





دانشکده علوم ریاضی

پایان نامه کارشناسی ارشد آمار

بلوک بندی خود کار پارامترها برای نمونه گیری مونت کارلوی زنجیر مارکوفی

نگارنده: فائزه کجوری

استاد راهنما

دکتر حسین باغیشنی

بهمن ۱۳۹۷

تقدیم به

همه‌ی کسانی که بر این نوشته ورق‌ی خواهند زد

...

سپاس‌گزاری

پاک‌خدایا! چه بزرگ است آن چه می‌بینیم از خلقت تو؛ و چه خرد است، بزرگی آن در کنار قدرت تو؛ و چه با عظمت است آن چه می‌بینیم از ملکوت تو، و چه ناچیز است برابر آن چه برمانهان است از سلطنت تو، و چه فراگیر است نعمت تو در این جهان؛ و چه اندک است در کنار نعمت‌های آن جهان.

خدایا! اگر در پرسش خود دمانم یاراه‌پرسیدن را ندانم، صلاح‌کارم را به من نیاودم را بدان چه رستگاری من در آن است متوجه فرما! که چنین کار از راه‌نمایی‌های تو ناشناخته نیست و از کفایت‌های تونه. و خیفه‌شاکردی خود می‌دانم تا مراتب سپاس و قدردانی خویش را با خلوص و صمیمیت هر چه تمام‌تر به محضر استاد عالی‌قدر، جناب آقای دکتر حسین باغی‌شنی تقدیم دارم که با سعی صدر در تمام مراحل این پایان‌نامه پشتیبان بنده بودند و مرا از راه‌نمایی‌های ارزنده و عالمانه‌شان در عرصه‌ی علم و اخلاق بهره‌مند نمودند. همچنین از اساتید، سرکار خانم دکتر اقبال و جناب آقای دکتر داود شاهسونی که زحمت داورى این پایان‌نامه را متقبل شدند کمال تشکر و سپاس‌گزاری را دارم. در پایان از پدر و مادر عزیزم، به پاس لطف‌های بی‌کرانشان کمال تشکر و قدردانی دارم.

تعهد نامه

این جانب **فائزه کجوری** دانشجوی کارشناسی ارشد رشته **آمار** دانشکده **علوم ریاضی** دانشگاه صنعتی شاهرود، نویسنده پایان نامه با عنوان **بلوک بندی خودکار پارامترها برای نمونه گیری مونت کارلوی زنجیر مارکوفی**، تحت راهنمایی **حسین باغیشنی** متعهد می شوم:

- تحقیقات در این پایان نامه توسط این جانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش های دیگر پژوهش گران، به مرجع مورد استفاده استناد شده است.
- مطالب این پایان نامه، تا کنون توسط خود، یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا 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 به دست آوردن نتایج اصلی پایان نامه تاثیرگذار بوده اند، در مقالات مستخرج از پایان نامه رعایت می گردد.
- در تمام مراحل انجام این پایان نامه، در مواردی که از موجود زنده (یا بافت های آن ها) استفاده شده است، ضوابط و اصول اخلاقی رعایت شده است.
- در تمام مراحل انجام این پایان نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته (یا استفاده شده است)، اصل رازداری و اصول اخلاق انسانی رعایت شده است.

فائزه کجوری

بهمن ۱۳۹۷

مالکیت نتایج و حق نشر

- تمام حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه های رایانه ای، نرم افزارها و تجهیزات ساخته شده) متعلق به دانشگاه صنعتی شاهرود می باشد. این مطلب باید به نحو مقتضی، در تولیدات علمی مربوطه ذکر شود.
- استفاده از اطلاعات و نتایج موجود در این پایان نامه بدون ذکر منبع مجاز نمی باشد.

چکیده

الگوریتم‌های مونت کارلوی زنجیر مارکوفی (MCMC)، مانند نمونه‌گیری گیبز و الگوریتم متروپلیس-هستینگز، روش‌هایی برای تولید نمونه از توزیع‌های پیچیده هستند. اگر در روش MCMC سرعت همگرایی کند شود، زمان اجرای الگوریتم طولانی خواهد شد. یکی از دلایلی که باعث کند شدن همگرایی الگوریتم‌های MCMC می‌شود، جداگانه به‌روز کردن مولفه‌هایی از توزیع است که به هم وابسته هستند. بلوک‌بندی گروهی مولفه‌هایی که به هم وابسته هستند و از سایر مولفه‌ها مستقل، یکی از روش‌های معمول برای گریز از چنین مشکلی است. بلوک‌بندی مناسب مولفه‌های موجود در توزیع، باعث افزایش سرعت همگرایی و کاهش زمان اجرا می‌شود. در این پایان‌نامه یک روش بلوک‌بندی خودکار معرفی می‌شود که از بلوک‌بندی سلیقه‌ای برحذر است و کارایی مناسبی دارد. در نهایت فرآیند بلوک‌بندی خودکار را با استفاده از داده‌های واقعی و مدل‌های شبیه‌سازی شده ارزیابی کرده و نتایج را با استفاده از الگوریتم‌های MCMC استاندارد مقایسه می‌کنیم.

کلمات کلیدی: الگوریتم متروپلیس-هستینگز، نمونه‌گیر گیبز، نمونه‌گیری بلوک‌بندی خودکار، پارامترهای وابسته، کارایی روش‌های نمونه‌گیری MCMC.

پیش‌گفتار

در دیدگاه استنباط بیزی، تمامی استنباط‌ها مبتنی بر توزیع پسین به‌دست می‌آیند. در اغلب موقعیت‌های کاربردی توزیع پسین صورت بسته‌ای ندارد و باید به شیوه‌ای آن را تقریب زد. یک رده از روش‌ها برای تقریب توزیع پسین، روش‌های نمونه‌گیری MCMC، به‌ویژه الگوریتم‌های متروپلیس-هستینگز و نمونه‌گیر گیبز، می‌باشد. این روش‌ها، بنابر زنجیرهای مارکوف طوری طراحی می‌شوند که با افزایش تعداد تکرارهای نمونه‌گیری یا به عبارتی با به‌روزرسانی حالت زنجیر از جایی به بعد، حالت‌های زنجیر حاصل به‌عنوان نمونه‌هایی از توزیع هدف، یعنی توزیع پسین، در نظر گرفته می‌شوند.

در مسایل واقعی، بعد متغیرهای پسین بالاست و انتخاب توزیع پیشنهادی چالش‌برانگیز است. در روش‌های MCMC، معمول است که به جای به‌روزرسانی هم‌زمان همه متغیرها، در هر مرحله فقط یک متغیر (یا یک بلوک از متغیرها) به‌روزرسانی می‌شود که به آن به‌روزرسانی مولفه به مولفه می‌گویند (نیل و رابرتز، ۲۰۰۶). الگوریتم‌های نمونه‌گیر گیبز و متروپلیس-هستینگز درون گیبز از این جمله هستند.

سرعت همگرایی الگوریتم‌های MCMC موجود برای تولید نمونه از توزیع پسین مدل‌های پیچیده، معمولاً کند هستند و معرفی الگوریتم‌های کارا برای سرعت بخشیدن همگرایی زنجیر به توزیع مانای خود، یک زمینه فعال تحقیقاتی محسوب می‌شود. بلوک‌بندی پارامترهای مدل و به‌روز کردن جدای هر بلوک، یکی از راه‌هایی است که برای این منظور پیشنهاد می‌شود (رابرت و کسلا، ۲۰۰۴). در این پایان‌نامه، ابتدا اهمیت بلوک‌بندی را در طراحی الگوریتم‌های MCMC کارا تشریح می‌کنیم. برای سنجش کارایی الگوریتم‌ها سه معیار را معرفی خواهیم کرد. بخش اصلی پایان‌نامه تشریح الگوریتم خودکار برای بلوک‌بندی پارامترهای توزیع هدف است که از کارایی بالایی برخوردار است. این الگوریتم توسط تورک و همکاران (۲۰۱۷) معرفی شده است. با توجه به این مقدمه، ساختار پایان‌نامه به‌صورت زیر است:

- در فصل اول، تعاریف و مفاهیم مقدماتی را بیان کرده و مدل‌های مورد استفاده برای تشخیص عملکرد الگوریتم را به اختصار معرفی می‌کنیم.
- در فصل دوم، به معرفی انواع کارایی الگوریتم‌های MCMC، شامل کارایی محاسباتی، کارایی الگوریتمی و کارایی کل، و نقش بلوک‌بندی پارامترها می‌پردازیم.
- در فصل سوم، الگوریتم بلوک‌بندی خودکار پیشنهادی را بازگو می‌کنیم.
- در فصل چهارم، فرآیند بلوک‌بندی خودکار را بر روی داده‌های شبیه‌سازی شده و داده‌های واقعی، اجرا کرده و نتایج را با الگوریتم‌های MCMC استاندارد مقایسه می‌کنیم.
- در پیوست پایان‌نامه نیز خلاصه‌ای در مورد نحوه استفاده از بسته‌افزار nimble در محیط R برای اجرای روش‌های پیشنهادی پایان‌نامه ارائه می‌کنیم.

لیست مقالات مستخرج از پایان نامه

۱. کجوری، ف.، باغیشتی، ح. (۱۳۹۷). "افزایش کارایی الگوریتم‌های MCMC با بلوک‌بندی بهینه مولفه‌ها"، چهل‌ونهمین کنفرانس ریاضی ایران، دانشگاه علم و صنعت.
۲. کجوری، ف.، باغیشتی، ح. (۱۳۹۷). "افزایش کارایی الگوریتم‌های MCMC با بلوک‌بندی پارامترهای مدل"، اولین سمینار دانشجویی آمار ایران، دانشگاه تربیت مدرس.

فهرست مطالب

ق فهرست تصاویر

ش فهرست جداول

۱	تعاریف و مفاهیم اولیه	۱
۱	۱.۱ مقدمه	۱
۲	۲.۱ رهیافت‌های استنباط آماری	۲
۳	۱.۲.۱ رهیافت کلاسیک	۳
۴	۲.۲.۱ رهیافت بیزی	۴
۶	۳.۱ الگوریتم‌های MCMC	۶
۷	۱.۳.۱ الگوریتم متروپلیس-هستینگز	۷
۱۱	۲.۳.۱ معیارهای همگرایی زنجیرهای MCMC	۱۱
۱۵	۴.۱ مدل‌های خطی تعمیم‌یافته	۱۵
۱۸	۵.۱ مدل‌های آمیخته خطی تعمیم‌یافته	۱۸
۱۸	۶.۱ مدل‌های فضای حالت	۱۸
۲۰	۷.۱ آمار فضایی	۲۰
۲۱	۸.۱ ساختار همبستگی فضایی	۲۱
۲۲	۱.۸.۱ تغییرنگار	۲۲
۲۳	۲.۸.۱ هم‌تغییرنگار	۲۳
۲۴	۹.۱ مدل‌های تغییرنگار پارامتری معتبر	۲۴
۲۷	۲ کارایی الگوریتم‌های MCMC و نقش بلوک‌بندی پارامترها	۲۷
۲۷	۱.۲ مقدمه	۲۷
۲۹	۲.۲ نمادگذاری و تعریف الگوریتم‌های MCMC	۲۹
۳۱	۳.۲ کارایی الگوریتمی	۳۱
۳۳	۱.۳.۲ کارایی از دست رفته نمونه‌گیری بلوکی	۳۳

۳۴	کارایی از دست رفته نمونه‌گیری مولفه به مولفه	۲.۳.۲
۳۶	کارایی محاسباتی	۴.۲
۳۷	محاسبات روش نمونه‌گیری مولفه به مولفه	۱.۴.۲
۳۷	محاسبات روش نمونه‌گیری بلوکی	۲.۴.۲
۳۸	مقایسه زمانی	۳.۴.۲
۳۹	کارایی کلی	۵.۲
۴۱	۳ بلوک‌بندی خودکار	
۴۱	مقدمه	۱.۳
۴۲	خوشه‌بندی	۲.۳
۴۳	انواع روش‌های خوشه‌بندی	۱.۲.۳
۴۴	روش‌های خوشه‌بندی سلسله‌مراتبی	۲.۲.۳
۴۹	الگوریتم‌های MCMC سازوار	۳.۳
۵۱	الگوریتم بلوک‌بندی خودکار	۴.۳
۵۵	۴ ارزیابی عملکرد بلوک‌بندی خودکار	
۵۵	مقدمه	۱.۴
۵۶	مثال‌های کاربردی	۲.۴
۵۶	مدل اثرات تصادفی	۱.۲.۴
۵۷	مدل اتورگرسیون	۲.۲.۴
۵۸	مدل فضای حالت گاوسی	۳.۲.۴
۵۹	مدل فضایی	۴.۲.۴
۶۰	مدل آمیخته خطی تعمیم یافته	۵.۲.۴
۶۱	تحلیل مثال‌های کاربردی	۳.۴
۶۳	مدل اثرات تصادفی	۱.۳.۴
۶۳	مدل اتورگرسیون	۲.۳.۴
۶۴	مدل فضای حالت گاوسی	۳.۳.۴
۶۵	مدل فضایی	۴.۳.۴
۶۶	مدل آمیخته خطی تعمیم یافته	۵.۳.۴
۶۷	کارایی حاصل از بلوک‌بندی خودکار	۴.۴
۶۸	تعداد دفعات مراجعه بیماران شهر اهواز	۵.۴
۷۳	نتیجه‌گیری و آینده تحقیق	۶.۴
۷۵	آ راهنمای استفاده از NIMBLE	
۷۵	مقدمه	۱.آ

۷۶		۲.آ	مراحل ساخت مدل در محیط NIMBLE
----	-------	-----	-------------------------------

۸۹		مراجع
----	--	-------

فهرست تصاویر

۲۳	۱.۱	نمودار تغییرنگار و هم تغییرنگار
	۱.۲	کارایی الگوریتمی MCMC برای مقادیر متفاوتی از بعد مدل (d) و همبستگی درون گروهی ρ . مقادیر $-\log(1-\rho)$ بر روی محور افقی رسم شده‌اند. ساختار ماتریس کوواریانس مدل شامل عناصر خارج قطر اصلی ثابت برابر ρ (سمت چپ) و همبستگی‌های کاهشی-نمایی $\rho^{ i-j }$ (سمت راست) است.
۳۵	۲.۲	زمان اجرای MCMC برای الگوریتم‌های تک بلوکی و مولفه به مولفه، برای ساختارهای مدل نرمال یک متغیره و گامای یک متغیره و نرمال چندمتغیره.
۳۹	۱.۳	نحوه اجرا دو روش خوشه‌بندی سلسله‌مراتبی پایین به بالا و بالا به پایین.
۴۵	۲.۳	شباهت بین دو خوشه در روش پیوند تکی برابر است با کمترین فاصله بین داده‌های دو خوشه.
۴۶	۳.۳	شباهت بین دو خوشه در روش پیوند کامل برابر است با بیشترین فاصله بین داده‌های دو خوشه.
۴۷	۴.۳	شباهت بین دو خوشه در روش پیوند میانگین برابر است با میانگین فاصله بین تمام اعضای یکی با تمام اعضای دیگری.
۴۸	۱.۴	کارایی الگوریتم‌های MCMC برای مجموعه مثال‌ها
۶۷	۲.۴	محل سکونت بیماران مراجعه‌کننده به بیمارستان آریا اهواز. مربع بزرگ قرمز
۶۸		محل بیمارستان آریا را نشان می‌دهد.
۷۲	۳.۴	اثر فضایی بیماران مراجعه‌کننده به بیمارستان آریا اهواز

