دانشکده علوم ریاضی

پایان‌نامه کارشناسی ارشد آمارریاضی

انتخاب متغیر در داده‌های بزرگ توسط مدل‌سازی غیرخطی

نگارنده: محبوبه محبی

استاد راهنما

دکتر داود شاهسونی

بهمن ۱۳۹۶

شماره:
تاریخ:

باسمه‌تعالی

مدیریت تحصیلات تکمیلی

فرم شماره (۳) صورتجلسه نهایی دفاع از پایان نامه دوره کارشناسی ارشد

با نام و یاد خداوند متعال، ارزیابی جلسه دفاع از پایان نامه کارشناسی ارشد خانم محبوبه محبی با شماره دانشجویی ۹۴۱۵۷۹۴ رشته آمار گرایش آمارریاضی تحت عنوان انتخاب متغیر در داده‌های بزرگ توسط مدل‌سازی غیرخطی که در تاریخ ۹۶/۱۱/۱۰ با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح ذیل اعلام می‌گردد:

قبول (با درجه: خیلی خوب) ☒ مردود ☐

نوع تحقیق: نظری ☒ عملی ☐

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضاء
۱ـ استادراهنمای اول	دکتر داود شاهسونی	دانشیار	
۲- استادراهنمای دوم			
۳- استاد مشاور			
۴- نماینده تحصیلات تکمیلی	دکتر احمد نزاکتی رضازاده	دانشیار	
۵- استاد ممتحن اول	دکتر محمد آرشی	دانشیار	
۶ـ استاد ممتحن دوم	دکتر حسین باغیشنی	استادیار	

نام و نام خانوادگی رئیس دانشکده: دکتر ابراهیم هاشمی

تاریخ و امضاء و مهر دانشکده:

معاونت علوم، تحقیقات و فناوری
دانشکده ریاضی

تبصره: در صورتی که کسی مردود شود حداکثر یکبار دیگر (در مدت مجاز تحصیل) می تواند از پایان نامه خود دفاع نماید (دفاع مجدد نباید زودتر از ۴ ماه برگزار شود).

تقدیم به

سرمایه‌های زندگیم؛ پدر و مادر مهربانم

همراه همیشگیم؛ خواهرم

تکیه‌گاه استوارم؛ برادرانم

و همه‌ی آن‌هایی که می‌خواهند بیشتر بدانند

باسپاس

سپاس و ستایش خدای یگانه و مهربانی را که وجودمان را آکنده از عطش پویایی و دانش پژوهشی ساخت. حمد و ثنا خدایی را که عشق به علم و پژوهش را در وجودمان نهاد و بدان طراوت بخشید.

در این جا از کلیه ی کسانی که مرا در انجام این پایان نامه یاری و همراهی کرده اند، کمال تشکر را دارم. بر خود واجب می دانم به سپاس زحمات بی دریغ استاد گرامی ام، آقای دکتر شاهسونی که در نهایت لطف و بزرگواری، با قبول راهنمایی این پایان نامه افتخار بزرگی را نصیب اینجانب کردند و تدوین و تحقیق این پایان نامه مرهون راهنمایی و شکیبایی ایشان است، کمال تشکر و قدردانی را داشته باشم؛ دلسوزی ها و همیاری های ایشان را ارج می نهم. از آقایان دکتر آرشی و دکتر باغیشنی که داوری پایان نامه را عهده دار شدند تشکر می نمایم. همچنین از کلیه ی اساتید دلسوز و بزرگوار گروه آمار آقای دکتر ربیعی، آقای دکتر نزاکتی و خانم دکتر اقبال به خاطر زحماتی که در طول دوره ی تحصیل متقبل شدند کمال تشکر و قدردانی را دارم.

در پایان از پدر و مادر بزرگوارم که نه می توانم موهایشان را که در راه عزت من سفید شد، سیاه کنم و نه برای دستهای پینه بسته شان که ثمره تلاش برای افتخار من است، مرهمی دارم. پس پروردگارا توفیقم ده هر لحظه شکرگزارشان باشم و ثانیه های عمرم را در عصای دست بودنشان بگذرانم.

از خواهر عزیزم که مهربانیش سایه سار زندگیم می باشد، او که اسوه صبر و پشتکار بوده و مشکلات مسیر را برایم تسهیل نموده و برادرانم همراه تمامی لحظات و قوت قلبی برای ادامه ی راهم بوده اند، سپاسی ویژه دارم.

محبوبه محبی
بهمن ۱۳۹۶

تعهد نامه

اینجانب **محبوبه محبی** دانشجوی کارشناسی ارشد رشته **آمار** دانشکده **علوم ریاضی** دانشگاه صنعتی شاهرود، نویسنده پایان‌نامه با عنوان **انتخاب متغیر در داده‌های بزرگ توسط مدل‌سازی غیرخطی** ، تحت راهنمایی **داود شاهسونی** متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های دیگر پژوهش‌گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان‌نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ‌جا ارایه نشده است.
- حقوق معنوی این اثر، به دانشگاه صنعتی شاهرود تعلق دارد، و مقالات مستخرج با نام ‘‘ دانشگاه صنعتی شاهرود ’’ یا “ Shahrood University of Technology ” به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به‌دست آوردن نتایج اصلی پایان‌نامه تاثیرگذار بوده‌اند، در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (یا بافت‌های آنها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

محبوبه محبی
بهمن ۱۳۹۶

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می‌باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان‌نامه بدون ذکر منبع مجاز نمی‌باشد.

چکیده

تحلیل داده‌های با بعد بالا در همه‌ی زمینه‌های مختلف علوم، صنعتی و تجاری، همواره با چالش انتخاب متغیرهای مهم روبرو است. روش‌های متداول انتخاب متغیر مبتنی بر ساختارهای خطی هستند اما در بسیاری از موارد و در بعد بالا ممکن است ارتباط ساختاری متغیر پاسخ و متغیرهای توضیحی خطی نباشد. همچنین این روش‌ها، متغیرهایی را که به تنهایی تاثیر ناچیزی بر متغیر پاسخ دارند، اما در ترکیب با سایر متغیرها، برای پیشگویی ممکن است مفید باشند را نادیده می‌گیرند. در این پژوهش با ارائه الگوریتمی چند مرحله‌ای مبتنی بر رگرسیون موضعی خطی، متغیرهای مهم در قالب زیرمجموعه‌هایی شناسایی و انتخاب شده و مدل پیشگویی برازش می‌شود.

کلمات کلیدی: رگرسیون موضعی، اعتبارسنجی متقاطع، ترکیبی از متغیرها، انتخاب متغیر

لیست مقالات مستخرج از پایان‌نامه

۱. محبی.، شاهسونی‌د، (۱۳۹۶)، ’’انتخاب متغیر و پیشگویی در داده‌های بزرگ توسط یک روش غیرخطی‘‘، کنفرانس روش‌های مدرن در قیمت‌گذاری بیمه‌ای و آمارهای صنعتی، دانشگاه بوعلی‌سینا، همدان، ایران

پیشگفتار

داده‌های بزرگ معمولا به مجموعه‌ای از داده‌ها اطلاق می‌شود که مدیریت، کنترل، چالش و پردازش آنها فراتر از توانایی ابزارهای نرم‌افزاری در یک زمان پذیرفتنی و مورد انتظار است. چالش اصلی در رابطه با داده‌های بزرگ شامل استخراج، ذخیره‌سازی، جستجو و آنالیز می‌باشد. مقیاس داده‌های بزرگ با توجه به رشد و فناوری روز افزون، مداوم در حال رشد است.
نمونه‌هایی از داده‌های بزرگ شامل تحقیقات در حوزه‌های مختلف تجاری و صنعتی و نیز شاخه‌های مختلف ژنتیک، نجوم، پزشکی، پژوهش‌های زمین‌شناسی و سایر علوم دیگر، از جمله داده‌ها با ابعاد بالا می‌باشد. ساختار این گونه داده‌ها به این صورت است که تعداد متغیرهای توضیحی بزرگتر از تعداد مشاهدات است. به دلیل ساختار این گونه داده‌ها با یک موضوع چالش برانگیز در تحلیل داده‌ها با بعد بالا، تحت عنوان انتخاب متغیر مواجه می‌شویم. در حال حاضر انتخاب متغیر بطور گسترده در حوزه‌های مختلف علوم مورد استفاده قرار می‌گیرد. در ۱۵ سال گذشته توجه ویژه‌ای به انتخاب متغیر شده است. که این روش‌ها بر فرض خطی بودن است.

در این پژوهش بااستفاده از یک روش غیرخطی انتخاب متغیر در داده‌های با بعد بالا، که در آن تعداد متغیرها بسیار بزرگتر از تعداد مشاهدات است مورد مطالعه قرار می‌گیرد.
با توجه به کاربرد موفق روش‌های انتخاب متغیر خطی برای بسیاری از مجموعه داده‌ها، اما ارزیابی وجود رابطه‌ی خطی در بعد بالا کاری بسیار دشوار است و ممکن است در بسیاری از موارد این فرض معنی‌دار نبوده و رابطه‌ی غیرخطی حاکم باشد. لذا استفاده از روش‌های خطی موثر نخواهد بود. علاوه بر این روش‌های متداول انتخاب متغیر منجر به حذف متغیرهایی می‌شوند که به تنهایی ارزش کمی دارند، اما در ترکیب با دیگر متغیرها ممکن است نتایج سودمندی داشته باشند.

در این پایان‌نامه پس از معرفی روش انتخاب متغیر با پذیره‌ی خطی بودن، به بیان ارائه یک روش انتخاب متغیر با پذیره‌ی غیرخطی بودن آن پرداخته، در پایان با روش‌های دیگر خطی مورد ارزیابی قرار می‌دهیم. با این مقدمه، ساختار پایان‌نامه به صورت زیر تنظیم شده است:

- در فصل اول تعاریف و مفاهیم اولیه در ارتباط با رگرسیون خطی و غیرخطی را مطرح می‌کنیم.
- در فصل دوم به بیان روش‌های انتخاب متغیر در یک مدل خطی می‌پردازیم. از جمله این روش‌ها کمترین توان‌های دوم و روش رگرسیون حداقل زاویه پرداخته و معیارهای مربوطه را به منظور ارزیابی عملکرد روش انتخاب متغیر پیشنهادی معرفی می‌کنیم.
- در فصل سوم ابتدا روش اعتبارسنجی متقاطع Leave-One-Out را بیان نموده و در ادامه به بیان جزیی‌تر الگوریتم انتخاب متغیر با پذیره‌ی غیرخطی می‌پردازیم.
- در فصل چهارم برای ارزیابی عملکرد روش پیشنهادی، مجموعه داده‌هایی در ابعاد بالا

و بسیار بالا را شبیه‌سازی کرده و توانایی الگوریتم را به منظور انتخاب صحیح متغیرها بررسی می‌کنیم. همچنین الگوریتم را بر روی مجموعه داده‌ی واقعی اجرا نموده و نتایج بدست آمده را با دیگر روش‌های انتخاب متغیر مقایسه می‌کنیم.

- در پیوست آ حاوی کدهای نوشته شده در محیط R برای باز تولید مثال‌های پایان نامه است.

فهرست مطالب

فهرست تصاویر

فهرست جداول

فصل ۱

مفاهیم و مقدمات

۱.۱ مقدمه

گسترش روز افزون فناوری اطلاعات، امکان ایجاد مجموعه داده‌هایی با بعد بالا را در تحقیقات حوزه‌های مختلف همچون صنعت و تجارت و نیز شاخه‌های مختلف علوم: زیست شناسی، نجوم، ژنتیک و پزشکی و سایر علوم دیگر پدید آورده است. ساختار گونه‌ای از این داده‌ها به این صورت است که بعد مجموعه متغیرهای توضیحی [۲] بسیار بزرگتر از اندازه نمونه است. این امر تحلیل کلان داده‌ها را با چالش جدی روبرو ساخته است. بنابراین برای تحلیل و تفسیر بهتر این داده‌ها نیاز به شناسایی متغیرهای موثر از طریق روش‌های انتخاب متغیر کارآمد هستیم. بدین ترتیب چالشی برای کشف روابط پنهان در داده‌ها و بکارگیری روش‌های متداول آماری و یادگیری ماشین[۳] ایجاد شود. از این رو، محققین با مسأله مهمی تحت عنوان «انتخاب متغیر[۴]» در هر دو زمینه‌ی یادگیری با راهنما[۵] (رگرسیون و رده بندی [۶]) یادگیری

---

[۲] Covariates
[۳] Machine learninig
[۴] Variable selection
[۵] Superviesed
[۶] Classification

بدون راهنما[۱] (خوشه‌بندی [۲]و تحلیل پیوند) مواجه هستند. تفاوت روش‌های باراهنما و بدون راهنما به وجود یا عدم وجود متغیر پاسخ برمی‌گردد. در روش‌های باراهنما متغیر پاسخ[۳] وجود دارد در حالیکه در روش‌های بدون راهنما این گونه نیست.

مسأله انتخاب متغیر [۴] یک چالش جدی در تحلیل و تفسیر کلان داده‌ها است. هدف از انتخاب متغیر در داده‌های بزرگ کاهش بعد و پیشگویی دقیق می‌باشد. فرآیند انتخاب متغیر در تحلیل و تفسیر مدل‌سازی‌های آماری بسیار حائز اهمیت است زیرا در اغلب کاربردها تعداد زیادی متغیر وجود دارد، که بسیاري از آن‌ها بار اطلاعاتي چندانی ندارند از این رو حذف چنین داده‌هایی خللی در مدل‌سازی ایجاد نمی‌کند و باعث افزایش حجم محاسباتی می‌شوند. علاوه بر این اطلاعات غیرمفید زیادی به همراه داده‌های مفید نیز ذخیره می‌شوند.

روش‌های متفاوتی در بحث انتخاب متغیر تاکنون ارائه شده است و مشکل اغلب این روش‌ها حجم بالای محاسبات است که فرآیند انتخاب متغیر را پیچیده می‌کند همچنین با ظهور فناوری‌های جدید تعداد مسائلی که در آن با حجم عظیمی از متغیرهای توضیحی مواجه هستیم رو به افزایش است، از این رو پژوهشگران به دنبال ابداع روش‌های جدید یا بهبود روش‌های قدیم برای حل مشکلات مربوط به انتخاب متغیر در داده‌ها با بعد بالا هستند.

روش‌های متداول انتخاب متغیر بر روی داده‌های بزرگ بطور موفقیت آمیز اعمال شده‌اند. انتخاب متغیر در بعد بالا و مبتنی بر فرض خطی بودن رابطه متغیر پاسخ با متغیرهای پیشگوهستند. علی‌رغم کاربرد موفق این روش‌ها در بعد بالا برای بسیاری از مجموعه داده‌ها، مشکل آن است که ارزیابی وجود رابطه خطی در بعد بالا کاری بسیار دشوار است و ممکن است در موارد بسیار زیادی این فرض معنی‌دار نبوده و رابطه غیرخطی حاکم باشد؛ لذا استفاده از روش‌های خطی، موثر نخواهند بود.

در این پایان‌نامه ضمن معرفی روشی جدید در انتخاب متغیر بر پایه فرض غیرخطی در داده‌های بزرگ به دنبال متغیرهایی تاثیرگذار خواهیم بود. این الگوریتم چند مرحله‌ای مبتنی بر رگرسیون موضعي خطی، متغیرهاي مهم و تاثیرگذار را شناسايي و انتخاب نموده، سپس مدل پیشگویی را برازش می‌دهد. در فصل‌های آتی الگوریتم را براساس مثال‌های شبیه‌سازی و مطالعات موردی بر روی مجموعه داده‌های واقعی بررسی خواهیم کرد.

---

[۱] Unsuperviesed

[۲] Clustering

[۳] Response

[۴] Feature selection

۱.۱.۱ پیشینه تحقیق

در مسأله‌ی رگرسیون، با هدف پیشگویی متغیر پاسخ y برحسب متغیرهای توضیحی، از حدود ۱۵ سال پیش توجه ویژه‌ای به مسئله‌ی انتخاب متغیر در چارچوب مدل‌های خطی شده است. این تحقیقات با ارائه روش لاسو در سال ۱۹۹۶ آغاز گشت. روش‌هایی چون روش‌های مختصات نزولی[۱](فو [۲]، ۱۹۹۸ ؛ فریدمن و همکاران [۳]، ۲۰۰۷)، smoothly clipped absolute deviation (فن و لی [۴]، ۲۰۰۱)، رگرسیون کمترین زاویه (افرون و همکاران [۵] ، ۲۰۰۴)، شبکه منعطف (ژو و هیستی [۶] ، ۲۰۰۵)، the adaptive lasso (ژو، ۲۰۰۶)، انتخابگر دانتزینگ (کاندس و تائو [۷] ، ۲۰۰۷)، the relaxed lasso (ماینزهازون [۸] ، ۲۰۰۷)، لاسو گروهی (یان و لی، [۹] ۲۰۰۶) و multi-step adaptative lasso (بولمن و مایر[۱۰]، ۲۰۰۸) ارائه شده است. همچنین از جمله مقالات مروری در این زمینه می‌توان به هیستی، تیشیرانی و فریدمن(۲۰۰۹)، فان و ال وی[۱۱] (۲۰۱۰) و بولمن و ون جیرو[۱۲](۲۰۱۱) اشاره کرد.

علی‌رغم کاربرد موفق این روش‌ها در بعد بالا برای بسیاری از مجموعه داده‌ها، مشکل آن است که ارزیابی وجود رابطه خطی در بعد بالا کاری بسیار دشوار است و ممکن است در موارد بسیار زیادی این فرض معنی‌دار نبوده و رابطه غیرخطی حاکم باشد؛ لذا استفاده از روش‌های خطی، موثر نخواهند بود. چالش‌های بسیاری در مسئله‌ی انتخاب متغیر با فرض غیرخطی وجود دارد از جمله تحیقات انجام شده در زمینه انتخاب متغیر غیرخطی می‌توان به مایر، ون جییر و بولمن (۲۰۰۹)، راوی‌کومار و همکاران[۱۳](۲۰۰۹)، هوانگ، هورورویتز و وی [۱۴] (۲۰۱۰) و فراتی، هال و وایو [۱۵] (۲۰۱۰) اشاره کرد.

---

[۱] Coordinate descent methods
[۲] Fu
[۳] Friedman et al
[۴] Fan and Li
[۵] (Efron et al
[۶] Zou and Hastie
[۷] Candes and ‘ Tao
[۸] Meinshausen
[۹] Yuan and Lin
[۱۰] Buhlmann and Meier
[۱۱] Lv
[۱۲] Van de Geer
[۱۳] Ravikumar et al
[۱۴] Huang, Horowitz, and Wei
[۱۵] Ferraty, Hall, and Vieu

۲.۱.۱ ضرورت تحقیق

در بسیاری از روش‌های انتخاب متغیر مطرح شده، رابطه‌ی متغیرهای توضیحی با متغیر پاسخ، خطی فرض می‌شود. اما در بسیاری از موارد، این پذیره معنی‌دار نبوده و ممکن است بین متغیرها و متغیر پاسخ رابطه‌ی غیرخطی حاکم باشد. لذا در چنین مواردی استفاده از روش‌های خطی موثر نخواهد بود. علاوه بر این در روش‌های متداول انتخاب متغیر، منجر به حذف متغیرهایی می‌شوند که ممکن است آن متغیر به تنهایی ارزش کمی داشته باشد اما در ترکیب با دیگر متغیرها اهمیت بیشتری پیدا کند.

الگوریتم پیشنهادی در این پایان‌نامه، در هر گام زیرمجموعه‌ای از بهترین متغیرهای توضیحی را انتخاب می‌کند. در فصل‌های آتی به جزئیات الگوریتم می‌پردازیم.

۲.۱ انتخاب متغیر

انتخاب متغیر در ابعاد بسیار بالا [۱] به یکی از مشکلات اساسی در آمار تبدیل شده است. برای مثال در داده‌های ژنی تعداد مشاهدات ممکن است خیلی کوچک و در عین حال تعداد عبارات بیان ژنی در آن به ده‌ها هزار ژن می‌رسد. بطوری که در بررسی اثرات متقابل این داده‌های ژنی، با میلیون‌ها ژن مواجه خواهیم شد([۳۵] و [۲۰]). در برخورد با این حجم عظیم از داده‌ها، شناسایی ویژگی‌های مهم و پیش‌بینی قابل اعتماد، به شدت احساس می‌شود.

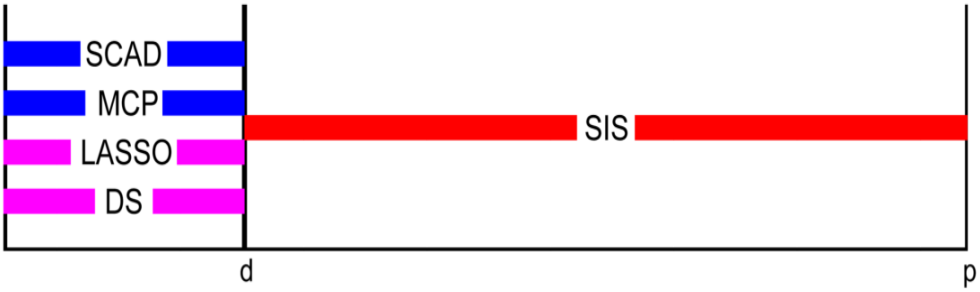
سه مسئله‌ی هزینه‌ی محاسبات، دقت آماری و تفسیرپذیری مدل از جمله مباحث مهم در به کارگیری شیوه‌های انتخاب متغیر یا طرح نظریه‌های جدید در مورد داده‌های با ابعاد بالا و بسیار بالا است.

ایده‌های ارائه شده در زمینه‌ی کاهش بعد به این صورت است که اندازه‌ی بعد داده‌ها را از یک مقیاس بزرگ یا خیلی بزرگ (به عنوان مثال $logp = O(n^a), a > 0$) به یک مقیاس کوچک‌تر d $(O(n^b), b > 0)$، توسط روش‌های قابل اعتماد و کارآمد، ابعاد متغیرها را کاهش دهیم و در ادامه فرآیند کاهش بعد، تکنیک‌های انتخاب متغیر را اجرا می‌کنیم. یکی از ابزارهای قدرتمند انتخاب متغیر در فضای بسیار بالا، استفاده از روش غربالگری متغیر است، که این روش ضمن کاهش بعد در داده‌ها با بعد بالا یا بسیار بالا، قادر به حفظ متغیرهای مهم است[۲۱]. در واقع فن و لیو (۲۰۰۸) روش غربالگری را تحت عنوان SIS [۲] به منظور کاهش بعد در مدل خطی ارائه دادند[۱۹]. این روش با استفاده از روش ضریب همبستگی پیرسون بین متغیرهای توضیحی و متغیر پاسخ، غربالگری را انجام می‌دهد. با استفاده از این روش تعداد متغیرهای توضیحی از بعد بسیار بزرگ p به بعد نسبتا متوسط d کاهش می‌یابد و سپس روش‌های انتخاب متغیر

[۱] Ultra-high dimensional

[۲] Sure Independence Screening

دیگر مانند لاسو[۳۴]، SCAD [؟]، MCP [۹] [1] و دیگر روش‌ها را بر روی متغیرهای توضیحی اجرا می‌کنیم (شکل ۱.۱).



شکل ۱.۱: تصویر طرح انتخاب متغیر در ابعاد بسیار بالا یا ابعاد بالا. نمایش داده‌ها در مقیاس بزرگ p .ابتدا غربالگری متغیرها براساس ضریب همبستگی صورت گرفته و پس از کاهش بعد در مقیاس متوسط d به منظور انتخاب متغیرهای مهم، از روش‌های انتخاب متغیر استفاده می‌شود.

یکی دیگر از روش‌های انتخاب متغیر توسط لی و همکاران (۲۰۱۲) براساس ضریب همبستگی فاصله‌ای بین متغیرهای توضیحی و متغیر پاسخ معرفی شد. آن‌ها ابتدا متغیرهای توضیحی را براساس همبستگی فاصله‌ای با متغیر پاسخ رتبه‌بندی کرده و سپس تعدادی از متغیرهای توضیحی با بیش‌ترین همبستگی فاصله‌ای برای ورود به مدل انتخاب کردند. روش‌های مطرح شده، همگی ابتدا کاهش بعد را انجام داده و پس از کاهش بعد با استفاده از روش‌های انتخاب متغیر، متغیرهای مهم را شناسایی می‌کنند.

اما روش ارائه شده در این پایان‌نامه شیوه‌ی انتخاب متغیر جدیدی را در ابعاد بالا و بسیار بالا مطرح می‌کند که ضمن کاهش بعد، همزمان انتخاب متغیر را براساس مدل‌های غیرخطی انجام می‌دهد. در فصل ۳ به طور کامل الگوریتم انتخاب متغیر پیشنهادی را شرح می‌دهیم.

۳.۱ رگرسیون

رگرسیون یکی از تکنیک‌های پرکاربرد و کلاسیک است که بطور گسترده در علوم مختلف از آن استفاده می‌شود. رگرسیون تقریبا در همه‌ی زمینه‌ها از جمله مهندسی، فیزیک، اقتصاد، مدیریت، علوم زیستی و سایر علوم کاربرد فراوان دارد. هدف آن بررسی اثر متغیرهای مستقل (پیش‌بینی کننده) روی متغیر وابسته (پاسخ) است. رگرسیون از لحاظ نوع تابع بین متغیرهای مستقل و پاسخ، به دو دسته رگرسیون خطی و رگرسیون غیرخطی تقسیم می‌شود.

[1] Measure Correlate Predict

۱.۳.۱ مدل رگرسیون خطی

زمانی که یک متغیر پاسخ روی یک پیش‌بینی کنندهرگرسیون شود، رگرسیون خطی ساده نامیده می‌شود. زمانی که تعداد پیش‌بینی کننده‌ها بیش از یک مورد باشد، این مدل، رگرسیون چندگانه نامیده می‌شود.

۱. رگرسیون خطی ساده

$$y_i = \beta_۰ + \beta_۱ x_i + \varepsilon \quad i = ۱, ۲, ..., n$$

۲. رگرسیون چند گانه

$$y = \beta_۰ + \beta_۱ x_۱ + \beta_۲ x_۲ + ... + \beta_i x_k + \varepsilon$$

۲.۳.۱ مدل رگرسیون غیرخطی

مدل‌های رگرسیونی خطی یک چارچوب وسیع و غنی را در بر می‌گیرد که نیاز تحلیل‌های زیادی را برآورده می‌کند و پاسخ می‌دهد. هرچند رگرسیون خطی برای همه مسائل نمی‌تواند مناسب باشد، زیرا بعضی از اوقات پاسخ و متغیرهای رگرسیونی با تابع غیرخطی معلوم به هم مربوط می‌شوند. مدل‌های رگرسیون غیرخطی به مدل‌هایی گفته می‌شود که حداقل شامل یک پارامتر غیرخطی باشند،

$$Y_i = X_i^{\theta} + \varepsilon_i$$

که θ پارامتر مورد نظر برای برآوردیابی است.

رگرسیون ناپارامتری

به حالتی از رگرسیون که بوسیله‌ی مدل زیر معرفی می‌شود توجه کنید،

$$Y_i = f(x_i) + \varepsilon_i$$

که ε_iها مستقل با میانگین صفر و واریانس $\delta^۲$ هستند و تابع $f(x_i)$ می‌تواند بدون پذیره‌ی فرضیه‌های پارامتری (بدون وابسته بودن به پارامتر) برآورد شود. این نوع رگرسیون، رگرسیون ناپارامتری نامیده می‌شود. انواع روش‌های متفاوت در رگرسیون ناپارامتری به منظور برآورد $f(x)$ وجود دارند. از جمله این روش‌ها می‌توان به رگرسیون موضعی [۱] و اسپلاین هموار [۲] اشاره کرد[۴].

[۱] Local Regression

[۲] Smother Spline

۳.۳.۱ برآورد تابع چگالی به روش هسته

نمونه‌ی تصادفی $X_۱, \cdots, X_n$ را از یک توزیع پیوسته با تابع چگالی احتمال $f(x)$ در نظر بگیرید. برآورد تابع چگالی یک متغیره به روش هسته عبارت است از:

$$\hat{f}(x,\lambda) = \frac{۱}{n\lambda}\sum_{i=۱}^{n} K(\frac{x - X_i}{\lambda})$$

که $K(x)$ تابع هسته نامیده می‌شود. به طوریکه

$$\int_{-\infty}^{+\infty} K(x)dx = ۱ \qquad (۱.۱)$$

باشد. λ یک عدد مثبت است که پارامتر هموارساز [۱] نامیده می‌شود و مقدار آن میزان هموارى در برآورد کرنل را تعیین می‌کند. همچنین $K(.)$ تابعی هموار و پیوسته است و اگر در شرط (۱.۱) صدق کند آن را تابع هسته می‌نامند. معمولا $K(x)$ به صورت یک تابع چگالی متقارن انتخاب می‌شود. در این صورت $\hat{f}(x,\lambda)$ هم یک تابع چگالی احتمال است که تمام ویژگی‌های پیوستگی و مشتق‌پذیری $K(x)$ را دارد. از دیگر ویژگی های تابع هسته، پیوستگی و داشتن واریانس متناهی است. در این پایان‌نامه از نماد λ برای نشان دادن پارامتر هموارسازی در برآوردگر هسته استفاده می‌کنیم[۴].

