روش کشف کلاه برداری از طریق پردازش کلان داده در سیستم‌های پرداخت اعتباری

محمد حسین مطیع بیرجندی

دانشکده مهندسی برق و کامپیوتر دانشگاه تهران moti.hosein@ut.ac.ir

چکیده:

کارت‌ها اعتباری و کارت‌های پیش پرداخته[1] بسیار محبوب و پر استفاده هستند. از این کارت‌ها بسادگی می‌توان جهت کلاهبرداری استفاده نمود و در زمینه کلاهبرداری بسیار آسیب پذیر هستند. هدف از این مقاله ارائه یک راه حل اتوماتیک و کارآمد جهت تشخیص این کلاهبرداری ها و تمرکز بر روی کاهش هشدارهایی که در اثر تشخیص اشتباه کلاهبرداری به صدا درمی‌آیند، می‌باشد. بر خلاف الگوریتم‌های موجود که براساس رفتار مشتری عمل می‌کنند و نمی‌توانند تا دنباله‌ای از کلاهبرداری ها را تشخیص دهند و یا با تغییر رفتار مشتری دچار خطا می‌شوند، تکنیک و الگوریتم پیشنهادی هر دو مشکل را حل کرده است. در انتها تکنیک ارائه شده بر روی مجموعه‌ی واقعی از تراکنش های اعتباری در بازه سال ۲۰۱۲-۲۰۱۰ اعمال شده است. این روش می‌تواند بصورت بی‌درنگ تقلب را تشخیص دهد.

مقدمه:

امروزه تراکنش‌های بانکی زیادی در هر روز صورت می‌پذیرد. با افزایش فروشگاه‌های آنلاین و افزایش خرید از آنها و نیز توسعه تجارت الکترونیک در دنیای امروز، تقلب و کلاهبرداری در معاملات بانکی و تراکنش‌های روزانه مشتریان بانک‌ها در حال افزایش است. سالانه میلیون‌ها دلار ضرر به صنعت بانکداری و مشتریان بانک‌ها در اثر کلاهبرداری‌های گوناگون وارد می‌شود. بعنوان مثال در سال ۲۰۱۶ بیش از ۱۶میلیارد دلار در ایالات متحده آمریکا به مشتریان بانکی ضرر وارد شد.[1] از سوی دیگر تعداد تراکنش‌های کارت‌های اعتباری در طول یک روز بسیار زیاد است، که این امر مستلزم استفاده از کلان‌داده و ابزاری‌های مدل سازی مرتبط با آن می‌باشد. در زمان خرید، زمانی که یک تراکنش جعلی تشخیص داده شد سیستم تشخیص کلاهبرداری می‌تواند بعنوان یک عامل پیشگیری کننده از انجام تراکنش ممانعت بعمل آورد. حتی تشخیص مبالغ پایین کلاهبرداری نیز اهمیت بالایی دارد زیرا که سارقین معمولا کارت‌های مسروقی را با مبالغ پایین تست می‌کنند.[2] روش‌های تشخیص کلاهبرداری را می‌توان به دو دسته اصلی تقسیم نمود: تشخیص سوءاستفاده و تشخیص ناهنجاری. در روش‌های تشخیص سوءاستفاده سیستم براساس تراکنش‌هایی که می‌دانیم در آنها کلاهبرداری رخ داده‌است، تمرین داده می‌شود و تنها می‌تواند کلاهبرداری‌هایی را که مشابه آن در داده‌هایی که روی آن آموزش داده شده است را تشخیص دهد. در روش‌های تشخیص ناهنجاری سیستم روی تراکنش‌های نرمال آموزش داده می‌شود لذا توانایی کشف کلاهبرداری با روش‌های جدید را داراست.[3] با افزایش روزافزون پیش بینی کننده‌های در دسترس، روش‌های یادگیری آماری که بطور موثر یک مدل پارسیونی[2] با عملکرد پیش‌بینی برتر معرفی می‌کنند، یک روش ایده‌آل برای توسعه سیستم‌های تشخیص

[1] Prepaid cards

[2] Parsimonious model

کلاهبرداری محسوب می‌شوند. اما از سوی دیگر حجم بالای داده موجب می‌شود تا پیاده سازی و توسعه ساده‌ترین روش ها بصورت موثر کاری سخت یا حتی غیر ممکن گردد. اگرچه در نگاه اول بنظر می‌رسد که سخت افزارهای جدید و تکنیک‌های جدید مورد استفاده در کلان‌داده مانند پردازش ابری، بسیاری از مشکلات را حل کرده‌اند اما مشکلاتی مانند هزینه منابع مصرفی، سربار محاسباتی و مسئله نقض حریم خصوصی و امنیت کاربران باعث می شوند تا عمده این روش‌ها کمتر جذاب و قابل استفاده باشند. در عوض یک روش ساده‌تر می‌تواند مشکلات کلان داده را بصورت الگوریتمی حل نماید. روش‌های گوناگون و مختلفی شامل نمونه برداری تصادفی، شکست و سرهم سازی و روش‌های یادگیری آنلاین برای تحلیل کلان‌داده توسعه داده شده است. درحالیکه برخی روش‌ها برای ساده سازی انتخاب مدل توسعه یافته‌اند، تمرکز در درجه اول بر روی تکنیک‌های منظم سازی بوده است، که می‌تواند در زمان کار با ساختمان داده‌های پیچیده مانند مجموعه داده همبسته غیرگوسی سخت و چالش برانگیز باشد. بنابراین به تکنیک‌های موثری نیاز داریم که بتوانند ساختمان‌داده‌های پیچیده مانند آنچه بیان شد را بصورت ساده‌تر تحلیل و پردازش کنند.