فهرست جداول

۵۱	الگوریتم بلوک‌بندی خودکار	۱.۳
۵۶	نسبت زنده ماندن نوزادان دو گروه از ۱۶ خوک	۱.۴
۵۷	مجموعه داده ice	۲.۴
۶۳	میعارهای کارایی برای مثال اثرات تصادفی در مجموعه داده litters	۳.۴
۶۴	میعارهای کارایی برای مثال اتورگرسیو در مجموعه داده سرطان پستان ایسلند	۴.۴
۶۵	میعارهای کارایی برای مثال فضای حالت شبیه‌سازی شده	۵.۴
۶۶	میعارهای کارایی برای مثال مدل فضایی برای گونه‌های گوش ماهی	۶.۴
	میعارهای کارایی برای مثال مدل آمیخته خطی تعمیم‌یافته برای داده‌های مراجعه‌کنندگان	۷.۴
۶۷	سالخورده	
۶۹	متغیرهای مختلف مجموعه داده بیماران مراجعه‌کننده به بیمارستان آریا	۸.۴
۷۰	میعارهای کارایی برای داده‌های تعداد مراجعه بیماران شهر اهواز	۹.۴
	نتایج استنباط پسینی حاصل از اجرای الگوریتم نمونه‌گیری خودکار برای داده‌های	۱۰.۴
۷۱	تعداد مراجعه بیماران شهر اهواز	
۷۶	مجموعه داده‌های Pump	۱.آ

فصل ۱

تعاریف و مفاهیم اولیه

۱.۱ مقدمه

استنباط بیزی یک چارچوب منسجم برای تحلیل مدل‌های آماری ایجاد می‌کند؛ در واقع، در این چارچوب همه استنباط‌ها مبتنی بر توزیع پسین مدل پایه‌ریزی می‌شود. با توجه به پیشرفت‌های امروزی در جمع‌آوری داده‌ها و در مدل‌های واقعی برای تحلیل آن‌ها، توزیع‌های پسین معمولاً صورت بسته ندارند و آن‌قدر پیچیده هستند که نتوان روش‌های استنباطی دقیق را به‌دست آورد. در این موارد، روش‌های تقریبی برای تقریب توزیع پسین به‌کار می‌روند که عموماً نسخه‌ای از روش‌های نمونه‌گیری مونت کارلویی هستند.

در مدل‌هایی که پیچیدگی مدل یا بعد پارامترهای مدل بالا است، روش استاندارد برای تولید نمونه از توزیع پسین الگوریتم‌های مونت کارلوی زنجیر مارکوفی^۱ (MCMC) است که به‌طور غیرمستقیم از توزیع پسین نمونه تولید می‌کنند (متروپلیس و همکاران، ۱۹۵۳؛ هستینگز، ۱۹۷۰). مکانیسم الگوریتم‌های MCMC مبتنی بر تولید نمونه وابسته از یک زنجیر مارکوف است که توزیع مانای آن توزیع پسین مورد نظر است. در این صورت اگر حجم نمونه مونت کارلویی تولیدشده به اندازه کافی بزرگ باشد، از جایی به بعد مقادیر زنجیر به‌عنوان نمونه‌هایی از توزیع مطلوب، یعنی توزیع پسین، در نظر گرفته می‌شوند.

سرعت همگرایی الگوریتم‌های MCMC موجود برای تولید نمونه از توزیع پسین مدل‌های

^۱Markov chain Monte Carlo

پیچیده، معمولاً کند هستند و معرفی الگوریتم‌های کارا برای سرعت بخشیدن به همگرایی زنجیر به توزیع مانای خود، یک زمینه فعال تحقیقاتی محسوب می‌شود. بلوک‌بندی پارامترهای مدل و به‌روز کردن جدای هر بلوک، یکی از راه‌هایی است که برای این منظور پیشنهاد می‌شود (رابرت و کسلا، ۲۰۰۴). این رهیافت نه تنها بر همگرایی زنجیر، بلکه بر روی رفتار آمیختگی^۲ و کارایی زنجیر نیز تاثیر خواهد داشت (رابرتز و ساهو، ۱۹۹۷). منگرسن و تویییدی (۱۹۹۶)، رابرتز و تویییدی (۱۹۹۶)، رابرتز و همکاران (۱۹۹۷) عملکرد الگوریتم‌های نمونه‌گیری بلوکی را با نمونه‌گیری تک متغیره مقایسه کرده‌اند. همچنین رابرتز و ساهو (۱۹۹۷) و رابرتز و روزنتال (۲۰۰۱) قسمتی از مباحث نظری مربوط به همگرایی بلوکی الگوریتم‌های MCMC را مورد بررسی قرار داده‌اند.

در استفاده از الگوریتم‌های MCMC بلوکی، این که کدام پارامترها در یک بلوک قرار بگیرند و با هم به‌روز شوند اهمیت ویژه‌ای دارد. به‌روز کردن جداگانه مولفه‌هایی که به هم وابسته هستند، اثر منفی بر کارایی الگوریتم دارد. به عبارت دیگر، بلوک‌بندی گروهی مولفه‌هایی که به هم وابسته هستند و از سایر مولفه‌ها مستقل، از روش‌های معمول برای افزایش کارایی محاسباتی الگوریتم‌های MCMC است.

با این مقدمه، در این فصل به معرفی الگوریتم‌های MCMC استاندارد می‌پردازیم و در آخر، به‌طور خیلی خلاصه، مدل‌هایی که برای تشریح عملکرد روش اصلی این پایان‌نامه در فصل چهارم مورد استفاده قرار می‌گیرند را معرفی می‌کنیم.

۲.۱ رهیافت‌های استنباط آماری

مجموعه روش‌های آماری که برای به‌دست آوردن نتایج در مورد ویژگی‌های مورد نظر داده‌ها، به کار می‌روند را استنباط آماری می‌گویند. مهم‌ترین هدف استنباط آماری کسب دانش، بر اساس اطلاعات داده‌شده در مورد موضوع یا مسأله مورد بررسی است. به‌طور کلی، دو رهیافت جامع برای انجام استنباط آماری وجود دارند:

(الف) رهیافت کلاسیک

(ب) رهیافت بیزی

که در ادامه به اختصار هر کدام را معرفی می‌کنیم.

^۲Mixing Properties

۱.۲.۱ رهیافت کلاسیک

در آمار کلاسیک پارامترهای مدل مقداری ثابت ولی معمولاً نامعلوم در نظر گرفته می‌شوند و هدف، ارزیابی رفتار روش‌های استنباطی بر اساس اصل تکرارپذیری^۳ است. یکی از روش‌های معمول برای برآورد پارامترهای نامعلوم مدل‌های آماری روش برآورد درست‌نمایی ماکسیمم^۴ (MLE) نام دارد که نخستین بار توسط گاوس (۱۸۲۱) به کار گرفته شده است.

تعریف ۱.۱. (تابع درست‌نمایی). برای هر مقدار مشاهده شده $X = x$ ، تابع درست‌نمایی x را تابع چگالی احتمال توام (تابع احتمال توام) x ، یعنی $f(x|\theta)$ ، تعریف می‌کنیم که به صورت تابعی از پارامتر θ در نظر گرفته می‌شود و آن را با نماد $L(\theta)$ نمایش می‌دهیم. بنابراین

$$L(\theta) = f(x|\theta).$$

تابع درست‌نمایی $L(\theta)$ لزوماً یک تابع چگالی احتمال (تابع احتمال) نیست، بلکه متناسب با تابع چگالی احتمال است. همچنین این تابع لزوماً نسبت به θ مشتق‌پذیر نیست. اگر $X_1, \dots, X_n$ یک نمونه تصادفی n تایی از توزیعی با خانواده چگالی‌های پارامتری

$$\{f(x|\theta) : \theta \in \Theta\}$$

و X_i ها مستقل از هم باشند، آن‌گاه

$$L(\theta) = \prod_{i=1}^n f(x_i|\theta)$$

در غیر این صورت تابع درست‌نمایی به صورت زیر است:

$$L(\theta) = f(x_1, \dots, x_n|\theta).$$

تعریف ۲.۱. (برآوردگر درست‌نمایی ماکسیمم). اگر $\delta(X)$ برآوردگری برای θ باشد، به طوری که

$$i) \quad P(\delta(x) \in \Theta) = 1 \quad \forall \theta \in \Theta \subseteq \mathbb{R}^p$$

$$ii) \quad L(\delta(X)) \geq L(\theta) \quad \forall \theta \in \Theta \subseteq \mathbb{R}^p$$

آن‌گاه $\delta(x)$ برآوردگر درست‌نمایی ماکسیمم θ تعریف می‌شود. معمولاً برآوردگر درست‌نمایی ماکسیمم θ را با $\hat{\theta}$ نمایش می‌دهند و لزوماً منحصر به فرد نیست. همچنین بر اساس تعریف $\hat{\theta}$ (پارسیان، ۱۳۹۱)

$$L(\hat{\theta}) = \sup_{\theta \in \Theta} L(\theta).$$

محاسبه تابع درست‌نمایی در رده مدل‌های پیچیده، معمولاً حل انتگرال‌های با بعد بالا را شامل می‌شود که دارای محاسباتی زمان‌بر و دشوار است. بنابراین این روش همیشه مورد استفاده قرار نمی‌گیرد. رهیافت بیزی که استنباط در آن مبتنی بر توزیع پسین است، می‌تواند یک جانشین مناسب باشد.

^۳Repeated principle

^۴Maximum likelihood estimation

۲.۲.۱ رهیافت بیزی

استنباط بیزی یک روش احتمالی است که مبنای آمار بیزی است و پایه‌های زیربنایی آن برای اولین بار توسط توماس بیز در سال ۱۷۶۳ ارائه شد. تفاوت اصلی آمار بیزی با آمار کلاسیک این است که در دیدگاه بیزی، پارامترهای نامعلوم نیز به عنوان کمیت‌های تصادفی در نظر گرفته می‌شوند و توزیع آماری مناسبی به آن‌ها تخصیص داده می‌شود که به آن توزیع پیشین^۵ می‌گویند و آن را با نماد $\pi(\theta)$ نمایش می‌دهند. در این حالت استنباط آماری بر مبنای قضیه بیز^۶ انجام می‌شود. در واقع، با ترکیب اطلاعات توزیع پیشین و اطلاعات نمونه (بر حسب تابع درست‌نمایی)، توزیع جدیدی حاصل می‌شود که به آن توزیع پسین^۷ می‌گویند. به بیان دیگر، ادغام اطلاعات پیشین و اطلاعات حاصل از مشاهدات موجب به‌روز شدن اطلاعات محقق از پارامتر شده و در نتیجه توزیع پسین حاصل می‌شود که استنباط بر مبنای آن انجام می‌شود. آمار بیزی یک دیدگاه از آمار است که علاوه بر اطلاعات موجود در نمونه‌ها، جوانب دیگری مانند اطلاعات پیشین را در نظر می‌گیرد. فرض کنید $\theta = (\theta_1, \dots, \theta_d)'$ بردار پارامترهای مورد نظر باشد که می‌خواهیم در مورد آن استنباط کنیم. در آمار کلاسیک تنها بر اساس نمونه‌ها درباره پارامتر θ اطلاعاتی به دست می‌آید. اما در روش بیزی با تلفیق نمونه‌ها و اطلاعات پیشین $\pi(\theta)$ ، توزیع پسین ساخته می‌شود و بر مبنای توزیع پسین به استنباط بیزی می‌پردازیم. تابع چگالی توزیع پسین به شکل زیر ساخته می‌شود:

$$\pi(\theta|x) = \frac{f(x, \theta)}{m(x)}$$

که در آن $f(x, \theta)$ چگالی توام θ و بردار مشاهدات x و برابر است با

$$f(x, \theta) = f(x|\theta)\pi(\theta)$$

و

$$m(x) = \int_{\Theta} f(x|\theta)\pi(\theta)d\theta$$

تابع چگالی کناری x و Θ نشان‌دهنده فضای پارامتر است.

در دیدگاه بیزی یکی از مولفه‌های مهم، تعیین توزیع پیشین $\pi(\theta)$ است. از این رو برخی از روش‌های تعیین آن را مرور می‌کنیم.

انواع توزیع‌های پیشین

انتخاب توزیع پیشین به دلیل آن که توزیع پسین را تحت تاثیر قرار می‌دهد، از اهمیت زیادی برخوردار است. در این قسمت به معرفی برخی از انواع توزیع‌های پیشین متداول می‌پردازیم.

^۵Prior distribution

^۶Bayes theorem

^۷Posterior distribution

۱. توزیع پیشین ناآگاهی بخش:

در اغلب مثال‌های کاربردی که هیچ‌گونه اطلاعی از پارامتر در دسترس نیست، توزیع پیشین را طوری در نظر می‌گیرند که کمترین اطلاع از پارامتر را وارد توزیع پسین کند. چنین توزیع‌هایی را توزیع ناآگاهی بخش^۸ می‌گویند (نتزوفراس، ۲۰۰۹). یکی از انواع توزیع‌های پیشین ناآگاهی بخش به صورت

$$\pi(\theta) \propto c$$

روی فضای پارامتر تعریف می‌شود که c یک مقدار ثابت است و در این صورت به آن توزیع یکنواخت یا تخت^۹ می‌گویند. در این حالت، توزیع پسین با تابع درستنمایی متناسب می‌شود و در نتیجه مد توزیع پسین با برآورد ماکسیمم درستنمایی در حالت کلاسیک معادل خواهد شد.

تذکر ۱. اگر توزیع پیشین طوری تعریف شود که

$$\int_{\Theta} \pi(\theta) = \infty$$

به آن، توزیع پیشین ناسره^{۱۰} می‌گویند. توزیع‌های ناسره در صورتی قابل استفاده هستند که منجر به یک توزیع پسین سره^{۱۱} شوند.

جزئیات بیشتر در مورد توزیع‌های پیشین ناآگاهی بخش در یانگ و برگر (۱۹۹۶) موجود است.

۲. توزیع پیشین مزدوج^{۱۲}:

پیشین مزدوج به توزیعی گفته می‌شود که در آن خانواده توزیع پیشین و پسین با هم یکی باشند (رابرتز، ۲۰۰۸).

تعریف ۳.۱. هرگاه F خانواده‌ای از توزیع‌ها با تابع چگالی احتمال $f(x|\theta)$ باشد، خانواده Π از توزیع‌های پیشین را خانواده مزدوج برای f گویند، هرگاه برای هر $f \in F$ و هر $\pi(\theta) \in \Pi$ آن‌گاه $\pi(\theta|x) \in \Pi$.

از انواع توزیع‌های پیشین دیگر می‌توان به توزیع‌های پیشین جفریز^{۱۳} (جفریز، ۱۹۶۱)، مرجع^{۱۴} (هیل، ۱۹۷۵)، و جورشدنی^{۱۵} (موکرجی و قوش، ۱۹۹۷) اشاره کرد.