۴.۳.۱ معرفی برآوردگر هسته

فرض کنید $(x_۱, y_۱), \ldots, (x_n, y_n)$ متغیرهای تصادفی دو بعدی مستقل و هم توزیع با تابع چگالی توأم $f(x,y)$ و تابع توزیع توأم $F(x,y)$ باشند به قسمی که رگرسیون غیرخطی $m(x)$ بصورت زیر موجود است

$$m(x) = E\left(Y_i | X_i = x\right)$$

مدل رگرسیونی بصورت زیر در نظر گرفته می‌شود،

$$Y_i = m(X_i) + \varepsilon_i$$

که ε_iها متغیرهای مستقل با میانگین صفر و واریانس $\delta^۲$ هستند. ما به دنبال برآورد تابع رگرسیونی $m(x)$ هستیم و فرض می‌کنیم توزیع تحت بررسی ناشناخته بوده و دارای فرم پارامتری باشد. بسیاری از نویسندگان مسائل رگرسیون ناپارامتری را مطالعه کرده‌اند از جمله نادارایا (۱۹۶۴)و واتسن (۱۹۶۴) یک برآوردگر هسته‌ای مشهور را پیشنهاد کردند که بسیار مورد استفاده قرار گرفت. این برآوردگر عبارت است از

$$m(x_\circ) = \sum_{i=۱}^{n} \frac{y_i K_\lambda(x_\circ, x_i)}{K_\lambda(x_\circ, x_i)}$$

[۱] Smother

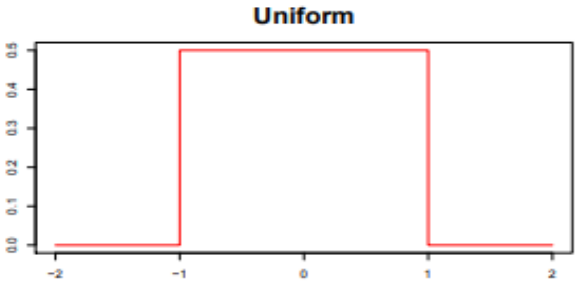
که در آن K به صورت زیر تعریف می‌شود:

$$K_\lambda(x_\circ, x_i) = D\left(\frac{(x_i - x_\circ)}{\lambda}\right) \qquad (۲.۱)$$

در رابطه‌ی (۲.۱)، K یک تابع چگالی متقارن حول صفر است و تابع هسته‌ای نامیده می‌شود و λ پهنای باند یا همان پارامتر هموارساز است. از جمله توابع وزن معروف می‌توان اشاره کرد به،

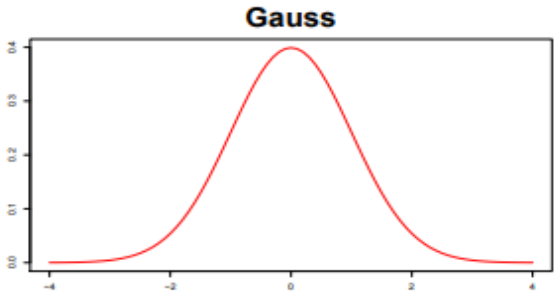
- تابع کرنل یکنواخت

$$K_\lambda(x_i, x_\circ) = \frac{x_i - (x_\circ - \lambda)}{۲\lambda}$$



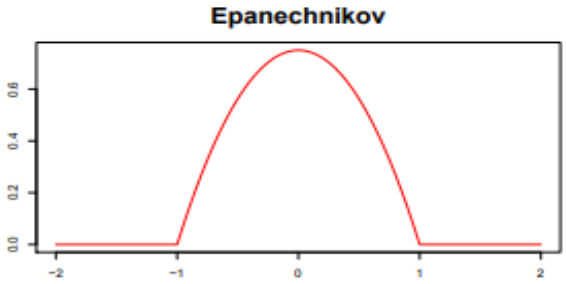
- تابع کرنل گوسی

$$K_\lambda(x_\circ, x_i) = \frac{۱}{\lambda}\exp\left[-\frac{\|x - x_\circ\|^۲}{۲\lambda}\right]$$



- تابع کرنل Epanechnikov

$$K_\lambda(x_i, x_\circ) = \frac{۳}{۴}\left(۱ - \left|\frac{(x_\circ - x_۱)}{\lambda}\right|\right)^۳$$



۵.۳.۱ رگرسیون موضعی

رگرسیون موضعی یکی از انواع روش‌های رگرسیون ساده‌ی ناپارامتری است. در این روش نیز مانند اکثر روش‌های رگرسیون ناپارامتری ناحیه‌ای را که داده‌ها در آن قرار گرفته‌اند به زیرناحیه‌هایی تقسیم می‌کنیم و سپس معادله‌ی خطی را به هر زیر ناحیه برازش می‌دهیم به طوری که نمودارها در انتهای هر زیرناحیه پیوسته باشد. معادله‌ی رگرسیون موضعی برای نقطه‌ی $x_\circ$ وقتی که متغیر مستقل x به $x_\circ$ نزدیک است به صورت زیر تعریف می‌شود [۱۲]:

$$\min_{\alpha(x_\circ),\beta(x_\circ)} \sum_{i=1}^{N} K_\lambda(x_\circ, x_i)[y_i - \alpha(x_\circ) - \beta(x_\circ)x_i]^2$$

معادله‌ی چند جمله‌ای موضعی وقتی که متغیر مستقل x_i به $x_\circ$ نزدیک است به صورت زیر تعریف می‌شود:

$$f(x_i, x_\circ) = \beta_\circ(x_\circ) + \sum_{j=1}^{p} \beta_j(x_i - x_\circ)^2.$$

مقادیر $\{\widehat{\beta}_\circ(x_\circ), \widehat{\beta}_1(x_\circ), \dots \widehat{\beta}_p(x_\circ)\}$ از مینیمم کردن رابطه‌ی زیر بدست می‌آید،

$$RSS = \min \sum_{i=1}^{n} w_i \left(y_i - \widehat{f}(x_i, x_\circ)\right)^2 \qquad (۳.۱)$$

که در آن

$$\widehat{f}(x_i, x_\circ) = \widehat{\beta}_\circ(x_\circ) + \sum_{i=1}^{p} \beta_i(x_\circ)x_i$$

w_i در رابطه‌ی (۳.۱) به صورت زیر تعریف می‌شود:

$$w_i = \frac{K_\lambda(x_\circ, x_i)}{\sum_{i=1}^{N} K_\lambda(x_\circ, x_i)}$$

فرم ماتریسی رابطه (۳.۱) را می‌توان به صورت زیر نوشت

$$RSS = \sum_{i=1}^{n} e_i^2 w_i = (y - \beta B)^T w(x_\circ)(y - \beta B) \qquad (۴.۱)$$

که در آن ماتریس وزن w، β و B بصورت زیر تعریف می‌شوند.

$$w(x_۰) = \begin{pmatrix} K_\lambda(x_۰, x_۱) & ۰ & \dots & ۰ \\ ۰ & K_\lambda(x_۰, x_۲) & \dots & ۰ \\ \vdots & \vdots & \ddots & \vdots \\ ۰ & ۰ & \dots & K_\lambda(x_۰, x_n) \end{pmatrix},$$

$$B = \begin{pmatrix} ۱ & x_۱ \\ \vdots & \vdots \\ ۱ & x_n \end{pmatrix}, \quad \beta(x) = (۱, x)^T$$

با مشتق‌گیری از رابطه‌ی (۴.۱) نسبت به پارامترهای β و معادل صفر قراردادن نتیجه، می‌توان برآوردهای $\hat{\beta}$ و $\hat{f}$ را به صورت $\widehat{f}(x_۰) = x_۰\widehat{\beta}$

$$\min Rss \quad \Longrightarrow \quad \frac{\partial RSS}{\partial \beta} = ۰ \quad \Longrightarrow \quad \widehat{\beta} = (B'wB)^{-۱}B'wy, \quad \widehat{f}(x_۰) = x\widehat{\beta}$$

با جایگذاری $\hat{\beta}$ در $\hat{f}$ داریم

$$\widehat{f}(x_۰) = (b'(x_۰))(B'wB)^{-۱}B'wy$$

اگر قرار دهیم

$$l_i(x_۰) = (b(x_۰))'(B'wB)^{-۱}B'w(x_۰)$$

آن‌گاه

$$\widehat{f}(x_۰) = \sum_{i=۱}^{N} l_i(x_۰)y_i \qquad (۵.۱)$$

که در مجموع l_i ها برابر یک خواهد شد، زیرا

$$B'wB = \begin{pmatrix} ۱ & \dots & ۱ \\ x_۱ & \dots & x_n \end{pmatrix} \begin{pmatrix} k_۱ & \dots & ۰ \\ \vdots & \ddots & \vdots \\ ۰ & \dots & k_n \end{pmatrix} \begin{pmatrix} ۱ & x_۱ \\ \vdots & \vdots \\ ۱ & x_n \end{pmatrix},$$

$$= \begin{pmatrix} k_۱ & k_۲ & \dots & k_n \\ x_۱k_۱ & x_۲k_۲ & \dots & x_nk_n \end{pmatrix} \begin{pmatrix} ۱ & x_۱ \\ \vdots & \vdots \\ ۱ & x_n \end{pmatrix} = \begin{pmatrix} \sum k_i & \sum x_ik_i \\ \sum x_ik_i & \sum x_i^۲k_i \end{pmatrix}$$

بنابراین

$$(B'wB)^{-1} = \underbrace{\frac{1}{\sum k_i \sum x_i^2 k_i - \left(\sum x_i k_i\right)^2}}_{A} \begin{pmatrix} \sum x_i^2 k_i & -\sum x_i k_i \\ -\sum x_i k_i & \sum k_i \end{pmatrix}$$

$$\frac{1}{A}(B'wB)^{-1}B' = \frac{1}{A}\begin{pmatrix} \sum x_i^2 k_i - x_1 \sum x_i k_i & \sum x_i^2 k_i - x_2 \sum x_i k_i & \dots & \sum x_i^2 k_i - x_n \sum x_i k_i \\ -\sum x_i k_i + x_1 \sum k_i & -\sum x_i k_i + x_2 \sum k_i & \dots & -\sum x_i k_i + x_n \sum k_i \end{pmatrix}$$

$$\frac{1}{A}(B'wB)^{-1}B'w = \frac{1}{A}\begin{pmatrix} k_1 \sum x_i^2 k_i - k_1 x_1 \sum x_i k_i & k_2 \sum x_i^2 k_i - k_2 x_2 \sum x_i k_i & \dots & k_n \sum x_i^2 k_i - k_n x_n \sum x_i k_i \\ -k_1 \sum x_i k_i + k_1 x_1 \sum k_i & -k_2 \sum x_i k_i + k_2 x_2 \sum k_i & \dots & -k_n \sum x_i k_i + k_n x_n \sum k_i \end{pmatrix}$$

$$\begin{aligned} \frac{1}{A}(1, x_0)(B'wB)^{-1}B'w &= \frac{1}{A}\begin{pmatrix} k_1 \sum x_i^2 k_i - k_1 x_1 \sum x_i k_i - x_0 k_1 \sum x_i k_i + x_0 k_1 x_1 \sum k_i \\ k_n \sum x_i^2 k_i - k_n x_n \sum x_i k_i - x_0 k_n \sum x_i k_i + x_0 k_n x_n \sum k_i \end{pmatrix}^T \times \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \frac{1}{A}\sum x_i^2 k_i \sum k_i - \sum x_i k_i - x_0 \sum x_i k_i \sum k_i + x_0 \sum k_i \sum x_i k_i = \frac{1}{A} \times A \end{aligned}$$

پس داریم:

$$\sum l_i(x_0) = 1.$$

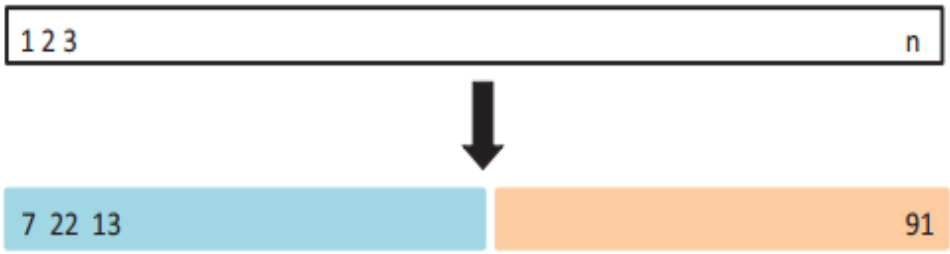
۴.۱ اعتبار سنجی متقابل

اعتبارسنجی [۱] متقابل مشهورترین روش در زمینه‌ی ارزیابی اعتبار، روش‌های آماری است. به عبارت دیگر روشی برای ارزیابی توان (پیشگویی) در یک مدل آماری می‌باشد. این تکنیک به طور ویژه در پیش‌بینی مدل‌های آماری مورد استفاده قرار می‌گیرد تا مشخص شود مدل مورد نظر تا چه اندازه در عمل مفید است.اعتبار سنجی متقابل شامل افراز مجموعه‌ها به دو زیر مجموعه آموزشی [۲]و مجموعه آزمون [۳] است. در این روش برازش را بر روی مجموعه آموزش صورت می‌گیرد و پیش‌بینی با استفاده از مجموعه آزمون صورت می‌گیرد. زمانی که حجم داده‌ها بزرگ باشد، معیارهای (آماره‌ها) برازش مدل برای پیش‌بینی مدل دقیق نیست.

۱.۴.۱ مجموعه‌های آموزشی و آزمون

یک روش برای اندازه‌گیری قدرت پیشگویی یک مدل، آزمودن آن بر روی مجموعه‌ای از داده‌ها است که در برازش مدل مورد استفاده قرار نگرفته است. محققان یادگیری ماشین و داده کاوی به چنین مجموعه‌ای مجموعه آزمون و به مجموعه‌ای که برای برازش مدل استفاده شده است، مجموعه آموزشی می‌گویند. رویکرد اعتبارسنجی که در شکل ۲.۱ نمایش داده شده

[۱] Cross validation
[۲] Testing set
[۳] Training set



شکل ۲.۱: اعتبار سنجی

است، یک استراتژی بسیار ساده از اعتبارسنجی است. در این روش مجموعه داده‌ها را به دو مجموعه تصادفی تقسیم می‌شود، مجموعه‌ی آموزشی، (قسمت آبی شکل ۲.۱ مشاهدات ۷، ۲۲، ۱۳،...) که برازش مدل بر روی آن صورت می‌گیرد و یک مجموعه آزمون (قسمت صورتی شکل ۲.۱ که شامل مشاهدات ۹۱،...) براساس روش یادگیری آماری متناسب با مجموعه آموزش، عملکرد آن را در مجموعه آزمون بررسی می‌کنیم.

استفاده از این رهیافت دارای دو عیب اساسی است:

- در مواردی که حجم داده‌ها کم باشد ممکن است در نظر گرفتن بخشی از آن‌ها به عنوان مجموعه آزمون مناسب نباشد.
- این رهیافت یک آزمایش آزمون کلی است.اگر تقسیم کردن داده‌ها به دو مجموعه آموزش و آزمون به طور مناسب انجام نشود می‌تواند گمراه کننده باشد.

در صورتی که این محدودیت را می‌توان با درنظر گرفتن روش‌های دیگر از مجموعه‌ی آموزشی و آزمون مرتفع کرد:

- اعتبارسنجی Leave-k-out
- اعتبارسنجی K-fold
- اعتبارسنجی Leave-One-Out

اعتبارسنجی K-fold

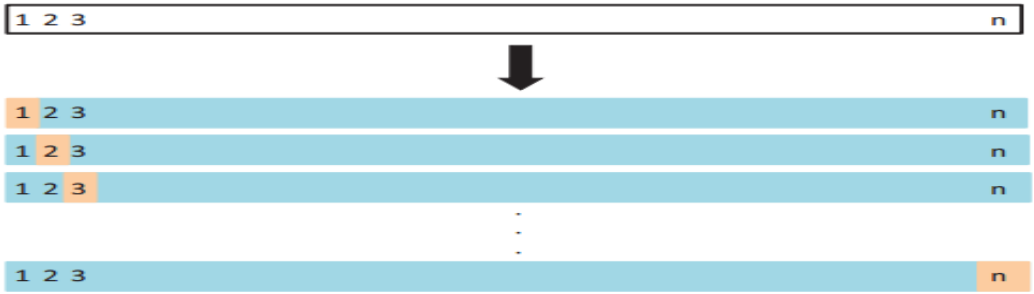
یکی دیگر از روش‌های اعتبارسنجی متقابل، اعتبارسنجی K-fold است که در آن نمونه‌ی اصلی به طور تصادفی به K زیرنمونه تقسیم می‌شود و در هر مرحله، یکی از آن‌ها به عنوان مجموعه‌ی آزمون در نظر گرفته می‌شود.

اعتبارسنجی Leave-k-out

یکی از صورت‌های اعتبارسنجی متقابل، شامل تشکیل مجموعه‌های آزمون با حجم K مشاهده در هر مرحله است.

اعتبارسنجی Leave-One-Out

اعتبارسنجی Leave-one-out رویکردی از روش اعتبارسنجی است که در آن مجموعه‌ی آزمون شامل یک مشاهده و بقیه‌ی مشاهدات $n-۱$ را به عنوان مجموعه‌ی آزمون در نظر گرفته می‌شود. روش اعتبار سنجی Leave-one-out را به اختصار با نماد LOOCV نشان می‌دهند.



شکل ۳.۱: مجموعه‌های آموزشی در اعتبارسنجی Leave-one-out

شکل ۳.۱ رویکردی از اعتبارسنجی LOO را نشان می‌دهد [۲۳]. همان طور که مشاهده می‌کنید $(x_۱,y_۱),\cdots,(x_n,y_n)$ تعداد کل مشاهدات باشد، در روش LOOCV مشاهدات به دو مجموعه تقسیم می‌شوند که مجموعه‌ی آزمون شامل مشاهده‌ی اول $(x_۱,y_۱)$ و مدل را با بقیه‌ی مشاهدات مجموعه آموزش $(n-۱)$، $(x_۲,y_۲),\cdots,(x_n,y_n)$ برازش داده می‌شود. سپس میانگین خطای $MSE_۱ = y_۱ - \hat{y}_۱$ برای مشاهده‌ی حذف شده محاسبه می‌شود. مجددا مشاهده‌ی دوم $(x_۲,y_۲)$ به عنوان مجموعه‌ی آزمون و $n-۱$ دیگر به غیر از مشاهده‌ی دوم $(x_۱,y_۱),(x_۳,y_۳),\cdots,(x_n,y_n)$ به عنوان مجموعه‌ی آموزش در نظر گرفته می‌شود، خطا برای مشاهده‌ی دوم $MSE_۲ = y_۲ - \hat{y}_۲$ محاسبه می‌شود. به همین ترتیب این روند را برای n مشاهده ادامه پیدا می‌کند. برای مشاهده‌ی i ام مجموعه‌ی آزمون (x_i,y_i) و مدل را با بقیه‌ی مشاهدات برازش می‌دهیم سپس خطای اعتبارسنجی $MSE_i = y_i - \hat{y}_i$ برای مشاهده‌ی حذف شده محاسبه می‌شود. برآورد آزمون خطای LOOCV برای n مشاهده بصورت زیر محاسبه خواهد شد:

$$CV_{(n)} = \frac{۱}{n}\sum_{i=۱}^{n} MSE_i$$

این روش به دلیل اینکه در هر مرحله تنها یک مشاهده حذف می‌شود، به غیر از مدل‌های خطی، می‌تواند خیلی زمان‌بر و پرهزینه باشد. اما با استفاده از وزن موضعی پیش‌بینی‌های LOO خیلی

ساده مانند پیش‌بینی‌های معمولی انجام می‌شود، به این معنی که محاسبه‌ی LOOCV زمان بیش‌تری را صرف محاسبه‌ی خطا نمی‌کند و این روش راه بسیار خوبی برای ارزیابی مدل‌ها خواهد بود.

۵.۱ امید ریاضی شرطی

اگر X و Y متغیرهای تصادفی تواما گسسته باشند، تابع احتمال شرطی X به شرط $Y = y$، برای همه‌ی مقادیر y که $P\{Y = y\} > ۰$ ، به صورت زیر تعریف می‌شود:

$$P_{X|Y}(x|y) = P\{X = x|Y = y\} = \frac{p(x,y)}{p_Y(y)}$$

بنابراین طبیعی است که در این حالت امید شرطی X به شرط $Y = y$ را برای همه‌ی مقادیر y که $p_Y(y) > ۰$ به صورت زیر تعریف می‌کنیم[۳]:

$$E[X|Y = y] = \sum_x xP\{X = x|Y = y\}$$
$$= \sum_x xP_{X|Y}\{x|y\}$$

به طور مشابه، اگر X و Y دارای تابع چگالی توأم $f(x,y)$ باشند، تابع چگالی شرطی X به شرط $Y = y$ برای همه‌ی مقادیر y که $f_Y(y) > ۰$ است را بصورت زیر تعریف می‌شود:

$$f_{X|Y}(x|y) = \frac{f(x,y)}{f_Y(y)}$$

در این حالت طبیعی است که امید شرطی X به شرط $Y = y$ را با فرض $f_Y(y) > ۰$ را به صورت زیر تعریف می‌کنیم:

$$E[X|Y = y] = \int_{-\infty}^{+\infty} x f_{X|Y}(x|y)dx \quad = \int_{-\infty}^{+\infty} x\frac{f(x,y)}{f_Y(y)}dx = \int_{-\infty}^{+\infty} x\frac{f(x,y)}{\int_{-\infty}^{+\infty} f(x,y)dx}dx \qquad (۶.۱)$$

فصل ۲

روش‌های معمول انتخاب متغیر در مدل‌های خطی

در فصل اول مفاهیم اولیه در ارتباط با موضوع رگرسیون پارامتری و ناپارامتری را مرور کردیم. در این فصل به بیان روش‌های دیگر انتخاب متغیر که برای مقایسه نتایج با الگوریتم اصلی در این پایان‌نامه از آن استفاده شده است، می‌پردازیم. در این فصل ابتدا به بررسی روش رگرسیون کمترین توان‌های دوم جزئی[۲] پرداخته و سپس الگوریتم آن را شرح می‌دهیم. در ادامه، روش رگرسیون کمترین زاویه که یک روش انتخاب متغیر خطی است، را معرفی می‌کنیم.

۱.۲ رگرسیون کمترین توان‌های دوم جزئی

روش کمترین توان‌های دوم جزئی که به اختصار آن را PLS نیز می‌نامند، یک روش آماری است که اخیراً در زمینه‌های مختلف به خصوص در مسائل شیمی سنجی محبوبیت فراوان پیدا کرده است. این روش در سال ۱۹۶۰ میلادی توسط یک اقتصاددان سوئیسی به نام هرمن ولد در زمینه اقتصادسنجی معرفی و به کار برده شد. هنگامی که تعداد متغیرهای توضیحی بسیار زیاد است و بین متغیرها هم‌خطی شدیدی وجود دارد، رگرسیون PLS در ساختن مدل‌هایی برای پیش‌بینی متغیر پاسخ بسیار مفید است[۲]. این روش یک روش خاص آنالیز در مدل‌هایی

[۲] Partial least squares regression

است که در آن‌ها تعداد متغیرهای توضیحی از نمونه‌ها بیش‌تر است. در حقیقت هدف رگرسیون کمترین توان‌های دوم جزئی پیش‌بینی متغیر پاسخ با مدلی است که براساس ترکیبات خطی از متغیرهای توضیحی تشکیل شده است. رگرسیون کمترین توان‌های دوم جزیی روشی از بازسازی مدل‌های رگرسیونی از نوع

$$\hat{y} = \beta_0 + \beta_1 Z_1 + \cdots + \beta_p Z_p$$

است که Z_i ها ترکیب خطی از متغیرهای توضیحی $x_1, \cdots, x_p$ هستند. اگر چه چندین نسخه از الگوریتم کمترین توان‌های دوم جزئی وجود دارد، ایده‌ی پایه‌ای آن‌ها یکسان است. موضوع مهم انتخاب مولفه‌های Z_i براساس معیارهای مشخص است. بنابراین هدف در رگرسیون کمترین توان‌های دوم جزئی، همانند رگرسیون مولفه‌های اصلی، رگرسیون پیش‌رو کاهش عبارات معادله پیش‌بینی است. در رابطه‌ی

$$\hat{y} = \beta_0 + \beta_1 Z_1 + \cdots + \beta_p Z_p$$

این امکان وجود دارد که تعداد مولفه‌ها کمتر از تعداد متغیرهای توضیحی باشند. در حقیقت رگرسیون کمترین توان‌های جزیی از طریق ساختن مولفه‌ها، مانعی برای استفاده از تعداد زیادی پارامتر در مدل خواهد شد. در روش رگرسیون کمترین توان‌های جزئی مولفه‌های $Z_1, \cdots, Z_p$، i ام مولفه دارای $maxCov(Z_i, y)$ انتخاب می‌شود. مطلوب است که مولفه‌های انتخاب شده نه تنها با y همبستگی بالایی داشته بلکه شامل اطلاعات زیادی از X هستند[۵]. این روش با الگوریتم نسبتا پیچیده‌ای تعریف می‌شود که فهم آن را کمی مشکل می‌کند. در اینجا ابتدا تفسیر ساده‌ای این الگوریتم در حالت دو متغیره بیان می‌شود.

۱.۱.۲ الگوریتم PLS

جزئیات الگوریتم دو متغیره به شرح زیر است:

- : ابتدا ضرب داخلی $\varphi_{11} =< x_1^{(0)}, y >$ را محاسبه کن
- : معادله‌ی $Z_1 = \varphi_{11} X_1 + \varphi_{12} X_2$ را محاسبه کن.
- : $\widehat{\theta} = \frac{<Z_1, y>}{<Z_1, Z_1>}$
- : رگرسیون Z_1 را روی y انجام دهید و $\widehat{y}^{(1)}$ را محاسبه کنید.

$$\widehat{y}^{(1)} = \widehat{y}^{(0)} + \widehat{\theta}_1 Z_1$$

- : باقیمانده حاصل از رگرسیون $x_1^{(0)}$ روی Z_1 و $x_2^{(0)}$ روی Z_1 را به دست آورده و برابر $x_1^{(1)}$ و $x_2^{(1)}$ قرار دهید که مؤلفه یا فاکتورهای جدید هستند. محاسبه‌ی مؤلفه‌های جدید

$$x_1^{(1)} = x_1^{(0)} - (\frac{< Z_1, x_1^{(0)} >}{< Z_1, Z_1 >}) Z_1 \qquad x_2^{(1)} = x_2^{(0)} - (\frac{< Z_1, x_2^{(0)} >}{< Z_1, Z_1 >}) Z_1$$

الگوریتم مربوط به ساختن مولفه‌ها در PLS دو متغیره است. قبل از شروع الگوریتم، متغیرهای توضیحی و پاسخ استاندارد می‌شود. معادله‌ی رگرسیونی به صورت زیر قابل نوشتن است:

$$\widehat{Y} = \beta_0 + \beta_1 Z_1$$

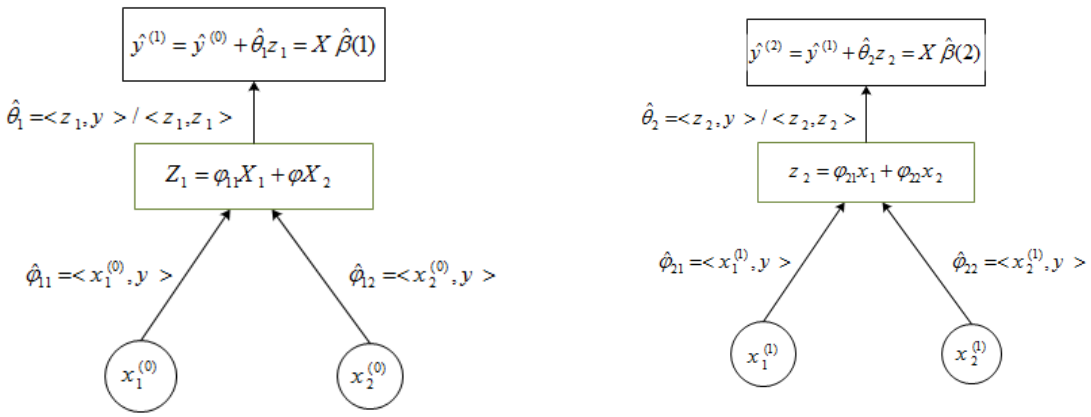
در شرح الگوریتم می‌توان بیان کرد:
ابتدا ضرب داخلی متغیرهای توضیحی با y محاسبه می‌کنیم(گام اول) و معادله‌ی

$$\widehat{y}^{(1)} = \widehat{y}^{(0)} + \widehat{\theta}_1 Z_1$$

را بدست می‌آوریم(گام دوم) اما در این معادله ارتباط بین X_1 و X_2 چشم پوشی شده است، از آنجا که Z_1 یک رابطه‌ی خطی از متغیرهای توضیحی است، خودش نیز یک پیشگوی مفید برای y می‌باشد اما Z_1 قادر نیست همه‌ی تغییرات y را توضیح دهد. آن بخش از تغییرات y که توسط Z_1 توضیح داده نمی‌شود با استفاده از مانده‌های رگرسیون Z_1 روی y قابل توجیح است(گام پنجم). متغیرهای توضیحی به طور بالقوه شامل اطلاعات مفیدتری نسبت به Z_1 در پیش‌بینی y هستند. به این ترتیب آن بخش از اطلاعات موجود در متغیرهای توضیحی که در Z_1 وجود ندارد نیز به وسیله‌ی مانده‌های حاصل از رگرسیون x_1 روی z_1 قابل توجیح است.