"انتخاب مرحله‌ای" یک روش انتخاب کلاسیک است که بعنوان یک جایگزین برای تکنیک‌های منظم سازی معروفی مانند Lasso که مجددا مورد توجه محققین قرار گرفته است، استفاده می‌شود. "انتخاب مرحله‌ای" با اینکه ارتباط قوی با تکنیک منظم سازی دارد، انعطاف پذیری بیشتری برای تعامل با ساختمان داده‌های پیچیده تر دارد. فرضیه پایه‌ای روش "انتخاب مرحله‌ای" اجرای روش "آهسته دمیدن" است. همانطور که از نام روش "انتخاب مرحله‌ای" انتظار داریم، فرایند انتخاب مرحله‌ای با یک مدل خالی شروع می‌شود و در طی چندین مرحله تکرار یادگیری موثر، مدل ساخته می‌گردد.

این پژوهش خانواده‌ای جدید از روش‌های مرحله‌ای تصادفی را معرفی می‌کند که از زیرنمونه برداری[۳] جهت حل مسئله انتخاب مدل در کلان‌داده با قابلیت پشتیبانی از ساختمان داده‌های پیچیده استفاده می‌کند. در ادامه ابتدا به بررسی مسائل تئوری پیرامون انتخاب مرحله‌ای می‌پردازیم. لذا در ابتدا روش انتخاب مرحله‌ای تصادفی را بررسی می‌کنیم. سپس به بررسی داده‌های خوشه‌ای می‌پردازیم. در انتهای این بخش به بررسی معادله تخمینی انتخاب مرحله‌ای تصادفی خواهیم پرداخت. با پایان این بخش به شبیه سازی داده‌های گوسی و باینری می‌پردازیم. در بخش بعد به بررسی تشخیص کلاهبرداری به کمک روش‌های بیان شده خواهیم پرداخت. در نهایت در بخش جمع بندی به بررسی و مقایسه نتایج روش‌های معرفی شده روی دو نوع داده‌ای که در بخش شبیه سازی بررسی شد، می‌پردازیم.

روش‌ها

روش زیر نمونه برداری (Sub-sampling)

روش های زیرنمونه‌برداری بر روی ایده انتخاب تصادفی داده جهت افزایش کارایی تمرکز دارند. ساده ترین این روش‌ها، انتخاب تصادفی داده از مجموعه داده اولیه و انجام کارهای دلخواه با آن می‌باشد. با توجه به ذات احتمالاتی این روش‌ها، معمولا چندین بار تکرار این فرآیند و بررسی نتایج بدست‌آمده با bootstrap های محبوب و معروف، بسیار رایج است. به تازگی توجه زیادی به این روش‌های تصادفی شده است. بعنوان نمونه یک افزونه برای روش رگرسیون زیرنمونه‌برداری کلاسیک منتشر کرده‌اند که "بهره‌برداری" نامیده می‌شود

[۳] Sub-sampling

(Leveraging for big data regression). در این روش توزیع مشاهدات یکنواخت نیست بلکه، یک مکانیزم مانند بهره‌برداری (leveraging) برای اطمینان از اینکه برخی مشاهدات بیشتر از برخی دیگر نمونه برداری شده‌اند، استفاده می‌شود. G. Vaughan در [۴] بیان کرده است که [۵] می‌گوید که زیرنمونه‌هایی که در فرآیند زیرنمونه‌برداری یا بهره‌برداری بدست آمده‌اند، می‌توانند بعنوان جایگزین برای مصور کردن تمام مجموعه داده مورد استفاده قرار بگیرند.

برخی کارها در کاوش بدنبال کارکردهای بالقوه برای روش‌های تصادفی در روش‌های یادگیری آماری انجام شده است. مثلا در [۴] آمده است که آقای .Ahmed et al در [۶] انتخاب کردن به روش زیرنمونه‌برداری/پایداری را معرفی کرده است که، عطف به روش رگرسیون لاسو که برای مشخص کردن عدم توازن کلاس‌ها در مدل انتخابی است، می‌باشد. براساس دانسته های نویسنده، روش‌های تصادفی برای انتخاب بصورت مرحله‌ای بررسی نشده‌اند.