^۸Noninformative prior

^۹Flat

^{۱۰}Improper prior distribution

^{۱۱}Proper prior distribution

^{۱۲}Conjugate prior distribution

^{۱۳}Jeffreys prior

^{۱۴}Reference prior

^{۱۵}Matching prior

همان‌طور که گفته شد، در دیدگاه بیزی همه استنباط‌ها بر اساس توزیع پسین به‌دست می‌آیند. در مواردی که صورت بسته توزیع پسین در دسترس نیست یا بعد پارامترهایش بالا باشد، برای محاسبه کمیت‌های مورد علاقه مبتنی بر آن، از روش‌های تقریبی استفاده می‌شود. برای تقریب توزیع‌های پسین از دو گروه از روش‌ها می‌توان استفاده کرد؛ روش‌های عددی و روش‌های مبتنی بر نمونه‌گیری^{۱۶}. از جمله روش‌های تقریب عددی، تقریب نیوتن-رافسون^{۱۷}، روش‌های تربیع‌بندی گاوسی^{۱۸} (آبرامویتز و استگن، ۱۹۷۲) و تقریب لاپلاس^{۱۹} (تیرنی و کدین، ۱۹۸۶) را می‌توان نام برد. روش‌های مبتنی بر نمونه‌گیری همان روش‌های شبیه‌سازی مونت کارلویی هستند که می‌توان روش‌های نمونه‌گیری رد و پذیرش^{۲۰} (کسلا و برگر، ۲۰۰۲)، نمونه‌گیری نقاط مهم^{۲۱} (هستبرگ، ۱۹۸۷) و نمونه‌گیری مونت کارلوی زنجیر مارکوفی^{۲۲} (MCMC) (هستینگز، ۱۹۷۰؛ رابرت و کسلا، ۲۰۰۴) را نام برد.

تقریب‌های حاصل از روش‌های عددی در صورتی که بعد توزیع پسین کمتر از ۱۰ باشد، خطای قابل قبولی دارند اما در مواردی که این بعد بالا باشد استفاده از این نوع روش‌ها جایز نیست. در این حالت از روش‌های مبتنی بر نمونه‌گیری برای تقریب توزیع پسین استفاده می‌شود که ممکن است سرعت محاسباتی آن‌ها خیلی کندتر از روش‌های عددی باشد. در روش‌های نمونه‌گیری مونت کارلویی، خطای تقریب، تابعی از بعد توزیع پسین نیست و به ازای هر بعدی، ثابت و از مرتبه $O(n^{-\frac{1}{2}})$ است، که n حجم نمونه مونت کارلویی است. در این روش‌ها، قانون قوی اعداد بزرگ^{۲۳} برای نمونه‌های تولیدشده مستقل (کسلا و برگر، ۲۰۰۲) و قضیه ارگودیک^{۲۴} برای نمونه‌های تولیدشده وابسته (زنجیر مارکوفی) (بیرخوف، ۱۹۴۲) همگرایی برآوردگرها به کمیت مورد نظر از توزیع پسین را تضمین می‌کنند.

توسعه و پیشرفت استنباط بیزی در سال‌های اخیر مدیون انقلاب کامپیوترها و توسعه قدرت پردازنده‌های کامپیوتری است که منجر به استفاده شایع از الگوریتم‌های MCMC شد. در بخش بعدی مهم‌ترین الگوریتم‌های MCMC را معرفی می‌کنیم.

۳.۱ الگوریتم‌های MCMC

الگوریتم‌های MCMC توسط متروپلیس و همکاران (۱۹۵۳) و هستینگز (۱۹۷۰) معرفی شدند که وظیفه آن‌ها تولید غیرمستقیم نمونه از یک توزیع احتمالی مفروض است. مکانسیم

^{۱۶}Sampling base methods

^{۱۷}Newton-Raphson approximation

^{۱۸}Gauss quadrature method

^{۱۹}Laplace approximation

^{۲۰}Reject-accept sampling

^{۲۱}Important sampling

^{۲۲}Markov chain Monte Carlo

^{۲۳}Strong law of large number

^{۲۴}Ergodic theorem

الگوریتم‌های MCMC مبتنی بر تولید نمونه وابسته از یک زنجیر مارکوف است که توزیع مانای^{۲۵} آن همان توزیع پسین مورد نظر است. در این صورت اگر حجم نمونه مونت کارلویی تولیدشده به اندازه کافی بزرگ باشد، از جایی به بعد حالت زنجیر حاصل به‌عنوان نمونه‌های توزیع هدف، یعنی توزیع پسین، در نظر گرفته می‌شود. از جمله معروف‌ترین الگوریتم‌های MCMC، دو الگوریتم متروپلیس-هستینگز^{۲۶} (MH) و نمونه‌گیری گیبز^{۲۷} (GS) هستند که آن‌ها را معرفی می‌کنیم.

۱.۳.۱ الگوریتم متروپلیس-هستینگز

این الگوریتم متغیرهای همبسته‌ای از یک زنجیر مارکوف را تولید می‌کند، به‌طوری که با ارزیابی یک قاعده احتمالی، مشاهداتی که توسط یک توزیع پیشنهادی^{۲۸} به‌عنوان نمونه‌های ممکن از توزیع پسین پیشنهاد می‌شوند را با احتمال معین می‌پذیرد. بر این اساس برای استفاده از این الگوریتم، به یک توزیع پیشنهادی نیاز است که انتخاب آن بر دقت و سرعت همگرایی الگوریتم مؤثر است.

فرض کنید $f(x)$ تابع چگالی احتمال توزیع هدف باشد که تولید نمونه به‌صورت مستقیم از آن سخت و امکان‌ناپذیر است. برای تولید نمونه از تابع چگالی احتمال هدف $f(x)$ ابتدا یک تابع چگالی پیشنهادی $q(y|x)$ که تکیه‌گاه آن با $f(x)$ یکسان و تولید نمونه از آن آسان است را انتخاب می‌کنیم. در نتیجه می‌توان الگوریتم MH را به‌صورت زیر بیان کرد:

۱. تولید y از تابع چگالی پیشنهادی $q(y|x)$.

۲. محاسبه احتمال پذیرش $\alpha(x^{(t)}, y)$ که در آن $\alpha(x^{(t)}, y) = \min \left\{ 1, \frac{f(y)}{f(x^{(t)})} \frac{q(x^{(t)}|y)}{q(y|x^{(t)})} \right\}$.

۳. قرار دادن $x^{(t+1)} = y$ با احتمال $\alpha(x^{(t)}, y)$ و قرار دادن $x^{(t+1)} = x$ با احتمال $1 - \alpha(x^{(t)}, y)$.

کارایی الگوریتم MH به تولید نمونه از تابع چگالی پیشنهادی وابسته است. از این رو نمونه خوب زمانی حاصل می‌شود که تابع چگالی پیشنهادی یک تقریب مناسب برای چگالی هدف یعنی $f(x)$ باشد. اگر توزیع پیشنهادی، مقادیری نزدیک به x را پیشنهاد کند، تقریباً همه پیشنهادها پذیرفته می‌شوند اما زنجیر خیلی کند حرکت می‌کند و زمان زیادی طول می‌کشد تا الگوریتم همگرا شود. همچنین اگر توزیع پیشنهادی مقادیر دور از x را پیشنهاد کند، پیشنهادها تقریباً همیشه رد می‌شوند و باز هم زنجیر به کندی همگرا می‌شود. توزیع پیشنهادی را می‌توان طوری تنظیم کرد که الگوریتم از نرخ پذیرش مناسبی برخوردار باشد (رابرتز و همکاران، ۱۹۹۷).

^{۲۵}Stationary distribution

^{۲۶}Metropolis-Hastings algorithm

^{۲۷}Gibbs sampling algorithm

^{۲۸}Proposal distribution

الگوریتم MH مستقل^{۲۹} (IMH) و الگوریتم MH قدم زدن تصادفی^{۳۰} (RWMH) دو نسخه ویژه از الگوریتم MH هستند که در ادامه معرفی می‌شوند.

الگوریتم متروپلیس-هستینگز مستقل

در حالتی که مشاهده پیشنهادی به‌طور مستقل از حالت فعلی زنجیر انتخاب شود، الگوریتم حاصل با این انتخاب را الگوریتم متروپلیس-هستینگز مستقل می‌نامند. در این حالت $q(y|x) = g(y)$ الگوریتم IMH به‌صورت زیر است:

۱. تولید y از تابع چگالی پیشنهادی $g(y)$.

۲. محاسبه احتمال چگالی پذیرش $\alpha(x^{(t)}, y)$ به‌صورت

$$\alpha(x^{(t)}, y) = \min \left[1, \frac{f(y)}{f(x^{(t)})} \frac{g(x^{(t)})}{g(y)} \right].$$

۳. قرار دادن $x^{(t+1)} = y$ با احتمال $\alpha(x^{(t)}, y)$ و قرار دادن $x^{(t+1)} = x^{(t)}$ با احتمال $1 - \alpha(x^{(t)}, y)$.

این نمونه‌گیری زمانی که چگالی پیشنهادی تقریباً با چگالی هدف همسان باشد، به‌خوبی عمل می‌کند، در غیر این‌صورت عملکرد خوبی نخواهد داشت. برای اطلاعات بیشتر از ویژگی‌های این الگوریتم می‌توانید به رابرت و کسلا (۲۰۱۰) مراجعه کنید.

الگوریتم متروپلیس-هستینگز قدم زدن تصادفی

در این الگوریتم برای تولید مقدار بعدی زنجیر، از نمونه تولیدشده قبلی استفاده می‌شود، که در آن یک کاوش در همسایگی^{۳۱} مقدار جاری زنجیر صورت می‌گیرد. اجرای این ایده با شبیه‌سازی $y = x^{(t)} + \epsilon_t$ کامل می‌شود، که در آن ϵ_t یک خطای تصادفی مستقل با تابع چگالی $g(\cdot)$ است. در این حالت توزیع پیشنهادی به‌صورت زیر است:

$$q(x|y) = g(x - y) = g(y - x).$$

اگر چگالی g حول صفر متقارن باشد (مثلاً در توزیع نرمال یا نرمال چندمتغیره با میانگین صفر، یا توزیع یکنواخت استاندارد) زنجیر مارکوف متناظر با آن، یک فرآیند قدم زدن تصادفی است. ساختار الگوریتم به شکل زیر است:

۱. تولید y از تابع چگالی پیشنهادی $g(y - x^{(t)})$.

^{۲۹}Independent Metropolis-Hastings algorithm

^{۳۰}Random walk Metropolis-Hastings algorithm

^{۳۱}Local exploration

۲. محاسبه احتمال پذیرش $\alpha(x^{(t)}, y) = \min \left[1, \frac{f(y)}{f(x^{(t)})} \right]$ که

۳. قرار دادن $x^{(t+1)} = y$ با احتمال $\alpha(x^{(t)}, y)$ و قرار دادن $x^{(t+1)} = x^{(t)}$ با احتمال $1 - \alpha(x^{(t)}, y)$.

در این الگوریتم احتمال پذیرش به ضابطه g وابسته نیست. یعنی برای یک جفت مفروض (x, y) ، احتمال پذیرش y در حالتی که از توزیع نرمال یا کوشی تولید شود یکسان خواهد بود. انتخاب g ‌های مختلف منجر به دامنه‌های مختلف مقادیر برای y ‌ها و در نتیجه نرخ‌های پذیرش متفاوت خواهد شد. بنابراین نمی‌توان گفت انتخاب g بر رفتار الگوریتم تاثیری ندارد.

نمونه‌گیر گیبز

نام نمونه‌گیر گیبز از مقاله معروف گمان و گمان (۱۹۸۴) گرفته شده است. در واقع نمونه‌گیری گیبز یک نسخه ویژه از الگوریتم متروپلیس-هستینگز است. استفاده شایع از این الگوریتم با کار گلفاند و اسمیت (۱۹۹۰) شروع شد که از این الگوریتم برای حل مسائلی در استنباط بیزی که قبلاً بدون حل مانده بودند، استفاده کردند. برای اطلاعات بیشتر می‌توان به رابرت و کسلا (۲۰۰۴) مراجعه کرد.

الگوریتم‌های نمونه‌گیر گیبز مسائل پیچیده (با بعد بالا) را به دنباله‌ای از مسائل ساده‌تر تقسیم می‌کنند که این مسائل ممکن است به وسیله الگوریتم MH قابل حل نباشند.

نمونه‌گیر گیبز دو مرحله‌ای

نمونه‌گیری دو مرحله‌ای یک زنجیر مارکوف از توزیع توام دو متغیره به روش زیر به دست می‌آید. اگر دو متغیر تصادفی x و y دارای تابع چگالی احتمال توام $f(x, y)$ با چگالی‌های شرطی $f_{y|x}$ و $f_{x|y}$ باشند، نمونه‌گیر گیبز دو مرحله‌ای یک زنجیر مارکوف $(x^{(t)}, y^{(t)})$ طبق مراحل زیر تولید می‌کند:

الگوریتم ۱. (نمونه‌گیری گیبز دو مرحله‌ای). مقدار مفروض $x^{(0)}$ را در نظر بگیر. برای $t = 1, 2, \dots$ نمونه‌های $(x^{(t)}, y^{(t)})$ را با اجرای دو مرحله زیر تولید کن:

$$1. \quad y^{(t)} \sim f_{y|x}(\cdot | x^{(t-1)})$$

$$2. \quad x^{(t)} \sim f_{x|y}(\cdot | y^{(t)})$$

الگوریتم ۱ در صورتی که چگالی‌های شرطی صورت بسته‌ای داشته باشند و تولید نمونه از آن‌ها ساده باشد، به سادگی اجرا می‌شود. اگر $(x^{(t)}, y^{(t)})$ از توزیع f باشند، آن‌گاه $(x^{(t+1)}, y^{(t+1)})$ نیز از توزیع f خواهند بود، به عبارت دیگر توزیع مانای این زنجیر، f است.

نمونه‌گیر گیبز چند مرحله‌ای

نمونه‌گیر گیبز چند مرحله‌ای تعمیمی از نمونه‌گیر گیبز دو مرحله‌ای است. فرض کنید که برای $p > 1$ متغیر تصادفی X را بتوان به صورت $X = (X_1, \dots, X_p)'$ نوشت که در آن مولفه‌های X_i ها ممکن است تک بعدی یا چند بعدی باشند. علاوه بر این فرض کنید از توابع چگالی متناظر با $f_1, \dots, f_p$ می‌توان به صورت زیر شبیه‌سازی کرد

$$X_i | x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_p \sim f_i(x_i | x_1, x_2, \dots, x_{i-1}, x_{i+1}, \dots, x_p), \quad i = 1, \dots, p.$$

الگوریتم نمونه‌گیر گیبز تحت انتقال از $X^{(t)}$ به $X^{(t+1)}$ با تکرارهای $t = 1, 2, \dots$ و با فرض داشتن $X^{(t)} = (x_1^{(t)}, \dots, x_p^{(t)})$ ، به صورت زیر است:

$$x_1^{(t+1)} \sim f_1(x_1 | x_2^{(t)}, \dots, x_p^{(t)}) \quad ۱.$$

$$x_2^{(t+1)} \sim f_2(x_2 | x_1^{(t+1)}, x_3^{(t)}, \dots, x_p^{(t)}) \quad ۲.$$

⋮

$$x_p^{(t+1)} \sim f_p(x_p | x_1^{(t+1)}, \dots, x_{p-1}^{(t+1)}) \quad ۳.$$

چگالی‌های $f_1, \dots, f_p$ را چگالی‌های شرطی کامل^{۳۲} می‌گویند. هر اندازه میزان خودهمبستگی بین نمونه‌های تولیدشده زیاد باشد، الگوریتم برای همگرایی، نیازمند تعداد تکرارهای بیشتری است. اما در حالت کلی همگرایی الگوریتم‌های MCMC به انتخاب توزیع‌های پیشین پارامترها بستگی دارد.