مانده‌های حاصل از رگرسیون $x_1^{(0)}$ و $x_2^{(0)}$ روی Z_1 به عنوان متغیرهای جدید در مرحله‌ی بعد به کار می‌رود و گام‌های ۱ تا ۵ تکرار می‌شود. این روند برای ساختن مؤلفه‌های بعدی نیز به همین صورت انجام می‌شود.و هر مؤلفه از مانده‌های حاصل از رگرسیون روی مؤلفه‌های قبلی خود به دست می‌آید.

شکل ۱.۲، فلوچارت الگوریتم PLS دو متغیره به شرح زیر است.



شکل ۱.۲: فلوچارت الگوریتم PLS دو متغیر

۲.۱.۲ چند متغیره PLS

حالت کلی الگوریتم مربوط به PLS چند متغیره به صورت زیر است:

گام اول : استاندارد سازی X_jـ ها با میانگین صفر و واریانس یک در نظر بگیرید.

$$\hat{y}^{\circ} = \bar{y}_{۱}, \quad x_j^{(\circ)} = x_j, \quad j = ۱, ۲, \ldots, p$$

گام دوم : m بار تکرار کن $m = ۱, ۲, \ldots, p$

(a) $Z_m = \sum_{j=۱}^{p} \hat{\varphi}_{m_j} x_j^{(m-۱)}$, که $\hat{\varphi}_{m_j} =< x_j^{(m-۱)}, y >$

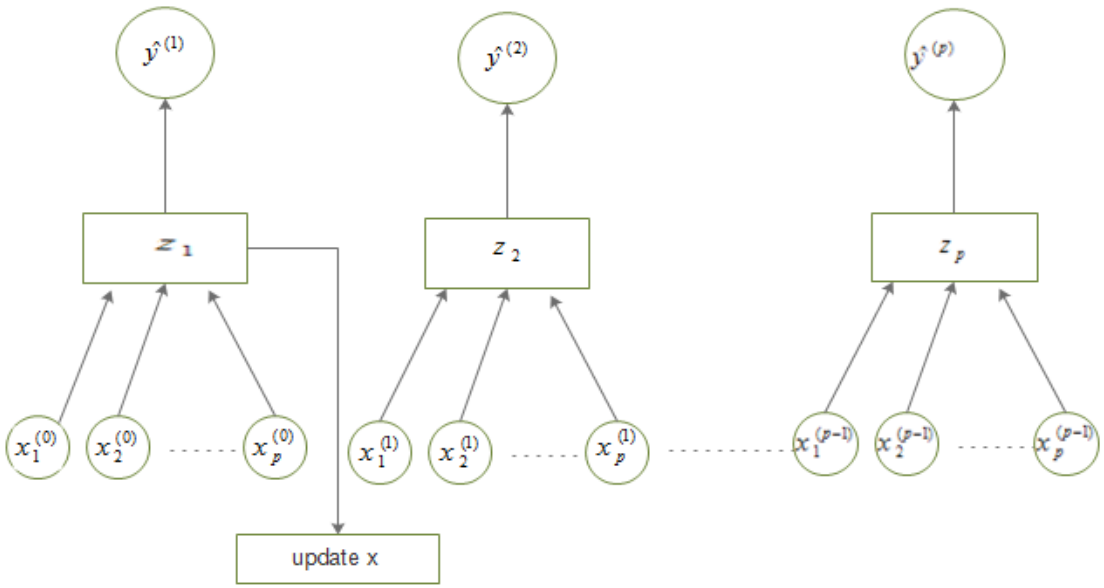
(b) $\hat{\theta}_{(m_۱)} = \frac{<Z_m, y>}{<Z_m, Z_m>}$

(c) $\hat{y}^m = \hat{y}(m-۱) + \hat{\theta}_m Z_m$

(d) باقیمانده حاصل از رگرسیون $x_j^{(m-۱)}$ روی Z به صورت زیر است.

$$x_j^{(m)} = x_j^{(m-۱)} - (\frac{< Z_m, x_j^{(m-۱)} >}{< Z_m, Z_m >}) Z_m$$

شکل ۲.۲، فلوچارت PLS چند متغیره به شرح زیر است:



شکل ۲.۲: فلوچارت الگوریتم PLS چند متغیره

۳.۱.۲ اعتبارسنجی متقابل براساس روش PLS

در روش PLS برای به دست آوردن تعداد مؤلفه‌های موجود در مدل، روشی مشابه روش اعتبار سنجی LOO به کار برده می‌شود. که چگونگی آن به شرح زیر است:
یک نمونه به n بخش تقسیم می‌شود و یک بخش به عنوان مجموعه آزمون حذف می‌شود و از $(n-1)$ بخش دیگر برای ساختن مؤلفه‌ها با استفاده از روش PLS استفاده می‌شود. متغیر Y را روی مؤلفه‌های به دست آمده رگرسیون کرده و پارامترهای مدل را به روش کمترین توان‌های دوم برآورد می‌کنیم. سپس مقادیر مجموعه آزمون را توسط این مدل پیش‌بینی کرده و خطای حاصل از پیش‌بینی را با استفاده از یک تابع زیان مناسب محاسبه می‌کنیم. این عمل را n مرتبه و هر بار برای یکی از n بخش به عنوان مجموعه آزمون تکرار می‌کنیم و مجموع خطاهای پیش‌بینی در این n مرتبه را به عنوان خطای کل مدل در نظر می‌گیریم.
حال مؤلفه‌ها را به مدل اضافه می‌کنیم تا این که مؤلفه‌ای این خطای کل را افزایش دهد. بدین ترتیب تعداد مؤلفه‌هایی که باید در مدل بمانند مشخص خواهند شد [۲].

۲.۲ رگرسیون کمترین زاویه

روش رگرسیون کمترین زاویه که به اختصار آن را LARS [۱] نیز می‌نامند، یک روش آماری است که اولین بار توسط افرون و همکاران در سال ۲۰۰۴ برای انتخاب متغیر براساس مدل خطی در داده‌های بزرگ مطرح شد. رگرسیون حداقل زاویه مدلی جدید براساس رگرسیون پیش‌رو است، این روش نسبت به روش‌های انتخاب متغیر دیگر کم هزینه‌تر و مفیدتر است[۱۱].

در روش رگرسیون پیش‌رو، مرحله‌ی اول متغیری که دارای بیش‌ترین همبستگی با متغیر پاسخ باشد وارد مدل می‌شود و در گام دوم متغیری که بیش‌ترین همبستگی را با باقیمانده‌ی مرحله‌ی اول e_{y,x_j} داشته باشد، به عنوان دومین متغیر وارد مدل می‌شود، این روند زمانی خاتمه می‌یابد که آخرین متغیر وارد مدل شود(منظور کل متغیرها وارد مدل شوند).

رگرسیون حداقل زاویه با رویکردی مشابه رگرسیون پیش‌رو، فرآیند انتخاب متغیر را انجام می‌دهد. در این روش معیار ورود متغیر به مدل همانند روش رگرسیون پیش‌رو است با این تفاوت که آن متغیری وارد مدل می‌شود که دارای ضریب همبستگی برابر با متغیر پاسخ داشته باشد، در گام دوم متغیری که ضریب همبستگی برابر با باقیمانده r_{y,x_k} گام اول داشته باشد به عنوان متغیر دوم وارد مدل می‌شود البته در گام دوم مقدار ضریب همبستگی به طور پیوسته به سمت مقدار همبستگی متغیر x_k حرکت می‌کند و به محض اینکه متغیر دیگری همبستگی برابر با همبستگی متغیر x_k داشته باشد وارد مدل می‌شود. این روند تا $m = min(n-1, p)$ ادامه دارد.

[۱] Least Angle Regression

۱.۲.۲ الگوریتم LARS

روش لارس با رویکردی شبیه به انتخاب پیش‌رو نسبت به ورود متغیرها به مدل عمل می‌کند. بدین ترتیب که در هر گام یک متغیر پیشبین به صورت زیر وارد مدل می‌شود [۱]:

گام اول: متغیرها با میانگین صفر و واریانس یک استانداردسازی شوند.

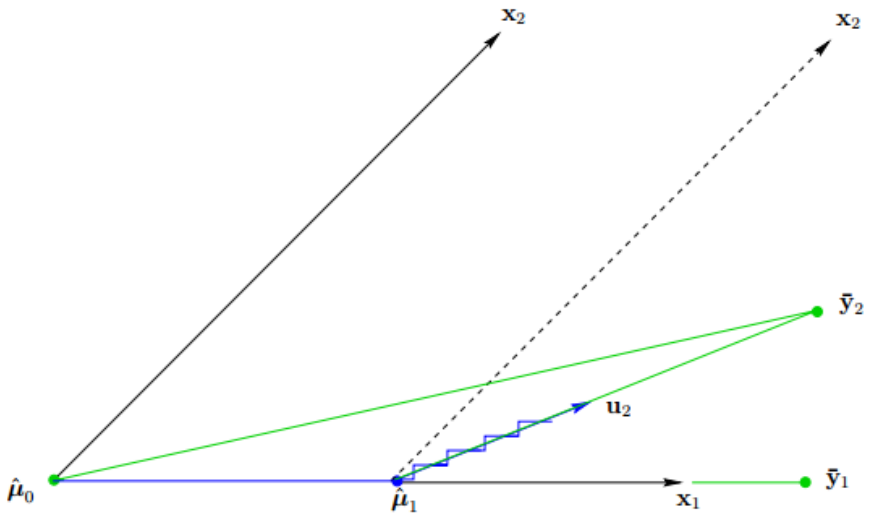
گام دوم: ابتدا همه ضرایب صفر در نظر گرفته می‌شوند، سپس متغیری انتخاب می‌شود که بیش‌ترین همبستگی را با متغیر وابسته داشته باشد، آن را x_j می‌نامیم.

گام سوم: سپس بزرگ‌ترین گام ممکن در جهت این متغیر (با تغییر ضریب مربوط به آن)، تا جایی که همبستگی متغیری دیگر مثل x_k با باقی‌مانده‌ها برابر با همبستگی متغیر x_j با باقی‌مانده‌ها شود برداشته می‌شود (این برخلاف روش پیش‌رو است که در آن تا جایی پیش می‌رویم که همبستگی متغیر x_j با مانده‌ها صفر شود).

گام چهارم: در مرحله‌ی بعد در مسیر متساوی الزاویه بین x_j و x_k پیش می‌رویم، تا جایی که همبستگی متغیری دیگر مثل x_m با باقی‌مانده برابر با همبستگی دو متغیر قبلی با باقی‌مانده‌ها شده و بتواند وارد مدل شود (توجه کنید که با حرکت در مسیر متساوی الزاویه، همبستگی دو متغیر x_1 و x_2 با مانده‌ها به یک میزان کاهش می‌یابد).

گام پنجم: این روند تا اضافه شدن m متغیر که $m = min(n-1, p)$ است، ادامه می‌یابد.

مراحل الگوریتم LARS را در حالت $p = 2$ با کمک شکل ۳.۲ تشریح می‌کنیم[۱]:
در شکل ۳.۲ $X = (x_1, x_2)$ و $\bar{y}_2$ برآورد حداقل مربعات بردار y است. بنابراین $\acute{X}(y - \hat{\mu}) = \acute{X}(\bar{y}_2 - \hat{\mu})$.



شکل ۳.۲: عملکرد روش LARS در حالت $p = 2$

الگوریتم LARS با $\hat{\mu}_0 = 0$ آغاز می‌شود. با توجه به شکل ۳.۲ $\bar{y}_2 - \hat{\mu}_0$ زاویه کمتری با متغیر

x_1 نسبت به x_2 می‌سازد، یعنی همبستگی x_1 با y بیش‌تر است، بنابراین x_1 وارد مدل می‌شود. سپس $\hat{\mu}_0$ در جهت x_1 تصحیح می‌شود یعنی در اولین گام خواهیم داشت:

$$\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma} x_2$$

که در آن $\hat{\gamma}_1$ طوری انتخاب می‌شودکه به ازای آن $\bar{y}_2 - \hat{\mu}_1$ نیمساز زاویه بین $x_1 - \hat{\mu}_1$ و x_2 باشد. اگر بردار u_2 بردار یکه‌ای باشد که در طول نیمساز قرار گیرد آن‌گاه برآورد بعدی رگرسیون کمترین زاویه در گام دوم به فرم زیر است:

$$\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 u_2$$

در این رابطه $\hat{\gamma}_2$ طوری انتخاب می‌شود که همبستگی x_1 و x_2 با مانده‌ها صفر شود، یعنی $\bar{y}_2 = \hat{\mu}_2$. برآوردهای بدست آمده در روش رگرسیون کمترین زاویه تا حدود زیادی به روش لاسو نزدیک است و بر خلاف روش لاسو، در روش LARS برآورد ضرایب نیازی به استفاده از الگوریتم‌های پیچیده ریاضی ندارد[۱].

فصل ۳

روش غیرخطی برای انتخاب متغیر

تاکنون در بحث روش‌های انتخاب متغیر، توجه ویژه‌ای به روش‌های انتخاب متغیر براساس مدل‌سازی خطی شده است. در این فصل به معرفی روشی نوین در انتخاب متغیر بر فرض غیرخطی می‌پردازیم. روش انتخاب متغیر پیشنهادی با ویژگی متمایز، به انتخاب متغیرهایی منجر می‌شود که به تنهایی تاثیر چندانی بر متغیر پاسخ ندارند اما زمانی که با متغیرهای دیگر ترکیب می‌شود تاثیر بسزایی بر متغیر پاسخ می‌گذارد. این گونه متغیرها توسط روش‌های انتخاب متغیر خطی دیگر نادیده گرفته می‌شوند. لذا در این فصل ابتدا توضیحی در مورد ابعاد متغیرها ارائه می‌دهیم و سپس الگوریتم اصلی را مورد مطاله قرار داده و نحوه‌ی انتخاب متغیرانتخاب را شرح می‌دهیم، در نهایت به بحث نظری الگوریتم خواهیم پرداخت. عنوان الگوریتم Nonparametric Variable Selection است، که مخفف عنوان آن را با نماد NOVAS نشان می‌دهیم و در تمامی مفاهیم پایان‌نامه، نام الگوریتم را با عنوان NOVAS مطرح می‌کنیم.

۱.۳ نمادگذاری

از آنجایی که روش انتخاب متغیر پیشنهادی شامل گام‌های متفاوتی است با نمادگذاری‌های متفاوتی روبرو هستیم لذا برای آشنایی با این نمادگذاری‌ها توضیح مختصری در این بخش داده می‌شود.

هر گام از الگوریتم، شامل زیرمجموعه‌های تولید شده از متغیرها است، بدین ترتیب

در الگوریتم مذکور، m بیانگر مراحل الگوریتم است و همچنین به منظور ساده‌سازی، به زیرمجموعه‌های تولید شده در هر گام نیز شماره می‌دهیم در واقع j شماره‌ی زیرمجموعه‌ی تولید شده در هر گام است که به صورت زیر نمایش داده می‌شود:

$$T_j^m = \{X_{j_1}, X_{j_2}, \cdots, X_{j_\ell}\}$$

رابطه بیانگر این است که الگوریتم در گام m شامل ℓ متغیر است و j مین زیرمجموعه تولید شده در گام است. به عنوان مثال زیرمجموعه‌ی زیرا در نظر بگیرید:

$$T_۴^۳ = \{X_۱, X_۹, X_{۱۱}\}$$

در تفسیر رابطه بالا می‌توان گفت الگوریتم در گام ۲ شامل ۳ متغیر است.

فرض کنید Y متغیر پاسخ و $X_۱, X_۲, ..., X_p$ متغیرهای توضیحی باشند. هدف ما مدل‌سازی پاسخ با فرض بزرگ (یا بسیار بزرگ) بودن بعد متغیرها، p، است همچنین فرض کنید Y_i مقدار پاسخ متناظر با مشاهده iام $(X_{i۱}, X_{i۲}, ..., X_{ip})$ باشد به طوری که $i \in M = \{۱, \cdots, n\}$ و $p >> n$. هدف ما آن است که با انتخاب هوشمندانه‌ی موثرترین متغیرها بر پاسخ، بتوانیم پیشگویی را براساس زیرمجموعه‌ی کوچکی از متغیرها انجام دهیم.

۲.۳ خطای اعتبارسنجی متقابل

با توجه به حجم عظیم داده‌ها در بعد بالا با ماتریس‌هایی مستطیلی مواجه می‌شویم، که تعداد ستون‌ها (متغیرها) بیش‌تر از تعدادسطرها (مشاهدات) است $p >> n$. در بحث رگرسیون معمولی تک متغیره، هدف ما بدست آوردن α, β است، که رگرسیون‌های معمولی در برآورد α, β داده‌ها با بعد بالا ناکارآمد است .به همین علت برای برآورد تابع در داده‌ها با بعد بالا از توابع ناپارامتری مانند رگرسیون خطی موضعی [۱]، اسپلاین[۲]ها و دیگر روش‌های غیرخطی استفاده می‌شود. در هر گام از الگوریتم برای برآورد از تابع ناپارامتری رگرسیون خطی موضعی استفاده می‌شود.

فرض کنید Y متغیر پاسخ و $X_۱, X_۲, ..., X_p$ متغیرهای توضیحی باشند. هدف ما مدل‌سازی متغیر پاسخ با فرض بزرگ (یا بسیار بزرگ) بودن بعد متغیرها، p، است همچنین فرض کنید Y_i مقدار پاسخ متناظر با مشاهده iام $(X_{i۱}, X_{i۲}, ..., X_{ip})$ باشد به طوری که $i \in M = \{۱, \cdots, n\}$ و $p >> n$. برآورد ناپارامتری متغیر پاسخ براساس رگرسیون موضعی خطی را با $\hat{\gamma}$ نشان می‌دهیم. الگوریتم روش پیشنهادی انتخاب متغیر به صورت سلسله‌مراتبی و در چندین مرحله انجام می‌گیرد به طوری که در مرحله m ام، زیرمجموعه‌های ℓتایی از متغیرهای توضیحی به صورت $\{X_{j_1}, X_{j_2}, \cdots, X_{j_\ell}\}$ در نظر گرفته شده و برآورد LOO تابع $\hat{\gamma}$ و خطای مربوطه محاسبه می‌شود.

[۱] Local Linear Regression

[۲] Spline

به عبارت دیگر $\hat{\gamma}^{-i}(X_{j_1},\cdots,X_{j_\ell})$ را برآورد بدست آمده از مجموعه مشاهدات $M-\{i\}$ فرض نموده و خطای اعتبارسنجی متقابل مربوط به آن مجموعه متغیر را به صورت زیر خواهد بود:

$$S(j_1,...,j_\ell)=\sum_{i=1}^{n}\{Y_i-\hat{\gamma}^{-i}_{j_1,...,j_\ell}(X_{ij_1},...,X_{ij_\ell})\}^2 w_l(X_{ij_1},...,X_{ij_\ell}) \qquad (۱.۳)$$

با توجه به رابطه‌ی (۱.۳) مشاهده می‌کنید که تابع $\hat{\gamma}^{-i}_{j_1,...,j_\ell}(X_{ij_1},...,X_{ij_\ell})$ یک برآوردگر LOOCV از مجموعه‌ی $M-\{i\}$ است. در اعتبارسنجی LOOCV، همه مشاهدات به غیر از مشاهده x_i نقش دارند، یعنی مشاهده (x_{ij_1},y_i) کنار گذاشته می‌شود و مدل را بر روی بقیه مشاهدات (یعنی مجموعه $M-\{i\}$) برازش می‌شود، که برآوردگر متناظر آن را با نماد $\hat{\gamma}^{-i}_{j_1,...,j_\ell}(X_{ij_1},...,X_{ij_\ell})$ نشان می‌دهیم. برآوردگر LOOCV از لحاظ اجرای الگوریتم خیلی زمان‌بر و پر هزینه است، اما زمانی که با رگرسیون موضعی برازش صورت گیرد، هزینه و زمان اجرای الگوریتم کاهش می‌یابد. $w(.)$ وزن مشاهده i ام متغیرهای $\{X_{j_1},X_{j_2},\cdots,X_{j_\ell}\}$ است که براساس توابع هسته متداول در بحث رگرسیون موضعی تعریف می‌شود (w_ℓ تابع نامنفی که به سمت یک میل می‌کند).

همان طور در بخش ۱.۳ اشاره شد به منظور ساده‌سازی در نمادگذاری فرض می‌کنیم $T=\{X_{j_1},...,X_{j_l}\}$ یک زیر مجموعه از $\{X_1,...,X_p\}$ باشد. فضای برداری $u=(u_1,\cdots,u_p)$ را روی یک میدان R^p تعریف می‌شود. با توجه به ساده‌سازی نماد گذاری ، u^T را ساده شده‌ی زیرمجموعه‌ی $(u_{j_1},\cdots,u_{j_\ell})$ بیان می‌کنیم.اکنون خلاصه شده‌ی رابطه (۱.۳) بصورت زیر است:

$$S(T)=\sum_{i=1}^{n}\{Y_i-\hat{\gamma}^{-i}_{\tau}(X_i^T)\}^2 w_{|T|}(X_i^T) \qquad (۲.۳)$$

که در آن $|T|$ اندازه‌ای از مجموعه‌ی T است.

$T_1,\cdots,T_j$، شامل زیرمجموعه‌ای از متغیرها است، که در هر زیرمجموعه، تعداد متفاوتی از متغیرها وجود دارد. با توجه به رابطه (۱.۳)، برای هریک از زیرمجموعه‌ها مقادیر

$$S(T_1),\cdots,S(T_j)$$

محاسبه می‌شود. در ادامه آنها به طور صعودی مرتب کرده، و در هر گام بهترین زیرمجموعه براساس خطای اعتبارسنجی انتخاب می‌شود، مراحل الگوریتم ادامه می‌یابد تا زمانی که بهترین زیرمجموعه از متغیرها که دارای کمترین خطای اعتبارسنجی $S(T)$ است، انتخاب شود.جزییات بیشتر در بخش ۳.۳ پرداخته می‌شود.

۳.۳ مروری بر اجماعی الگوریتم پیشنهادی

روش انتخاب متغیر پیشنهادی شامل مراحل متفاوتی است و در هر مرحله تعدادی متغیر برای ورود به مرحله‌ی بعد انتخاب می‌شود. در این بخش علاوه بر مرور اجماعی الگوریتم، نحوه‌ی جدید کاهش بعد در داده‌های بزرگ مطرح می‌شود. الگوریتم به شرح زیر است:
در گام نخست تمامی متغیرها به صورت زیرمجموعه‌های تک عضوی $\{1\}, ..., \{p\}$ مورد بررسی قرار می‌گیرند و تعدادی از متغیرهای موثر بر پاسخ، برای ورود به گام دوم انتخاب می‌شوند.
گام دوم شامل ترکیبات دوتایی متغیرهای موثر است که به صورت زیرمجموعه‌های دوعضوی $\{X_{j_{\ell-1}}, X_{j_\ell}\}$ مورد واکاوی قرار می‌گیرند.
با انجام یک غربالگری مجدد، موثرترین ترکیبات دوتایی برای ورود به گام بعدی شناسایی می‌شوند. گام سوم که از اجتماع زیرمجموعه‌های دو عضوی گام قبل بدست می‌آید، موجب تشکیل زیرمجموعه‌های ۳ یا ۴ عضوی از متغیرها به صورت $\{X_{j_1}, X_{j_2}, X_{j_3}, X_{j_4}\}$ می‌شود، با یک روش تکراری شبیه به قبل، موثرترین متغیرها در هر گام شناسایی می‌شوند و برای ورود به گام بعدی انتخاب می‌شود در نهایت با اعمال یک شرط توقف، این روند با انتخاب بهترین زیرمجموعه از متغیرها خاتمه می‌یابد.

همان طور که بیان شد متغیرهای تشکیل دهنده در هر گام از اجتماع زیرمجموعه متغیرهای مرحله‌ی قبل بدست می‌آید؛ لذا اگر همه‌ی متغیرها در هر مرحله وارد شود و بخواهیم بر روی همه‌ی متغیرها الگوریتم اعمال شود، در این صورت با حجم بزرگ محاسباتی $O(p^2)$ مواجه می‌شویم. بررسی بر روی این حجم از داده‌ها دارای محاسبات پیچیده، سنگین و گران خواهد بود. اکنون با ترفندی به دنبال کاهش تعداد متغیرها در هر مرحله هستیم. به این منظور در گام اول بعد از رتبه‌دهی متغیر و مرتب کردن آن‌ها بطور صعودی، تنها $\sqrt{p}$ متغیر اول را برای تشکیل مجموعه‌های دوعضوی انتخاب می‌شود. بنابراین تنها تعداد $O(\sqrt{p}^2) = O(p)$ زیر مجموعه از این طریق ساخته می‌شود. در نهایت تعداد مجموعه‌های دوعضوی از متغیرها نیز کاهش یافته و محاسبات برای انتخاب متغیر در گام دوم آسان‌تر صورت می‌گیرد.

به طور کلی فضای $O(p)$ توصیف خوبی برای کران‌های بالا در منابع محاسباتی نیست و توانایی محاسبات الگوریتم را محدود می‌کند، به این دلیل از فضای $O(q)$ به عنوان کران که دارای منابع محاسباتی نسبتا گسترده است، در نظر گرفته می‌شود.
q ممکن است با افزایش n، با مقادیر بیش‌تر از p به سمت p نزدیک می‌شود و یا با مقادیر کمتر از p به p نزدیک می‌شود. در واقع q به سمت p میل می‌کند. این استدلال مشکلاتی را ایجاد می‌کند لذا $q = p$ در نظر گرفته و در محاسبات الگوریتم از فضای $O(q)$ استفاده می‌شود.
در این صورت الگوریتم از میان p متغیر در گام اول، $\sqrt{q}$ متغیر اول برای ورود به گام دوم انتخاب می‌شوند که منجر به تشکیل زیرمجموعه‌های دو عضوی می‌شود و این روند در گام‌های بعدی که منجر به تشکیل زیرمجموعه‌هی سه یا چهار عضوی یا بیشتر از چهار عضو خواهد شد(و متعاقبا برای ساختارهای پیش‌بینی در مراتب بالاتر صدق می‌کند).

مرحله‌ی اول الگوریتم شامل همه‌ی متغیرهای اصلی است. برای ورود به گام دوم $\sqrt{p}$ متغیر از متغیرهای اصلی انتخاب می‌شوند اما ممکن است بعضی از این متغیرها کارآمد و مفید نباشد اما همراه با $\sqrt{p}$ متغیر اول وارد مرحله‌ی بعد شوند اکنون الگوریتم به دنبال پیدا کردن متغیرهای پنهان در میان $\sqrt{p}$ است.
پیدا کردن این ویژگی پنهان در میان متغیرها کاری بس دشوار است لذا این محدودیت را با استفاده از روش $\sqrt{p}$ و زیرمجموعه‌های به وجود آمده در هر مرحله انجام می‌شود . مزیت رویکرد این روش بدین صورت است :
در مرحله‌ی اول الگوریتم اگر یک متغیر در میان $\sqrt{p}$، زیرمجموعه‌های تک عضوی مفید باشد، در مراحل بعدی الگوریتم با استفاده از تکنیک $\sqrt{p}$، به مرحله‌ی بعد ورود پیدا می‌کند. همان طور که مطرح شد زیرمجموعه از اجتماع زیرمجموعه‌های گام قبل بدست می‌آیند این روش تشکیل زیرمجموعه باعث می‌شود تا فرصتی برای شناسایی آن دسته از متغیرهایی که تنها زمانی تاثیر چندانی ندارند اما در ترکیب با دو یا چند متغیر دیگر نتایج سودمندی خواهند داشت.
در روش‌های متداول انتخاب متغیر، انتخاب ویژگی براساس روش‌های خطی است. متغیرهایی که به تنهایی تاثیر ندارند اما در ترکیب با دیگر متغیرها تاثیرگذارند. در روش‌های خطی نادیده گرفته شوند. انتخاب متغیر در الگوریتم مطرح شده براساس روش غیرخطی است که این روش می‌تواند جایگزین مناسبی برای حل مشکل در روش‌های خطی باشد.