روش مرحله‌ای

روش انتخاب مرحله‌ای یک روش کلاسیک انتخاب مدل است که به تازگی توجه بسیاری را به خود جلب کرده‌است. ساختار پایه‌ای تمام روش‌های مرحله‌ای بدین صورت است که با یک مدل تهی (خالی) که با بردار ضرایب $\beta^{[0]} = 0$ توصیف شده است، آغاز می‌شوند. سپس پس از چندین بار پیمایش بردار ضرایب براساس گام‌های ساده‌ی یادگیری که محاسبه شده اند، بروزرسانی می‌شود. یک مثال برای تکنیک مرحله‌ای، روش انتخاب مرحله‌ای رو به جلو (forward stagewise selection) است. در این روش از یک مدل خطی چند متغییره استفاده می‌شود. در پیمایش t ام در این روش، الگوریتم i را اینگونه تعریف می‌کند: $i = \arg\max r_j^{[t]}$ تعریف می‌کند که $r_j^{[t]}$ یک همبستگی میان j امین covariate و باقی‌مانده مدل براساس مقادیر فعلی ضرایب $\beta^{[t-1]}$ است. سپس بردار ضرایب مطابق فرمول زیر بروزرسانی می‌گردد.

$$\beta_i^{[t]} = \beta_i^{[t-1]} + \epsilon \, sig(r_i^{[t]})$$

منظور از ϵ در این فرمول، سایز ثابت گام یادگیری می‌باشد که می‌بایست به اندازه کافی کوچک باشد و برای تمام $i \neq j$ داریم $\beta_j^{[t]} = \beta_j^{[t-1]}$. دنباله یا "مسیر" (path) بدست آمده از تقریب ضرایب می‌تواند مشابه روش‌های تنظیمی (regularization) برای انجام انتخاب مدل مورد استفاده قرار گیرد.

میان روش‌های مرحله‌ای و روش‌های تنظیم(regularization) ملاحظه شده یک ارتباط قوی برقرار است. بعنوان مثال، نشان داده شده است که مسیری که توسط روش انتخاب مرحله‌ای رو به جلو تولید می‌شود به شرطی که $\epsilon \to 0$ برقرار باشد، به مسیری که توسط lasso تولید می‌شود میل می‌کند. بعلاوه ثابت شده است که دسته کلی تکنیک‌های مرحله‌ای با تقریب خوبی با همتایان خود در تکنیک‌های regularization برابری می‌کنند. حتی در صورتی که این ارتباط قوی میان تکنیک‌های مرحله‌ای و تکنیک‌های regularization را کنار بگذاریم، روش‌های مرحله‌ای هم در کاربری‌های مختلف منعطف هستند و هم از نظر محاسبات کامپیوتری بهینه هستند.

روش انتخاب مرحله‌ای تصادفی

ما با مشاهده کاربردهای زیرنمونه‌برداری و ترکیب آن با روش مرحله‌ای، روش انتخاب مرحله‌ای تصادفی را مطرح می‌کنیم. مدل را اینگونه بیان می‌کنیم: $Y_i \in \mathbb{R}^1$ پاسخ ما و $X_i^T \in \mathbb{R}^p$ برای covariates مشاهدات می‌باشد که $i = 1,2, \dots, \tilde{n}$ است. f تابع ضرر (loss

function) از فرمی نامعین، بدون شکستگی و محدب است. برای f می‌توان گردایان را محاسبه کرد و نامعین نیست. $\pi^{[t]} = \{\pi_1^{[t]}, \dots, \pi_{\tilde{n}}^{[t]}\}$ احتمال نمونه برداری برای هر مشاهده $t = 1,2, \dots$ باشد و $\{\pi^{[t]}(i)\}_1^{\tilde{r}}$ نشان دهنده یک نمونه رندم با سایز $\tilde{r} = \lfloor q\tilde{n} \rfloor$ از عداد طبیعی از ۱ تا $\tilde{n}$ که q نسبت زیرنمونه‌برداری است که در بازه 0 تا 1 قرار دارد. در صورتی که $q = 1$ باشد، آنگاه روش مرحله‌ای بصورت کلاسیک و قابل پیش بینی خواهد بود. ما نمونه تصادفی تمام مجموعه داده را در گام t ام بصورت $\{Y_{\pi^{[t]}(i)}, X_{\pi^{[t]}(i)}\}_i^{\tilde{r}}$ نمایش میدهیم. با شروع از $\beta^{[0]} = 0$ برای $t = 1,2, \dots$ داریم:

$$Draw\ Random\ sample\ \{\pi^{[t]}(i)\}_1^{\tilde{r}}$$

$$\delta^{[t]} = arg_{\delta \in \mathbb{R}^p} \min\left(\nabla f^{[t]}\left(\beta^{[t-1]}\right), \delta\right) subject\ to\ \varphi(\delta) \leq \epsilon$$

$$\beta^{[t]} = \beta^{[t-1]} + \delta^{[t]}$$

که $f^{[t]}(\beta)$ تابع ضرری است که در β تنها با استفاده از $\{Y_{\pi^{[t]}(i)}, X_{\pi^{[t]}(i)}\}_i^{\tilde{r}}$ محاسبه شده است. برای سادگی ما از نرم نوع ۱ برای φ استفاده کرده‌ایم که در فرمت بروزرسانی $\delta^{[t]}$ که در تحقیقات گذشته بخوبی مطالعه شده است، تاثیر گذار است. اما با این حال همچنان می‌توان از خیلی از توابع پنالتی محدب استفاده نمود.