الگوریتم متروپلیس-هستینگز درون گیبز

در حالتی که شبیه‌سازی از یک چگالی شرطی کامل بسیار دشوار باشد، می‌توان در مراحل مختلف نمونه‌گیر گیبز از الگوریتم متروپلیس-هستینگز استفاده کرد. فرض کنید شبیه‌سازی از

$$X_i^{(t+1)} \sim f_i(x_i | x_1^{(t+1)}, \dots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \dots, x_p^{(t)})$$

مشکل باشد. در این مرحله از نمونه‌گیر گیبز، از الگوریتم متروپلیس-هستینگز استفاده می‌شود که توزیع مانای آن به شکل $f_i(x_i | x_1^{(t+1)}, \dots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \dots, x_p^{(t)})$ است. این نوع الگوریتم ترکیبی از دو الگوریتم گیبز و MH است که به آن الگوریتم MH درون گیبز^{۳۳} می‌گویند. در مواردی که بعد یک یا چند مرحله از نمونه‌گیری گیبز ترکیبی بزرگ‌تر از یک باشد، الگوریتم MCMC را الگوریتم بلوکی^{۳۴} نیز می‌گویند. در استفاده از الگوریتم‌های MCMC بلوکی،

^{۳۲} Full conditional

^{۳۳} Metropolis-within-Gibbs algorithm

^{۳۴} Blocked algorithm

این که کدام پارامترها در یک بلوک قرار بگیرند و با هم به‌روز شوند اهمیت ویژه‌ای دارد. جداگانه به‌روز کردن مولفه‌هایی که به هم وابسته هستند، اثر منفی بر کارایی الگوریتم دارد. به عبارت دیگر، بلوک‌بندی گروهی مولفه‌هایی که به هم وابسته هستند و از سایر مولفه‌ها مستقل، از روش‌های معمول برای افزایش کارایی محاسباتی الگوریتم‌های MCMC است، که این موضوع را در این پایان‌نامه مورد بررسی قرار می‌دهیم و در فصل دوم و سوم به مطالعه بیشتر آن می‌پردازیم.

۲.۳.۱ معیارهای همگرایی زنجیرهای MCMC

در مواردی که نتوان از چگالی هدف $f(\cdot)$ به‌طور مستقیم نمونه تولید کرد، می‌توان از یک زنجیر مارکوف مانا، که توزیع مانای آن همان $f(\cdot)$ است، به‌صورت غیرمستقیم برای این کار استفاده کرد. از آنجایی که نمونه‌های تولیدشده از زنجیر مارکوف وابسته هستند، در نتیجه کیفیت یک نمونه مستقل با حجمی مشابه را ندارند. برای سنجش کیفیت نمونه‌های تولیدشده، از معیار حجم نمونه موثر^{۳۵} (ESS) می‌توان استفاده کرد (گلمن و همکاران، ۱۹۹۸؛ رابرت و کسلا، ۲۰۰۴). هر قدر مقدار ESS محاسبه‌شده برای هر پارامتر به تعداد نمونه‌های تولیدشده نزدیک‌تر باشد، نمونه‌های تولیدشده از کیفیت بالاتری برخوردار هستند.

تعریف ۴.۱. اگر n تعداد نمونه‌های تولیدشده از زنجیر باشد، آن‌گاه حجم نمونه موثر برای پارامتر مورد نظر به‌صورت زیر محاسبه می‌شود:

$$ESS = \frac{n}{1 + 2 \sum_{k=1}^{\infty} p_k}$$

که در آن p_k مقدار خودهمبستگی در k امین تاخیر^{۳۶} است. در واقع ESS کمیتی است که تعداد نمونه‌های مستقل به‌دست آمده از زنجیر را تخمین می‌زند. در مباحث نظری زنجیر مارکوف، در صورتی می‌توان نمونه تولید کرد که زنجیر دارای یک توزیع مانای $f(\cdot)$ باشد. در نتیجه، هسته مارکوف K و توزیع مانای $f(\cdot)$ در رابطه زیر صدق می‌کنند (رابرت و کسلا، ۲۰۰۴):

$$\int_x K(x, y) f(x) dx = f(y).$$

وجود یک توزیع مانا، ویژگی تحویل‌ناپذیری^{۳۷} را بر هسته K تحمیل می‌کند. یعنی K ، صرف نظر از مقدار اولیه $x^{(0)}$ ، اجازه حرکت بر روی تمام وضعیت‌های فضای حالت زنجیر را خواهد داشت. ویژگی بعدی زنجیر مارکوف بازگشتی^{۳۸} بودن آن است. به این معنی که زنجیر می‌تواند به هر مجموعه دلخواه به تعداد متناهی بازگردد. در این حالت توزیع مانای یک توزیع حدی^{۳۹}

^{۳۵}Effective sample size

^{۳۶}Lag

^{۳۷}Irreducibility

^{۳۸}Recurrency

^{۳۹}Limiting distribution

است. یعنی توزیع $x^{(t)}$ ، به ازای هر مقدار اولیه $x^{(0)}$ ، به توزیع مانای $f(\cdot)$ همگرا است. به این ویژگی، ارگودیک بودن نیز می‌گویند. طبق قضیه ارگودیک

$$\frac{1}{T} \sum_{t=1}^T h(x^{(t)}) \rightarrow E_f[h(x)]$$

که در آن $h(\cdot)$ یک تابع اندازه‌پذیر^{۴۰} است.

شبیه‌سازی از یک زنجیر MCMC به دو بخش تقسیم می‌شود: قبل از همگرایی زنجیر و بعد از همگرایی زنجیر. بخش قبل از همگرایی به دوره سوزاندن^{۴۱} معروف است و به‌عنوان بخشی از زنجیر که در آن نمونه‌های تولیدشده را نمی‌توان به‌عنوان یک معرف خوب از توزیع پسین در نظر گرفت، معرفی می‌شود. پیدا کردن دوره سوزاندن و حذف نمونه‌ها باعث صرفه‌جویی در زمان خواهد شد. در بخش بعد از همگرایی، نمونه‌های تولیدشده از $f(\cdot)$ برای استنباط در مورد کمیت‌های مورد نظر استفاده می‌شوند. این موضوع نظارت بر همگرایی را با مقایسه استنباط‌های به‌دست آمده از چند دنباله تولیدشده مستقل با نقاط شروع متفاوت، پیشنهاد می‌کند (گلמן و روبین، ۱۹۹۲).

روش‌های متعددی برای بررسی همگرایی زنجیرهای MCMC وجود دارند. از جمله این روش‌ها می‌توان به معیارهای همگرایی گلמן و روبین (۱۹۹۲)، جی‌وک (۱۹۹۲) و هیدل برگ و ولچ (۱۹۸۳) اشاره کرد. برای یک زنجیر مشخص، این معیارها توسط توابعی از بسته‌های موجود در نرم‌افزار R، از جمله بسته coda (پلامر و همکاران، ۲۰۰۶) قابل محاسبه هستند.

معیار همگرایی گلמן و روبین

گلמן و روبین (۱۹۹۲) روشی برای ارزیابی همگرایی شبیه‌سازی یک زنجیر مارکوف به توزیع مانای هدف ارائه کردند، که در آن دو یا چند زنجیر به‌صورت موازی اما با مقادیر شروع متفاوت که از پراکندگی زیادی نسبت به توزیع پسین برخوردار باشند، تولید می‌شوند. همگرایی زمانی رخ می‌دهد که واریانس درون زنجیرها با واریانس بین زنجیرها یکسان باشد. اساس کار این معیار همگرایی، برآورد و مقایسه این دو نوع واریانس است. به‌طور کلی دو برآوردگر برای برآورد واریانس توزیع مانا وجود دارند. اولی متوسط واریانس تجربی درون هر زنجیر که با W نمایش داده می‌شود و دوم واریانس تجربی همه زنجیرهای آمیخته‌شده که به‌صورت زیر است:

$$\hat{\sigma}^2 = \frac{(n-1)B}{n} + \frac{W}{n}$$

و در آن B واریانس تجربی بین زنجیرها است. اگر زنجیر همگرا باشد، هر دو برآوردگر نااریب هستند. در غیر این صورت روش اول واریانس را کم و روش دوم واریانس را بیش برآورد

^{۴۰} Measurable

^{۴۱} Burn-in

می‌کند. بر این اساس معیار همگرایی گلمن و روبین به صورت

$$R = \sqrt{\frac{(d+3)\hat{V}}{(d+1)W}}$$

تعریف می‌شود، که در آن

$$\hat{V} = \hat{\sigma}^2 + \frac{B}{mn}, \quad d = \frac{2\hat{V}}{\text{Var}(\hat{V})}$$

و m و n به ترتیب تعداد و طول زنجیرها هستند. مقادیر خیلی بزرگ‌تر از ۱ معیار R نشان از عدم همگرایی زنجیر است.

معیار همگرایی جی‌وک

جی‌وک در سال ۱۹۹۲ یک آماره آزمون تشخیص همگرایی که برابری میانگین‌های بخش اول و آخر زنجیرهای مارکوف را اندازه‌گیری می‌کند، ارائه داده است. این دو بخش، به طور معمول، ۱۰ درصد اول و ۵۰ درصد آخر زنجیر در نظر گرفته می‌شوند. آزمون تشخیص همگرایی برای تعیین این که چه تعداد از تکرارهای آغازی باید نادیده گرفته شوند، مورد استفاده قرار می‌گیرد. به عبارتی این آزمون به عنوان یک برآوردگر برای دوره سوزاندن شناخته می‌شود.

اگر n تعداد تکرارها و n_0 برابر با اولین تکراری باشد که بخواهیم با انجام این آزمون بررسی کنیم که زنجیر همگرا می‌شود یا نه، باید آماره آزمون جی‌وک که یک Z -امتیاز^{۴۲} استاندارد است، محاسبه شود. فرض کنید $\theta(X)$ کمیت مورد دلخواه باشد و $\theta^t = \theta(X^{(t+n_0)})$. اگر n_A تکرار اول، تعداد تکرارهای بخش A و n_B تکرار آخر، تعداد تکرارهای بخش B ، که در آن n^* مقداری است که $n_B = n - n^* + 1$ باشند، آن‌گاه بخش‌های A و B و میانگین‌های آن‌ها به صورت زیر تعریف می‌شوند:

$$A = \{t : 1 \leq t \leq n_A\}$$

$$B = \{t : n^* \leq t \leq n\}$$

$$1 < n_A < n^* < n, \quad \frac{n_A + n_B}{n} < 1$$

$$\bar{\theta}_A = \frac{1}{n_A} \sum_{t \in A} \theta^t$$

$$\bar{\theta}_B = \frac{1}{n - n^* + 1} \sum_{t \in B} \theta^t.$$

آزمون جی‌وک بیان‌گر این مطلب است که اگر زنجیر در n_0 همگرا شود، آن‌گاه نمونه‌های تولیدشده از توزیع مانای زنجیر گرفته شده‌اند و در نتیجه میانگین‌های محاسبه‌شده از اولین

^{۴۲} Z-score

و آخرین بخش زنجیر، معادل هستند. بنابراین آماره جی وک که به طور مجانبی دارای توزیع نرمال استاندارد است، عبارتست از

$$Z_n = \frac{\bar{\theta}_A - \bar{\theta}_B}{\sqrt{\frac{\hat{S}_{\theta}^A}{n_A} + \frac{\hat{S}_{\theta}^B}{n-n^*+1}}} \rightarrow N(0, 1) \quad n \rightarrow \infty$$

که در آن $\frac{\hat{S}_{\theta}^B}{n-n^*+1}$ واریانس‌های مجانبی $\bar{\theta}_A$ و $\bar{\theta}_B$ هستند. با توجه به مطالب ذکر شده می‌توان فرضیه صفر، فرضیه برابری میانگین‌ها، را روی فضای حالت θ آزمون کرد. در این صورت، اگر مقدار $|Z_n|$ بزرگ باشد، آنگاه فرضیه صفر رد می‌شود و این یعنی زنجیر هنوز همگرا نشده است.

معیار همگرایی هیدلبرگ-ولچ

این معیار توسط هیدلبرگ و ولچ (۱۹۸۳) مطرح شد. در این روش روند همگرایی زنجیر در دو مرحله صورت می‌گیرد. اگر زنجیر در مرحله اول همگرا شود، وارد مرحله دوم همگرایی می‌شود. در مرحله اول، آمار آزمون کرامر وان میس ^{۴۳} (وان میس، ۱۹۳۱) محاسبه می‌شود. برای این آماره، آزمون فرضیه

$$\begin{cases} H_0 : & \text{زنجیر دارای توزیع مانا است} \\ H_1 : & \text{زنجیر دارای توزیع مانا نیست} \end{cases}$$

در نظر گرفته و به صورت زیر محاسبه می‌شود:

$$T = \int_0^1 B_n^{\gamma}(t) dt$$

به طوری که، با توجه به تعریف

$$Y_0 = 0, \quad Y_n = \sum_{j=0}^n X^j, \quad \bar{X} = \frac{1}{n} \sum_{j=0}^n X^j$$

$B_n(t)$ به صورت زیر به دست می‌آید:

$$B_n(t) = \frac{Y_{[nt]} - [nt]\bar{X}}{\sqrt{n\hat{S}(0)}} \quad 0 \leq t \leq 1$$

که در آن $[x]$ بزرگترین عدد صحیح است که از x بیشتر نیست. در این مرحله اگر فرضیه صفر پذیرفته شود، آنگاه زنجیر همگراست و وارد مرحله دوم همگرایی می‌شود. در صورتی که فرضیه صفر رد شود، ۱۰٪ زنجیر قطع شده و آماره آزمون دوباره محاسبه می‌شود. این روند تا زمانی که فرضیه صفر پذیرفته شود یا نیمی از زنجیر قطع شود، ادامه می‌یابد.

^{۴۳} Cramer-Von Mises statistic

در مرحله دوم، برای زنجیر پذیرفته‌شده در مرحله قبل یک فاصله اطمینان ۹۵ درصدی حول میانگین در نظر گرفته می‌شود. اگر شرط زیر برقرار باشد

$$\frac{[\text{طول بازه}] / 5}{\text{میانگین}} \leq \epsilon$$

فرضیه صفر پذیرفته می‌شود. در نتیجه زنجیر همگرا خواهد بود. همچنین مقدار ϵ توسط کاربر مشخص می‌شود. این مرحله به آزمون نیم‌فاصله^{۴۴} معروف است. در ادامه، به‌طور خیلی خلاصه، مدل‌هایی که برای تشریح عملکرد روش اصلی این پایان‌نامه در فصل چهارم مورد استفاده قرار می‌گیرند را معرفی می‌کنیم.

۴.۱ مدل‌های خطی تعمیم‌یافته

مدل‌های خطی تعمیم‌یافته^{۴۵} (GLMs)، تعمیم انعطاف‌پذیری از رگرسیون خطی ساده است که میانگین متغیر پاسخ را با یک تابع پیوند^{۴۶} $g(\cdot)$ به متغیرهای تبیینی رگرسیونی متصل می‌کند. مدل‌های خطی تعمیم‌یافته توسط نلدر و ودرین در سال ۱۹۷۲ به‌عنوان یک چارچوب منسجم برای تجمیع مدل‌های آماری گوناگون شامل رگرسیون خطی، رگرسیون لجستیک و رگرسیون پواسون پیشنهاد شدند.