۴.۳ جزییات الگوریتم

در بخش ۳.۳ به مرور اجماعی الگوریتم و نحوه‌ی کاهش بعد در هر گام از الگوریتم پرداخته شد. اکنون به جزییات الگوریتم و نحوه‌ی دقیق انتخاب متغیر در هر مرحله از الگوریتم پرداخته می‌شود.

گام اول(پیش‌بینی براساس تک متغیره):

گام نخست الگوریتم شامل غربالگری تکی متغیرها است به این صورت که رگرسیون موضعی را بر هر یک از مجموعه متغیرهای

$$T_۱ = \{۱\}, T_۱ = \{۲\}, \cdots, T_۱ = \{p\}$$

اجرا کرده و مقادیر خطای اعتبارسنجی یعنی

$$S(T_۱), S(T_۲), \cdots, S(T_p)$$

را بدست آورده و آنها را مرتب می‌کنیم. فرض کنید $T_{(۱)}, T_{(۲)}, \cdots, T_{(p)}$ مجموعه مرتب شده‌ی متغیرها برحسب خطای اعتبارسنجی S هستند بطوریکه

$$S(T_{(۱)}) \leq S(T_{(۲)}) \leq \cdots \leq S(T_{(p)})$$

که

$$S(T_{(j)}) = \arg\min_j S(T_j)$$

$$S(T_j) = \sum_{i=1}^{n} (y_i - \hat{\gamma}_j^{-i}(X_j))^2 w_\ell(X_{ij}) \quad j = 1, \cdots, p \qquad (۳.۳)$$

$\hat{\gamma}_j^{-i}$ تابعی یک متغیره و برآورد LOO رگرسیون موضعی بدست آمده از داده‌های $M - \{i\}$ است. در اینجا می‌توان گفت که $T_{(1)}$ بهترین پیشگو در غربالگری یک متغیره است و بعد از آن $T_{(2)}, T_{(3)}, \cdots$ در جایگاه‌های بعدی واقع خواهند شد.
متغیرهای $T_{(1)}, T_{(2)}, T_{(3)} \cdots$، موثرترین متغیرها در گام ۱ نامیده شده و از آنها برای شروع گام دوم استفاده می‌کنیم. یک پیشنهاد برای انتخاب تعداد متغیرهای موثر در گام اول به منظور ورود به گام دوم $p_1 = \sqrt{q}$ است که در بعد بسیار بالا بسیار مفید است به طوری که اگر $p = 10,000$ آن‌گاه فقط ۱۰۰ متغیر اول موثر هستند. به عنوان مثال، با داشتن۱۶ متغیر $X_1, \cdots, X_{16}$، اگر

$$S(T_{(6)}) \leq S(T_{(1)}) \leq S(T_{(15)}) \leq \cdots$$

آن‌گاه متغیرهای X_{15}، X_1 ، X_6 موثرترین متغیرها نامیده شده و X_6 بهترین پیشگو در گام اول است.

گام دوم (پیش‌بینی براساس دو متغیره):

در این گام، به منظور کاهش هزینه‌ی محاسبات و دقت در نتایج محاسبات به دنبال محدود کردن فضای $O(p)$ فضای کل متغیرها هستیم. بدین منظور فضای را به $O(q)$ تغییر می‌دهیم. در مرحله‌ی قبل پس از مرتب کردن خطای اعتبار سنجی تعداد $p_1 = \sqrt{q}$ از مرحله‌ی اول برای ورود به مرحله‌ی دوم انتخاب کرده، بدین ترتیب هنگام بررسی اثرات متقابل یا زیرمجموعه‌های دو عضوی متغیرها با حجم عظیمی از داده‌ها مواجه نیستیم.غربالگری متغیرها به صورت ترکیبات دو تایی انجام می‌شود، تعداد مجموعه‌ها در این مرحله از طریق $k = \frac{1}{2} p_1(p_1 - 1)$ محاسبه می‌شود. با فرض اینکه در این گام ترکیبات دوتایی موثرترین متغیرهای گام ۱ مد نظر باشد داریم

$$\begin{array}{llll} T_1^2 = T_{(1)} \cup T_{(2)} & T_{p_{\ell+1}}^2 = T_{(2)} \cup T_{(3)} & & \\ T_2^2 = T_{(1)} \cup T_{(3)} & T_{p_{\ell+2}}^2 = T_{(2)} \cup T_{(3)} & \cdots & T_k^2 = T_{(p_1-1)} \cup T_{(p_1)} \\ \vdots & \vdots & & \cdots \\ T_{p_1}^2 = T_{(1)} \cup T_{(p_1)} & \vdots & & \end{array}$$

به عنوان مثال برای ۴ متغیر موثر در گام ۱ داریم:

$$\begin{array}{lll} T_{(1)}^2 = \{6,1\} & T_{(3)}^2 = \{6,10\} & T_{(5)}^2 = \{1,12\} \\ T_{(2)}^2 = \{6,15\} & T_{(4)}^2 = \{14,2\} & T_{(6)}^2 = \{3,8\} \end{array}$$

با انجام رگرسیون موضعی بر هریک از مجموعه متغیرهای $T_۱^۲, T_۲^۲, \cdots, T_k^۲$ و محاسبه خطای اعتبارسنجی

$$S(T_۱^۲), S(T_۲^۲), \cdots, S(T_k^۲)$$

و مرتب‌سازی آنها می‌توان بهترین زوج پیشگو و موثرترین زوج‌های پیشگو را تعیین کرد. فرض کنید $T_{(۱)}^۲, T_{(۲)}^۲, \cdots, T_{(k)}^۲$ مجموعه مرتب شده بر حسب خطای S هستند بطوریکه

$$S(T_{(۱)}^۲) \le S(T_{(۲)}^۲) \le \cdots \le S(T_{(k)}^۲)$$

$$S(T_{(j)}) = \arg\min_j S(T_j)$$

$$S(T_j^۲) = \sum_{i=۱}^{n} (y_i - \hat{\gamma}_j^{-i}(X_\ell))^۲ w_\ell(X_{ij}) \quad j = ۱, \cdots, p \qquad (۴.۳)$$

که $\hat{\gamma}_j^{-i}$ تابع دو متغیر و برآورد LOO رگرسیون موضعی y بر زوج متغیر X_ℓ, X_m $(T_j^۲ = \{\ell, m\})$ مبتنی بر داده‌های $M - \{i\}$ است. در اینجا $T_{(۱)}^۲$ شامل بهترین زوج پیشگو و

$$T_{(۱)}^۲, T_{(۲)}^۲, T_{(۳)}^۲, \cdots, T_{(q)}^۲$$

موثرترین زوج متغیرها نامیده می‌شود و بدین صورت غربالگری‌های زوج متغیرها نامیده می‌شود $p_۱ = p_۲ = \sqrt{k}$.

در ادامه‌ی مثال قبل فرض کنید

$$S(T_۴^۲) \le S(T_۶^۲) \le S(T_۳^۲) \le S(T_۵^۲) \le S(T_۱^۲) \le S(T_۲^۲)$$

$$T_{(۱)}^۲ = T_۴^۱ \cup T_۱^۱ = \{۱, ۱۵\}$$
$$T_{(۲)}^۲ = T_۶^۱ \cup T_۲^۱ = \{۱۰, ۱۵\}$$
$$T_{(۳)}^۲ = T_۳^۱ \cup T_۵^۱ = \{۶, ۱۰\}$$
$$T_{(۳)}^۲ = T_۳^۱ \cup T_۵^۱ = \{۶, ۱۰\}$$

$p_۱ = p_۲ = \sqrt{k}$ از متغیرها پس از غربالگری برای ورود به گام سوم انتخاب می‌شوند

گام سوم (پیش‌بینی براساس سه یا چهار متغیره):

در این گام غربالگری متغیرها به صورت ترکیبات (مجموعه) سه یا چهارتایی انجام می‌شود که تعداد این مجموعه‌ها همانند مراحل قبل با استفاده از رابطه‌ی $k_۱ = \frac{۱}{۲} p_۲(p_۲ - ۱)$ محاسبه می‌شود. با فرض اینکه در این گام، ترکیبات سه یا چهارتایی موثرترین متغیرهای گام ۲ مد نظر باشد داریم

$$\begin{array}{llll}
T_۱^۳ = T_{(۱)} \cup T_{(۲)} & T_{p_{\ell+۱}}^۲ = T_{(۲)} \cup T_{(۳)} & & \\
T_۲^۳ = T_{(۱)}^۲ \cup T_{(۳)}^۲ & T_{p_{\ell+۲}}^۳ = T_{(۲)}^۲ \cup T_{(۳)}{}^۲ & \cdots & T_{k_۱}^۳ = T_{(p_۲-۱)}^۳ \cup T_{(p_۲)}{}^۳ \\
\quad \vdots & \quad \vdots & & \cdots \\
T_{p_۲}^۳ = T_{(۱)}^۲ \cup T_{(p_۲)} & & \vdots &
\end{array}$$

به عنوان مثال برای ۴ زیرمجموعه‌یی دو عضوی موثر در گام دوم داریم

$$T_1^3 = T_3^2 \cup T_6^2 = \{10, 6, 15\}$$

$$T_2^3 = T_4^2 \cup T_6^2 = \{10, 1, 15\}$$

$$T_3^3 = T_4^2 \cup T_3^2 = \{10, 1, 15, 6\}$$

با انجام رگرسیون موضعی بر هریک از مجموعه متغیرهای

$$T_1^3, T_2^3, T_3^3 \cdots, T_{p_2}^3$$

و محاسبه‌ی خطای اعتبارسنجی

$$S(T_1^3), S(T_2^3), S(T_3^3), \cdots, S(T_{p_2}^3)$$

و مرتب‌سازی آنها می‌توان بهترین زوج متغیر پیشگو و موثرترین زوج‌های پیشگو را تعیین کرد. فرض کنید

$$T_{(1)}^3, T_{(2)}^3, T_{(3)}^3 \cdots, T_{(p_2)}^3$$

مجموعه مرتب شده برحسب خطای S هستند بطوریکه

$$S(T_{(1)}^3) \leq S(T_{(2)}^3) \leq S(T_{(3)}^3) \leq \cdots \leq S(T_{(p_2)}^3)$$

$$S(T_{(j)}) = \arg\min_j S(T_j)$$

$$S(T_j^2) = \sum_{i=1}^{n} (y_i - \hat{\gamma}_j^{-i}(X_\ell, X_m, X_n))^2 w_3(X_\ell, X_m, X_n) \quad j = 1, \cdots, p \qquad (۵.۳)$$

که $\hat{\gamma}_j^{-i}$ تابعی سه یا چهار متغیره در برآورد LOO رگرسیون موضعی y بروی مجموعه متغیرهای سه یا چهار عضوی $(T_j = \ell, m, n)$ X_ℓ, X_m, X_n مبتنی بر داده‌های $M - \{i\}$ است.در اینجا $T_{(1)}^3$ شامل بهترین زیرمجموعه از متغیرها و

$$T_{(1)}^3, T_{(2)}^3, T_{(3)}^3 \cdots, T_{(p_2)}^3$$

موثرترین زیرمجموعه متغیرهای سه یا چهار عضوی نامیده می‌شود و بدین صورت غربالگری‌های زیرمجموعه متغیرهای سه یا چها عضوی نیز انجام می‌شود.

در گام m ام رگرسیون موضعی را بروی زیرمجموعه‌های $T_1^{m-1}, T_2^{m-1}, \cdots, T_{p_m}^{m-1}$ اجرا کرده. تعداد زیرمجموعه‌های هر کدام بصورت

$$T_k^m = \{j_1, j_2, j_3, \cdots\} \qquad m \leq |T| \leq 2^{m-1}$$

در هر مرحله تعداد

$$p_\ell^* \leq \frac{۱}{۲} p_{\ell-۱}(p_{\ell-۱}-۱)$$

زیرمجموعه بدست می‌آید و از میان p_ℓ^* اول از زیرمجموعه‌های جدید بوجود آمده، انتخاب می‌کنیم.تعداد اعضای زیرمجموعه‌ها به دلیل اجتماع دو زیرمجموعه در هر مرحله در حال افزایش است و مقادیر خطای اعتبارسنجی یعنی

$$S(T_۱^{m-۱}), S(T_۲^{m-۱}), \cdots, S(T_{p_m}^{m-۱})$$

را بدست آورده و آنها را همانند مراحل قبل مرتب می‌کنیم فرض کنید $T_{(۱)}^{m-۱}, T_{(۲)}^{m-۱}, \cdots, T_{(p_m)}^{m-۱}$ مجموعه‌ی مرتب شده‌ی متغیرها برحسب خطای S هستند به طوریکه

$$S(T_{(۱)}^{m-۱}) \leq S(T_{(۲)}^{m-۱}) \leq \cdots \leq S(T_{(p_\ell)}^{m-۱})$$

که

$$S(T_{(j)}) = \arg\min_j S(T_j)$$

$$S(T_j^۲) = \sum_{i=۱}^{n} (y_i - \hat{\gamma}_j^{-i}(X_{j_۱}, X_{j_۲}, \cdots, X_{j_\ell}))^۲ w_\ell(X_{j_۱}, X_{j_۲}, \cdots, X_{j_\ell})) \quad j = ۱, \cdots, p \quad (۶.۳)$$

$\hat{\gamma}_j^{-i}$ تابعی ℓ متغیر و برآورد LOO رگرسیون موضعی بدست آمده از داده‌های $M-i$ است. در انتهای هر مرحله بهترین پیشگوها انتخاب می‌شود $T_۱^m = T_۱, ..., T_{p_۱^*}^m = T_{p_۱^*}$. این روند ادامه دارد تا زمانی که الگوریتم متوقف شود. الگوریتم زمانی متوقف می‌شود که تفاوت دو مقدار خطا کمتر از یک مقدار از پیش تعیین شده تعیین می‌شود:

$$\frac{S(T_۱^m) - S(T_۱^{m+۱})}{S(T_۱^m)} \leq t \quad (۷.۳)$$

$t = t(n)$ یک آستانه قابل انتخاب توسط کاربر می‌باشد که حداقل مقدار خطا را برای ورود به مرحله‌ی بعدی تعیین می‌کند.

۱.۴.۳ آستانه‌ی t

آستانه[۱]ی (t) به عنوان یک پارامتر برای توقف الگوریتم می‌باشد. این آستانه در مراحل الگوریتم به کار می‌رود. در واقع پس از پایان هر مرحله از الگوریتم رابطه (۷.۳) با مقدار t مقایسه می‌شود. اگر مقدار رابطه (۷.۳) کمتر از مقدار t باشد، الگوریتم وارد گام بعد می‌شود و بطور پیش فرض از مقدار $t = ۰/۰۵$ در الگوریتم استفاده می‌شود.

[۱] THRESHOLD

۵.۳ خواص نظری

در این بخش به مباحث نظری الگوریتم و اثبات آن‌ها می‌پردازیم.

مروری بر الگوریتم :

همان طور که در بخش ۴.۳ مطرح شد، الگوریتم پیشنهادی در هر گام زیرمجموعه‌هایی از متغیرهای توضیحی (p ـ بعد) را رتبه‌دهی می‌کند. تعداد اعضای زیرمجموعه‌های انتخاب شده در هر گام از الگوریتم متغیر است و فرایند تشکیل زیرمجموعه تا زمانی که الگوریتم خاتمه یابد، ادامه دارد. در نظر بگیرید

$$T = \{j_1, \cdots, j_l\} \subseteq \{1, \cdots, p\}$$

T زیرمجموعه‌ای از متغیرها است که در هر گام (T در هر مرحله وابسته نیست و میزان تعداد متغیر در زیرمجموعه متفاوت است) انتخاب می‌شوند. و آن را با نماد $T^{\ell}_{j_\ell}$ نمایش می‌دهیم.
در مراحل بعدی الگوریتم همان طور که در بخش ۴.۳ بیان شد. زیرمجموعه‌های تشکیل شده در هر گام شامل ترکیبی (اجتماع)از بهترین زیرمجموعه‌ها موجود در گام قبل است.

در هر مرحله پس از تعیین زیرمجموعه‌های جدید دو عضوی، سه عضوی، چهارعضوی و زیرمجموعه‌ها در گام ℓ الگوریتم شامل

$$l \leq |T| \leq 2^{l-1}$$

عضو متغیر خواهند بود، این زیرمجموعه‌ها را براساس خطای اعتبار سنجی رتبه دهی می‌شوند. این روند پردازش تا زمانی که الگوریتم زیرمجموعه‌ای از متغیرها که دارای کمترین مقدار خطای اعتبارسنجی $S(T)$ باشد، ادامه پیدا می‌کند.

در مورد فضای برداری p ـ بعدی مطرح شده، مجموعه‌ی $u = (u_1, \cdots, u_p)$ از فضای میدانی R^p را به عنوان فضای کل متغیرها معرفی کردیم. u^T یک زیرمجموعه‌ای به طول ℓ در نظر میگیریم:

$$u^T = (u_{j_1}, u_{j_2}, ..., u_{j_\ell})$$

جزییات بیشتر در مورد مجموعه u^T [۱۵] آمده است.

۱.۵.۳ محاسبه‌ی امید ریاضی و واریانس γ

در این بخش ابتدا توابعی را که برای اثبات امید ریاضی و واریانس γ به لحاظ احتمالی لازم است مطرح می‌کنیم.

فرض کنید f تابع چگالی بردار p ـ متغیره باشد. همچنین تابع $\phi_{j_1,\cdots,j_\ell}(x_1, \cdots, x_p)$ تابعی از زیرمجموعه‌های تولید شده در هر گام از الگوریتم است. در واقع تابع ϕ متناسب با تابع

چگالی احتمال f است. به عبارت دیگر

$$\phi_{j_1,\cdots,j_\ell}(x_1,\cdots,x_p) \approx f(x_1,\cdots,x_p)w_\ell(x_{j_1},\cdots,x_{j_\ell}) \qquad (۸.۳)$$

که w_ℓ تابع وزن (هسته) نامنفی، همگرا به یک است. همچنین تابع $\Psi_{j_1,\cdots,j_\ell}(x_{j_1},\cdots,x_{j_\ell})$ انتگرال تابع $\phi_{j_1,\cdots,j_\ell}(x_1,\cdots,x_p)$ به صورت زیر تعریف می‌شود:

$$\int \phi_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell}) = \psi_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell}) \quad i \in \{\ j_1,\ldots,j_\ell\} \qquad (۹.۳)$$

لم ۱.۵.۳. فرض کنید تابع $\phi_{j_1,\cdots,j_\ell}(x_1,\cdots,x_p)$ ، با تابع چگالی f متناظر باشد در این صورت می‌توان فرض کرد که مقادیر تابع $\gamma_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell})$ برابر با مقادیر امید شرطی $E(g(X_i)|X_{ij_1}=x_{ij},\ldots,X_{ij_\ell}=x_{ij})$ خواهد بود.

برهان. در رابطه‌ی (۱۰.۳) مشاهده می‌کنید که انتگرال‌گیری را نسبت به x'' صورت گرفته و x'' شامل بردار $p-\ell$ می‌باشد. لذا مطابق با آنچه بیان شد و عبارت (۶.۱) ، می‌توان تابع γ را براساس امید شرطی نوشت:

$$\gamma_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell}) = \frac{\int g(x_1,\ldots,x_r)\phi_{j_1,\ldots,j_\ell}(x_1,\ldots,x_p)dx''}{\psi_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell})} \qquad (۱۰.۳)$$

رابطه‌ی (۱۰.۳) را می‌توان به صورت زیر بدست آورد:

$$\begin{aligned} E(g(X_i)|X_{ij_1}=x_{ij},\ldots,X_{ij_\ell}=x_{ij_\ell}) &= \int g(x_1,\ldots,x_r)f_{g(x_1,\ldots,x_r)|x_{j_1},\ldots,x_{j_\ell}}(x_1,\ldots,x_p)dx'' \\ &= \int g(x_1,\ldots,x_r)\frac{f_{g(x_1,\ldots,x_r)|x_{j_1},\ldots,x_{j_\ell}}(x_1,\ldots,x_p)}{\int f_{x_{j_1},\ldots,x_{j_\ell}}(x_1,\ldots,x_p)}dx'' \end{aligned}$$

با توجه به رابطه (۸.۳) دو تابع f و ϕ متناسب هستند لذا می‌توان نوشت:

$$E(g(X_i)|X_{ij_1}=x_{ij_1},\ldots,X_{ij_\ell}=x_{ij_\ell}) = \int \frac{g(x_1,\ldots,x_r)\phi_{j_1,\ldots,j_\ell}(x_1,\ldots,x_p)}{\int \phi_{j_1,\ldots,j_\ell}(x_1,\ldots,x_p)}dx'' \qquad (۱۱.۳)$$

همان طور که در ابتدا بیان شد در صورتی که تابع $\phi_{j_1,\cdots,j_\ell}(x_1,\cdots,x_p)$ ، با تابع چگالی f متناظر باشد در این صورت می‌توان فرض کرد که مقادیر تابع $\gamma_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell})$ برابر با مقادیر امید شرطی $E(g(X_i)|X_{ij_1}=x_{ij},\ldots,X_{ij_\ell}=x_{ij})$ است و همچنین مطابق با رابطه‌ی (۹.۳) نتیجه می‌شود

$$\gamma_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell}) = \int \frac{g(x_1,\ldots,x_r)\phi_{j_1,\ldots,j_\ell}(x_1,\ldots,x_p)}{\psi_{j_1,\ldots,j_\ell}(x_{j_1},\ldots,x_{j_\ell})}dx'' \qquad (۱۲.۳)$$

□

در رابطه (۱۲.۳) انتگرال بر روی بقیه‌ی متغیرها به جز متغیرهای انتخابی شده صورت می‌گیرد به عبارت دیگر آن متغیرهای انتخاب شده $(x_{j_1},...,x_{j_\ell})$ از میان $(x_1,...,x_p)$ حذف می‌شوند.
واریانس برای یک متغیر بصورت زیر تعریف می‌شود:

$$u_0 = E\{g(X) - Eg(X)\}^2$$

و اکنون واریانس را برای زیرمجموعه‌های انتخابی متغیرها محاسبه می‌کنیم

$$u_\ell(j_1,...,j_\ell) = \int \{g(x_1,...,x_r) - \gamma_{j_1,...,j_\ell}(x_{j_1},...,x_{j_\ell})\}^2 \times \phi_{j_1,...,j_\ell}(x_1,...,x_p)dx_1...dx_p$$

فرض کنید برای هر زیرمجموعه از M_ℓ در میان فضای برداری R^ℓ در نظر گرفته می‌شود.

۲.۵.۳ برآورد تابع γ

در این بخش به نحوه‌ی برآورد تابع γ می‌پردازیم با توجه به آن چه در فصل ۱ در مورد رگرسیون موضعی و نحوه‌ی برآورد آن بیان شده است در این بخش نحوه‌ی بدست آوردن تابع $\hat{\gamma}$ را بیان می‌کنیم.
در بخش ۲.۳ $\hat{\gamma_{j_l}}$ را به عنوان برآورد رگرسیون خطی موضعی بیان کردیم نحوه‌ی برآورد $\hat{\gamma_{j_l}}$ بصورت زیر است:

$$\hat{\gamma}_{j_1,...,j_l}(x) = \bar{Y} + \{\bar{X} - x\}^T \hat{\sum}^{-1} \{\bar{X} - x\}$$

مقادیر $\bar{Y}$، $\bar{X}$ و $\hat{\sum}$ به صورت زیر محاسبه می‌شوند

$$\bar{X} = \frac{\sum_i K\{(x-X_j)/h\}X_i}{\sum_i K\{(x-X_j)/h\}} \quad , \quad \bar{Y} = \frac{\sum_i K\{(x-X_j)/h\}Y_i}{\sum_i K\{(x-X_j)/h\}}$$

$$\hat{\sum} = \frac{\sum_i \{(X_i - X)(X_i - X)^T\}K\{(x - X_i)/h\}X_i}{\sum_i K\{(x - X_i)/h\}}$$

که $\hat{\Sigma(x)}$ ماتریس کوواریانس است. در ماتریس $\Sigma = \hat{Cov}(x)$ ماتریس کوواریانس $n \times p$ برای x است. که شامل واریانس‌های $x_1, \cdots, x_p$ روی قطر اصلی و کوواریانس هر x_i با هر x_j $i \neq j$ در خارج قطر اصلی قرار دارند.

$$\hat{\sum} = \begin{pmatrix} \sigma^2_{X_1} & \sigma^2_{X_1X_2} & \cdots & \sigma^2_{X_1X_p} \\ \sigma^2_{X_2X_1} & \sigma^2_{X_2} & \cdots & \sigma^2_{X_2X_p} \\ \vdots & \vdots & \vdots & \vdots \\ \sigma^2_{X_1X_p} & \sigma^2_{X_2X_p} & \cdots & \sigma^2_{X_nX_p} \end{pmatrix}$$

$K(u_1,...,u_\ell)=K(u_1),...,K(u_\ell)$ ، K_1 یک تابع یکنواخت و کران‌دار با تابع چگالی احتمال متقارن و h نیز پهنای باند است.

مجموعه متغیرهای تشکیل شده در هر گام $x_1,x_2,\ldots,x_\ell$ مستقل از یکدیگر و دارای تابع چگالی $g(x)$ است. w_ℓ یک تابع وزن نامنفی متقارن از مجموعه M_ℓ است.تابع $\gamma_{j_1,\cdots,j_\ell}$ زمانی دارای عملکرد مناسب است که برآورد متغیرها ی

$$1\le j_1\le ...\le j_\ell\le r\quad 1\le \ell\le r-1$$

برابر با صفر و تقریبا نزدیک به مقدار g انتظار می‌رود. زمانی که برآورد متغیرها صفر شود بیان کننده‌ی این موضوع است که متغیرهای بی اثر را در هر مرحله حذف می‌کند. از آنجایی که هیچ یک از متغیرها $X=(X_1,\cdots,X_p)$ بطور خاص مزیت قابل توجهی نسبت به دیگر متغیرها ندارد به همین دلیل هیچ فرصتی برای انتخاب متغیر صحیح وجود ندارد ، برای حل این مشکل، تابع وزن‌های

$$w_1,w_2,\cdots,w_\ell$$

را در نظر گرفته می‌شود، این تابع وزن‌ها متقارن و از آن‌ها در رابطه‌ی

$$S(j_1,...,j_\ell)=\sum_{i=1}^{n}\{Y_i-\hat{\gamma}^{-i}_{j_1,...,j_\ell}(X_{ij_1},...,X_{ij_\ell})\}^2 w_l(X_{ij_1},...,X_{ij_\ell})$$

استفاده شده است. همچنین برای حذف متغیرهای بی‌اثر در هر مرحله از الگوریتم از تابع وزن w_ℓ استفاده می‌شود، لذا شرط بردار $(j_1,j_2,\cdots,j_r)$ انتخاب شده از متغیرها

$$1\le j_1\le ...\le j_\ell\le p\quad 1\le \ell\le r$$

منحصربه فرد هستند و خطای حاصل از این زیرمجموعه‌ی انتخاب شده نیز به طور منحصربه فرد نیز دارای کمترین خطا است. با توجه به شرط مذکور، به تنهایی تضمین کننده‌ی توقف الگوریتم در رابطه‌ی

$$\frac{S(T_1^\ell)-S(T_1^{\ell+1})}{S(T_1^\ell)}\le t$$

نیست در واقع با استفاده از این شرط نمی‌توان اثبات کرد که مجموعه انتخاب شده به طور منحصربه فرد دارای کمترین خطا است. با توجه به شرط زیر

- برای هر مقدار $\eta>0$ و $1\le\ell\le r$ شماری از ℓـ بردار متمایز $(j_1,\cdots,j_\ell)$ با $1\le j_1<\ldots<j_\ell\le p$ که $u_\ell(j_1,\cdots,j_\ell)>n^{\eta-\{4/(\ell+4)\}}$ از مقدار $\sqrt{q}$ خیلی کوچکتر است
- برای هر $1\le\ell\le r$ شامل همه‌ی ℓـ بردار اعداد صحیح انتخاب شده از $1,\cdots,r$ است.

می‌توان بیان کرد که مجموعه انتخاب شده دارای کمترین خطای اعتبارسنجی است . با این شرط و دیگر شرط‌های مطرح شده در (فراتی و هال؛۲۰۱۵) [۱۵]، با احتمال بالا می‌توان بیان کرد که زیرمجموعه‌های تک عضوی

$$\{j_1\}, \cdots, \{j_p\}$$

و زیرمجموعه‌های دو عضوی (j_1, j_2) $1 \leq j_1 \leq j_2 \leq r$ به همین ترتیب تا مرحله rـ چند متغیره $(1, 2, \ldots, r)$ با احتمال یک همگرا هستند.

قضیه ۱.۵.۳. $g(X_i) = E(Y_i|X_i)$ یک تابع از r تا از اولین متغیرها در بردار X_i است. الگوریتم در مرحله‌ی r با احتمال همگرایی به یک وقتی که $n \longrightarrow \infty$ می‌یابد متوقف می‌شود.