انتخاب ها از دنباله احتمالات $\pi^{[t]}$ و نمونه برداری با جایگزینی یا بدون آن می‌تواند براساس مسئله ما باشند و حکم کلی ندارند. در [4] برای قسمت زیرنمونه‌برداری در روش stochastic gradient boosting آقای Friedman در[7] از یک توزیع یکنواخت و نمونه‌برداری بدون جایگزینی استفاده کرده است. از سوی دیگر در [5] یک توزیع از زیرنمونه‌ها برای روش leverage linear regression با استفاده از قدرت مشاهدات ایجاد کرد تا مقادیر احتمالات را دیکته کرده و نمونه‌برداری با جایگزینی انجام دهد. بعلاوه می‌توانیم توابع توزیع تطبیقی را براساس کارآیی مدل مانند قدرمطلق خطا (absolute error) یا misclassification که جایگزین توزیع نمونه‌برداری می‌شوند تا بر روی مشاهداتی که پیش بینی آن ها سخت است تمرکز کند، را تصور کنیم. فرکانس نمونه برداری نیز می‌تواند تنظیم شود. مثلا ممکن است یک زیرنمونه قبل از پیمایش اول مورد استفاده مجدد قرار بگیرد و یا زیرنمونه‌های جدید می توانند در بازه‌های مشخص برداشته شوند.

در عمل اندازه گام یا ϵ و مکانیزم پایان الگوریتم باید با دقت مورد توجه قرار گیرد. ارتباط میان الگوریتم‌های مرحله‌ای و regularization به سایز کوچک گام ها تکیه دارد. از طرفی سایز کوچک گام‌ها موجب اتلاف محاسبات و منابع محاسباتی می‌شود. استفاده از مجموعه ساده‌ای از قوانین پایان که براساس تعداد ویژگی‌ها و تعداد پیمایش ها هستند بطور کلی توصیه می‌شوند. اول، دانستن پیش بینی اینکه چه تعدادی از ویژگی‌های ممکن است بر روی پاسخ تاثیر دارند، می‌تواند به دانستن تعداد پیش بینی کننده‌های مورد استفاده در مدل کمک کند. سپس حداکثر تعداد پیمایش‌ها باید مشخص شود.

داده‌های خوشه‌ای

بعنوان مشکل انگیزشی در این پژوهش، داده ها به فرمت خوشه‌ای هستند که ما انتظار داریم مشاهدات برای یک دارنده کارت بشدت همبستگی داشته باشند. استفاده از این همبستگی می‌تواند بهره وری روش ما را بهبود بخشد. یک روش برای کار کردن با دادههای همبسته غیر گوسی استفاده از Generalized Estimation Equation یا GEE هاست. می‌توان به GEE ها بعنوان نسخه جامعیت داده شده از

Generalized Linear Model یا GLM ها نگریست، که از آن بطور مستقیم جهت مدل سازی داده‌های خوشه بندی شده که این باور وجود دارد که میان خوشه‌ها یک همبستگی وجود دارد. تفاوت اصلی این است که برخلاف GLM، در GEE فرض نمی‌شود یک likelihood کامل یا یک ساختار توزیعی داریم. در عوض فقط یک ساختار متوسط حاشیه‌ای و واریانس بر اساس یک پیش بینی کننده خطی و یک پارامتر پراکنش مزاحم فرض می شود که وجود دارد.

معادله تخمینی انتخاب مرحله‌ای

ترکیب کردن GEE ها در فریمورک تصادفی مرحله‌ای در ابتدا نیاز دارد تا معادلات تخمینی را برای گرادیان‌های نامعین جایگزین کنیم. سپس تغییرات کمی اعمال می‌گردد تا خوشه بندی در نظر گرفته شود و بازدهی در روش معادلات تخمینی انتخاب تصادفی مرحله‌ای بدست آید. بطور مشابه برای روش تصادفی مرحله‌ای داریم: $\pi^{[t]} = \{\pi_1^{[t]}, \dots, \pi_{\tilde{n}}^{[t]}\}$ احتمالات نمونه‌برداری برای هر خوشه به ازای $t = 1,2, \dots$ و $\{\pi^{[t]}(i)\}_1^r$ نشان دهنده یک نمونه رندم با سایز $r = \lfloor qn \rfloor$ از عداد طبیعی از ۱ تا n است که q نسبت زیرنمونه‌برداری است که در بازه 0 تا 1 قرار دارد. نمونه تصادفی کل مجموعه داده در مرحله t بصورت $\{Y_{\pi^{[t]}(i)}, X_{\pi^{[t]}(i)}\}_i^r$ داده شده است و معادله تخمینی که براساس این نمونه از داده در β و v محاسبه شده است بصورت $U^{[t]}(\beta, v)$ نمایش داده می‌شود. با شروع از $\beta^{[0]} = 0$ برای $t = 1,2, \dots$ داریم:

$$Draw\ Random\ sample\ \{\pi^{[t]}(i)\}_1^r$$

$$Given\ \beta^{[t-1]}, update\ the\ nuisance\ parameters\ using\ only\ \{Y_{\pi^{[t]}(i)}, X_{\pi^{[t]}(i)}\}_i^r\ to\ obtain\ v^{[t]}$$

$$\delta^{[t]} = arg_{\delta \in \mathbb{R}^p} \min \left(U_{[0]}^{[t]}(\beta^{[t-1]}, v^{[t]}), \delta \right) subject\ to\ \varphi(\delta) \leq \epsilon$$

$$\beta^{[t]} = \beta^{[t-1]} + \delta^{[t]}$$

$$\text{که } U_{[0]}^{[t]}(\beta^{[t-1]}, v^{[t]}) = (U_1^{[t]}(\beta^{[t-1]}, v^{[t]}), \dots, U_p^{[t]}(\beta^{[t-1]}, v^{[t]}))$$

مشابه فرم کلی روش تصادفی مرحله‌ای، تعداد زیادی تابع پنالتی φ وجود دارند که می‌توانند مورد استفاده قرار بگیرند اما ما تنها بر روی نرم ۱ یعنی ℓ_1 تمرکز می‌کنیم.

مزیت شاخص و یکتا در کار کردن با داده های خوشه بندی شده این است که بجای اینکه مشاهده‌های جداگانه را نمونه برداری کنیم، براساس خوشه نمونه برداری انجام می‌دهیم. در نمونه برداری خوشه‌ای مدت زمان محاسبه کاهش می‌یابد و استفاده از منابع محاسباتی کاهش می‌یابد. علت این امر آن است که در هر گام تنها تعداد ثابت و کمی از خوشه‌ها باید در نظر گرفته شوند.

شبیه‌سازی

داده‌های گوسی

ما فرایند شبیه سازی را در جهت طولی و با سایز خوشه $k_i = k = 4$ با $p = 500\ covariates$ انجام دادیم. ما دو مجموعه داده با سایز های مختلف را که $n = 100$ خوشه و $n = 1000$ خوشه آزمایش کردیم. ما در ابتدا حالتی را در نظر می‌گیریم که پاسخ از توزیع گوسی روی covariate ها پیروی می‌کند. تمام covariate ها بصورت مستقل از روی توزیع نرمال استاندارد تولید شده‌اند. از 500 تا از

