Annals of Applied Statistics, 5(1):468-485, 2011.

59. John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 31:210-227, 2009.

60. Huan Xu, Constantine Caramanis, and Shie Mannor. Sparse algorithms are not stable: A no-free-lunch theorem. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(1):187-193, 2012.

61. J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classiffication. In IEEE Conference on Computer Vision and Pattern Recognition, 2009.

62. Guo-Xun Yuan, Kai-Wei Chang, Cho-Jui Hsieh, and Chih-Jen Lin. A comparison of optimization methods and software for large-scale ℓ_1-regularized linear classiffication. Journal of Machine Learning Research, 11:3183-3234, 2010.

63. Guo-Xun Yuan, Chia-Hua Ho, and Chih-Jen Lin. Recent advances of large-scale linear classiffication. Submitted, 2011.

64. Tong Zhang. Statistical behavior and consistency of classiffication methods based on convex risk minimization. The Annals of Statistics, 32:56-134, March 2004.

65. Tong Zhang and Frank J. Oles. Text categorization based on regularized linear classification methods. Information Retrieval, 4:5-31, 2001.

Abstract*

The ℓ_1-regularized square loss minimization problem has recently gained much attention. This optimization principle has two main applications in the machine learning literature. Specically, the lasso or basis pursuit de-noising (although the two are not entirely equivalent in practice) is the optimization principle used for solving the linear inverse problem: $y = Xa$, under convex sparsity constraints. When the lasso is used for regression and classification, y is a vector of outputs. When it is used for sparse coding and feature learning, or in the context of sparse representation classification, y is the feature vector or signal itself.

The use of lasso for regression is already well-established. In this thesis, we argue that the use of lasso for classification also has its advantages. One might think that the square loss is not appropriate for the classification task, however, theoretical results show that all convex loss functions are Fisher consistent. Additionally, square loss minimization, like logistic loss minimization, and unlike hinge loss minimization, gives estimates of the posterior probability. The value of the posterior probability at a point tells us about the confidence of the classifier in its prediction. Another benefit of the lasso for classification is that ℓ_1-regularization leads to a sparse classifier, that once trained, can be evaluated quickly. Additionally, one has direct control over the sparsity of the solution through the regularization parameter. The only problem with the lasso is the stability of its solutions (Wang et al., 2011).

The second part of the thesis, is on the use of the lasso for signal and feature representation. The lasso or basis pursuit de-noising is also an integral part of the sparse representation classification method. We extend this method to the regression setting. Through experimental results we argue that one can easily achieve the same or even better results using simpler methods like k-nearest neighbor classification which is also better motivated theoretically. We conclude that ℓ_1-regularized square loss minimization is not worth it.

Key Words: square loss minimization, ℓ_1-regularization, binary classification, sparse representation.

Amirkabir University of Technology
(Tehran Polytechnic)

Department of Computer Engineering and Information Technology

MSc Thesis

Title
Efficient Classification Based on
Sparse Regression

By
Pardis Noorzad

Supervisor
Dr. Mohammad Rahmati

Advisor
Dr. Nasrollah Moghaddam Charkari
Dr.Mohammad Mehdi Ebadzadeh

July 2012