برهان. اثبات به [۱۵] مراجعه شود □

فصل ۴

مطالعه شبیه‌سازی و داده‌های واقعی

در فصل سوم، روش انتخاب متغیر در ابعاد بالا و ابعاد بسیار بالا مبتنی بر رهیافت غیرخطی و توسط رگرسیون موضعی معرفی شد. در این فصل برای ارزیابی عملکرد روش پیشنهادی، مجموعه داده‌هایی در ابعاد بالا و بسیار بالا را شبیه‌سازی کرده و توانایی الگوریتم را به منظور انتخاب صحیح متغیرها بررسی می‌کنیم. سپس موضوع توقف اجرای الگوریتم را با مقادیر متفاوت برای آستانه توقف، مورد ارزیابی قرار می‌دهیم. همچنین الگوریتم را بر روی مجموعه داده‌ی واقعی اجرا نموده و نتایج بدست آمده را با دیگر روش‌های انتخاب متغیر مقایسه می‌کنیم.

۱.۴ شبیه‌سازی

الگوریتم بیان شده در این پایان‌نامه از لحاظ محاسباتی بسیار قوی است و اجرای آن با یك مجموعه داده‌ی بزرگ ممکن است بسیار زمان‌بر باشد. لذا راه حل پیشنهادی برای بالا بردن سرعت محاسبات، موازی‌سازی الگوریتم است. منظور از موازی‌سازی بدین معنا است که می‌توان هر بار قسمتی از برنامه را روی پردازنده‌های متفاوت اجرا کرد و در آخر برای کسب نتیجه‌ی مطلوب، نتایج را کنار هم قرار داد. اجرای روش مذکور با استفاده از رایانه‌های دارای پردازشگر چندهسته‌ای امکان‌پذیر است که البته بیشتر رایانه‌های امروزی دارای چنین ویژگی هستند. این ویژگی شرایطی را فراهم می‌کند تا به توان برنامه‌های غیرمستقل را همزمان

پردازش کرد.
برای اجرای روش انتخاب متغیر پیشنهادی، از زبان برنامه‌نویسي R، که توسط گروهی از متخصصان علم آمار (تیم هسته توسعه نرم افزار R [۳۱]) توسعه یافته است، استفاده می‌شود. این نرم افزار رایگان بوده و بر روی انواع سیستم عامل‌ها قابل نصب است. از دیگر مزایای این نرم افزار فراهم کردن زمینه‌ای بی نظیر برای اجرای روش‌های آماری است و قابلیت توسعه همچون دیگر نرم افزارهای باز را دارد.

در اینجا از بسته‌ی doSNOW که یک ساختار جدید موازی‌سازی در محاسبات شبیه‌سازی نرم افزار R است، استفاده می‌شود[۳۳]. در این ساختار عملیات تکرار بر روی پردازنده / هسته‌های متعدد رایانه اجرا می‌شود که موجب کاهش در زمان اجرای الگوریتم است. تمامی نتایج شبیه‌سازی و مجموعه داده‌های واقعی که در این پژوهش ارائه شده است با استفاده از یک رایانه ۵ هسته‌ای با پردازشگر ۲ GHz و ۴ گیگابایت حافظه بدست آمده است.

۱.۱.۴ شبیه‌سازی مدل‌های رگرسیونی

اکنون به منظور بررسی کارآیی الگوریتم روش انتخاب متغیر، از مدل زیر برای شبیه‌سازی استفاده می‌کنیم.

$$Y = \gamma(X_1, ..., X_p) + \varepsilon$$

از آنجا که ساختارهای مختلفی برای γ در نظر گرفته شده است، لذا برای سهولت در ارجاع، آن را به صورت زیر در نظر می‌گیریم:

$$Y^m = \gamma^{[m]}(X_1 \cdots, X_p) + \varepsilon^{[m]} \qquad , m = 1, \cdots, 5 \qquad (۱.۴)$$

که $[m]$ بیانگر شماره‌ی مدل است. ساختارهای مختلف تابع γ به صورت زیر درنظر گرفته می‌شود:

[1] R Development Core Team

$$\gamma^{[1]}(X_1,\cdots,X_p) = X_1^2 + X_2^2 + X_3^2$$

$$\gamma^{[2]}(X_1,\cdots,X_p) = |X_1X_2| + |X_1X_3| + |X_2X_3|$$

$$\gamma^{[3]}(X_1,\cdots,X_p) = |X_1X_2X_3| \qquad (۲.۴)$$

$$\gamma^{[4]}(X_1,\cdots,X_p) = \frac{|X_1X_2|+X_{13}^2}{2+X_1X_2X_3}$$

$$\gamma^{[5]}(X_1,\cdots,X_p) = \frac{|X_1X_2|+|X_1X_3|}{2+|X_2X_3|}$$

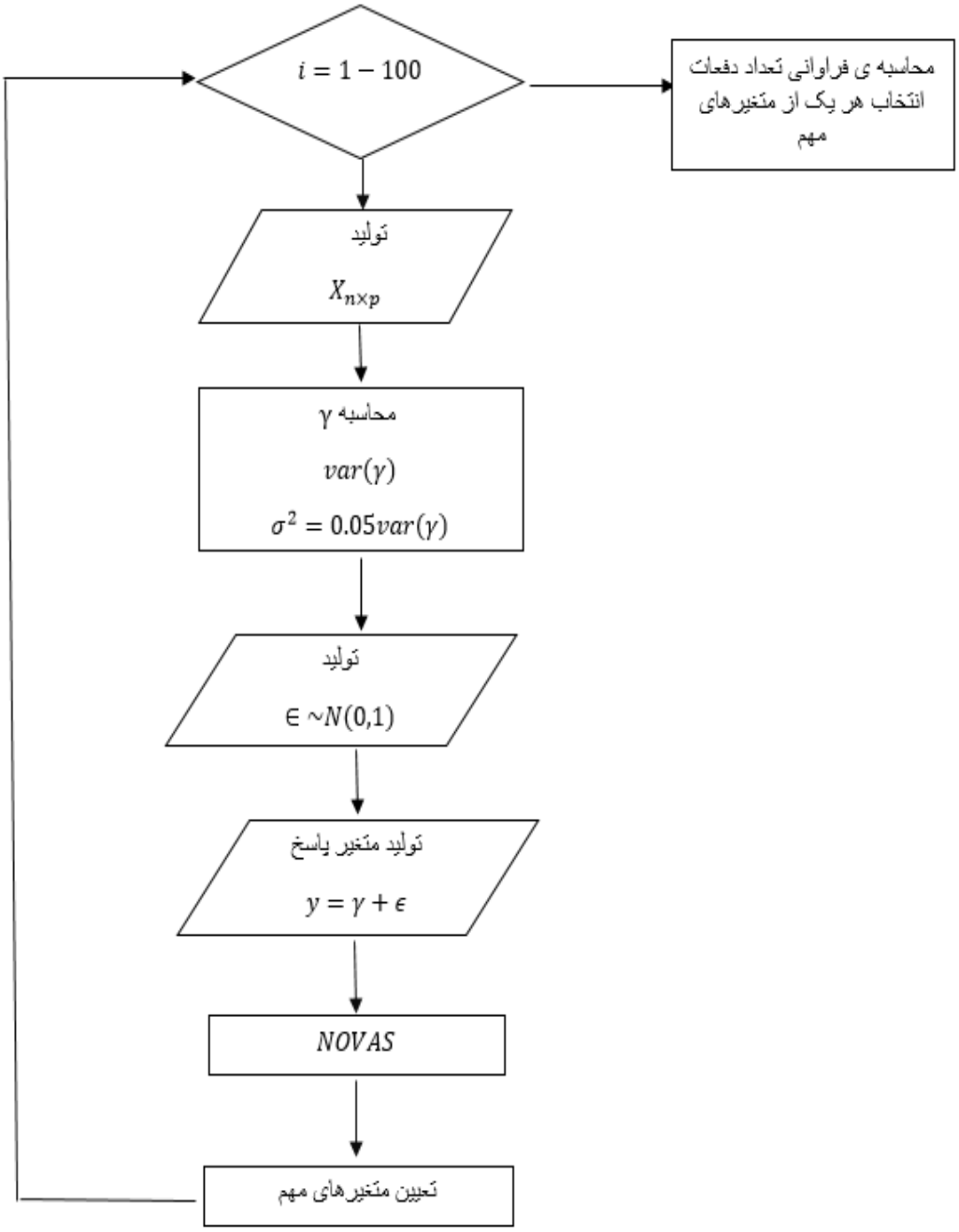
ساختار تمامی مدل‌های در نظر گرفته شده برای تابع γ، و در نتیجه متغیر پاسخ، تنها وابسته به سه متغیر X_1, X_2, X_3 است که این سه متغیر به عنوان متغیرهای با اهمیت، و بقیه متغیرها بی‌اهمیت در نظر گرفته شده‌اند. در اینجا هدف ما بررسی توانایی الگوریتم NOVAS در شناسایی سه متغیر با اهمیت در مدل است.

برای ارزیابی عملکرد الگوریتم، مقادیر مختلف از حجم نمونه $n = 50, 100, 150, 200$ را برای p های مختلف $p = 100, 1000, 10000$ که معرف ابعاد متوسط، بالا، بسیاربالا است نیز در نظر می‌گیریم. با توجه به فلوچارت ۱.۴، جزئیات شبیه‌سازی مدل‌های مختلف به قرار زیر است:

- مقادیر p متغیرتوضیحی به طور مستقل براساس توزیع یکنواخت $U[-1,1]$ در اندازه n تولید می‌شوند؛
  - براساس مدل‌های در نظر گرفته شده (۲.۴)، $\gamma^{[m]}$ محاسبه می‌شود؛
  - مقادیر خطا $\varepsilon_i^{[m]}$ از توزیع نرمال $N(0, \sigma_m^2)$ تولید می‌شوند که

$$\sigma_m^2 = 0/05(var\{\gamma_{1,2,3}^m(X_1, X_2, X_3)\})$$

- مقادیر متغیر پاسخ توسط رابطه $y = \gamma + \varepsilon$ تولید می‌شوند.
- برای پایش اثرات تصادفی داده‌ها، مراحل فوق به تعداد ۱۰۰ بار انجام شده و فراوانی تعداد دفعاتی که الگوریتم توانسته متغیرهای X_1, X_2, X_3 را به درستی انتخاب کند، محاسبه می‌شود؛



شکل ۱.۴: فلوچارت بررسی الگوریتم NOVAS توسط داده‌های شبیه‌سازی

۲.۱.۴ نتایج شبیه‌سازی

در این بخش نتایج اعمال الگوریتم NOVAS را بر داده‌های شبیه‌سازی مورد ارزیابی قرار می‌دهیم. همان طور که در (۲.۴) مشخص است تنها سه متغیر $\{X_1, X_2, X_3\}$ در ساخت متغیر پاسخ تاثیرگذارند و بقیه متغیرها بی‌اهمیت هستند و هدف ما آن است که ببینیم آیا

روش پیشنهادی قادر به تشخیص این سه متغیر است یا خیر.

برای هر یک از مدل‌های (۲.۴)، با (n,p) مشخص و از قبل تعیین شده، الگوریتم NOVAS را اجرا نموده و متغیرهای انتخاب شده را تعیین می‌کنیم. سپس به منظور پایش خطای تصادفی، این کار را ۱۰۰ بار تکرار کرده و تعداد دفعاتی که متغیرهای مهم $\{X_1, X_2, X_3\}$ انتخاب شده‌اند را محاسبه می‌کنیم. برای انجام تحلیل حساسیت روش نسبت به مقادیر مختلف اندازه نمونه n و تعداد متغیرها p، اعمال مذکور را برای n های مختلف و همچنین ابعاد متوسط (۱۰۰)، ابعاد بالا (۱۰۰۰) و ابعاد بسیار بالا (۱۰۰۰۰) نیز به طور مشابه تکرار می‌کنیم. در هر اجرا، مقدار $t = ۰/۰۵$ برای شرط توقف الگوریتم در نظر گرفته شده است. فلوچارت ۱.۴ اجرای الگوریتم و نتایج در جداول ۱.۴ تا ۵.۴ ثبت شده‌اند.

الف: مدل ۱

$$\gamma^{[1]}(X_1, \cdots, X_p) = X_1^2 + X_2^2 + X_3^2 \qquad (۳.۴)$$

همان‌طور که در جدول ۱.۴ مشاهده می‌شود الگوریتم NOVAS توانسته است متغیرهای مهم $\{X_1, X_2, X_3\}$ در مدل ۱ که دارای ساختاری بسیار ساده است را به ازای $n = ۱۰۰, ۱۵۰, ۲۰۰$ و $p = ۱۰۰, ۱۰۰۰, ۱۰,۰۰۰$ در هر یک از ۱۰۰ بار تکرار به طور همزمان به درستی شناسایی کند، یعنی در بعد بسیار بالا، الگوریتم برای n های ۱۰۰, ۱۵۰, ۲۰۰ بدون خطا عمل کرده است، اما در حجم نمونه‌ی بسیار کم $n = ۵۰$ و $p = ۱۰۰۰$، متغیرهای $\{X_1, X_2, X_3\}$ به ترتیب ۹۴، ۹۲ و ۹۵ بار انتخاب شده‌اند ولی ۸۲ بار از ۱۰۰ بار تکرار، هر سه متغیر به طور همزمان شناسایی شده‌اند و به ازای $p = ۱۰۰۰$ نیز متغیرهای $\{X_1, X_2, X_3\}$ به ترتیب ۸۷، ۸۵ و ۸۲ بار انتخاب شده‌اند که فقط ۶۸ بار، هر سه متغیر به طور همزمان شناسایی کرده است همچنین برای $p = ۱۰۰۰۰$ نیز سه متغیر $\{X_1, X_2, X_3\}$ به ترتیب ۴۷، ۳۶، ۴۳ بار شناسایی شده‌اند این در حالی است که تنها ۱۶ متغیر به طور همزمان انتخاب شده است. با توجه به نتایج بدست آمده از الگوریتم به ازای حجم نمونه‌ی بسیار کم $n = ۵۰$ در ابعاد متوسط، بالا و بسیار بالا، نشان دهنده‌ی عملکرد خوب الگوریتم در ابعاد بسیار بالا است.

جدول ۱.۴: تعداد دفعاتی که متغیرهای X_1, X_2, X_3 مدل ۱، در ۱۰۰بار تکرار، به عنوان متغیر مهم شناسایی شده‌اند

	n=۵۰	n=۱۰۰	n=۱۵۰	n=۲۰۰
مدل ۱				
p=۱۰۰	۸۲	۱۰۰	۱۰۰	۱۰۰
p=۱۰۰۰	۶۸	۱۰۰	۱۰۰	۱۰۰
p=۱۰۰۰۰	۱۶	۱۰۰	۱۰۰	۱۰۰

ب: مدل ۲

$$\gamma^{[۲]}(X_۱,\cdots,X_p)=|X_۱X_۲|+|X_۱X_۳|+|X_۲X_۳|$$

نتایج جدول ۲.۴ بیانگر این موضوع است که با تغییر ساختار مدل، همچنان الگوریتم در ابعاد بالا و بسیار بالا با $n=۱۰۰,۱۵۰,۲۰۰$ متغیرهای مهم را به طور همزمان به درستی تشخیص داده که نشان از دقت بالای الگوریتم در ابعاد بالا است. البته به ازای $n=۵۰$ در ابعاد بسیار بالا $(p=۱۰۰۰۰)$، سه متغیر مهم $\{X_۱,X_۲,X_۳\}$ به ترتیب ۳۲، ۳۳ و ۲۶ انتخاب شده که تنها یک بار از ۱۰۰ بار تکرار، هر سه متغیر با هم شناسایی شده‌اند، با توجه به ساختار مدل، کارآیی الگوریتم در بعد بسیار بالا کاهش یافته است. برای $(n,p)=(۵۰,۱۰۰)$ سه متغیر $\{X_۱,X_۲,X_۳\}$ به تعداد دفعات ۸۰، ۷۵ و ۸۲ بار انتخاب شده‌اند که ۵۱ بار از ۱۰۰ بار تکرار الگوریتم، متغیرهای مهم به طور همزمان شناسایی شده‌اند، با افزایش p به بعد بالا یعنی ۱۰۰۰، متغیرهای $\{X_۱,X_۲,X_۳\}$ به ترتیب ۵۰، ۴۹ و ۴۷ انتخاب شده‌اند که ۱۱ بار به طور همزمان شناسایی شده‌اند.

جدول ۲.۴: تعداد دفعاتی که متغیرهای $X_۱,X_۲,X_۳$ مدل ۲، در ۱۰۰ بار تکرار، به عنوان متغیر مهم شناسایی شده‌اند

	n=۵۰	n=۱۰۰	n=۱۵۰	n=۲۰۰
مدل ۲				
p=۱۰۰	۵۱	۱۰۰	۱۰۰	۱۰۰
p=۱۰۰۰	۱۱	۱۰۰	۱۰۰	۱۰۰
p=۱۰۰۰۰	۱	۹۹	۱۰۰	۱۰۰

ج: مدل ۳

$$\gamma^{[۳]}(X_۱,\cdots,X_p)=|X_۱X_۲X_۳|$$

جدول ۳.۴ بیان می‌کند که با توجه به ساختار مدل و پیچیدگی آن به لحاظ وجود اثر متقابل سه عاملی، همچنان الگوریتم در ابعاد بالا و بسیار بالا $p=۱۰۰۰,۱۰۰۰۰$ به ازای $n=۱۵۰,۲۰۰$ متغیرهای مهم را ۹۹ بار به طور همزمان درست تشخیص داده است. تاثیر پیچیدگی مدل را می‌توان در نتایج نسبتا ضعیف ستون اول مربوط به $n=۵۰$ ملاحظه کرد که برای بعد بسیار بالا، $p=۱۰۰۰۰$، هریک از متغیرهای $X_۱,X_۲,X_۳$ به ترتیب ۱۴، ۱۴ و ۱۵ بار درست تشخیص داده‌اند ولی هیچ یک از متغیرهای مهم به طور همزمان شناسایی نشده‌اند. همچنین برای حجم نمونه $n=۱۰۰$، نتایج حاصله بسیار قابل قبول است.

جدول ۳.۴: تعداد دفعاتی که متغیرهای $X_۱, X_۲, X_۳$ مدل ۳، در ۱۰۰ بار تکرار به عنوان متغیر مهم شناسایی شده‌اند

	n=۵۰	n=۱۰۰	n=۱۵۰	n=۲۰۰
مدل ۳				
p=۱۰۰	۱۰	۷۱	۱۰۰	۱۰۰
p=۱۰۰۰	۳	۵۵	۹۹	۹۹
p=۱۰۰۰۰	۰	۲۱	۹۹	۹۹

د: مدل ۴

$$\gamma^{[۴]}(X_۱, \cdots, X_p) = \frac{|X_۱X_۲| + X_۳^۲}{۲ + X_۱X_۲X_۳}$$

در عین اینکه مدل ۴ دارای ساختار بسیار پیچیده‌تری نسبت به سه مدل قبلی است، جدول ۴.۴ موید این مطلب است که نتایج مربوط به $n = ۱۵۰, ۲۰۰$ بسیار رضایت‌بخش است و برای $n = ۱۰۰$ عملکرد متوسط رو به بالایی را نشان داده است. نتایج برای $n = ۵۰$ همان طور که انتظار میرود ضعیف است اما نسبت ۱۰۰۰۰ متغیر به ۵۰ مشاهده را که عددی بسیار کوچکی است را نباید از نظر دور داشت.

جدول ۴.۴: تعداد دفعاتی که متغیرهای $X_۱, X_۲, X_۳$ مدل ۵، در ۱۰۰ با تکرار به عنوان متغیر مهم شناسایی شده‌اند

	n=۵۰	n=۱۰۰	n=۱۵۰	n=۲۰۰
مدل ۴				
p=۱۰۰	۱۲	۷۴	۹۷	۱۰۰
p=۱۰۰۰	۱	۵۹	۹۳	۹۹
p=۱۰۰۰۰	۰	۵۰	۶۴	۸۹

ه: مدل ۵

$$\gamma^{[۵]}(X_۱, \cdots, X_p) = \frac{|X_۱X_۲| + |X_۱X_۳|}{۲ + |X_۲X_۳|}$$

به طور قطع، انتخاب متغیر این مدل دارای ساختار غیرجمعی اثرات متقابل دو عاملی است و نسبت به ۴ مدل قبلی دارای پیچیدگی خاصی است؛ لذا الگوریتمی که بتواند این سه متغیر مهم را به طور همزمان شناسایی کند، حتما از قابلیت ویژه‌ای برخوردار است. همان طور که در جدول ۵.۴ نشان داده شده است، در حجم $n = ۲۰۰$ و در بعد بسیار بالا $p = ۱۰۰۰۰$، الگوریتم توانسته ۷۰ بار متغیرهای مهم را به طور همزمان درست تشخیص دهند، بسیار مطلوب است. در اندازه نمونه $n = ۱۵۰$ و به ازای بعد متوسط، از کارآیی قابل قبولی برخوردار بوده و حتی مقدار ۴۸ برای $p = ۱۰۰۰$ نشان از قابلیت روش انتخاب متغیر در ساختارهای پیچیده است.

البته به دلیل ساختار پیچیده‌ی مدل، نتایج حجم‌های نمونه‌ی ۵۰ و ۱۰۰ رضایت بخش نیست.

جدول ۵.۴: تعداد دفعاتی که متغیرهای $X_۱, X_۲, X_۳$ مدل ۵، در ۱۰۰ بار تکرار، به عنوان متغیر مهم شناسایی شده‌اند

	n=۵۰	n=۱۰۰	n=۱۵۰	n=۲۰۰
مدل ۵				
p=۱۰۰	۲	۱۶	۷۲	۹۴
p=۱۰۰۰	۲	۱۶	۵۳	۸۸
p=۱۰۰۰۰	۰	۶	۴۸	۶۹

از نتایج بدست آمده مشخص است که الگوریتم در ابعاد بسیار بالا قابلیت تشخیص عالی و بدون نقص با افزایش n دارد، البته این نکته نیز قابل توجه است که هر چه مدل در نظر گرفته شده دارای پیچیدگی بیشتر باشد نتایج الگوریتم متمایز خواهد بود.
بنابراین تغییر در اندازه نمونه‌ها نشان دهنده این موضوع است، که برای داشتن نتایج عالی و بدون نقص در ابعاد بالا و بسیار بالا، نیاز به اندازه نمونه‌های بزرگتری است که الگوریتم بتواند متغیرهای مهم را شناسایی کنند.

۳.۱.۴ تاثیر آستانه‌ی t در انتخاب متغیر

آستانه‌ی[۱] t به عنوان یک پارامتر برای توقف الگوریتم NOVAS، در هر گام از اجرا به کار می‌رود. در واقع پس از اتمام هر مرحله از الگوریتم، شرط توقف در رابطه‌ی (۳.۱.۴) یعنی

$$\frac{S(T_۱^m) - S(T_۱^{m+۱})}{S(T_۱^m)} \leq t$$

بررسی می‌شود.
همان طور که در جدول ۶.۴ مشخص است اندازه‌های متفاوت t در کاهش زمان اجرای الگوریتم موثر خواهد بود بطوریکه برای مقادیر کوچک t

$$t = ۰/۰۱, ۰/۰۵, ۰/۱, ۰/۱۵, ۰/۲$$

اجرای الگوریتم به ازای $n = ۱۰۰$ و $p = ۱۰۰۰$ نزدیک به ۵ ساعت زمان می‌برد اما برای مقادیر بزرگ t

$$t = ۰/۲۵, ۰/۳, ۰/۳۵, ۰/۴, ۰/۴۵, ۰/۵$$

این زمان به ۳ ساعت کاهش می‌یابد.

[۱] Thereshold

زمانی که مقدار کوچک برای t در نظر گرفته می‌شود مقدار خطای اعتبارسنجی به آرامی کاهش یافته و باعث افزایش تعداد متغیرهای انتخابی در هر مرحله می‌شود.

همان‌طور که در جدول ۶.۴ مشاهده می‌کنید، الگوریتم NOVAS برای مقادیر $t \leq ۰/۲$ توانسته با عملکرد عالی متغیرهای مهم $X_۱, X_۲, X_۳$ را به طور همزمان ۱۰۰ بار به درستی شناسایی کرده است، اما به ازای مقادیر $t > ۰/۲$ الگوریتم کارآیی چندان خوبی برای شناسایی متغیرهای مهم نداشته به طوری که برای $t = ۰/۵$، در هر یک از مدل‌های ۳، ۴ و ۵ نتوانسته متغیرهای مهم را به طور همزمان انتخاب کند؛ لذا نتایج حاصل، عاملی است برای تشویق اینکه از پیش فرض $t = ۰/۰۵$ به عنوان مقدار آستانه‌ی توقف الگوریتم استفاده شود.

جدول ۶.۴: تعداد دفعاتی که متغیرهای $X_۱, X_۲, X_۳$، در ۱۰۰ بار تکرار، به ازای مقادیر متفاوت t، عنوان متغیر مهم شناسایی شده‌اند $(p = ۱۰۰۰)$

t	۰.۰۱	۰.۰۵	۰.۱۵	۰.۲	۰.۲۵	۰.۳	۰.۵
مدل ۱ $(n = ۱۰۰)$	۱۰۰	۱۰۰	۱۰۰	۱۰۰	۹۴	۳۳	۸۶
مدل ۲ $(n = ۱۰۰)$	۱۰۰	۱۰۰	۱۰۰	۷۴	۳۲	۲۵	۵
مدل ۳ $(n = ۱۵۰)$	۱۰۰	۹۹	۱۰۰	۶۷	۲۲	۳۸	۰
مدل ۴ $(n = ۱۰۰)$	۶۳	۹۳	۴۱	۳۰	۱۷	۴	۰
مدل ۵ $(n = ۱۰۰)$	۲۱	۲۰	۷۳	۸	۲	۱۱	۰

۴.۱.۴ تاثیر نسبت Noise-To-Signal

همواره داده‌های واقعی دارای مقادیری است که باعث بالا رفتن میزان خطا و در نتیجه باعث کاهش کارآیی روش‌های انتخاب متغیر می‌شود. Noise-To-Signal اشاره به نسبت اطلاعات مفید به داده‌های نادرست و غیرمرتبط دارد، برای اینکه داده‌های شبیه‌سازی شده نزدیک به داده واقعی باشد از Noise-To-Signal استفاده می‌کنیم.نسبت Noise-To-Signal را به صورت زیر محاسبه می‌شود:

$$nsr = \sigma_m^۲ / \mathrm{var}\{\gamma_{۱,۲,۳}^{[m]}(X_{i۱}, X_{i۲}, X_{i۳})\}$$

تمامی نتایج شبیه‌سازی براساس $nsr = ۰/۰۵$ می‌باشد. به منظور بررسی تاثیر این مقدار در کارایی الگوریتم، مجددا نتایج شبیه‌سازی را به ازای مقادیر متفاوت $nsr = ۰/۰۵, ۰/۱, ۰/۲, ۰/۴$ برای $p = ۱۰۰۰$ و $n = ۱۰۰$ بدست می‌آوریم.با توجه به جدول ۷.۴ عملکرد الگوریتم زمانی که $nsr = ۰/۱$ در نظر گرفته می‌شود، با قابلیت بسیار بالا متغیرهای انتخابی را شناسایی می‌کند، همچنین برای $nsr = ۰/۲$ نیز الگوریتم قادر به تشخیص درست متغیرهای انتخابی است با اینکه مقدار $Noise$ افزایش یافته است.

جدول ۷.۴: تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم درست شناسایی شده‌اند زمانی که $p = 1000$

nsr	۰.۰۵	۰.۱	۰.۲
مدل ۱ $(n = 100)$	۱۰۰	۱۰۰	۱۰۰
مدل ۲ $(n = 100)$	۱۰۰	۸۷	۴۳
مدل ۳ $(n = 150)$	۹۹	۱۰۰	۹۳
مدل ۴ $(n = 200)$	۹۹	۹۶	۹۳
مدل ۵ $(n = 200)$	۸۸	۸۶	۶۴

۵.۱.۴ مدل رگرسیونی

در این بخش کارآیی الگوریتم انتخاب متغیر را در مدل‌های ناپارامتری (مدل غیرخطی)، نیمه‌پارامتری (مدل شبه‌خطی) و پارامتری (مدل خطی) بررسی می‌کنیم. بدین منظور از مدل رگرسیونی زیر استفاده می‌شود:

$$Y = \gamma^{\alpha}(X_1, X_2, X_3) + \varepsilon_i$$

که ساختار γ^{α} را به صورت زیر در نظر می‌گیریم:

$$\gamma^{\alpha}(X_1, X_2, X_3) = 3 + \alpha(X_1 + X_2 + X_3) + (1 - \alpha)(X_1^2 + X_2^2 + X_3^2) + \varepsilon_i \qquad (۴.۴)$$

با توجه به بخش ۱.۱.۴ X_i ها نیز دارای توزیع $U[-1, 1]$ است، همچنین خطای تصادفی ε_i نیز مستقل و دارای توزیع $N(0, \sigma^2)$ است. شبیه‌سازی مدل‌های پارامتری، نیمه‌پارامتری و ناپارامتری با اندازه نمونه‌ی $n = 50$ و تعداد متغیر توضیحی $p = 1000$ و $nsr = 0/1$ در نظر گرفته می‌شود. اگر $\alpha = 0$ در نظر بگیریم، ساختار تابع γ به مدل ناپارامتری (غیرخطی) تبدیل می‌شود، با در نظر گرفتن $\alpha = 0/35$ ساختار تابع γ به مدل نیمه‌پارامتری تبدیل می‌شود، مدل نیمه‌پارامتری شامل مدل‌های رگرسیونی است که مدل‌های پارامتری و نیمه‌پارامتری را ترکیب می‌کند و زمانی که $\alpha = 1$ در نظر گرفته شود ساختار تابع γ به مدل پارامتری (خطی) تبدیل می‌شود. در رابطه (۴.۴) همانند دیگر ساختارهای غیرخطی بخش ۱.۱.۴ تنها متغیرهای اول، دوم و سوم برای مدل‌های متفاوت رگرسیونی با اهمیت هستند و دیگر متغیرها بی‌اهمیت محسوب می‌شوند. شبیه‌سازی مدل‌های متفاوت رگرسیونی به منظور بررسی کارآیی الگوریتم NOVAS در شرایط مدل‌های خطی، غیرخطی و شبه‌خطی است.