در بررسی اجمالی مدل‌های خطی تعمیم‌یافته، متغیر پاسخ y از یک توزیع خاص در خانواده توزیع‌های نمایی تولید می‌شود که شامل طیف بزرگی از توزیع‌های احتمالی مانند نرمال، دو جمله‌ای و پواسون می‌شود. هر مدل خطی تعمیم‌یافته دارای شکل کلی به‌صورت زیر است:

$$g(\mu_i) = g[E(y_i|x_i)] = x_i'\beta$$

که در آن x_i بردار متغیرهای تبیینی با بعد p و β بردار پارامترهای ضرایب رگرسیونی با بعد p است. تابع g نیز تابع پیوند است. به‌طور کلی این مدل دارای سه مؤلفه اصلی است:

۱. توزیع احتمال از خانواده توزیع‌های نمایی

۲. پیش‌گوی خطی^{۴۷}

$$\eta = x'\beta = \sum_{j=1}^p x_{ij}\beta_j$$

۳. تابع پیوند g که پیش‌گوی خطی را به میانگین متغیر پاسخ مرتبط می‌کند. به‌عبارتی

$$E(y|x) = \mu = g^{-1}(\eta).$$

^{۴۴}Halfwide test

^{۴۵}Generalized linear models

^{۴۶}Link function

^{۴۷}Linear predictor

اگر بخواهیم با جزئیات بیشتر این سه مولفه را معرفی کنیم، می توان نوشت

۱. توزیع احتمال

تابع چگالی احتمال متغیر پاسخ y با میانگین $E(y) = \mu$ و پارامترهای θ و τ به شکل زیر نمایش داده می شود:

$$f(y|\theta, \tau) = \exp \left\{ \frac{b(\theta)T(y) - A(\theta)}{d(\tau)} + h(y, \tau) \right\}$$

که در آن τ پارامتر پراکندگی است و به طور معمول، معلوم فرض می شود و معمولاً به واریانس توزیع مربوط می باشد. توابع $A(\theta)$ ، $T(y)$ ، $b(\theta)$ ، $h(y, \tau)$ و $d(\tau)$ نیز معلوم هستند.

۲. پیش گوی خطی

پیش گوی خطی یک ترکیب خطی از متغیرهای تبیینی رگرسیونی $x = (x_1, \dots, x_p)'$ را تشکیل می دهد. این ترکیب را می توان به صورت زیر نوشت:

$$\eta = x'\beta = \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_p x_p.$$

معمولاً متغیر تبیینی x_1 برای هر مشاهده برابر ۱ در نظر گرفته می شود تا پارامتر عرض از مبدا نیز در مدل وارد شود.

۳. تابع پیوند

یک تابع پیوند چگونگی وابستگی $E(y) = \mu$ را به $x'\beta$ نشان می دهد و معمولاً به صورت $g(\mu)$ نوشته می شود. انتخاب تابع پیوند تا حدی اختیاری است و رابطه آن با پیش گوی خطی به صورت زیر است:

$$g(\mu) = \eta = x'\beta.$$

ساده ترین تابع پیوند، $g(\mu) = \mu$ است که یک تابع پیوند همانی نامیده می شود و رگرسیون خطی کلاسیک را نتیجه می دهد. در واقع، این تابع، میانگین را به طور مستقیم مدل سازی کرده و مدلی شبیه مدل رگرسیون معمولی پدید می آورد که معمولاً برای متغیر پاسخ پیوسته مورد استفاده قرار می گیرد:

$$\mu = \eta = x'\beta.$$

مدل رگرسیون پواسون

یکی از اعضای پرکاربرد مدل های خطی تعمیم یافته، مدل رگرسیون پواسون است. این مدل، معمولاً، برای تحلیل داده های شمارشی به کار می رود. در مدل رگرسیون پواسون متغیر پاسخ یک متغیر شمارشی است و از توزیع پواسون پیروی می کند. زمینه های کاربردی بسیاری برای

رگرسیون پواسون وجود دارند. این کاربردها وقتی پیش می‌آیند که پاسخ مورد علاقه در مساله تنها مقادیر صحیح نامنفی $0, 1, 2, 3, \dots$ را اختیار می‌کند. به عنوان مثال‌هایی از رگرسیون پواسون می‌توان به تعداد دفعات اهدای خون یا تعداد دفعات ترک اعتیاد اشاره کرد. توزیع احتمال پواسون برای متغیر تصادفی Y به صورت زیر است:

$$f(y, \mu) = \frac{\mu^y e^{-\mu}}{y!} \quad y = 0, 1, 2, \dots$$

در مدل رگرسیون پواسون تابع پیوند طوری انتخاب می‌شود که معکوس آن یک تابع مثبت باشد. بنابراین تابع پیوند لگاریتمی یک انتخاب طبیعی است. یعنی

$$\ln(\mu) = x'\beta$$

که در آن β و x به ترتیب بردارهای p بعدی از پارامترها و متغیرهای تبیینی هستند. لگاریتم تابع درست‌نمایی این مدل برای n مشاهده به صورت زیر است:

$$\log L(\beta) = \sum_{i=1}^n \{-\exp(x'_i \beta) + y_i x'_i \beta - \log(y_i!)\}.$$

محاسبه برآورد پارامترها مستلزم حل معادلات غیرخطی

$$\sum_{i=1}^n \{y_i - \exp(x'_i \hat{\beta})\} x_i = 0$$

توسط روش کم‌ترین توان‌های دوم تکراری وزنی^{۴۸} (IWLS) است. می‌توان نشان داد که برآوردگر پارامتر β توسط معادله

$$\hat{\beta} = (X'WX)^{-1} X'Wy$$

به صورت روش تکراری IWLS و به صورت عددی محاسبه می‌شود که در آن ماتریس وزن W قطری با اعضای

$$w_{ii} = \frac{1}{\text{Var}(y_i)} \left(\frac{\partial \mu_i}{\partial \eta_i} \right)^2$$

و X ماتریس طرح حاصل از متغیرهای تبیینی برای همه مشاهدات است. برآورد پارامترها، تحت شرایط نظم، به طور مجانبی دارای توزیع نرمال با میانگین β و ماتریس واریانس A^{-1} است. به عبارت دیگر $\sqrt{n}(\hat{\beta} - \beta) \rightarrow N(0, A^{-1})$ که در آن

$$A = \lim_{n \rightarrow \infty} E \left[\frac{1}{n} \sum_i \exp(x'_i \beta) x_i x'_i \right].$$

علاوه بر این ویژگی سازگار بودن برآوردهای پارامترها نیز برقرار است.

^{۴۸}Iteratively re-weighted least squares

۵.۱ مدل‌های آمیخته خطی تعمیم‌یافته

یکی از پذیره‌های اساسی مدل‌های خطی تعمیم‌یافته، استقلال بین مشاهدات است. اما در مواردی مانند داده‌های فضایی که پاسخ‌ها به هم وابسته هستند، این پذیره برقرار نیست. در این موارد، تعمیمی از این مدل‌ها معروف به مدل‌های آمیخته خطی تعمیم‌یافته^{۴۹} (GLMMs) مورد استفاده قرار می‌گیرد که توسط بریسلو و کلیتون (۱۹۹۳) به کار گرفته شد. در این موارد که بین مشاهدات همبستگی وجود دارد، معمولاً پذیره استقلال مشاهدات به استقلال شرطی تعدیل و همبستگی بین آن‌ها با اضافه کردن اثرات تصادفی از طریق متغیرهای پنهان به مدل لحاظ می‌شود. به عبارتی با افزودن اثرات تصادفی به پیشگوی خطی در یک GLM، یک خانواده از مدل‌های منعطف که در موقعیت‌های مختلفی کاربرد دارند، تشکیل می‌شود. در این مدل‌ها، متغیر پاسخ مستقل شرطی بوده و دارای توزیعی از خانواده توزیع‌های نمایی است. صورت پیش‌گوی خطی در یک GLMM به صورت

$$g(\mu) = \eta = x'\beta + z'u$$

تبدیل می‌شود، که در آن u بردار اثرات تصادفی با بعد q است که معمولاً دارای توزیع نرمال فرض می‌شود. همچنین z بردار متغیرهای تبیینی متناظر با اثرات تصادفی و هم بعد با u است. مؤلفه $x'\beta$ نیز مشابه پیش‌گوی خطی مدل GLM است. به عنوان یک مثال خاص، یک مدل عرض از مبدا تصادفی را معرفی می‌کنیم.

مثال ۱.۱. در یک مدل GLM اگر پیش‌گوی خطی η به صورت زیر باشد:

$$\eta = x'\beta + u = (\beta_1 + u) + \beta_2 x_2 + \dots + \beta_p x_p$$

که در آن u یک اثر تصادفی با توزیع $N(0, \sigma_u^2)$ است، به مدل نتیجه‌شده، مدل عرض از مبدا تصادفی می‌گویند که عضوی از رده GLMM است.

۶.۱ مدل‌های فضای حالت

داده‌هایی که از مشاهده یک پدیده در طول زمان به دست می‌آیند، به عنوان داده‌های سری‌های زمانی شناخته می‌شوند. سری‌های زمانی در علوم مختلفی مانند ژئوفیزیک، اقتصاد، مهندسی ارتباطات و هواشناسی در سطح وسیعی مورد استفاده قرار می‌گیرند. با گسترش دامنه کاربردهای سری‌های زمانی نیاز دانش‌پژوهان به آشنایی هرچه بیشتر با روش‌ها و تکنیک‌های جدید مدل‌بندی و تحلیل در این زمینه بیشتر می‌شود.

^{۴۹} Generalized linear mixed models

رده پیکاربرد از مدل‌های مفید برای در نظر گرفتن رفتارهای تجربی واقعی و تحلیل سری‌های زمانی، رده مدل‌های فضای حالت^{۵۰} است (دوربین و کوپمن، ۲۰۰۱). مدل فضای حالت ابتدا در رشته‌های مهندسی و به‌ویژه مباحث مربوط به سیستم‌های کنترلی مورد استفاده قرار گرفته است. این مدل‌ها شامل دو معادله مشاهده^{۵۱} و حالت^{۵۲} هستند. مدل‌های فضای حالت برای پیش‌بینی، مدل‌بندی و برآورد متغیرهای غیرقابل مشاهده به کار می‌روند. به عنوان مثال، ماهیت مسائل و پدیده‌های اقتصادی در طول زمان و مکان در حال تغییر است. به همین دلیل محدودیت در مدل‌بندی و روش‌های برآورد مانع از دستیابی به استنباط‌های درست و درک واقعیت‌های اقتصادی می‌شود. مدل‌های فضای حالت یک رده معمول برای استفاده در چنین مواردی است. این مدل برای پیش‌بینی، مدل‌سازی و برآورد متغیرهای غیرقابل مشاهده نظیر درآمد دائمی، انتظارات، مشاهدات از دست رفته، خطاهای اندازه‌گیری و برآورد پارامترهای متغیر با زمان به کار می‌رود. از جمله منابعی که به بیان مباحث نظری و جنبه‌های کاربردی مدل‌های فضای حالت پرداخته‌اند، هاروی (۱۹۸۹)، وست و هریسون (۱۹۹۷)، دوربین و کوپمن (۲۰۰۱) و میگون و همکاران (۲۰۰۵) قابل اشاره‌اند.

مدل فضای حالت گاوسی

مدل فضای حالت گاوسی یکی از مدل‌های استاندارد رده مدل‌های فضای حالت است که کاربردهای زیادی در علوم مختلف دارد. ساختار کلی این مدل به شکل زیر است:

$$y_t = Z_t \alpha_t + \epsilon_t; \quad \epsilon_t \sim N(0, H_t)$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \xi_t; \quad \xi_t \sim N(0, Q_t) \quad t = 1, \dots, n. \quad (1.1)$$

این مدل شامل دو معادله است. به معادله اول، معادله مشاهده، سیگنال یا اندازه‌گیری گفته می‌شود. این معادله بیان‌گر ارتباط بین متغیرهای قابل مشاهده و غیرقابل مشاهده (متغیرهای حالت α_t) است، به طوری که متغیر y_t در این معادله قابل مشاهده و اندازه‌گیری است و مقادیر آن از قبل مشخص هستند. در معادله مشاهده y_t یک بردار $(n \times 1)$ از متغیرهای مشاهده‌شده در زمان t و α_t یک بردار $(k \times 1)$ شامل متغیرهای غیرقابل مشاهده یا متغیرهای حالت است. همان‌طور که از نام متغیر غیرقابل مشاهده معلوم است، این متغیر به صورت مستقیم قابل اندازه‌گیری نیست و مقادیر آن از قبل مشخص نیستند. برای پی بردن به ماهیت این متغیر از مقادیر مربوط به متغیر قابل مشاهده y_t استفاده می‌شود. دومین معادله در (۱.۱)، تغییرات متغیر حالت α_t را در طول زمان نشان می‌دهد. این معادله، معادله حالت یا معادله انتقال نامیده می‌شود. ماتریس‌های Z_t و T_t ، R_t در سیستم معادلات (۱.۱)، ماتریس پارامترها و به ترتیب با ابعاد $(k \times r)$ ، $(k \times k)$ و $(n \times k)$ هستند. به این ماتریس‌ها، ماتریس‌های سیستمی

^{۵۰} State space

^{۵۱} Observation equation

^{۵۲} Latent equation

نیز گفته می‌شود. ماتریس‌های سیستمی شامل پارامترهای مدل هستند که در صورت نامعلوم بودن، باید برآورد شوند. این پارامترها ویژگی‌های تصادفی مدل را تبیین می‌کنند. همچنین بردارهای $\epsilon_t(n \times 1)$ و $\xi_t(k \times 1)$ جملات خطا و دوبه‌دو از هم مستقل هستند. ساختار معادله اول (۱.۱) یک مدل رگرسیون خطی است که بردار ضریب α_t تحت زمان تغییر می‌کند و مدل دوم نشان‌دهنده یک مدل اتو رگرسیو مرتبه اول است.

۷.۱ آمار فضایی

در بسیاری از روش‌های آماری پذیره استقلال مشاهدات یک پذیره اساسی است، زیرا نظریه‌های بسیاری بر اساس آن بنا شده‌اند. اما در عمل با داده‌هایی مانند سری‌های زمانی یا داده‌های فضایی^{۵۳} روبه‌رو می‌شویم که به‌نوعی به یک‌دیگر وابسته هستند. داده‌هایی که برحسب موقعیت قرار گرفتن آن‌ها در فضای مورد مطالعه همبسته باشند و این نوع همبستگی تابعی از فاصله بین موقعیت آن‌ها باشد، داده‌های فضایی نامیده می‌شوند. به‌عنوان مثالی از داده‌های فضایی می‌توان به میزان گازهای آلاینده هوا در نقاط مختلف یک شهر اشاره کرد. داده‌ها با موقعیت مکانی نزدیک‌تر، دارای شباهت بیشتری هستند که این امر سبب بروز همبستگی فضایی بین داده‌ها می‌شود. به دلیل وجود همبستگی فضایی، روش‌های متداول آماری برای تحلیل چنین داده‌هایی قابل استفاده نیستند. لذا شاخه‌ای به نام آمار فضایی به منظور مدل‌سازی ساختار وابستگی و تحلیل داده‌های فضایی گسترش یافته است (محمدزاده، ۱۳۹۱). در واقع آمار فضایی شاخه‌ای از آمار است که در آن به بررسی متغیرهایی پرداخته می‌شود که دارای ساختار همبستگی فضایی هستند و تلاش می‌شود این ساختار که همان ارتباط بین مقادیر متغیر و فاصله و جهت قرارگیری آن‌هاست، تعیین و برای افزایش دقت در تحلیل آن‌ها مورد استفاده قرار گیرد.