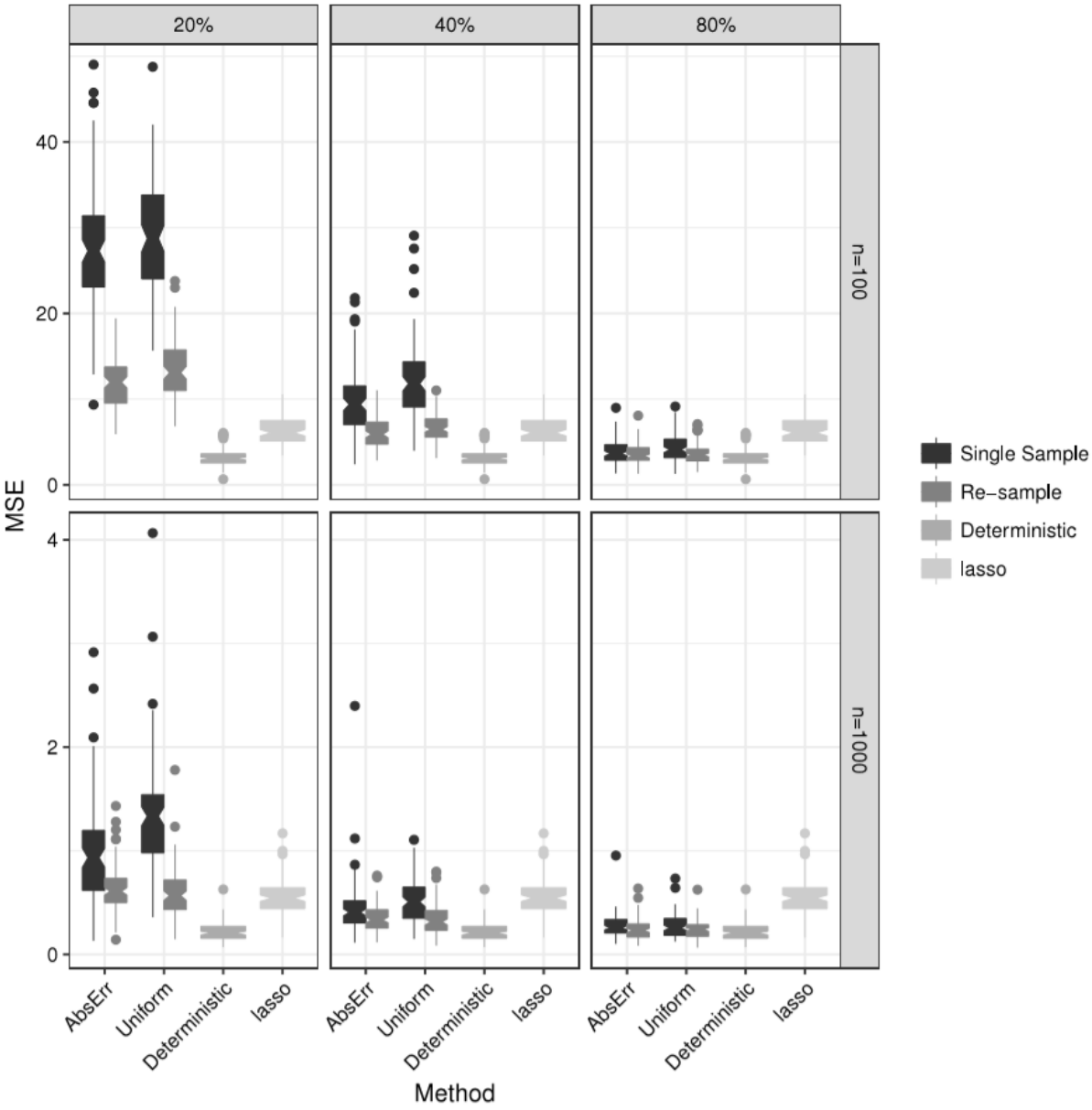
ضرایب، 10 تا از توزیع یکنواخت در بازه 1.5 تا 2.5 نمونه‌برداری شده بودند و مابقی صفر بودند. هر بردار پاسخ با فرمت $k \times 1$ برای هر خوشه از توزیع نرمال چند متغییره با میانگین $\beta_0 + X_i\beta$ و مقدار $\beta_0 = 1$ و ماتریس کوواریانس Σ_x که دارای المان‌های مورب σ_y^2 است و همبتسگی میان جفت خوشه ها با مقدار 0.6 ، تولید شده است.

				MSE		FP		FN		Time		Iterations	
$n = 100$	Re-Sample	20%	lasso	6.39	(1.57)	0.08	(0.01)	0.00	(0.01)	0.06	(0.01)		
			Deterministic	3.20	(1.02)	0.01	(0.00)	0.00	(0.00)	10.69	(1.79)	75.59	(2.86)
			AbsErr	11.97	(3.15)	0.03	(0.01)	0.01	(0.03)	3.53	(0.65)	72.07	(5.66)
			Uniform	13.51	(3.35)	0.04	(0.02)	0.02	(0.04)	2.27	(0.44)	71.89	(5.34)
		40%	AbsErr	6.10	(1.76)	0.02	(0.01)	0.00	(0.00)	5.82	(1.01)	75.76	(4.21)
			Uniform	6.64	(1.69)	0.02	(0.01)	0.00	(0.02)	4.49	(0.80)	75.45	(3.29)
		80%	AbsErr	3.64	(1.20)	0.01	(0.01)	0.00	(0.00)	10.46	(1.54)	75.60	(2.83)
			Uniform	3.66	(1.20)	0.01	(0.00)	0.00	(0.00)	9.15	(1.24)	75.51	(2.81)
	Single Sample	20%	AbsErr	27.74	(6.85)	0.02	(0.01)	0.38	(0.18)	1.97	(0.34)	69.13	(4.03)
			Uniform	28.70	(6.77)	0.02	(0.01)	0.41	(0.19)	1.91	(0.37)	66.28	(4.44)
		40%	AbsErr	9.78	(3.84)	0.01	(0.01)	0.03	(0.07)	3.97	(0.72)	72.90	(4.06)
			Uniform	12.05	(4.38)	0.02	(0.01)	0.04	(0.07)	3.78	(0.60)	69.98	(4.17)
		80%	AbsErr	3.88	(1.34)	0.01	(0.00)	0.00	(0.00)	8.44	(1.28)	75.78	(2.86)
			Uniform	4.39	(1.53)	0.01	(0.01)	0.00	(0.00)	8.43	(1.39)	74.39	(3.13)
$n = 1000$	Re-Sample	20%	lasso	0.56	(0.16)	0.08	(0.02)	0.00	(0.00)	0.80	(0.13)		
			Deterministic	0.22	(0.09)	0.00	(0.00)	0.00	(0.00)	245.49	(24.41)	85.88	(3.02)
			AbsErr	0.63	(0.24)	0.00	(0.01)	0.00	(0.00)	55.54	(10.45)	106.24	(14.47)
			Uniform	0.59	(0.25)	0.00	(0.01)	0.00	(0.00)	39.12	(8.09)	109.93	(18.44)
		40%	AbsErr	0.35	(0.13)	0.00	(0.00)	0.00	(0.00)	100.66	(15.66)	100.99	(11.87)
			Uniform	0.34	(0.15)	0.00	(0.00)	0.00	(0.00)	87.41	(15.65)	107.00	(16.44)
		80%	AbsErr	0.24	(0.10)	0.00	(0.00)	0.00	(0.00)	213.08	(29.12)	91.86	(7.55)
			Uniform	0.25	(0.09)	0.00	(0.00)	0.00	(0.00)	204.20	(29.32)	95.29	(10.56)
	Single Sample	20%	AbsErr	0.97	(0.49)	0.00	(0.00)	0.00	(0.00)	25.69	(3.40)	80.92	(2.35)
			Uniform	1.32	(0.52)	0.00	(0.00)	0.00	(0.00)	25.03	(3.06)	78.96	(2.25)
		40%	AbsErr	0.44	(0.26)	0.00	(0.00)	0.00	(0.00)	66.58	(7.99)	87.71	(3.55)
			Uniform	0.52	(0.22)	0.00	(0.00)	0.00	(0.00)	66.38	(7.82)	86.28	(3.11)
		80%	AbsErr	0.27	(0.11)	0.00	(0.00)	0.00	(0.00)	173.94	(18.31)	85.96	(3.37)
			Uniform	0.27	(0.11)	0.00	(0.00)	0.00	(0.00)	171.53	(17.86)	84.73	(2.93)