همان‌طور که در جدول ۸.۴ مشاهده می‌کنید در مدل غیرخطی الگوریتم از ۱۰۰ بار تکرار ۵۳ متغیرهای مهم X_1, X_2, X_3 را به طور همزمان شناسایی کرده، همچنین در مدل شبه‌خطی نیز ۵۴ بار الگوریتم توانسته متغیرهای مهم را به طور همزمان شناسایی کند اما در مدل خطی الگوریتم تنها یک بار توانسته هر سه متغیر مهم را با هم تشخیص دهد.

جدول ۸.۴: تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم در مدل‌های مختلف رگرسیونی درست شناسایی شده‌اند $(p = 1000, n = 50)$

	$\alpha = 0$ (غیرخطی)	$\alpha = 0/35$ (شبه‌خطی)	$\alpha = 1$ (خطی)
NOVAS	۵۳	۵۴	۱

با توجه به جدول ۸.۴ مشخص است که الگوریتم در مدل خطی تنها یک بار توانسته هر سه متغیر را به طور همزمان شناسایی کند، در صورتی که جدول ۹.۴ نشان می‌دهد هر یک از متغیرهای مهم X_1, X_2, X_3 به ترتیب ۸۸، ۹۰ و ۹۲ بار انتخاب شده‌اند اما از این میان تنها یک بار هر سه متغیر با هم انتخاب شده‌اند. ارزیابی الگوریتم با مدل‌های متفاوت رگرسیونی بیانگر این موضوع است که الگوریتم قابلیت عالی و بدون نقص در مدل‌های غیرخطی و شبه‌خطی دارد چنانچه که الگوریتم کارآیی عالی خود را در مدل خطی ندارد.

جدول ۹.۴: تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم درست شناسایی شده‌اند، ستون others حاوی متغیرهای دیگر که بیش‌تر از سه بار تکرار شده‌اند. $p = 1000$

	$\alpha = 0$ (غیرخطی)				$\alpha = 0/35$ (شبه‌خطی)				$\alpha = 1$ (خطی)			
Covariates	۱	۲	۳	others	۱	۲	۳	others	۱	۲	۳	others
NOVAS	۷۸	۷۳	۸۱	۰	۸۸	۸۶	۸۱	۰	۹۲	۹۰	۸۸	۵

۲.۴ مطالعه داده‌های واقعی ژنومیک

در این بخش، کارآیی الگوریتم انتخاب متغیر را برای یک مجموعه داده‌ی واقعی مورد مطالعه قرار می‌دهیم. این داده‌ها از طریق بسته mixOmics [۱۰] در نرم افزار R قابل دریافت هستند. داده‌های ژنی مربوط به ۶۴ موش صحرایی مبتلا به مسمومیت کبدی در اثر استفاده از داروی استامینوفن کدئین است، که به روش خوشه‌بندی توسط باشل و همکاران مورد مطالعه قرار گرفته است [۷]. در این پژوهش تعداد ۳۱۱۶ ژن با هدف شناسایی تاثیرگذاری اثرات استامینوفن بر روی نُه متغیر پاسخ بالینی: نیتروژن اوره[۱] (BUN)، پروتئین توتال[۲] (TP)، آلبومین[۳] (ALB)، آلانین آمینوترانسفراز [۴] (ALT)، آسپارتات ترانس آمیناز[۵] (AST)، آلکالین

[۱] Blood urea nitrogen
[۲] Total protien
[۳] Albumin
[۴] Alanine aminotransferase
[۵] Aspartate aminotransferase

فسفاتاز[۱] (ALP)، اسیدهای صفراوی [۲] (TBA)، سوربیتول دهیدروژناز[۳] (SDH) و کُلِسترِول[۴] (CHOL) را مورد مطالعه قرار می‌دهیم. هدف از انتخاب متغیر در این مجموعه داده، بررسی توانایی الگوریتم در شناسایی ژن‌های تاثیرگذار بر متغیرهای پاسخ بالینی اندازه‌گیری شده است.

۱.۲.۴ نتایج اجرای الگوریتم

جدول ۱۰.۴ حاوی جزئیات نتایج اجرای الگوریتم NOVAS در مراحل اول تا چهارم با هدف پیشگویی متغیر پاسخ نیتروژن اوره (BUN) بر حسب متغیرهای توضیحی (ژن) است.
در گام اول الگوریتم که انتخاب پیشگو به صورت تک متغیره است، ژن شماره ۱۱۶۵ در مرحله‌ی نخست به عنوان بهترین متغیر با خطای اعتبار سنجی $CV = ۶/۸۳$ انتخاب شده است. برای ورود به گام دوم شرط توقف الگوریتم بایستی بررسی شود. طبق رابطه (۷.۳) داریم:

$$\frac{S(\hat{T}_1^{(1)}) - S(\hat{T}_1^{(2)})}{S(\hat{T}_1^{(1)})} = \frac{۶/۳۸ - ۵/۲۲}{۶/۳۸} = ۰/۱۸ \not\leq t = ۰/۰۵$$

از آنجا که این مقدار بزرگتر از حد آستانه‌ای پیش فرض $t = ۰/۰۵$ است می‌توان وارد گام دوم الگوریتم شد. در گام دوم تعداد $\sqrt{۳۱۱۶} \approx ۵۶$ متغیر از موثرترین متغیرهای گام اول در نظر گرفته شده و رگرسیون موضعی با هریک از ترکیبات دوتایی آن‌ها اجرا می‌شود و در نهایت، الگوریتم پیشنهادی NOVAS ژن‌های شماره‌ی ۱۸۶۶ و ۲۰۵۰ را به عنوان بهترین زوج متغیر با خطای اعتبارسنجی $CV = ۵/۲۲$ انتخاب می‌کند. مجددا برای ورود به گام بعد داریم:

$$\frac{S(\hat{T}_1^{(2)}) - S(\hat{T}_1^{(3)})}{S(\hat{T}_1^{(2)})} = \frac{۵/۲۲ - ۳/۷۶}{۵/۲۲} = ۰/۲۷ \not\leq t = ۰/۰۵$$

همانند گام قبل، مقدار بدست آمده، از آستانه‌ی مورد نظر بیش‌تر است، لذا مراحل انتخاب متغیر الگوریتم ادامه دارد و وارد گام سوم می‌شویم. گام سوم برای انتخاب بهترین پیشگو، شامل زیرمجموعه‌های سه یا چهار عضوی است.

در گام سوم الگوریتم، خطای اعتبارسنجی برای زیرمجموعه‌های سه یا چهار عضوی محاسبه شده و مجموعه متغیری که دارای کمترین خطای اعتبارسنجی بوده است، انتخاب می‌شود. با توجه به جدول ۱۰.۴ مدل شامل ژن‌های $۱۹۵۷, ۱۸۳۷, ۱۱۶۷, ۱۰۰۰$ دارای کمترین خطای اعتبار سنجی در این مرحله بوده و به عنوان بهترین متغیرها انتخاب می‌شوند. مجددا برای ورود به گام بعدی شرط توقف را بررسی می‌کنیم:

$$\frac{S(\hat{T}_1^{(3)}) - S(\hat{T}_1^{(4)})}{S(\hat{T}_1^{(3)})} = \frac{۳/۷۶ - ۳/۲۷}{۳/۷۶} = ۰/۱۳ \not\leq t = ۰/۰۵$$

[۱] Alkaline Phosphatase
[۲] Total bile acids
[۳] Sorbitol dehydrogenase
[۴] Cholesterol

در گام چهارم الگوریتم نیز که متغیرها از اجتماع متغیرهای مراحل قبل بدست آمده‌اند، خطای اعتبار سنجی محاسبه شده و آن مجموعه متغیرهایی که کمترین خطا را دارند انتخاب می‌شوند. علاوه بر متغیرهای انتخاب شده در گام سوم، ژن شماره ۱۸۸۹ نیز انتخاب می‌شود. الگوریتم در این مرحله خاتمه می‌یابد زیرا:

$$\frac{S(\hat{T}_1^4) - S(\hat{T}_1^5)}{S(\hat{T}_1^4)} = \frac{۳/۲۷ - S(\hat{T}_1^5)}{۳/۲۷} \leq t = ۰/۰۵$$

تعداد ژن‌های انتخابی در مرحله‌ی نهایی الگوریتم NOVAS در واقع بیانگر این مطلب هستند که به غیر از این پنج متغیر، مجموعه متغیر دیگری در این گام وجود ندارد که بتواند خطا را به کمترین مقدار خود برساند. همان طور که در جدول ۱۰.۴ ملاحظه می‌شود. هیچ یک از متغیرهای انتخاب شده در گام ۴، ژن‌های منتخب درگام‌های ۱و ۲ نیستند و لذا می‌توان نتیجه گرفت که ژن‌های انتخاب شده در گام ۴، به تنهایی بهترین نبوده بلکه در ترکیب با سایر ژن‌های ردیف ۴ می‌تواند بهترین ترکیب موثر ۵ تایی را تشکیل دهند.

جدول ۱۰.۴: نتایج اجرای روش پیشنهادی انتخاب متغیر برای متغیر پاسخ BUN

شماره گام (m)	ژن‌های انتخاب شده در هر گام	$S(T)$
۱	۱۱۶۵	۶/۸۳
۲	۱۸۶۶ – ۲۰۵۰	۵/۲۲
۳	۱۰۰۰ – ۱۱۶۷ – ۱۸۳۷ – ۱۹۵۷	۳/۷۶
۴	۱۰۰۰ – ۱۱۶۷ – ۱۸۳۷ – ۱۸۹۹ – ۱۹۵۷	۳/۲۷

جداول ۱۱.۴ تا ۱۴.۴ بهترین متغیرهای انتخاب شده در هر گام از اجرای روش NOVAS به همراه خطای اعتبارسنجی برای متغیرهای پاسخ پروتئین توتال (TP)، آلبومین (ALB)، آلانین آمینوترانسفراز (ALT)، سوربیتول دهیدروژناز (SDH) را نشان می‌دهد.

جدول ۱۱.۴: نتایج اجرای روش پیشنهادی برای متغیر پاسخ TP

شماره گام (m)	ژن‌های انتخاب شده در هر گام	$S(T)$
۱	۲۹۲۳	۰/۰۹۶
۲	۲۱۷۳ – ۲۹۷۱	۰/۰۷۵
۳	۹۴۱ – ۲۴۱۲ – ۲۹۲۳	۰/۰۵۵
۴	۱۱۵۹ – ۱۹۷۰ – ۲۰۲۰ – ۲۱۷۳ – ۲۹۲۳ – ۲۹۲۷ – ۲۹۷۱	۰/۰۴۵

جدول ۱۲.۴: نتایج اجرای روش پیشنهادی برای متغیر پاسخ ALB

شماره گام (m)	ژن‌های انتخاب شده در هر گام	$S(T)$
۱	۲۰۲۳	۰/۰۳۲
۲	۸۹۰ – ۲۸۶۴	۰/۲۵
۳	۱۰۳۸ – ۱۹۹۲ – ۲۶۶۹ – ۲۹۱۶	۰/۲۰
۴	۱۰۳۸ – ۱۱۶۵ – ۱۹۹۲ – ۲۱۰۵ – ۲۶۶۹ – ۲۸۶۷	۰/۱۶
۵	۱۰۳۸ – ۱۱۶۵ – ۱۹۹۲ – ۲۰۲۰ – ۲۱۰۵ – ۲۶۶۹ – ۲۸۶۷ – ۲۹۲۱	۰/۰۱۵

جدول ۱۳.۴: نتایج اجرای روش پیشنهادی برای متغیر پاسخ ALT

شماره گام (m)	ژن‌های انتخاب شده در هر گام	$S(T)$
۱	۱۸۸۳	۱۴۲۵۷۶۹
۲	۱۸۸۳ – ۱۹۱۴	۴۵۴۰۶۰
۳	۱۸۴۶ – ۱۸۸۳ – ۱۹۰۹ – ۲۰۴۲	۱۸۱۹۵۳/۸
۴	۱۰۲۲ – ۸۸۱ – ۱۸۸۳ – ۱۹۰۹ – ۱۹۲۹ – ۱۹۴۳ – ۲۰۴۲	۸۰۸۰۸/۲۳
۵	۱۸۴۶ – ۱۸۷۱ – ۱۸۸۳ – ۱۹۰۹ – ۱۹۱۰ – ۱۹۱۱ – ۱۹۱۵ – ۱۹۲۱ – ۲۰۴۲	۶۰۶۲۱/۶۱

جدول ۱۴.۴: نتایج روش پیشنهادی برای متغیر پاسخ SDH

شماره گام (m)	ژن‌های انتخاب شده در هر گام	$S(T)$
۱	۹۹۳	۱۱۱۷۷/۸۲
۲	۷۶۴ – ۱۶۲۴	۳۹۸۷/۳۴
۳	۷۶۴ – ۱۱۴۵ – ۱۶۲۴ – ۱۸۶۶	۲۲۹۹/۱۱
۴	۷۶۴ – ۱۰۵۰ – ۱۱۴۵ – ۱۶۲۴ – ۱۹۴۰ – ۱۹۹۶	۱۵۰۲/۷۵۱
۵	۷۶۴ – ۱۱۴۵ – ۱۶۲۴ – ۱۸۶۶ – ۱۹۴۰ – ۱۹۹۲ – ۱۹۹۶ – ۲۸۹۴	۱۴۰۴/۶۹

برای سایر متغیرهای پاسخ AST, ALP, TBA, CHOL، الگوریتم NOVAS اجرا شده و متغیرهای انتخاب شده در آخرین گام از اجرای الگوریتم در جدول ۱۵.۴ آمده است. در خصوص متغیر پاسخ CHOL مشاهده می‌شود که از میان ۳۱۱۶ ژن تنها سه متغیر (ژن) $X_{۱۸۳۶}, X_{۱۸۷۵}, X_{۲۰۴۴}$ به عنوان متغیرهای مهم و تاثیرگذار بر میزان تولید کلسترول انباشته شدن این ماده پس از مصرف استامینوفن کدئین در بدن تاثیرگذار بوده است. همچنین برحسب نوع متغیر پاسخ، تعداد متغیرهای انتخاب شده موثر بر آن، متفاوت می‌باشد به طوری که برای پاسخ‌های AST، ALP و TBA به ترتیب تعداد ۱۰، ۸ و ۸ متغیر انتخاب شد

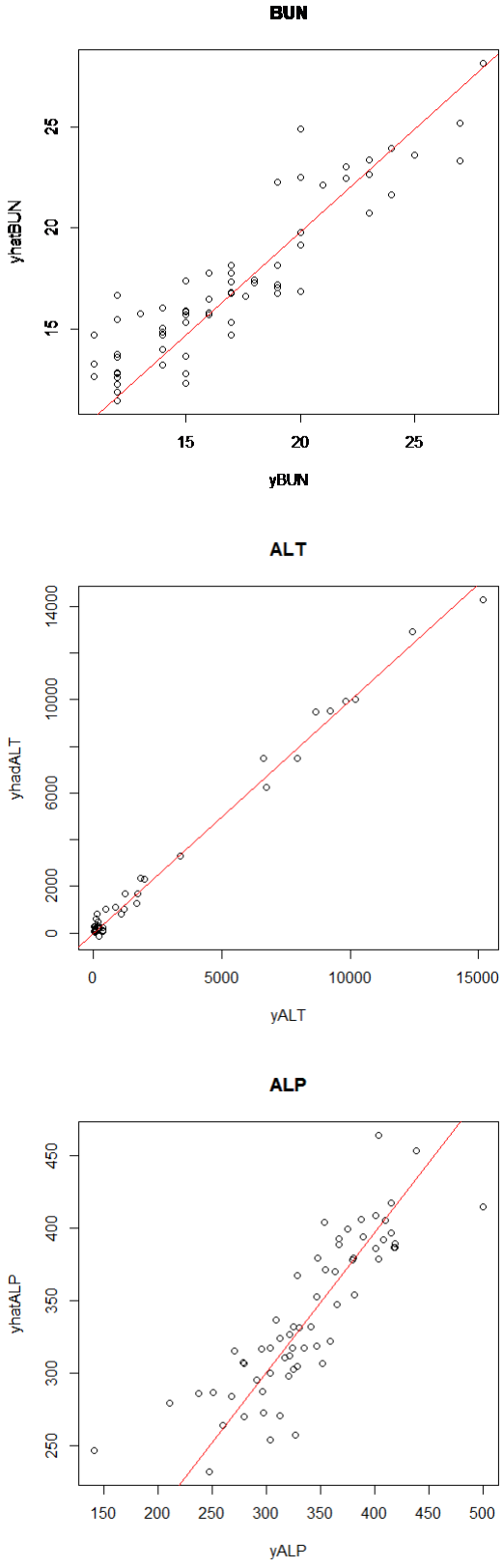
مجموعه ژن‌های انتخابی در هر گام از الگوریتم NOVAS، دارای کمترین مقدار خطای اعتبارسنجی به طور منحصربه فرد هستند، به این معنی که به غیر از مجموعه ژن‌های انتخاب

جدول ۱۵.۴: بهترین زیر مجموعه پیشگو متغیرها براساس LOOCV در گام نهایی الگوریتم

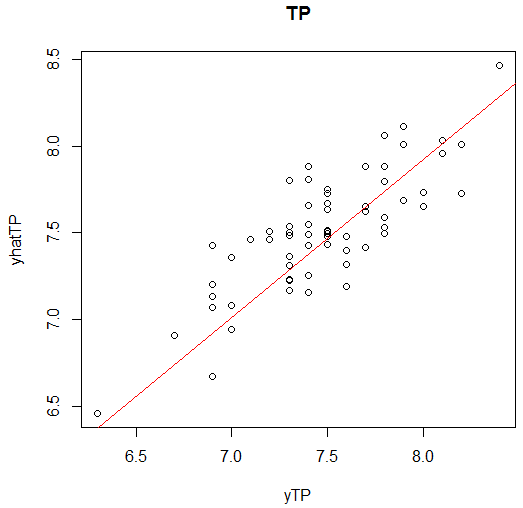
شماره گام	ژن‌های انتخاب شده در گام نهایی	$S(T)$
AST	۹۷۷ – ۱۱۱۶ – ۱۱۶۱ – ۱۳۳۵ – ۱۸۲۶ – ۱۸۹۱ – ۱۹۰۹ – ۱۹۶۱ – ۲۱۹۷ – ۲۲۰۱	۳۱۸۶۸۲/۴
ALP	۱۰۶۴ – ۱۴۸۴ – ۱۸۱۷ – ۱۸۲۳ – ۲۰۰۷ – ۲۳۸۵ – ۲۸۱۹	۱۰۴۳/۷۳
TBA	۱۸۹۱ – ۱۹۱۳ – ۱۹۱۶ – ۱۹۱۷ – ۱۹۵۴ – ۲۲۰۰ – ۲۲۰۵	۳۹/۷۳
CHOL	۱۸۳۶ – ۱۸۷۵ – ۲۰۴۴	۴۰/۶۳

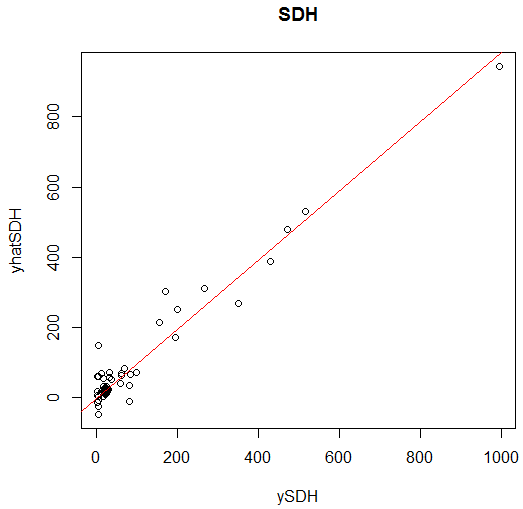
شده در هر گام از الگوریتم، ژن‌های دیگری وجود ندارد که دارای کمتری خطا نسبت به زیرمجموعه انتخاب شده باشد لذا این مجموعه متغیرها بطور منحصربه‌فرد مینیمم هستند.

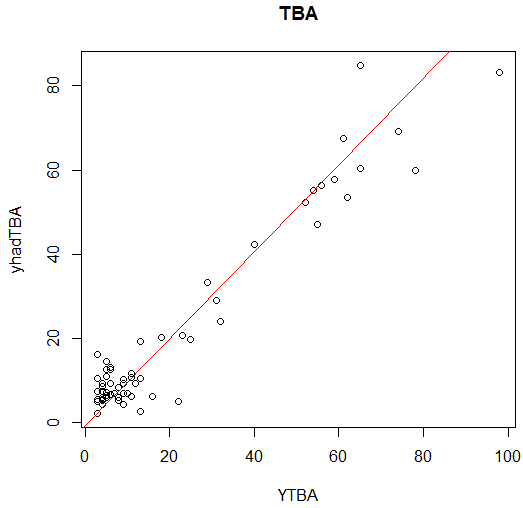
شکل ۲.۴، نمودار مقادیر واقعی در برابر مقادیر برازش شده متغیرهای پاسخ را نشان می‌دهد. مدل برازش شده برای هر یک از متغیرهای پاسخ متغیرهای انتخاب شده در گام آخر هستند که در جداول ۱۰.۴ تا ۱۴.۴ آمده است. به طور کلی همان‌گونه که مشخص است، ژن‌های انتخاب شده در بین ۳۱۱۶ ژن به خوبی توانسته‌اند تغییرات پاسخ را بیان کنند.



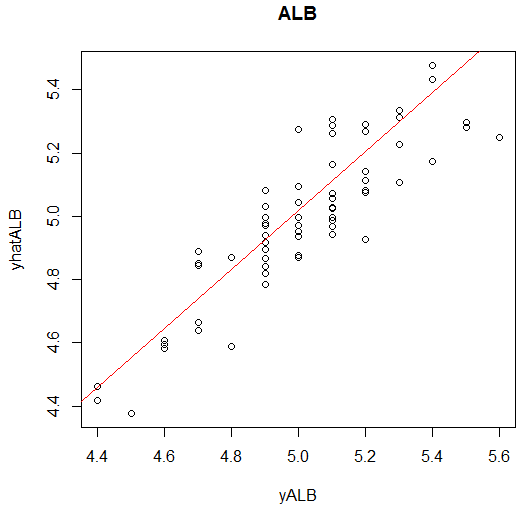
شکل ۲.۴: مقادیر واقعی در مقابل مقادیر برازش شده برای متغیرهای پاسخ BUN,ALT,ALP

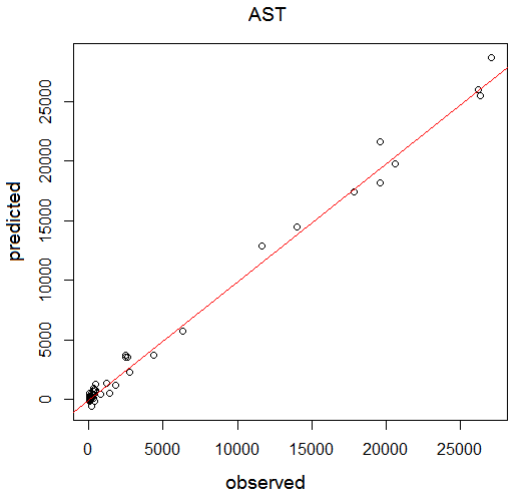


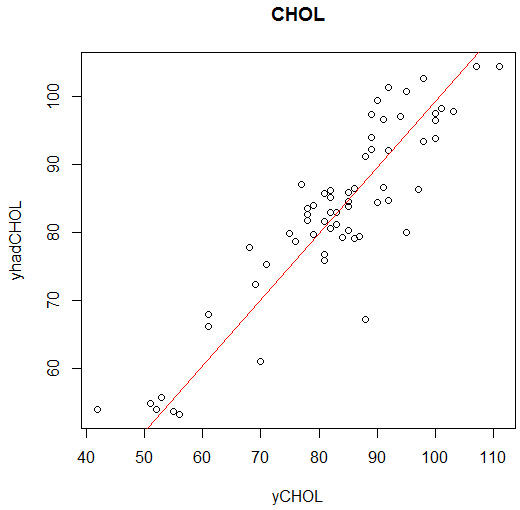




ادامه‌ی شکل ۲.۴ برای متغیر پاسخ TP,SDH,TBA







ادامه شکل ۲.۴ برای متغیر پاسخ ALB,AST,CHOL

۲.۲.۴ مقایسه روش NOVAS با دیگر روش‌ها

در این تحقیق از آماره اعتبارسنجی متقابل و نمودار پراکنش به عنوان معیارهایی برای ارزیابی کارآیی الگوریتم NOVAS نسبت به دیگر روش‌های انتخاب متغیر، استفاده می‌شود.

۱. اعتبارسنجی متقابل LOO :
استفاده از معیار اعتبارسنجی LOO به منظور بررسی بهتر دقت پیشگویی الگوریتم است. در اعتبارسنجی متقابل با در نظر گرفتن مجموعه‌های آزمون به تعداد حجم نمونه‌ی اصلی، که هر بار شامل یک مشاهده است، باعث می‌شود که برآوردهای پارامترها برای ارزیابی دقت مدل محاسبه شده تقریبا مشابه برآوردهایی باشند که از نمونه‌ی اصلی به دست می‌آیند؛ لذا این نوع اعتبارسنجی برای بررسی دقت الگوریتم معیاری مناسب است به منظور مقایسه با دیگر روش‌های مطرح شده.

۲. ارزیابی براساس نمودار پراکنش:
به منظور مطالعه‌ی رابطه‌ی بین متغیرهای انتخاب شده توسط روش پیشنهادی با دیگر روش‌های مطرح شده در این پایان‌نامه از نمودار پراکنش استفاده می‌شود. نمودار پراکنش در تحلیل رگرسیون اهمیت زیادی دارد و نقش بسیار مهمی در تحلیل و نمایش ارتباط بین متغیرها ایفا می‌کند. این نمودار چگونگی خوشه‌ای شدن در اطراف یک خط مستقیم یا یک نوع منحنی مشخص را نشان می‌دهد و نیز برداشتی عینی از میزان پراکندگی داده‌ها پیرامون خط یا منحنی را فراهم می‌کند. در صورتی که نمودار نشان دهنده‌ی این باشد که داده‌ها تقریبا (نه لزوما) در امتداد یک خط مستقیم (نیمساز ۴۵ درجه) پراکنده باشند.

۳.۲.۴ مقایسه روش NOVAS با PLS

در این بخش نحوه‌ی عملکرد NOVAS را در مقابل روش PLS مورد مقایسه قرار می‌دهیم در واقع به دنبال بررسی دقت پیشگویی الگوریتم NOVAS نسبت به روش PLS هستیم.

برای انتخاب بهترین مولفه در مدل از روش اعتبارسنجی متقابل با یک نمونه خارج شده (LOO) استفاده می‌شود. در این روش پس از خارج کردن یک نمونه برای $n-1$ مشاهده دیگر مدل رگرسیونی حداقل مربعات جزیی برازش داده می‌شود و با این کار همه‌ی نمونه‌های اعتبارسنجی مدل استفاده خواهد شد. در نهایت بهترین مدل برازش داده شده براساس حداقل جذر مربعات خطای پیش‌بینی (Root Mean Square Error of Prediction) معرفی می‌شود همچنین CV برآورد اعتبارسنجی متقابل و adjCV یک نوع اصلاح شده‌ی برآورد اعتبارسنجی متقابل است [۲۷]. (با اعتبارسنجی LOO تقریبا تفاوتی وجود ندارد [۲۸].) رگرسیون PLS را بر روی مجموعه داده‌های ژنی نسبت به هر یک از متغیرهای پاسخ بدست می‌آوریم.