آمار فضایی شاخه نسبتاً جدیدی از آمار است که در چند دهه اخیر مورد توجه بسیاری از متخصصین آمار قرار گرفته است. در شکل‌گیری آمار فضایی افراد زیادی فعالیت داشته‌اند. به‌عنوان مثال در ابتدا استیودنت (۱۹۰۷) در بررسی توزیع ذرات معلق در مایعات پیشنهاد کرد داده‌ها در سطوح مختلف به‌طور جداگانه به‌دست آورده شوند. فیشر (۱۹۳۵) با توجه به این که کرت‌های نزدیک به هم از لحاظ محصول شباهت بیشتری نسبت به کرت‌های دور از هم دارند، به وجود همبستگی فضایی در طرح آزمایش‌های کشاورزی اشاره کرد. همچنین دیوید (۱۹۷۷) در مورد وابستگی فضایی ذخایر معدنی مطالب بسیاری ارائه کرده است. مشابه داده‌های سری‌های زمانی، داده‌های فضایی مشاهده‌شده از نظر احتمالی باید همبسته باشند، که بتوان آن‌ها را تحلیل کرد. به‌طور کلی داده‌های فضایی را می‌توان به سه دسته تقسیم کرد:

^{۵۳} Spatial data

۱. داده‌های زمین‌آماري^{۵۴}: این نوع داده‌ها در مکان‌های ثابت و مشخص در ناحیه‌ای چگال^{۵۵} مشاهده می‌شوند. در این نوع داده‌ها متغیر مورد مطالعه می‌تواند پیوسته یا گسسته باشد، مثل ثبت میزان بارش باران در مکان‌های مختلف از یک ناحیه. هدف از استنباط این داده‌ها، پیش‌گویی پاسخ در مکان‌های مشاهده‌نشده است.
۲. داده‌های شبکه‌ای^{۵۶}: این نوع داده‌ها مربوط به نواحی غیرچگال هستند، یعنی بین مکان‌های مشاهده‌شده، مکان جدیدی نمی‌توان یافت. این مکان‌ها ممکن است منظم یا نامنظم باشند، مثل تعداد حوادث رانندگی و تعداد اتباع خارجی غیرمجاز. هدف از تحلیل داده‌های فضایی شبکه‌ای مدل‌بندی احتمالی مشاهدات است.
۳. الگوهای نقطه‌ای^{۵۷}: در این نوع داده‌ها مکان مشاهده‌شده، خود یک متغیر تصادفی است. این داده‌ها را می‌توان به سه دسته به‌طور کامل تصادفی فضایی^{۵۸}، منظم^{۵۹} و خوشه‌ای^{۶۰} تقسیم و آن‌ها را مدل‌بندی کرد. یک مثال برای این نوع مشاهدات، محل تخلفات رانندگی در یک مسابقات رالی است.

۸.۱ ساختار همبستگی فضایی

واریانس و کوواریانس در آمار کلاسیک به‌ترتیب میزان پراکندگی و رابطه خطی داده‌ها را نشان می‌دهند. در آمار فضایی نیز می‌توان معیارهایی تعریف کرد که مشابه واریانس و کوواریانس بیان‌گر ساختار همبستگی فضایی داده‌ها باشند. معمولاً داده‌های فضایی که از موقعیت‌های مجاور به‌دست می‌آیند دارای همبستگی بزرگ‌تری بوده و هر چه فاصله بین موقعیت‌ها بیشتر شود، همبستگی نیز کاهش می‌یابد. علاوه بر این‌ها جهت قرارگیری داده‌ها ممکن است در مقادیر مشاهده‌شده تاثیرگذار باشد. ارتباطی که بین مقادیر مختلف متغیرها و فاصله و جهت قرارگیری آن‌ها برقرار می‌شود، ساختار فضایی^{۶۱} نامیده می‌شود. بررسی و مطالعه این ساختار یکی از مسائل مهم آمار فضایی به‌شمار می‌رود. با توجه به این که دو داده نزدیک به هم احتمالاً مقادیری مشابه‌تر از دو داده دور از هم دارند، متوسط تفاضل مقادیر $Y(s)$ و $Y(s+h)$ که به فاصله h از هم قرار گرفته‌اند، می‌تواند معیار خوبی برای بیان اندازه این تشابه و وابستگی متقابل مقادیر میدان تصادفی در دو موقعیت s و $s+h$ باشد. هرچه متوسط این تفاضل‌ها کوچک‌تر و در نتیجه واریانس مقادیر میدان تصادفی کوچک‌تر باشد، می‌توان گفت $Y(s)$ و

^{۵۴} Geostatistical data

^{۵۵} Dense region

^{۵۶} Lattice data

^{۵۷} Point pattern

^{۵۸} Complete spatial randomness

^{۵۹} Regular

^{۶۰} Cluster

^{۶۱} Spatial structure

$Y(s+h)$ به یکدیگر نزدیک‌ترند و شباهت آن‌ها بیشتر است. هر چه متوسط تفاضل‌ها بیشتر یا واریانس مقادیر میدان تصادفی بزرگ‌تر باشد، تشابه آن‌ها کم‌تر است. از معیارهای مهم در آمار فضایی می‌توان به تغییرنگار^{۶۲} و هم‌تغییرنگار^{۶۳} اشاره کرد که بیان‌گر ساختار پراکندگی و وضعیت قرار گرفتن مشاهدات نسبت به یکدیگر هستند.

۱.۸.۱ تغییرنگار

واریانس اختلاف بین مقادیر $Y(s_i)$ و $Y(s_j)$ که به فاصله $h = s_j - s_i$ از یکدیگر قرار گرفته‌اند و می‌تواند همبستگی متقابل مقادیر میدان تصادفی در این دو نقطه را بیان کند، تغییرنگار^{۶۴} نامیده می‌شود و به صورت

$$\gamma(h) = \text{Var}(Y(s) - Y(s+h))$$

تعریف می‌شود. تابع $\gamma(h)$ نیز نیم تغییرنگار^{۶۵} نامیده می‌شود. کوچک بودن تغییرنگار نشانه وابستگی زیاد و بزرگ بودن آن نشانه وابستگی کم میدان تصادفی است. تغییرنگار را می‌توان به عنوان معیاری برای نمایش تاثیرپذیری مقادیر میدان تصادفی در نقاط مختلف فضای مورد مطالعه به کار برد. اگر تغییرنگار برای تمام مقادیر h تقریباً ثابت باشد، داده‌ها همبستگی فضایی ندارند یا همبستگی فضایی آن‌ها بسیار ضعیف است. برعکس، شیب نمودار تغییرنگار بیان‌گر همبستگی فضایی داده‌هاست. در آمار فضایی تغییرنگار معیاری برای تعیین تاثیرپذیری مقادیر میدان تصادفی در نقاط مختلف فضای مورد مطالعه است. تغییرنگار دارای سه پارامتر به شرح زیر است.

۱. دامنه^{۶۶}

بازه‌ای که در خارج از آن، تغییرنگار به حالت افقی درآمده و ثابت می‌ماند، دامنه نامیده می‌شود. خارج از دامنه، مشاهدات کم و بیش اثری بر هم ندارند، یعنی تقریباً ناهمبسته هستند. از این‌رو داده‌های واقع در خارج از دامنه با روش‌های کلاسیک آماری قابل تحلیل هستند.

۲. ازاره^{۶۷}

به‌طور معمول تغییرنگار تابعی افزایشی از تاخیر $\|h\|$ است و ممکن است به کران بالایی منتهی شود. چنین کرانی ازاره تغییرنگار نامیده شده و عبارت است از

$$\lim_{\|h\| \rightarrow \infty} \gamma(h) = c.$$

^{۶۲}Variogram

^{۶۳}Covariogram

^{۶۴}Variogram

^{۶۵}Semi-variogram

^{۶۶}Range

^{۶۷}Sill

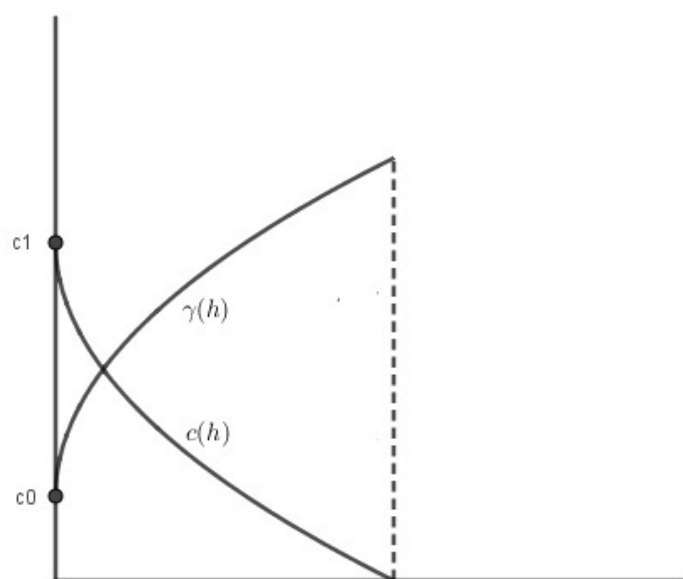
با توجه به طبیعت داده‌های فضایی که انتظار می‌رود در آن‌ها با افزایش فاصله مکانی، همبستگی داده‌ها کاهش یابد و در نتیجه تغییرنگار ثابت بماند، سعی می‌شود که بیشتر مدل‌های کران‌دار به کار گرفته شوند. ازاره به عنوان یکی از پارامترهای مهم تابع تغییرنگار، از روی مشاهدات برآورد می‌شود. هر چند تعیین کران‌دار بودن تغییرنگار از روی مشاهدات مشکل است، در عمل با رسم نمودار تغییرنگار تجربی مقدار c برآورد می‌شود. مقداری که تابع تغییرنگار تجربی بعد از رسیدن به دامنه، حول آن نوسان می‌کند، ازاره را نمایش می‌دهد.

۳. اثر قطعه‌ای^{۶۸}

مقدار تغییرنگار در مبدا مختصات ($h = 0$)، اثر قطعه‌ای نامیده شده و به صورت

$$c_0 = \lim_{h \rightarrow 0} \lambda(h)$$

تعریف می‌شود.



شکل ۱.۱: نمودار تغییرنگار و هم تغییرنگار

۲.۸.۱ هم تغییرنگار

می‌دانیم که در آمار کلاسیک، کوواریانس میزان شباهت رفتار دو متغیر را نشان می‌دهد. در آمار فضایی نیز کمیتی شبیه به کوواریانس وجود دارد که میزان تشابه وابستگی بین دو متغیر

^{۶۸}Nugget effect

را بر اساس فاصله بین آن‌ها بیان می‌کند. هر چه فاصله زیاد باشد وابستگی بین دو متغیر کم و برعکس هر چه فاصله بین آن دو کم شود، وابستگی زیاد خواهد شد. کوواریانس دو متغیر $Y(s)$ و $Y(s+h)$ به صورت

$$\begin{aligned} C(h) &= \text{cov}(Y(s), Y(s+h)) \\ &= E\{[Y(s) - E(Y(s))][Y(s+h) - E(Y(s+h))]\} \end{aligned}$$

تعریف می‌شود و آن را هم‌تغییرنگار می‌نامند. در عمل تابع هم‌تغییرنگار نامعلوم است و باید بر اساس مشاهدات موجود برآورد شود.

۹.۱ مدل‌های تغییرنگار پارامتری معتبر

مدل‌های معتبر تغییرنگار توابعی هستند که در شرط همیشه منفی شرطی^{۶۹} صدق می‌کنند. جورنل و هویج‌برگتس (۱۹۷۸) مدل‌های پارامتری مختلفی را معرفی کردند. مدل‌های تغییرنگار شامل دو گروه با ازاره و فاقد ازاره هستند. در تحلیل کاربردی داده‌ها از مدل‌های پارامتری کوواریانس برای مدل‌بندی ساختار وابستگی استفاده و بر روی تغییرنگار تجربی برازش داده می‌شود. از جمله مهم‌ترین مدل‌های تغییرنگار معتبر پارامتری مدل‌های نمایی، کروی^{۷۰} و گاوسی هستند، که به اختصار به تعریف آن‌ها می‌پردازیم.

۱. مدل گاوسی: تابع نیم‌تغییرنگار نمایی به صورت

$$\gamma(h) = c_0 + c(1 - e^{-\frac{\|h\|^2}{a^2}}), \quad h \in R^d, \quad d \geq 1$$

است و با سه پارامتر a ، c و c_0 مشخص می‌شود، که در آن a دامنه، c_0 اثر قطعه‌ای و $c + c_0$ ازاره است.

۲. مدل نمایی: تابع نیم‌تغییرنگار نمایی

$$\gamma(h) = c_0 + c(1 - e^{-\frac{\|h\|}{a}}), \quad h \in R^d, \quad d \geq 1$$

دارای سه پارامتر a ، c و c_0 است. دامنه آن به‌طور معمول بسیار بزرگ است. وقتی داده‌ها در ناحیه مورد بررسی دارای روند باشند یا نسبت به ابعاد ناحیه تحت مطالعه دارای دامنه بزرگی باشند، تغییرنگار آن‌ها از مدل نمایی پیروی می‌کند.

۳. مدل کروی: تابع نیم‌تغییرنگار کروی

$$\gamma(h) = \begin{cases} c_0 + c(\frac{3}{4} - \frac{\|h\|}{a} + \frac{1}{4} \frac{\|h\|^3}{a^3}) & 0 < \|h\| \leq a, \quad h \in R^d, \quad d = 1, 2, 3 \\ c_0 + c & \|h\| > a \end{cases}$$

^{۶۹}Conditional negative definite

^{۷۰}Spherical

دارای سه پارامتر a ، c و c_0 است. به‌طور مقدار مدل‌های تغییرنگار بدون ازاره با افزایش فاصله هم‌چنان افزایش پیدا می‌کند.

فصل ۲

کارایی الگوریتم‌های MCMC و نقش بلوک‌بندی پارامترها

۱.۲ مقدمه

الگوریتم‌های MCMC، تبدیل به هسته روش‌های محاسباتی بسیاری از تحلیل‌های آماری شده‌اند. این الگوریتم‌ها علاوه بر روش محاسباتی پیش‌فرض تحلیل‌های بیزی، عضوی از سایر روش‌های محاسباتی ترکیبی مثل روش EM^۱ (کافو و همکاران، ۲۰۰۵) و همسانه‌سازی داده‌ها^۲ (له‌له و همکاران، ۲۰۰۷) محسوب می‌شود. با توجه به وسعت کاربرد این الگوریتم‌ها، نسخه‌هایی از این الگوریتم‌ها در نرم‌افزارهای آماری، به‌عنوان یک ابزار جعبه سیاه^۳، تعبیه شده‌اند. اما معمولاً نسخه‌های پیش‌فرض موجود در نرم‌افزارهای مختلف آماری به روش‌های نمونه‌گیری MCMC کارا در مدل‌های پیچیده آماری منجر نمی‌شوند. از طرفی اجرای آن‌ها برای مدل‌های سلسله‌مراتبی پیچیده با حجم بزرگی از داده‌ها نیاز به زمان اجرای طولانی بر حسب چند ساعت یا حتی چند روز دارد. با این توضیح، زمان لازم برای اجرای الگوریتم‌های MCMC یک عامل محدودکننده محسوب می‌شود که می‌تواند برای یک مجموعه

^۱Monte Carlo expectation maximization

^۲Data cloning

^۳Black-box

داده مشخص، ۱) استفاده از مدل‌های پیچیده واقعی را محدود کند و تنها مدل‌های ساده قابل برازش و استفاده باشند؛ یا ۲) محاسبه معیارهای ارزیابی مدل مثل اعتبارسنجی متقابل (گنتینگ و رفتی، ۲۰۰۷) و معیارهای پیش‌گوی پسینی^۴ (هیورت و همکاران، ۲۰۰۶) را غیرممکن سازد. بنابراین، هر نوع اصلاح و تعدیل کاربردی برای کارایی الگوریتم‌های MCMC می‌تواند به اصلاح اساسی تحلیل‌های عملی مدل‌های سلسله‌مراتبی با حجم داده بزرگ، منتهی شود.