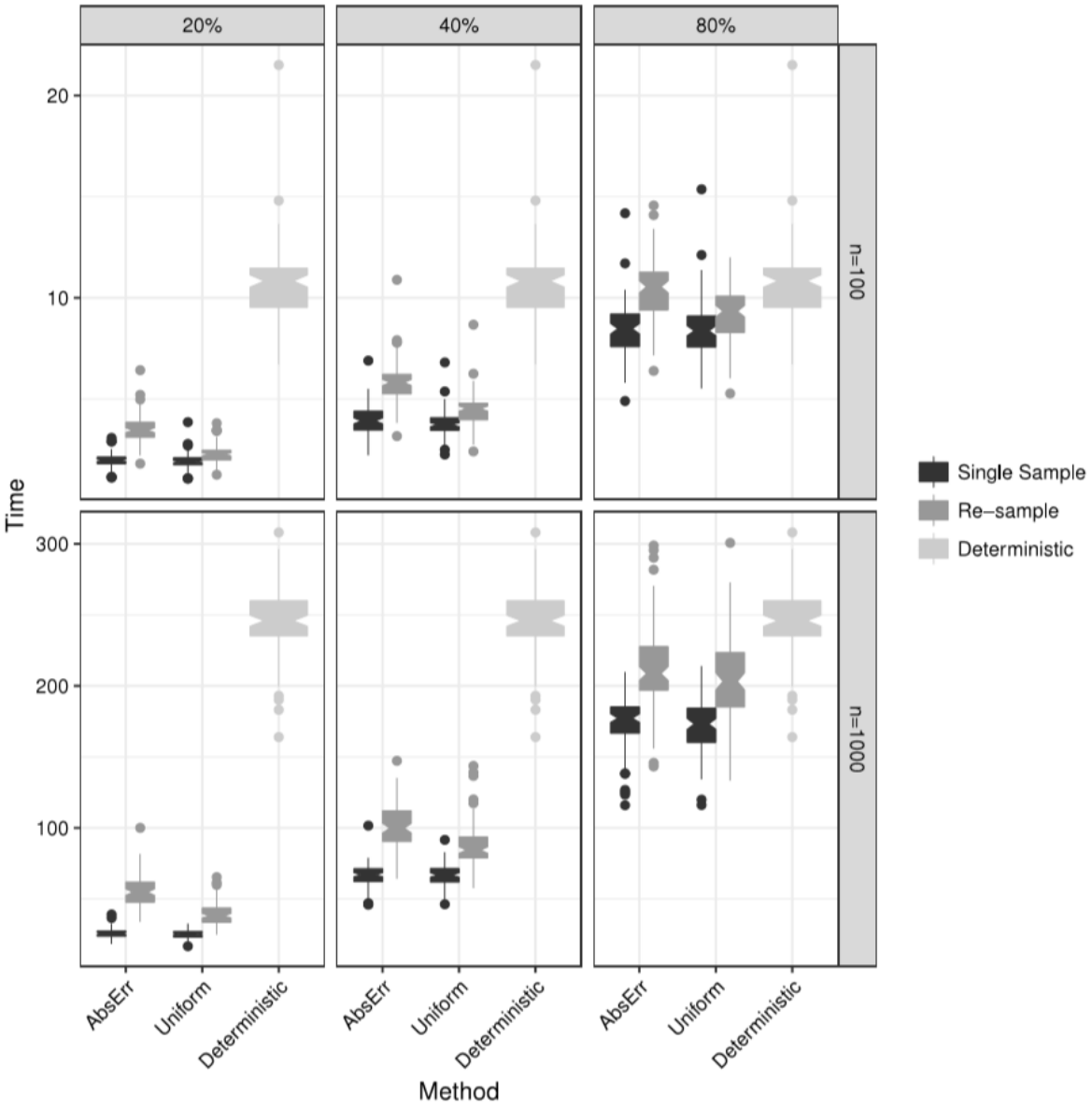
جدول ۱: نتایج شبیه سازی داده‌های گوسی برای تعداد ۱۰۰ و ۱۰۰۰ خوشه