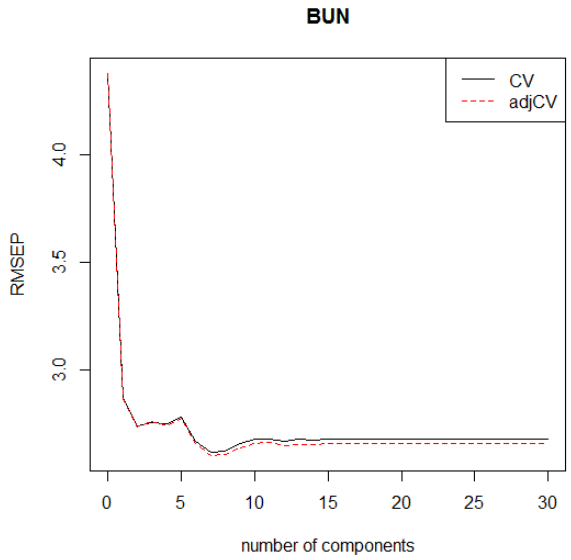
متغیر پاسخ BUN:

جدول ۱۶.۴ حاوی مقادیر تجمعی درصد سهم واریانس هر یک از مولفه‌ها را از مجموع واریانس متغیرهای توضیحی بیان می‌کند. ۶۰/۸۷ درصد از مجموع واریانس متغیرها را توسط مولفه‌ی اول بیان می‌شود. مولفه‌های اول و دوم، ۷۳/۳۲ درصد و هفت مولفه‌ی اول، ۹۶/۱۰ درصد تغییرات متغیرهای توضیحی را بیان می‌کند لذا سایر مولفه‌ها از محاسبات بعدی حذف می‌شوند.

جدول ۱۶.۴: مقادیر درصد سهم واریانس مولفه‌های روش PLS از کل واریانس متغیرهای توضیحی

	$C_۱$	$C_۲$	$C_۳$	$C_۴$	$C_۵$	$C_۶$	$C_۷$
BUN	۶۰/۸۷	۷۳/۳۲	۷۷/۰۱	۸۶/۰۴	۹۰/۱۲	۹۳/۸۹	۹۶/۱۰

شکل ۳.۴، نمودار مقادیر جذر میانگین مربعات خطای پیشگویی را به ازای مولفه‌های اول تا سی‌ام روش PLS نشان می‌دهد. همان‌طور که مشخص است، هفتمین مولفه، کمترین خطا را ایجاد کرده است.



شکل ۳.۴: نمودار مقادیر ریشه میانگین مربعات خطای پیشگویی (RMSE) برای متغیر پاسخ BUN

متغیر پاسخ TP:

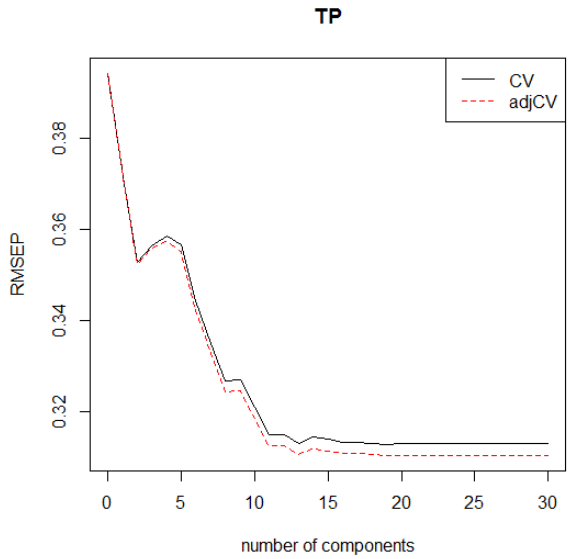
با توجه به جدول ۱۷.۴ حاوی مقادیر تجمعی درصد سهم واریانس هر یک از مولفه‌ها را از مجموع واریانس متغیرهای توضیحی بیان می‌کند. ۳۰/۷۸درصد از مجموع واریانس متغیرها را توسط مولفه‌ی اول بیان می‌شود. مولفه‌های اول و دوم، ۴۵/۰۳ درصد و ده مولفه‌ی اول،

۹۹/۳۰درصد تغییرات متغیرهای توضیحی را بیان می‌کند لذا سایر مولفه‌ها از محاسبات بعدی حذف می‌شوند.

جدول ۱۷.۴: مقادیر درصد سهم واریانس مولفه‌های روش PLS از کل واریانس متغیرهای توضیحی

	$C_۱$	$C_۲$	$C_۳$	$C_۴$	$C_۵$	$C_۶$	$C_۷$	$C_۸$	$C_۹$	$C_{۱۰}$
TP	۳۰/۷۸	۴۵/۰۳	۵۴/۴۶	۷۰/۳۵	۸۲/۰۱	۸۹/۱۸	۹۳/۹۴	۹۷/۲۰	۹۸/۶۹	۹۹/۳۰

شکل ۴.۴، نمودار مقادیر جذر میانگین مربعات خطای پیشگویی را به ازای مولفه‌های اول تا سی‌ام روش PLS نشان می‌دهد. همان‌طور که مشخص است، دهمین مولفه، کمترین خطا را ایجاد کرده است.



شکل ۴.۴: نمودار مقادیر ریشه میانگین مربعات خطای پیشگویی (RMSE) برای متغیر پاسخ TP

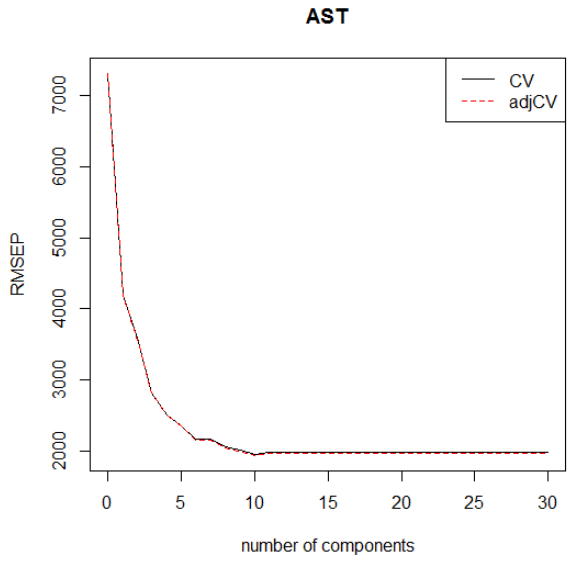
متغیر پاسخ AST:

با توجه به جدول ۱۸.۴ حاوی مقادیر تجمعی درصد سهم واریانس هر یک از مولفه‌ها را از مجموع واریانس متغیرهای توضیحی بیان می‌کند.۷۲/۴۳ درصد از مجموع واریانس متغیرها را توسط مولفه‌ی اول بیان می‌شود. مولفه‌های اول و دوم، ۸۳/۰۳ درصد و ده مولفه‌ی اول، ۹۹/۹۳درصد تغییرات متغیرهای توضیحی را بیان می‌کند لذا سایر مولفه‌ها از محاسبات بعدی حذف می‌شوند.

جدول ۱۸.۴: مقادیر درصد سهم واریانس مولفه‌های روش PLS از کل واریانس متغیرهای توضیحی

	$C_۱$	$C_۲$	$C_۳$	$C_۴$	$C_۵$	$C_۶$	$C_۷$	$C_۸$	$C_۹$	$C_{۱۰}$
AST	۷۲/۴۳	۸۳/۰۳	۹۲/۵۹	۹۴/۹۹	۹۶/۹۰	۹۸/۵۱	۹۹/۱۹	۹۹/۶۵	۹۹/۸۳	۹۹/۹۳

شکل ۵.۴، نمودار مقادیر جذر میانگین مربعات خطای پیشگویی را به ازای مولفه‌های اول تا سی‌ام روش PLS نشان می‌دهد. همان‌طور که مشخص است، دهمین مولفه، کمترین خطا را ایجاد کرده است.



شکل ۵.۴: نمودار مقادیر ریشه میانگین مربعات خطای پیشگویی (RMSE) برای متغیر پاسخ AST

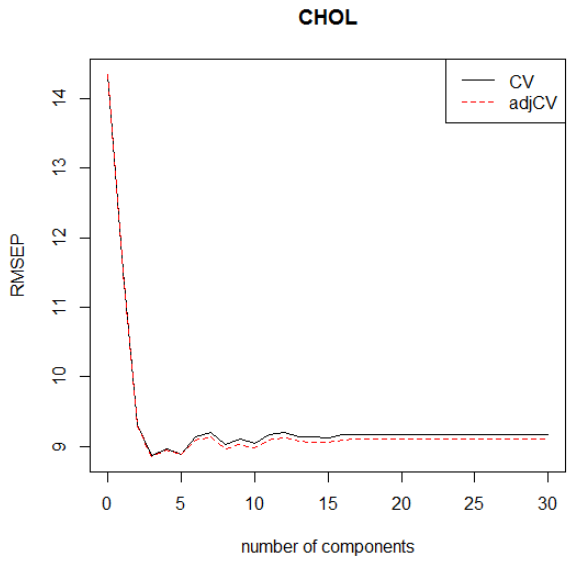
متغیر پاسخ CHOL:

با توجه به جدول ۱۹.۴ حاوی مقادیر تجمعی درصد سهم واریانس هر یک از مولفه‌ها را از مجموع واریانس متغیرهای توضیحی بیان می‌کند. ۴۳/۸۲درصد از مجموع واریانس متغیرها را توسط مولفه‌ی اول بیان می‌شود. مولفه‌های اول و دوم، ۷۱/۷۵ درصد و هشت مولفه‌ی اول، ۹۷/۶۷ درصد تغییرات متغیرهای توضیحی را بیان می‌کند لذا سایر مولفه‌ها از محاسبات بعدی حذف می‌شوند.

جدول ۱۹.۴: مقادیر درصد سهم واریانس مولفه‌های روش PLS از کل واریانس متغیرهای توضیحی

	$C_۱$	$C_۲$	$C_۳$	$C_۴$	$C_۵$	$C_۶$	$C_۷$	$C_۸$
CHOL	۴۳/۸۲	۷۱/۷۵	۷۷/۱۲	۸۴/۹۱	۸۸/۷۶	۹۳/۹۴	۹۶/۴۸	۹۷/۶۷

شکل ۶.۴، نمودار مقادیر جذر میانگین مربعات خطای پیشگویی را به ازای مولفه‌های اول تا سی‌ام روش PLS نشان می‌دهد. همان‌طور که مشخص است، هشتمین مولفه، کمترین خطا را ایجاد کرده است.



شکل ۶.۴: نمودار مقادیر ریشه میانگین مربعات خطای پیشگویی (RMSE) برای متغیر پاسخ CHOL

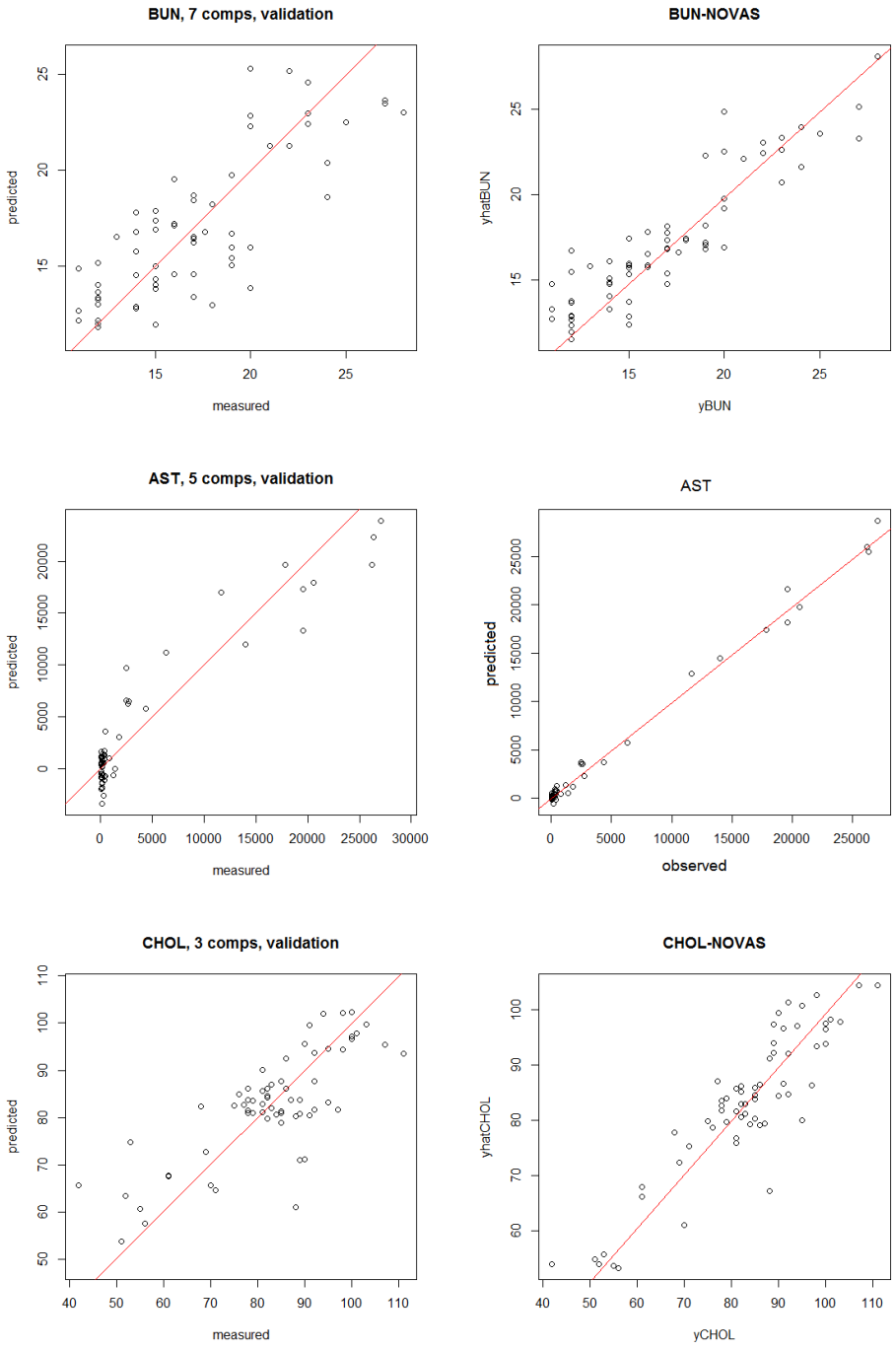
جدول ۲۰.۴ حاوی خطای محاسبه‌شده براساس اعتبارسنجی Leave-one-out به منظور مقایسه‌ی دقت دو روش NOVAS و PLS است. همان‌طور که مشهود است، نتایج خطای محاسبه شده PLS برای متغیر پاسخ BUN برابر ۶/۸۳ است در حالی‌که میزان خطا براساس الگوریتم NOVAS برای همین متغیر برابر ۳/۲۷ است و لذا عملکرد الگوریتم NOVAS بهتر از روش PLS است. برای سیر متغیرهای پاسخ نیز مشاهده می‌شود که الگوریتم NOVAS نتایج مناسب‌تری نسبت به PLS دارد.

جدول ۲۰.۴: مقادیر خطای اعتبارسنجی LOOCV محاسبه شده برای مقادیر اندازگیری بالینی به روش‌های NOVAS و PLS

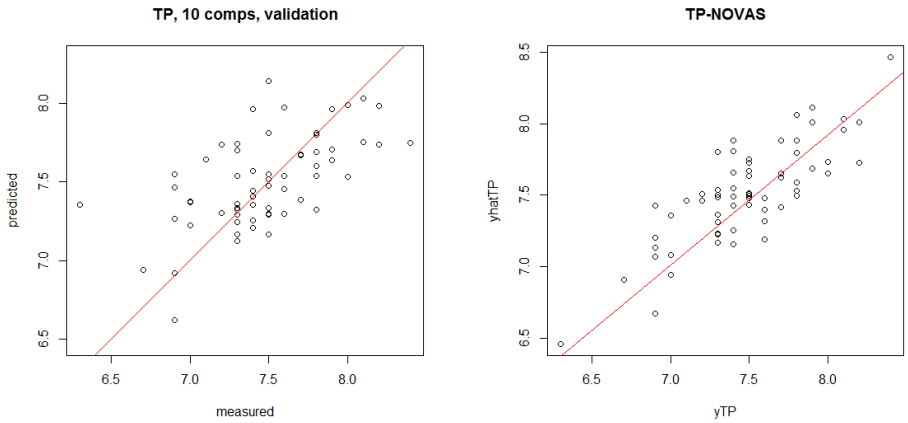
متغیرهای پاسخ	PLS	NOVAS
BUN	۶/۸۳	۳/۲۷
TP	۰/۱۰۲	۰/۰۴۵
ALB	۰/۳۶۸	۰/۰۱۵
ALT	۱۲۹۴۷۴۴	۶۰۶۲۱
SDH	۲۲۰۰۱/۵۱	۱۴۰۴/۷
AST	۵۵۷۴۴۶۱	۳۱۸۶۸۲
ALP	۳۰۷۲/۷۹۵	۱۰۴۳/۷
TBA	۱۳۷/۷۶	۳۰/۷۳
CHOL	۸۰/۴۸	۴۰/۶۸

برای مشاهده عملکرد پیشگویی، نمودار پراکنش مقادیر واقعی و پیشگویی شده متغیرهای پاسخ براساس دو روش NOVAS و PLS در شکل ۷.۴ رسم شده‌اند. شکل‌های سمت راست مربوط به روش NOVAS و شکل‌های سمت چپ مربوط به PLS می‌باشند. تعداد متغیرهای انتخاب شده مدل NOVAS برای متغیرهای پاسخ TP، BUN، AST و CHOL به ترتیب ۷، ۵، ۱۰ و ۴ در گام آخر الگوریتم است و همچنین مدل PLS برای هر یک از متغیرهای پاسخ، براساس مولفه‌ای که کمترین خطا را ایجاد می‌کند، رسم شده‌اند. (جداول ۱۶.۴ تا ۱۹.۴ و شکل‌های ۳.۴ تا ۶.۴).

بار دیگر یادآوری می‌شود که هر چقدر تراکم نقاط در اطراف خط نیمساز نمودارها بیش‌تر باشد و نقاط به این خط نزدیکتر باشند نشان از مطلوبیت پیشگویی‌ها است. از این رو با توجه به نمودارهای شکل ۷.۴ می‌توان نتیجه گرفت که برای ۴ متغیر پاسخ، پیشگویی‌های روش NOVAS نسبت به PLS از مطلوبیت بیش‌تری برخوردار است.



شکل ۷.۴: شکل مقادیر واقعی متغیرهای پاسخ BUN، AST و CHOL در برابر مقادیر پیشگویی شده با دو روش NOVAS (نمودارهای سمت راست)و PLS (نمودارهای سمت چپ)



ادامه شکل ۷.۴ برای نمودار مقادیر واقعی متغیر پاسخ TP، در برابر مقادیر پیشگویی شده با دو روش NOVAS (نمودار سمت راست)و PLS (نمودار سمت چپ)

۴.۲.۴ مقایسه روش NOVAS با LARS

جدول ۲۱.۴ حاوی مقادیر خطای اعتبارسنجی برای روش‌های NOVAS و رگرسیون کمترین زاویه LARS است. مشاهده می‌شود که برای همه‌ی متغیرهای پاسخ، روش NOVAS دارای خطای پیشگویی بسیار کمتری نسبت به روش LARS است و لذا دقت آن به مراتب بالاتر است این نشان می‌دهد که ژن‌های انتخاب شده توسط NOVAS در بین ۳۱۱۶ ژن، مدل پیشگویی دقیق‌تری را نسبت به ژن‌های انتخابی LARS، بوجود آورد.

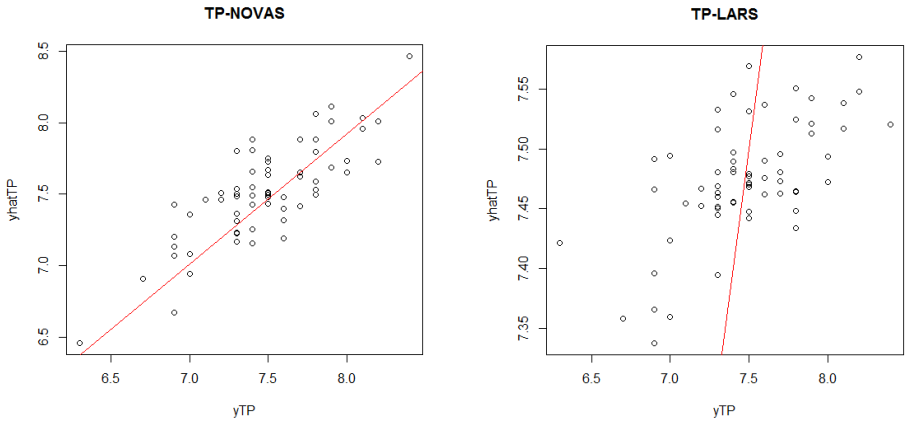
جدول ۲۱.۴: مقادیر خطای اعتبارسنجی LOOCV محاسبه شده برای مقادیر اندازگیری بالینی به روش‌های NOVAS و LARS

متغیرهای پاسخ	LARS	NOVAS
BUN	۸/۶۷	۳/۲۷
TP	۰/۱۱۷	۰/۰۴۵
ALB	۰/۰۴۴	۰/۰۱۵
ALT	۱۶۸۲۸۱۵	۶۰۶۲۱
SDH	۱۹۷۷۳/۷۵	۱۴۰۴/۷
AST	۹۲۵۳۴۳۲	۳۱۸۶۸۲
ALP	۳۱۷۸/۲۸۲	۱۰۴۳/۷
TBA	۱۵۲/۹۷	۳۰/۷۳
CHOL	۹۴/۹۶	۴۰/۶۸

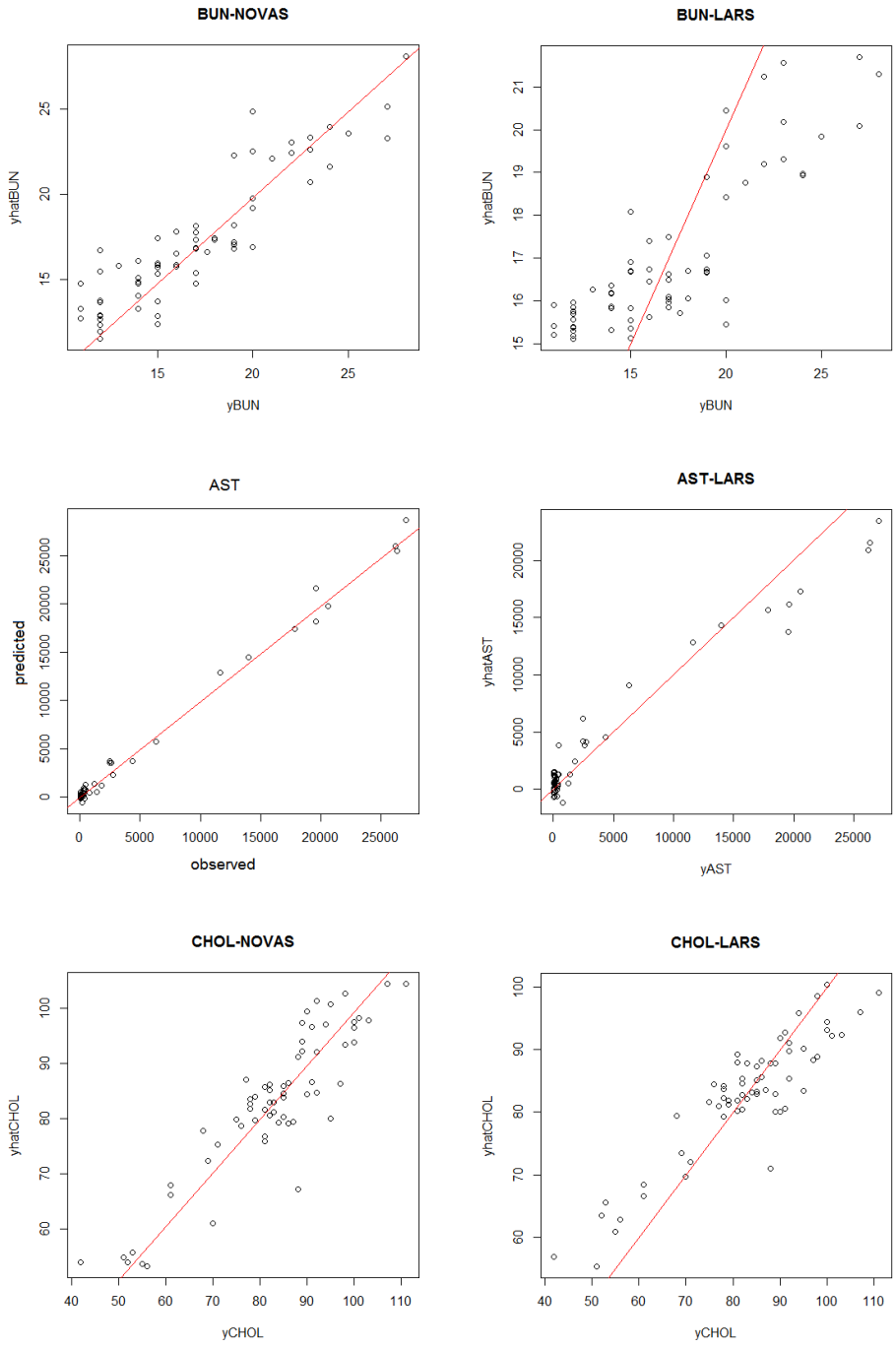
شکل ۸.۴، نمودار پراکنش مقادیر واقعی پاسخ را در مقابل مقادیر پیشگویی شده برای همه‌ی متغیرهای پاسخ در دو روش NOVAS (نمودارهای سمت چپ) و LARS (نمودارهای سمت راست) نشان می‌دهد. برای پاسخ‌های TP، AST، CHOL، و BUN برتری روش NOVAS کاملا مشهود است زیرا تراکم نقاط در اطراف خط نیمساز بیش‌تر بوده و فاصله‌ی کمتری تا این خط دارند. در خصوص پاسخ TP، روش LARS با $R^2 = ۰/۳۹$، در مقایسه با NOVAS که $R^2 = ۰/۸۰$ بسیار نامطلوب است. جدول ۲۲.۴ حاوی مقادیر ضریب تعیین در هر دو مدل عبارتند از:

جدول ۲۲.۴: مقادیر ضریب همبستگی R^2 روش‌های NOVAS و LARS

متغیرهای پاسخ	LARS	NOVAS
BUN	۰/۷۱	۰/۹۰
TP	۰/۳۹	۰/۸۰
AST	۰/۹۶	۰/۹۹
CHOL	۰/۸۲	۰/۹۰



شکل ۸.۴: نمودار مقادیر واقعی متغیر پاسخ TP در برابر مقادیر پیشگویی شده با دو روش NOVAS (نموار سمت چپ)و LAR (نمودار سمت راست)، و ضریب تعیین در این دو روش به ترتیب برابر با $R^2 = ۰/۸۰$ و $R^2 = ۰/۳۹$ است. تعداد متغیرهای انتخاب شده به ترتیب ۷ و ۶۳ است



ادامه شکل ۸.۴ برای نمودار مقادیر واقعی متغیرهای پاسخ BUN، AST و CHOL در برابر مقادیر پیشگویی شده با دو روش NOVAS (نمودارهای سمت چپ)و LAR (نمودارهای سمت راست)، و ضریب تعیین در این دو روش به ترتیب برابر با $R^۲ = ۰/۹۰$، $R^۲ = ۰/۹۹$، $R^۲ = ۰/۹۰$ و $R^۲ = ۰/۷۱$، $R^۲ = ۰/۹۶$، $R^۲ = ۰/۸۲$ است. تعداد متغیرهای انتخاب شده به ترتیب ۵، ۱۰، ۴ و ۶۳ متغیر برای LARS است.

نتیجه‌گیری و پیشنهادات

۳.۴ نتیجه‌گیری

در این پایان‌نامه شیوه‌ای جدید برای انتخاب متغیر و کاهش بعد در مدل‌های غیرخطی معرفی شد. در یک مطالعه شبیه‌سازی کارآیی این نوع شیوه‌ی انتخاب متغیر با تعریف توابعی از سه متغیر X_1, X_2, X_3 مورد ارزیابی قرار گرفت و نتایج آن معرفی گردید. همچنین شیوه‌ی مذکور به منظور انتخاب ژن‌های مهم و تاثیرگذار در حوزه‌ی پزشکی اعمال گردید. نتایج حاصل از آن با نتایج اخذ شده روش PLS و روش انتخاب متغیر LARS مورد مقایسه قرار گرفت. با توجه به نتایج به دست آمده از مطالعه‌ی شبیه‌سازی و مجموعه داده‌های واقعی می‌توان نتیجه گرفت که شیوه‌ی انتخاب متغیر NOVAS در داده‌ها با ابعاد بالا و بسیار بالا ضمن کاهش بعد می‌تواند متغیرهای مهم و تاثیرگذار را شناسایی کند علاوه بر متغیرهایی که به تنهایی تاثیر ندارند اما در ترکیب با دیگر متغیرها تاثیرگذار هستند را به خوبی شناسایی می‌کند.