در میان نسخه‌های مختلف الگوریتم‌های MCMC که برای افزایش کارایی آن‌ها توسعه یافته‌اند، یکی از پایه‌ای‌ترین رهیافت‌ها نمونه‌گیری بلوکی^۵ است (رابرتز و ساهو، ۱۹۹۷؛ سارگنت و همکاران، ۲۰۰۰). در الگوریتم‌های بلوکی، چند بلوک از پارامترهای توزیع هدف تشکیل شده و مولفه‌های هر یک از بلوک‌ها به‌طور توأم و هم‌زمان (و جدا از مولفه‌های سایر بلوک‌ها) در فرآیند نمونه‌گیری به‌روز می‌شوند.

در نظریه الگوریتم‌های MCMC معروف است که به‌روزرسانی توأم دو یا چند مولفه از توزیع هدف که همبسته هستند، می‌تواند کارایی نمونه‌گیری را، نسبت به زمانی که هر مولفه به‌طور مجزا به‌روز می‌شوند، افزایش دهد (لیو و همکاران، ۱۹۹۴). علی‌رغم درک عمیق مفید و کاربردی بودن این ایده پایه‌ای برای طراحی الگوریتم‌های MCMC بلوکی کارا، تا سال‌های اخیر، روشی خودکار برای انتخاب تعداد بلوک‌ها و مولفه‌های درون هر بلوک، برای بهینه‌سازی یا حداقل افزایش قابل ملاحظه کارایی الگوریتم نمونه‌گیری، وجود نداشته است. در سال ۲۰۱۷، تورک و همکارانش اولین قدم برای معرفی چنین روشی را برداشتند که در فصل سوم به تشریح روش پیشنهادی آن‌ها خواهیم پرداخت.

هدف این فصل، بیان تعریفی از کارایی برای سنجش عملکرد الگوریتم‌های MCMC است. برای این کار، ابتدا دو نوع کارایی را معرفی می‌کنیم:

۱. کارایی الگوریتمی^۶ که حاصل نمونه‌های مستقل موثر از توزیع هدف است.
 ۲. کارایی محاسباتی^۷ که به محاسبات مورد نیاز الگوریتم برای تولید نمونه‌های MCMC مربوط می‌شود.
- ترکیب این دو معیار کارایی، به معرفی یک معیار کارایی کل^۸ می‌انجامد که برای مقایسه الگوریتم‌های MCMC مختلف به کار گرفته می‌شود.
- در مقایسه روش‌های نمونه‌گیری بلوکی با روش‌های نمونه‌گیری یک‌متغیره مولفه به مولفه^۹، کارهای نظری موجود مانند منگرسون و توییدی (۱۹۹۶)، رابرتز و توییدی (۱۹۹۶) و رابرتز

^۴Posterior predictive

^۵Blocking sampling

^۶Algorithmic efficiency

^۷Computational efficiency

^۸Overall efficiency

^۹Univariate componentwise sampling

و همکاران (۱۹۹۷)، نتایج متعددی را نسبت به کارایی الگوریتم‌های MCMC فراهم کرده‌اند؛ اما به چند دلیل یک ارزیابی جامع و کامل وجود ندارد.

۱. اول، آن‌ها از نرخ‌های همگرایی MCMC به‌عنوان معیاری برای مقایسه استفاده می‌کنند و زمان محاسباتی لازم برای نمونه‌گیری بلوکی را نادیده می‌گیرند. این نوع مقایسه نقص دارد، زیرا هر معیاری برای اندازه‌گیری کارایی الگوریتم‌های MCMC باید هر دوی نرخ همگرایی و زمان محاسباتی لازم برای بهبود نرخ همگرایی را شامل شود. این نگاه می‌تواند تصویری متفاوت از کارایی واقعی یک الگوریتم نمونه‌گیری را ارائه دهد. به‌عنوان مثال، بهبود نرخ همگرایی با یک زمان محاسباتی بسیار طولانی، به وضوح نمی‌تواند یک الگوریتم بهینه را نتیجه دهد.

۲. دوم، زمان‌های محاسبات برای اجرای مراحل مختلف یک الگوریتم نمونه‌گیری، در سیستم‌های عامل مختلف می‌توانند متغیر باشند. دلیل آن هم به عواملی مانند پردازنده، مهندسی و ساختار حافظه و استفاده از بسته‌افزارهای جبر خطی کارا بستگی دارد. بنابراین علی‌رغم مقایسه‌های نظری توسعه‌یافته، بهترین روش‌های نمونه‌گیری بلوکی به سیستم عامل رایانه‌ای که الگوریتم در آن اجرا می‌شود وابسته است.

۳. سوم، نتایج نظری موجود معتقدند انتخاب دستی بلوک‌ها قابل دفاع است، زیرا شناخت ساختار مدل سلسله‌مراتبی به این شناخت منتهی می‌شود که بتوان تشخیص داد کدام مولفه‌های توزیع هدف (پسین) همبسته هستند. اما، به‌طور کلی، شناخت قبلی در مورد مولفه‌هایی که همبسته خواهند بود کار دشواری است. همین دشواری ایده بلوک‌بندی خودکار مولفه‌های توزیع را ایجاد کرده است.

در ادامه، ابتدا رده الگوریتم‌های مختلف بلوکی را تعریف و نمادگذاری لازم را معرفی می‌کنیم.

۲.۲ نمادگذاری و تعریف الگوریتم‌های MCMC

در این بخش ابتدا فضای الگوریتم‌های MCMC معتبر را تعریف می‌کنیم. فرض کنید $\mathcal{M}$ یک مدل سلسله‌مراتبی ثابت و معلوم با بردار مولفه‌های $\Theta = \{\theta_1, \dots, \theta_n\}$ باشد، که می‌توانند شامل پارامترها و مولفه‌های (تصادفی) پنهان مدل باشند. از این به بعد به مجموعه Θ ، پارامترهای مدل $\mathcal{M}$ می‌گوییم.

یک الگوریتم MCMC را بر حسب طرح نمونه‌گیری آن روی Θ تعریف می‌کنیم. فرض کنید b هر زیرمجموعه ناتهی (بلوکی) از Θ باشد و $u \in U$ هر روش نمونه‌گیری MCMC معتبر، مانند نمونه‌گیری لایه‌ای^{۱۰} و نمونه‌گیری گیبز، باشد (گیلکس، ۲۰۰۵). یک روش

^{۱۰}Slice sampling

نمونه‌گیری MCMC معتبر مثل $\psi = u(b)$ را به‌عنوان اعمال روش نمونه‌گیری u روی b تعریف می‌کنیم، به‌طوری که b برای همه پذیره‌های روش u برقرار باشد. به‌عنوان مثالی از این پذیره‌ها می‌توان به مزدوج بودن روش نمونه‌گیری اشاره کرد. افزون بر برقراری ویژگی‌های استاندارد زنجیرهای مارکوف (رابرت و کسلا، ۲۰۰۴)، یک الگوریتم MCMC معتبر را به‌عنوان هر مجموعه از روش‌های نمونه‌گیری $\Psi = \{\psi_1, \dots, \psi_k\}$ تعریف می‌کنیم، به‌طوری که برای $i = 1, \dots, k$ ، $\psi_i = u_i(b_i)$ که در شرط $\bigcup_{i=1}^k b_i = \Theta$ صدق کنند. این تعریف الگوریتم MCMC به این معنی است که الگوریتم هر پارامتر مدل را حداقل توسط یک روش نمونه‌گیری به‌روز می‌کند.

زنجیر نمونه‌های تولیدشده با اجرای متوالی Ψ را با $X^{(0)}, X^{(1)}, \dots$ نشان می‌دهیم، به‌طوری که نمونه $X^{(i)}$ بیانگر حالت مدل در مرحله i ام، یعنی $\Theta = X^{(i)}$ است. همچنین $X^{(0)}$ مجموعه مقادیر اولیه زنجیر برای پارامترهای مدل است. علاوه بر این، $X^{(i)} = (X_1^{(i)}, \dots, X_d^{(i)})$ و $X_k = \{X_k^{(0)}, X_k^{(1)}, \dots\}$ زنجیر عددی حاصل از نمونه‌های θ_k برای $k = 1, \dots, d$ است. تمرکز ما در این پایان‌نامه بر روی مجموعه محدودشده از روش‌های نمونه‌گیری

$$U_{\circ} = \{u_{\text{scalar}}, u_{\text{block}}\}$$

است، که در آن منظور از روش نمونه‌گیری یک‌متغیره (مولفه به مولفه) متروپلیس-هستینگز قدم زدن تصادفی سازوار است، که از این به بعد به آن روش نمونه‌گیری مولفه به مولفه^{۱۱} می‌گوییم (متروپلیس، ۱۹۵۳ و هستینگز، ۱۹۷۰)؛ و منظور از u_{block} تعمیم چندمتغیره الگوریتم u_{scalar} است که از این به بعد به آن روش نمونه‌گیری بلوکی خواهیم گفت (هاریو و همکاران، ۱۹۹۳) و فرض می‌کنیم هر الگوریتم $\psi = u_{\text{block}}(b)$ در شرط $|b| > 1$ صدق می‌کند. این شرط به آن معنی است که تعداد مولفه‌های هر بلوک بیش از یک است.

در ادامه فرض می‌کنیم الگوریتم u_{scalar} به‌طور سازوار پارامتر مقیاس توزیع پیشنهادی را تنظیم می‌کند و الگوریتم بلوکی u_{block} ماتریس کواریانس توزیع پیشنهادی را تنظیم می‌کند (رابرتز و روزنتال، ۲۰۰۹).

در این پایان‌نامه، برای مدل سلسله‌مراتبی $\mathcal{M}$ با پارامترهای Θ ، تمرکز ما بر روی مجموعه الگوریتم‌های MCMC $\Psi_{\mathcal{M}}$ است که شامل همه الگوریتم‌های به شکل $\Psi = \{\psi_1, \dots, \psi_k\}$ است، به‌طوری که $\psi_i = u_i(b_i)$ هر $u_i \in U_{\circ}$ ، و مجموعه‌های b_i افزایی از Θ را تشکیل می‌دهند. دو الگوریتم خاص از مجموعه الگوریتم‌های $\Psi_{\mathcal{M}}$ که در دو کران مجموعه قرار می‌گیرند، الگوریتم‌های مولفه به مولفه، Ψ_{scalar} ، و تک بلوکی، Ψ_{block} ، هستند. اولی شامل d روش نمونه‌گیری مولفه به مولفه است، یعنی $\Psi_{\text{scalar}} = \{\psi_1, \dots, \psi_d\}$ ، به‌طوری که $\psi_i = u_{\text{scalar}}(\theta_i)$. دومی که فقط یک تک بلوک برای تمام پارامترها است، یعنی $\Psi_{\text{block}} = \{u_{\text{block}}(\Theta)\}$. در نهایت فرض می‌کنیم پارامترهای θ_i پیوسته هستند.

^{۱۱}Scalar sampling

۳.۲ کارایی الگوریتمی

کارایی الگوریتمی مستقل از هر گونه نیاز محاسباتی قابل تعریف است. معیار کارایی که در این جا تعریف خواهد شد، تا حدودی، بیانگر بهترین آمیختگی^{۱۲} یا به طور معادل کمترین خودهمبستگی یا بزرگترین اندازه نمونه موثر^{۱۳} (ESS) است، که بدون در نظر گرفتن زمان لازم برای تولید یک مجموعه از نمونه‌ها اندازه‌گیری می‌شود.

در این بخش، ابتدا معیار کارایی الگوریتمی روش‌های نمونه‌گیری MCMC را، بر اساس زمان خودهمبستگی تجمیع‌شده^{۱۴}، تعریف می‌کنیم. سپس به مطالعه استفاده از Ψ_{scalar} یا Ψ_{block} برای انتخاب‌های خاصی از مدل $\mathcal{M}$ می‌پردازیم و تاثیر آن‌ها را بر معیار کارایی معرفی شده می‌سنجیم.

معیار کارایی الگوریتمی که تورک و همکاران (۲۰۱۷) تعریف کردند، مشابه کار رابرتز و روزنتال (۲۰۰۱) از تقسیم ESS بر طول زنجیر نتیجه می‌شود. با این تعریف، این معیار بیانگر نرخ تولید نمونه‌های مستقل موثر به ازای هر نمونه MCMC است. اندازه نمونه موثر یک زنجیر MCMC به صورت

$$ESS = \frac{N}{\tau}$$

تعریف می‌شود، که در آن N طول زنجیر و τ خودهمبستگی زمانی تجمیع‌شده است. برای یک زنجیر مولفه به مولفه با نمونه‌های $X_0, X_1, \dots$ که فرض می‌کنیم به توزیع مانای خود همگرا شده است، استراتسما و همکاران (۱۹۸۶) زمان خودهمبستگی تجمیع‌شده را به صورت

$$\tau = 1 + 2 \sum_{i=1}^{\infty} \text{Corr}(X_0, X_i)$$

تعریف کردند. در این تعریف، τ به عنوان تعداد متوسط نمونه‌های MCMC لازم برای تولید یک نمونه مستقل از توزیع هدف قابل تفسیر است.

بنابراین می‌توان معیار کارایی الگوریتمی را τ^{-1} تعریف کرد. برای درک این معیار به مثال زیر توجه کنید.

مثال ۱.۲. یک فرآیند اتورگرسیو مرتبه p را در نظر بگیرید. مدل این فرآیند به صورت زیر است:

$$X_t = \mu + \pi_1 X_{t-1} + \dots + \pi_p X_{t-p} + a_t; \quad a_t \sim N(0, \sigma_a^2)$$

که در آن ضرایب اتورگرسیو هستند و طوری تعیین می‌شوند که مدل مانا باشد. در این مدل، معیار کارایی الگوریتمی به صورت زیر محاسبه می‌شود (وی، ۲۰۰۶):

$$\tau^{-1} = \frac{(1 - \mathbf{1}'_p \boldsymbol{\pi}_{1:p})^2}{1 - \boldsymbol{\rho}'_{1:p} \boldsymbol{\pi}_{1:p}}$$

^{۱۲} Mixing

^{۱۳} Effective sample size

^{۱۴} Integrated autocorrelation time

که در آن $\mathbf{1}_p$ برداری به طول p از ۱ ها است، $\pi_{1:p} = (\pi_1, \dots, \pi_p)'$ بردار ضرایب اتورگرسو است، و $\rho_{1:p} = (\rho_1, \dots, \rho_p)'$ مقادیر خودهمبستگی تا تاخیر p است. از آن جا که پارامترهای مدل و ضرایب خودهمبستگی نامعلوم هستند، می‌توان با کمک روش برآورد یول-واکر^{۱۵} (وی، ۲۰۰۶) آن‌ها را برآورد کرد و معیار τ را برآورد کرد. بنابراین

$$\hat{\tau}^{-1} = \frac{(1 - \mathbf{1}_p' \hat{\pi}_{1:p})^2}{1 - \hat{\rho}_{1:p}' \hat{\pi}_{1:p}}$$

که در آن $\hat{\pi}_{1:p}$ و $\hat{\rho}_{1:p}$ برآوردهای متناظر بردارهای ضرایب خودهمبستگی و ضرایب اتورگرسو هستند.