جدول ۱ گزارش نتایج شبیه سازی در محیط و پارامترهایی است که بیان شد. در این شبیه سازی تمام تکنیک های موجود اجرا و تست شده اند و در جدول ۱ نتایج‌شان در کنار یکدیگر آمده است.

در تصویر ۱ نمودار جعبه‌ای اندازه‌گیری های پیشگویانه قابل ملاحظه است. در این نمودار زمان اجرای تمام تکنیک‌ها با مرجعیت روش-های قابل پیش بینی و لاسو رسم گردیده است. پیاده سازی لاسو توسط glmnet انجام شده است و بیشتر محاسباتش را از طریق زبان Fortran انجام می‌دهد. درحالیکه تکنیک های بیان شده، با زبان R پیاده سازی شده‌اند که به مراتب از زبان‌های کامپایلری کند تر هستند و Fortran یک زبان کامپایلری است. بعلاوه لاسو در حالت کلی فرض می‌کند که مشاهدات مستقل هستند، که این فرض موجب ساده تر شدن پیچیدگی محاسبات نسبت به حالتی که فرض شود یک همبستگی میان خوشه‌ها وجود دارد. این تفاوت ها موجب تفاوت زمانی غیر قابل مقایسه‌ای میان روش لاسو و روش‌های بیان شده در این مقاله شده است که این به این علت در تصویر ۲ روش لاسو حذف شده است.



تصویر ۱: نمودار جعبه‌ای اندازه‌گیری‌های پیشگویانه روی ۱۰۰ خوشه



تصویر ۲: نمودار جعبه‌ای زمان اجرای الگوریتم‌ها برحسب ثانیه بر روی ۱۰۰ خوشه

جمع بندی

حجم بسرعت در حال افزایش و پیچیدگی داده نشان می‌دهد که چگونه روش‌های قدیمی پیش بینی دوباره بازآفرینی می‌شوند و بهبود می‌یابند تا بتوانند با چالش های جدید مواجه شوند. با اینکه روش‌های سنتی regularization گام‌های عالی برداشته بودند اما برخی داده ساختارها مانند longitudinal همچنان ایجاد چالش می‌کنند. بعلاوه اعمال مستقیم این روش‌ها غیر قابل قیاس پذیری است، بنابراین یک زیرنمونه برداری ساده می‌تواند اجرا شود تا یک مجموعه داده بزرگ را مدیریت کند، اما انجام این کار بسیاری از اطلاعات را بدون

استفاده می‌گذارد. روش انتخاب مرحله‌ای تصادفی می‌تواند یک زیرنمونه‌برداری جدید در دو جهت را معرفی کند، تا منابع محاسباتی را حفظ کند و خطای پیش بینی را کمینه کند. یک زیرنمونه می‌تواند انتخاب شود و روش مرحله‌ای روی تنها آن زیرنمونه اعمال گردد.

محدودیت‌های کار ارائه شده در این مقاله مسیرهای جذابی را برای مطالعات آینده نمایش می‌دهد. روش انتخاب مرحله‌ای تصادفی می‌تواند از هر نوع تابع پنالتی محدبی استفاده کند اما اینجا تنها از نرم ۱ استفاده شده است. پیاده سازی این روش ها با کمک سایر این توابع می‌تواند اجازه‌ی استفاده از مدل‌های پیچیده تر در آینده را بدهد.

منابع:

[1] M. Kozubovska, "Breaking up big banks," *Research in International Business and Finance*, vol. 41, pp. 198–219, Oct. 2017.

[2] W. N. Robinson and A. Aria, "Sequential fraud detection for prepaid cards using hidden Markov model divergence," *Expert Systems with Applications*, vol. 91, pp. 235–251, Jan. 2018.

[3] L. Seyedhossein and M. R. Hashemi, "Mining information from credit card time series for timelier fraud detection," in *2010 5th International Symposium on Telecommunications*, 2010, pp. 619–624.

[4] G. Vaughan, "Efficient big data model selection with applications to fraud detection," *International Journal of Forecasting*, Jun. 2018.

[5] P. Ma and X. Sun, "Leveraging for big data regression," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7, no. 1, pp. 70–76, Jan. 2015.

[6] I. Ahmed, A. Pariente, and P. Tubert-Bitter, "Class-imbalanced subsampling lasso algorithm for discovering adverse drug reactions," *Statistical Methods in Medical Research*, vol. 27, no. 3, pp. 785–797, Mar. 2018.

[7] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, Feb. 2002.