۴.۴ پیشنهادات آینده

به منظور انجام تحقیقات و پژوهش‌های آینده در زمینه‌ی انتخاب متغیر غیرخطی می‌توان از راه‌کارهای ذیل بهره گرفت

۱. از آنجا که روش‌های انتخاب متغیر خطی مختلف ممکن است نتایج بسیار متفاوتی داشته باشند لذا می‌توان کارآیی الگوریتم غیرخطی را با دیگر روش‌های انتخاب متغیر خطی بررسی نمود.

۲. از آنجایی که انجام الگوریتم پیشنهاد با تابع ناپارامتری رگرسیون موضعی انجام شد، می‌توان از تابع‌های ناپارامتری دیگر برای محاسبه‌ی خطای اعتبارسنجی متقابل استفاده کرد.

پیوست آ

دستورات نرم افزار R

در این پیوست دستورات مربوط به محاسبه‌ی اعتبارسنجی و همین طور الگوریتم NOVAS ارائه شده است.
دستورات مربوط فراخوانی بسته‌های مورد نیاز برای خواندن داده‌ها در R

```
###loadin package####
library(splines)
library(fds)
library(mixOmics)
### preparing data ###
data(liver.toxicity)
X =as.matrix(liver.toxicity$gene)
Y =liver.toxicity$clinic
```

دستورات مربوط به محاسبه‌ی اعتبارسنجی و نمودار پیش‌بینی براساس Leave-one-out

```
novas(Y[,1], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
```

```
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)
#
yhatBUN=novas(Y[,1], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),

maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values
yBUN=Y[,1]

plot(yBUN,yhatBUN,main = "BUN/NOVAS",type = "p")
title(main = "BUN/NOVAS")
abline(lm( yBUN~ yhatBUN), col = 'red')
yBUN=Y[,1]
cor.test(yhatBUN,yBUN)
#####################################################
novas(Y[,3], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)
#
yhatTP=novas(Y[,3], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values
yTP=Y[,3]
plot(yTP,yhatTP,type = "p")
title(main = "TP/NOVAS")
abline(lm( y3~ yhad3), col = 'red')
cor.test(yhatTP,yTP)
############
novas(Y[,4], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),

maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)
#
yhatALB=novas(Y[,4], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),

maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values

yALB=Y[,4]
plot(yALB,yhatALB,type = "p")
```

```
title(main = "ALB/NOVAS")
abline(lm( yALB~ yhatALB), col = 'red')
cor.test(yhatALB,yALB)

#################################
resultsALT=novas(Y[,5], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),

maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)
#
yhatALT=novas(Y[,5], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values

yALT=Y[,5]
plot(yALT,yhadALT,type = "p")
title(main = "ALT/NOVAS")
abline(lm( yALT~ yhatALT), col = 'red')
cor.test(yhatALT,yALT)

###############################################
novas(Y[,6], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)
#
yhatSDH=novas(Y[,6], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values

ySDH=Y[,6]
plot(ySDH,yhatSDH,type = "p")
title(main = "SDH/NOVAS")
abline(lm( ySDH~ yhatSDH), col = 'red')
cor.test(yhatSDH,ySDH)
#########################################
novas(Y[,7], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),

maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)
```

```
#
yhatAST=novas(Y[,7], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values

yAST=Y[,7]
plot(yAST,yhatAST,type = "p")
title(main = "AST/NOVAS")
abline(lm( yAST~ yhatAST), col = 'red')
cor.test(yhatAST,yAST)

##########################
yhatCHOL=novas(Y[,10], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),

maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values
yCHOL=Y[,10]
plot(yCHOL,yhatCHOL,type = "p")
title(main ="CHOL")
abline(lm( yCHOL~ yhatCHOL), col = 'red')
cor.test(yhatCHOL,yCHOL)

####################################
yhatALP=novas(Y[,8], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values
yALP=Y[,8]
plot(yALP,yhatALP,type = "p")
title(main ="ALP")
title(main ="ALP/NOVAS")
abline(lm( yALP~ yhatALP),col = 'red')
############################################
yhatTBA=novas(Y[,9], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values
yTBA=Y[,9]
plot(yTBA,yhatTBA,type = "p")
title(main ="CHOL")
```

```
title(main ="CHOL/NOVAS")
abline(lm(YTBA~ yhadTBA), col = 'red')
cor.test(yhatCHOL,yCHOL)
##############################################
yhadTBA=novas(Y[,9], X,  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$Estimated.values
YTBA=Y[,9]
plot(YTBA,yhadTBA,type = "p")
abline(lm(YTBA~ yhadTBA), col = 'red')
```

دستورات مربوط به PLS

```
#.............................. PLS for ALB.............
#......

ALBfit=plsr(ALB ~ X, ncomp = 30, validation = "LOO")
plot(RMSEP(ALBfit), legendpos = "topright")

#......... number of component

ALBcv=mean((ALB-ALBfit$validation$pred[,,13])^2)
ALBcv
ALBcv1=mean((ALB-ALBfit$validation$pred[,,12])^2)
ALBcv1
ALBcv2=mean((ALB-ALBfit$validation$pred[,,14])^2)
ALBcv2
ALBcv3=mean((ALB-ALBfit$validation$pred[,,10])^2)
ALBcv3
ALBcv4=mean((ALB-ALBfit$validation$pred[,,15])^2)
ALBcv4
ALBcv5=mean((ALB-ALBfit$validation$pred[,,9])^2)
ALBcv5
ALBcv6=mean((ALB-ALBfit$validation$pred[,,16])^2)
ALBcv6
```

```
ALBcv7=mean((ALB-ALBfit$validation$pred[,,18])^2)
ALBcv7
ALBcv8=mean((ALB-ALBfit$validation$pred[,,20])^2)
ALBcv8
ALBcv9=mean((ALB-ALBfit$validation$pred[,,22])^2)
ALBcv9
ALBcv10=mean((ALB-ALBfit$validation$pred[,,24])^2)
ALBcv10
ALBm=summary(ALBfit)

plot(ALBfit, ncomp = 18, asp = 1, line = TRUE)

Rsquare=(cor(ALB,ALBfit$fitted.values[,,18]))^2
Rsquare
#######################
#.............................. PLS for AST.............
#......

ASTfit=plsr(AST ~ X, ncomp = 30, validation = "LOO")
plot(RMSEP(ASTfit), legendpos = "topright")

#......... number of component

ASTcv=mean((AST-ASTfit$validation$pred[,,10])^2)
ASTcv
ASTcv1=mean((AST-ASTfit$validation$pred[,,11])^2)
ASTcv1
ASTcv2=mean((AST-ASTfit$validation$pred[,,14])^2)
ASTcv2
ASTcv3=mean((AST-ASTfit$validation$pred[,,6])^2)
ASTcv3
ASTcv4=mean((AST-ASTfit$validation$pred[,,5])^2)
ASTcv4
```

```
ASTm=summary(ASTfit)
plot(ASTfit, ncomp = 5, asp = 1, line = TRUE)

Rsquare=(cor(AST,ASTfit$fitted.values[,,5]))^2
Rsquare
#########################
ALTfit=plsr(ALT ~ X, ncomp = 30, validation = "LOO")
plot(RMSEP(ALTfit), legendpos = "topright")

ALTcv=mean((ALT-ALTfit$validation$pred[,,10])^2)
ALTcv
ALTcv1=mean((ALT-ALTfit$validation$pred[,,5])^2)
ALTcv1
ALTcv2=mean((ALT-ALTfit$validation$pred[,,6])^2)
ALTcv2
ALTcv3=mean((ALT-ALTfit$validation$pred[,,4])^2)
ALTcv3
ALTm=summary(ALTfit)
plot(ALTfit, ncomp = 5, asp = 1, line = TRUE)

Rsquare=(cor(ALT,ALTfit$fitted.values[,,5]))^2
Rsquare
##################
CHOLfit=plsr(CHOL ~ X, ncomp = 30, validation = "LOO")
plot(RMSEP(CHOLfit), legendpos = "topright")



#......... Look at the figure. you can see the min cv was obtained for
#......... number of component=7

TBAcv=mean((CHOL-CHOLfit$validation$pred[,,3])^2)
TBAcv
TBAcv1=mean((CHOL-CHOLfit$validation$pred[,,2])^2)
TBAcv1
```

```
TBAcv2=mean((CHOL-CHOLfit$validation$pred[,,4])^2)
TBAcv2
TBAcv3=mean((CHOL-CHOLfit$validation$pred[,,10])^2)
TBAcv3
TBAcv4=mean((TBA-TBAfit$validation$pred[,,14])^2)
TBAcv4
TBAcv5=mean((TBA-TBAfit$validation$pred[,,20])^2)
TBAcv5
TBAcv6=mean((TBA-TBAfit$validation$pred[,,2])^2)
TBAcv6
TBAm=summary(CHOLfit)

plot(CHOLfit, ncomp = 3, asp = 1, line = TRUE)

Rsquare=(cor(CHOL,CHOLfit$fitted.values[,,3]))^2
Rsquare
```

دستورات مربوط به LARS

```
############RESPONSE 1 BUN############
################################################
lars.BUN <-lars(y=BUN, x=X,type="lar",use.Gram=FALSE)
summary(lars.BUN)
cv.lars.BUN<-cv.lars(y=BUN,x= X,K=64,type="lar",mode="fraction",use.Gram=FALSE)
##min CV###
min.CV.BUN=min(cv.lars.BUN$cv)
min.CV.BUN
#####plot yhat
s.cv.2 <-cv.lars.BUN$index[min(which(cv.lars.BUN$cv < limit))]
predictBUN<-predict(lars.BUN, newx=X, s=s.cv.2, mode="fraction")
plot(x=BUN, y=predictBUN$fit,xlab="yBUN", ylab="yhatBUN")
title(main = "BUN/LARS")
axis(2,at=seq(0,25,5))
abline(a=0,b=1,col = 'red')
Rsquare=(cor(BUN,predictBUN$fit))^2
```

```
Rsquare
################################
#######response 2 TP#########
####################################
lars.TP <-lars(y=TP, x=X,type="lar",use.Gram=FALSE)
summary(lars.TP)
cv.lars.TP<-cv.lars(y=TP,x= X,K=64,type="lar",mode="fraction",use.Gram=FALSE)
##min CV###
min.CV.TP=min(cv.lars.TP$cv)
min.CV.TP
#####plot yhat
s.cv.2 <-cv.lars.TP$index[min(which(cv.lars.TP$cv < limit))]
predictTP<-predict(lars.TP, newx=X, s=s.cv.2, mode="fraction")
plot(x=TP, y=predictTP$fit,
xlab="yTP", ylab="yhatTP")
abline(a=0,b=1)
title(main = "TP/LARS")
Rsquare=(cor(TP,predictTP$fit))^2
Rsquare
################################
#######RESPONSE 6 AST#######
################################
lars.AST <-lars(y=AST, x=X,type="lar",use.Gram=FALSE)
summary(lars.AST)
cv.lars.AST<-cv.lars(y=AST,x= X,K=64,type="lar",mode="fraction",use.Gram=FALSE)
##min CV###
min.CV.AST=min(cv.lars.AST$cv)
min.CV.AST
#####plot yhat
s.cv.2 <-cv.lars.AST$index[min(which(cv.lars.AST$cv < limit))]
predictAST<-predict(lars.AST, newx=X, s=s.cv.2, mode="fraction")
plot(x=AST, y=predictAST$fit,
xlab="yAST", ylab="yhatAST")
title(main = "AST/LARS")
```

```
abline(a=0,b=1)
Rsquare=(cor(AST,predictAST$fit))^2
Rsquare
#########################
########33response 8 TBA
#########################
lars.TBA <-lars(y=TBA, x=X,type="lar",use.Gram=FALSE)
summary(lars.TBA)
cv.lars.TBA<-cv.lars(y=TBA,x= X,K=64,type="lar",mode="fraction",use.Gram=FALSE)
##min CV###
min.CV.TBA=min(cv.lars.TBA$cv)
min.CV.TBA
#####plot yhat
s.cv.2 <-cv.lars.TBA$index[min(which(cv.lars.TBA$cv < limit))]
predictTBA<-predict(lars.TBA, newx=X, s=s.cv.2, mode="fraction")
plot(x=TBA, y=predictTBA$fit,
xlab="yTBA", ylab="yhatTBA")
title(main = "TBA/LARS")
abline(a=0,b=1)
Rsquare=(cor(TBA,predictTBA$fit))^2
Rsquare
#########################
############################
#######response 9 CHOL#####
############################
lars.CHOL <-lars(y=CHOL, x=X,type="lar",use.Gram=FALSE)
summary(lars.CHOL)
cv.lars.CHOL<-cv.lars(y=CHOL,x= X,K=64,type="lar",mode="fraction",use.Gram=FALSE)
##min CV###
min.CV.CHOL=min(cv.lars.CHOL$cv)
min.CV.CHOL
#####plot yhat
s.cv.2 <-cv.lars.CHOL$index[min(which(cv.lars.CHOL$cv < limit))]
predictCHOL<-predict(lars.CHOL, newx=X, s=s.cv.2, mode="fraction")
```

```
plot(x=CHOL, y=predictCHOL$fit,
xlab="yCHOL", ylab="yhatCHOL")
title(main = "CHOL/LARS")
abline(a=0,b=1)
Rsquare=(cor(CHOL,predictCHOL$fit))^2
Rsquare
#################################
```

دستورات مربوط به شبیه‌سازی (تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای $X_۱, X_۲, X_۳$ به عنوان متغیر مهم درست شناسایی شده‌اند) براساس مدل ۱

```
library(doSNOW)
# signal-to-noise ratio set to 0.05
snr.model = 0.05
# Proceed to 100 runs
niter = 100
model1=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod1.snr.mod0.05.n.p= list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression = COVARIATES[, 1]^2 + COVARIATES[, 2]^2 + COVARIATES[, 3]^2
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod1.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}
table=table(c(novas.mod1.snr.mod0.05.n.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod1.snr.mod0.05.n.p,setequal,1:3))/niter
list(table=table, mean=mean, sum=sum)
}
```

```
model22(50,10000)
```

دستورات مربوط به شبیه‌سازی (تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم درست شناسایی شده‌اند) براساس مدل ۲

```
library(doSNOW)
signal-to-noise ratio set to 0.05
snr.model = 0.05
# Proceed to 100 runs
niter = 100
model22=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod2.snr.mod0.05.n.p = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= abs(COVARIATES[, 1] * COVARIATES[, 2])
 + abs(COVARIATES[, 1] * COVARIATES[, 3])
 + abs(COVARIATES[, 2] * COVARIATES[, 3])
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod2.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}
table=table(c(novas.mod2.snr.mod0.05.n50.p100,recursive=T))
mean=sum(sapply(novas.mod2.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod2.snr.mod0.05.n.p,setequal,1:3))
list(table=table, mean=mean, sum=sum)
}
```

دستورات مربوط به شبیه‌سازی (تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم درست شناسایی شده‌اند) براساس مدل ۳

```
model3=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod3.snr.mod0.05.n.p= list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= abs(COVARIATES[, 1] * COVARIATES[, 2] * COVARIATES[, 3])
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod3.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
 Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}


table=table(c(novas.mod3.snr.mod0.05.n.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod3.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod3.snr.mod0.05.n.p,setequal,1:3))
list(table=table, mean=mean, sum=sum)
}
```

دستورات مربوط به شبیه‌سازی (تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم درست شناسایی شده‌اند) براساس مدل ۴

```
model4=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod4.snr.mod0.05.n.p= list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= (abs(COVARIATES[, 1] * COVARIATES[, 2])
+ COVARIATES[, 3]^2) / (2 + COVARIATES[, 1] * COVARIATES[, 2] * COVARIATES[, 3])
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
```

```
Responses <- Regression + Error
novas.mod4.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
 Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}

table=table(c(novas.mod4.snr.mod0.05.n.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod4.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod4.snr.mod0.05.n.p,setequal,1:3))
list(table=table,mean=mean,sum=sum)
}
```

دستورات مربوط به شبیه‌سازی (تعداد دفعات از ۱۰۰ بار تکرار که متغیرهای X_1, X_2, X_3 به عنوان متغیر مهم درست شناسایی شده‌اند) براساس مدل ۵

```
model5=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod5.snr.mod0.05.n.p = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= (abs(COVARIATES[, 1] * COVARIATES[, 2]) + abs(COVARIATES[, 1] * COVARIATES[, 3]
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod5.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES, Bwknn.ratio = seq(0.05, 0
maxsteps = 6, exponent = 0.5, threshold = 0.1, screening = F)$model
}
# details the set of selected variables with the number of times, out of 100, that each of th
table=table(c(novas.mod5.snr.mod0.05.n.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod5.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod5.snr.mod0.05.n.p,setequal,1:3))
list(table=table,mean=mean,sum=sum)
```

```
}
```

دستورات مربوط به تغییر پارامتر آستانه‌ی t

تغییر مقدار ۰/۰۱, ۰/۰۵, ۰/۱, ۰/۱۵, ۰/۲, ۰/۳, ۰/۴ $= threshold$ برای مدل ۱

```
model1=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod1.snr.mod0.05.n.p= list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression = COVARIATES[, 1]^2
 + COVARIATES[, 2]^2 + COVARIATES[, 3]^2
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod1.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.01, screening = F)$model
}
table=table(c(novas.mod1.snr.mod0.05.n50.p100,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod1.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod1.snr.mod0.05.n.p,setequal,1:3))
list(table=table,mean=mean,sum=sum)
}
```

دستورات مدل ۲

$threshold =$ ۰/۰۱, ۰/۰۵, ۰/۱, ۰/۱۵, ۰/۲, ۰/۳, ۰/۴

```
model2=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod2.snr.mod0.05.n50.p100 = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
```

```
Regression= abs(COVARIATES[, 1] * COVARIATES[, 2]) +
 abs(COVARIATES[, 1] * COVARIATES[, 3]) + abs(COVARIATES[, 2] * COVARIATES[, 3])
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod2.snr.mod0.05.n50.p100[[ii]] = novas(Responses,
 COVARIATES,
Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}
table=table(c(novas.mod2.snr.mod0.05.n50.p100,recursive=T))
mean=sum(sapply(novas.mod2.snr.mod0.05.n50.p100,setequal,1:3))/niter
sum=sum(sapply(novas.mod2.snr.mod0.05.n50.p100,setequal,1:3))
list(table=table, mean=mean, sum=sum)
}
```

دستورات مدل ۳ ۴ $threshold = ۰/۰۱, ۰/۰۵, ۰/۱, ۰/۱۵, ۰/۲, ۰/۳, ۰/۴$

```
model3=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod3.snr.mod0.05.n.p = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= abs(COVARIATES[, 1] * COVARIATES[, 2] * COVARIATES[, 3])
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod3.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
 Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}

table=table(c(novas.mod3.snr.mod0.05.n1.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
```

```
mean=sum(sapply(novas.mod3.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod3.snr.mod0.05.n.p,setequal,1:3))
list(table=table, mean=mean, sum=sum)
}
```

دستورات مدل ۴ ۴/۰, ۳/۰, ۲/۰, ۱۵/۰, ۱/۰, ۰۵/۰, ۰۱/۰ = $threshold$

```
model44=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod4.snr.mod0.05.n.p = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= (abs(COVARIATES[, 1] * COVARIATES[, 2])
 + COVARIATES[, 3]^2) / (2 + COVARIATES[, 1] * COVARIATES[, 2] * COVARIATES[, 3])
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod4.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}
table=table(c(novas.mod4.snr.mod0.05.n.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod4.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod4.snr.mod0.05.n.p,setequal,1:3))
list(table=table, mean=mean, sum=sum)
}
```

دستورات مدل ۵ ۴/۰, ۳/۰, ۲/۰, ۱۵/۰, ۱/۰, ۰۵/۰, ۰۱/۰ = $threshold$

```
model5=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod5.snr.mod0.05.n.p = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
```

```
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression= (abs(COVARIATES[, 1] * COVARIATES[, 2])
 + abs(COVARIATES[, 1] * COVARIATES[, 3])) / (2 + abs(COVARIATES[, 2] * COVARIATES[, 3]))
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
Responses <- Regression + Error
novas.mod5.snr.mod0.05.n.p[[ii]] = novas(Responses, COVARIATES,
 Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}
table=table(c(novas.mod5.snr.mod0.05.n.p,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod5.snr.mod0.05.n.p,setequal,1:3))/niter
sum=sum(sapply(novas.mod5.snr.mod0.05.n.p,setequal,1:3))
list(table=table,mean=mean,sum=sum)
}
```

دستورات مربوط به جدول NOISE-TO-SIGNAL به ازای مقادیر $nsr = ۰/۰۵, ۰/۱, ۰/۲$

```
snr.model = 0.05,0.1,0.2
# Proceed to 100 runs
niter = 100
#########
# MODEL 1
#########
model1=function(n,p){
COVARIATES = matrix(0, nrow = n, ncol = p)
novas.mod1.snr.mod0.05.n50.p100 = list()
for(ii in 1:niter){
cat("iteration ",ii,"\n")
for(jj in 1:p) COVARIATES[, jj] = runif(n, min = -1, max = 1)
Regression = COVARIATES[, 1]^2 + COVARIATES[, 2]^2
 + COVARIATES[, 3]^2
sdev.model <- sqrt(snr.model * var(Regression))
Error <- rnorm(n, sd = sdev.model)
```

```
Responses <- Regression + Error
novas.mod1.snr.mod0.05.n50.p100[[ii]] = novas(Responses, COVARIATES,
  Bwknn.ratio = seq(0.05, 0.5, length = 5),
maxsteps = 6, exponent = 0.5, threshold = 0.05, screening = F)$model
}
table=table(c(novas.mod1.snr.mod0.05.n50.p100,recursive=T))
# print the number of times, out of 100, that NOVAS selected the correct model
mean=sum(sapply(novas.mod1.snr.mod0.05.n50.p100,setequal,1:3))/niter
sum=sum(sapply(novas.mod1.snr.mod0.05.n50.p100,setequal,1:3))
list(table=table,mean=mean,sum=sum)
}
```

مراجع

[۱] احمدی، ع. و طالبی، ه. (۱۳۸۹)، مقایسه‌ی روش‌های برآوردیابی و انتخاب متغیر در مدل‌های رگرسیونی با استفاده از داده‌های شبیه‌سازی، *مجله اندیشه آماری* ، سال پانزدهم، **۲**، ۳۹-۲۹.

[۲] شاهینی، ا. و سرمد، م. (۱۳۸۸)، رگرسیون کمترین توان‌های دوم، *مجله گزیده مطالب آماری*، سال بیستم، شماره ۲، ۳۰۴-۲۸۹

[۳] راس، شلدون، مبانی احتمال، ترجمه احمد پارسیان، علی همدانی، *انتشارات شیخ بهایی*،۱۳۹۲، ویرایش هشتم.

[۴] خسروی، م،(۱۳۸۷)،**''استنباط آماری تابع رگرسیون ناپارامتری''**، پایان‌نامه کارشناسی ارشد، دانشگاه شیراز.

[۵] گودرزی م، (۱۳۸۹)،**''تاثیر موضعی و ماتریس تاثیر استاندارد شده در رگرسیون کمترین توان‌های جزیی''**، پایان‌نامه کارشناسی ارشد، دانشگاه شهید چمران اهواز.

[6] Buhlmann, P., Van De Geer, S. (2011). ***Statistics for high-dimensional data: methods, theory and applications***. Springer Science Business Media.

[7] Bushel, P. R., Wolfinger, R. D., Gibson, G. (2007). Simultaneous clustering of gene expression data with clinical chemistry and pathological evaluations reveals phenotypic prototypes. *BMC Systems Biology*, **1(1)**, 1-15.

[8] Candes, E., Tao, T. (2007). The Dantzig selector: statistical estimation when p is much larger than n. *The Annals of Statistics*, **35(6)**, 2313-2351.

[9] Carta, J. A., Velázquez, S., and Cabrera, P. (2013). A review of measure-correlate-predict (MCP) methods used to estimate long-term wind characteristics at a target site. *Renewable and Sustainable Energy Reviews*, 27, 362-400.

[10] Dejean, S., Gonzalez, I., Le Cao, K.-a., Monget, P., and Coquery, J. (2011). "mixOmics: Omics data integration project. R package version 3.0," available at http://CRAN.R-project.org/ package=mixOmics.

[11] Efron, B., Hastie, T., Johnstone, I., Tibshirani, R. (2004). Least angle regression. *The Annals of statistics*, **32(2)**, 407-499.

[12] Fan, J., Gijbels, I. (1996). ***Local Polynomial Modelling and Its Applications***: Monographs on Statistics and Applied Probability 66 (Vol. 66). CRC Press.

[13] Fan, J., Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, **96(456)**, 1348-1360.

[14] Fan, J., Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, **20(1)**, 101-148.

[15] Ferraty, F., Hall, P. (2015). An algorithm for nonlinear, nonparametric model choice and prediction. *Journal of Computational and Graphical Statistics*,**24(3)**, 695-714.

[16] Ferraty, F., Hall, P., Vieu, P. (2010). Most-predictive design points for functional data predictors. *Biometrika*, **97(4)**, 807-824.

[17] Friedman, J., Hastie, T., Höfling, H., Tibshirani, R. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics*, **1(2)**, 302-332.

[18] Fu, W. J. (1998). Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, **7(3)**, 397-416.

[19] Fan, J., and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(5), 849-911.

[20] Fan, J., and Ren, Y. (2006). Statistical analysis of DNA microarray data in cancer research. *Clinical Cancer Research*, **12(15)**, 4469-4473.

[21] Fan, J., and Fan, Y. (2008). High dimensional classification using features annealed independence rules. *Annals of statistics*,**36(6)** , 2605.

[22] Geladi, P., Kowalski, B. R. (1986). Partial least-squares regression: a tutorial. *Analytica chimica acta*, **185**, 1-17.

[23] Hastie, T., Tibshirani, R., and Friedman, J. (2009), The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.), New York: Springer.

[24] Huang, J., Horowitz, J. L., Wei, F. (2010). Variable selection in nonparametric additive models. *Annals of statistics*, **38(4)**, 2282-2313.

[25] Le Cao, K. A., Rossouw, D., Robert-Granie, C., and Besse, P. (2008). A Sparse PLS for Variable Selection when integrating omics data. *Statistical applications in genetics and molecular biology*. **7**. 35

[26] Martens, H., Geladi, P. (1989). Multivariate calibration. *John Wiley*.

[27] Mevik, B. H., Wehrens, R., and San Michele all'Adige, T. N. (2015). *Introduction to the pls Package. Help Section of The "pls" package of RStudio Software*, 1-23.

[28] Mevik, B. H., and Cederkvist, H. R. (2004). *Mean squared error of prediction (MSEP) estimates for principal component regression (PCR) and partial least squares regression (PLSR). Journal of Chemometrics*, 18(9), 422-429.

[29] Meier, L., Van de Geer, S., Bühlmann, P. (2009). High-dimensional additive modeling. *The Annals of Statistics*, **37(6B)**, 3779-3821.

[30] Meinshausen, N. (2007). Relaxed lasso. *Computational statistics and data analysis*.**52**, 374–393.

[31] R Development Core Team (2011), R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0. Available at http://www.R-project.org/

[32] Ravikumar, P., Lafferty, J., Liu, H., and Wasserman, L. (2009). Sparse additive models. *Journal of the royal statistical society*. **71**, 1009–1030.

[33] Revolution Analytics (2011), "doSNOW: Foreach Parallel Adaptor for the Snow Package. R Package Version 1.0.5," available at http://CRAN.R-project.org/package=doSNOW

[34] Tibshirani, R. (1996). Regression analysis and selection via the lasso. *Journal of the Royal Statistical Society*. **58**, 267–288

[35] Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, **99(10)** , 6567-6572.

[36] Yuan, M., Lin, Y. (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **68(1)**, 49-67.

[37] Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association*, **101(476)**, 1418-1429.

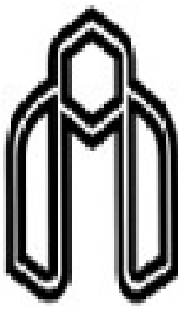
[38] Zou, H., Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **67(2)**, 301-320.

[39] Zou, H., Li, R. (2008). One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, **36(4)**, 1534-1541.

Abstract

Analysis of high-dimensional data in all fields of science, industry and commerce is always faced with the challenge of choosing important variables. Common methods for selecting variables are based on linear structures, but in many cases and in the above, the structural relationship between the response variable and the explanatory variables may not be. Also, these methods ignore variables that alone have little effect on the variable, but in combination with other variables for prediction. In this research, by providing a multi-stage algorithm based on linear local regression, important variables are identified and selected in the form of subsets, and the prediction model is fitted.

Keywords: Local regression, Cross validation, Combination of Variables, Variable selection



Shahrood University of Technology

Faculty of Mathematical Sciences

M.Sc. Thesis in Mathematical Statistics

Selection of variables in big data by nonlinear modeling

By: Mahboubeh Mohebbi

Supervisor

Dr. Davood Shahsavani

January 2018