ویژگی‌های معیار τ^{-1}

با توجه به معیار کارایی الگوریتمی تعریف‌شده، می‌توان ویژگی‌های زیر را برای آن برشمرد:

۱. این معیار، تعداد نمونه‌های موثر به ازای هر نمونه (مستقل) واقعی را اندازه می‌گیرد (تامپسون، ۲۰۱۰).

۲. معیاری است که سرعت همگرایی میانگین توابع دلخواه، که براساس نمونه‌های تولیدشده محاسبه می‌شوند، را به مقادیر نظری امید ریاضی آن توابع تحت توزیع مانای خود، می‌سنجد (رابرتز و ساهو، ۱۹۹۷)

۳. منطبق بر شهود طبیعی، مقادیر بزرگ آن نشان‌دهنده عملکرد بهتر الگوریتم است.

برای الگوریتم MCMC Ψ که روی مدل $\mathcal{M}$ با پارامترهای θ عمل می‌کند، کارایی الگوریتمی برای هر $\theta \in \Theta$ به صورت

$$A(\Psi, \theta) = \tau^{-1}$$

تعریف می‌شود، که در آن τ زمان خودهمبستگی تجمیع‌شده نمونه‌های θ است که با تکرار Ψ تولید شده‌اند. در واقع کارایی الگوریتم MCMC Ψ برای به‌روز کردن بردار θ که در فضای Θ تغییر می‌کند، با معیار $A(\Psi, \theta) = \tau^{-1}$ سنجیده می‌شود.

از آن جایی که اغلب موارد یک الگوریتم MCMC نمونه‌هایی تولید می‌کند که برای تعداد زیادی از پارامترهای مدل دارای آمیختگی خوب و برای تعداد اندکی دارای آمیختگی ضعیفی است، از معیار

$$A(\Psi) = \min_{\theta \in \Theta} A(\Psi, \theta)$$

برای سنجش کارایی الگوریتمی استفاده می‌کنیم. در واقع آمیختگی ضعیف حتی تعداد اندکی از پارامترهای مدل، اعتبار تمام نمونه‌های پسینی تولیدشده را زیر سوال می‌برد. به همین

دلیل، با اتخاذ یک رهیافت سختگیرانه، به دنبال الگوریتمی هستیم تا کارایی الگوریتمی را برای پارامترهایی از مدل ماکسیمم کند که ضعیف‌ترین آمیختگی را نتیجه می‌دهند.

اکنون برای روش‌های نمونه‌گیری مولفه به مولفه و بلوکی، شرایطی را در نظر می‌گیریم که انتخاب آن‌ها مناسب نیست و با این کار ناکارایی الگوریتمی آن‌ها را مورد بررسی قرار می‌دهیم.

۱.۳.۲ کارایی از دست رفته نمونه‌گیری بلوکی

الگوریتم MCMC، $\Psi_{block} \in \Psi_{\mathcal{M}}$ را در نظر بگیرید. اجرای Ψ_{block} بردار پیشنهادی $X^* \in \mathbb{R}^d$ را تولید می‌کند که به‌طور توأم این بردار پیشنهادی X^* پذیرفته یا رد می‌شود. موقعیتی را در نظر بگیرید که در روش نمونه‌گیری u_{block} ، مقدار پیشنهادی X^* از توزیع $N_d(\mu, \sigma_d^2 \Sigma)$ تولید شود، که در آن μ و Σ بر حسب وضعیت جاری زنجیر MCMC و ویژگی‌های توزیع مانای هدف، تغییر می‌کنند، ولی متناسب با d تغییر نمی‌کنند. رابرتز و همکاران (۱۹۹۷) نشان دادند برای دستیابی به نرخ پذیرش بهینه جانبی، و در نتیجه دستیابی به آمیختگی بهینه زنجیر نمونه، باید $\sigma_d^2 \propto d^{-1}$. با این نتیجه نظری، نرخ که به ازای آن الگوریتم Ψ_{block} فضای $\mathbb{R}^d$ را به خوبی کاوش کند و آمیختگی خوبی داشته باشد، متناسب است با d^{-1} . با همین استدلال معیار $A(\Psi_{block})$ متناسب است با d^{-1} .

این نتیجه، به‌طور هم‌ارز، برای روش‌های نمونه‌گیری بلوکی $\psi = u_{block}(b)$ که بر روی زیرمجموعه‌های $b \subset \Theta$ عمل می‌کنند، قابل تعمیم است. در واقع، کارایی الگوریتمی (برای عناصر b) که با استفاده از ψ به‌دست می‌آید، متناسب است با عکس تعداد عناصر موجود در b . این نتیجه یک اثر مهم روی نمونه‌گیری بلوکی دارد:

نتیجه ۱.۲. در مقایسه الگوریتم‌های بلوکی و مولفه به مولفه، در صورتی که تمام شرایط (مثل اثر همبستگی‌های پسینی) یکسان باشند، برای داشتن حجم نمونه موثر مشابه با نمونه‌های تولیدشده توسط نمونه‌گیری مولفه به مولفه، یک روش نمونه‌گیری بلوکی با بعد k باید k برابر نمونه بیشتر تولید کند (رابرتز و روزنتال، ۲۰۰۱). این ناکارایی به‌طور ذاتی در روش‌های نمونه‌گیری بلوکی وجود دارد و میزان آن تابعی از بعد هر نمونه‌گیر بلوکی است.

برای توضیح بیشتر، به مثال زیر توجه کنید.

مثال ۲.۲. فرض کنید $X = (X_1, \dots, X_d)'$. الگوریتم مولفه به مولفه Ψ_1 را به صورت زیر

تعریف می‌کنیم:

$$\begin{aligned}\Psi_1 &= X_1^{(t)} | X_2^{(t-1)}, \dots, X_d^{(t-1)} \\ &X_2^{(t)} | X_1^{(t)}, X_3^{(t-1)}, \dots, X_d^{(t-1)} \\ &\vdots \\ &X_d^{(t)} | X_1^{(t)}, X_2^{(t)}, \dots, X_{d-1}^{(t)}.\end{aligned}$$

به سادگی، بنا به نتیجه قبلی، می‌توان نوشت $A(\Psi_1) \propto 1$. حال اگر الگوریتم بلوکی Ψ_2 را به صورت زیر تعریف کنیم:

$$\begin{aligned}\Psi_2 &= (X_1, X_2)^{(t)} | (X_3, \dots, X_d)^{(t-1)} \\ &\vdots \\ &(X_{d-1}, X_d)^{(t)} | (X_1, \dots, X_{d-2})^{(t)}\end{aligned}$$

خواهیم داشت $A(\Psi_2) \propto \frac{1}{2}$. بنابراین $\text{ESS}(\Psi_2) = 2 \text{ESS}(\Psi_1)$. در نتیجه هر چقدر بعد بردار پارامترها بزرگ‌تر شود، بهتر است تعداد بلوک‌ها نیز بیشتر شود تا کارایی کمتری از بین برود.

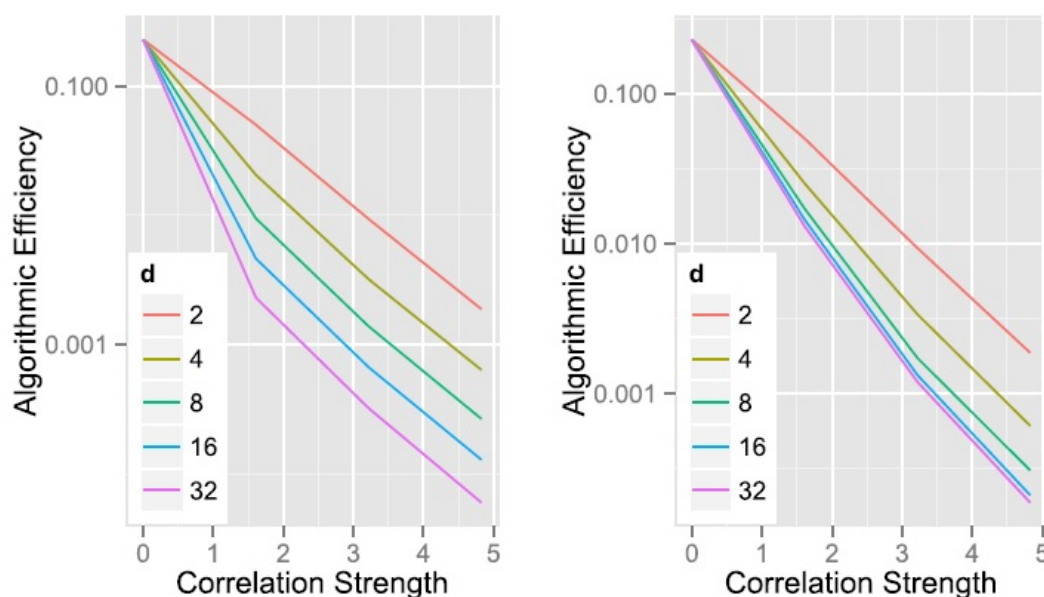
۲.۳.۲ کارایی از دست رفته نمونه‌گیری مولفه به مولفه

برای مطالعه حجم کارایی الگوریتمی که ممکن است در یک الگوریتم نمونه‌گیری مولفه به مولفه تحت $\Psi_{\text{scalar}} \in \Psi_{\mathcal{M}}$ از بین برود، در این بخش توزیع‌های پسین همبسته را در نظر می‌گیریم. در تحقیقات مختلف به خوبی درک شده است که همبستگی قوی پسینی مولفه‌های توزیع پسین می‌تواند باعث کند شدن سرعت همگرایی نمونه‌گیری MCMC شود (رابرتز و ساهو، ۱۹۹۷؛ گیلکس، ۲۰۰۵). در این مواقع، نمونه‌گیری بلوکی می‌تواند اثر چنین مشکلی را کاهش داده و سرعت همگرایی را افزایش دهد. البته ماهیت این ناکارایی، به‌ویژه به‌عنوان تابعی از شدت همبستگی و تعداد مولفه‌هایی که با هم همبسته هستند، مشخص نیست. برای بررسی نحوه عملکرد این عوامل بر کارایی الگوریتمی، در این بخش، یک مطالعه شبیه‌سازی انجام شده است.

فرض کنید توزیع هدف $N_d(\circ, \Sigma)$ باشد که در آن ماتریس کوواریانس Σ (که در این جا همان ماتریس همبستگی هم هست) شامل مقادیر ۱ روی قطر اصلی و مقادیر ρ ، با $|\rho| < 1$ ، خارج از قطر اصلی است. در این مثال، کارایی الگوریتمی پارامترهای تکی مدل تحت نمونه‌گیری مولفه به مولفه، یعنی $A(\Psi_{\text{scalar}}, \theta)$ ، برای $\theta \in \Theta$ مد نظر است.

شکل ۱.۲ (سمت چپ) نتایج حاصل از شبیه‌سازی را نمایش می‌دهد. منحنی افقی مقادیر $-\log(1 - \rho)$ را نشان می‌دهد. مقادیر مثبت در این محور نشان‌دهنده آن است که ρ به‌طور

نمایی به کران $\rho = 1$ در حال نزدیک شدن است یا این که همبستگی بین پارامترها کامل است. به طور تجربی، مشاهده می کنید که وقتی ρ به سمت یک میل می کند یا d بزرگ می شود، $A(\Psi_{scalar}, \theta)$ به صفر نزدیک می شود. این رفتار در نمودار سمت چپ شکل ۱.۲ (که مقادیر همبستگی و کارایی نتیجه شده بر حسب مقیاس لگاریتم- لگاریتم رسم شده اند) مشهود است. همان طور که انتظار داشتیم، زمانی که $\rho = 0$ ، تمام مقادیر d کارایی الگوریتمی یکسانی را نتیجه می دهند. اما زمانی که $\rho > 0$ ، معیار $A(\Psi_{scalar}, \theta)$ به صورت نمایی به سمت صفر نزول می کند. حتی برای d ثابت، کارایی الگوریتمی تحت Ψ_{scalar} چنان چه ρ به یک میل کند، می تواند ضعیف باشد.



شکل ۱.۲: کارایی الگوریتمی MCMC برای مقادیر متفاوتی از بعد مدل (d) و همبستگی درون گروهی ρ . مقادیر $-\log(1-\rho)$ بر روی محور افقی رسم شده اند. ساختار ماتریس کوواریانس مدل شامل عناصر خارج قطر اصلی ثابت برابر ρ (سمت چپ) و همبستگی های کاهشی- نمایی $\rho^{|i-j|}$ (سمت راست) است.

مثال شبیه سازی اول در بالا، یک حالت فرین است که در آن فرض کردیم توزیع هدف دارای بلوک های بزرگ دلخواه با همبستگی های جفتی دلخواه بالا بین مولفه های آن ها است. به عنوان مثال دوم، همان توزیع نرمال چندمتغیره d بعدی را در نظر بگیرید که عناصر ماتریس کوواریانس Σ در آن به صورت $\sigma_{ij} = \rho^{|i-j|}$ ، با $|\rho| < 1$ ، هستند. این ساختار کوواریانسی می تواند برای داده های سری زمانی یا داده های فضایی رخ دهد (بنرجی و همکاران، ۲۰۰۳). کارایی الگوریتمی اولین پارامتر مدل در شکل ۱.۲ (سمت راست) نشان داده شده است. همان طور که از این شکل نیز مشهود است، نتیجه ای مشابه مثال اول به دست آمده است.

با ترکیب مثال‌های نظری و شبیه‌سازی، مشاهده می‌شود که کارایی الگوریتمی به دست آمده تحت Ψ_{scalar} یا Ψ_{block} ، به‌طور نه چندان واضحی، به بعد مدل و ساختار همبستگی پسین وابسته است. در مثال‌های شبیه‌سازی فقط دو ساختار همبستگی ساده و به‌شدت سیستماتیک در نظر گرفته شدند. اما در عمل، هر مدل دلخواه می‌تواند دارای یک ساختار همبستگی پیچیده (و البته نامعلوم) باشد. این مساله یک مطالعه کامل تحلیلی از کارایی الگوریتمی روش‌های MCMC را دشوار می‌سازد. در این‌جا فقط دو الگوریتم فرین در مجموعه همه الگوریتم‌های Ψ_M را در نظر گرفتیم، یعنی Ψ_{block} و Ψ_{scalar} . اما دلخواه آن است که بتوان نتایج کلی برای کارایی الگوریتمی را برای مجموعه کامل Ψ_M به دست آورد که کار پیچیده و مشکلی است.

۴.۲ کارایی محاسباتی

در بخش ۳.۲ کارایی الگوریتمی MCMC الگوریتم Ψ را برای تولید نمونه‌های مستقل، بدون توجه به زمان محاسبات، بررسی کردیم. در این بخش، کارایی محاسباتی مورد نظر ماست که بر اساس مدت زمان اجرای الگوریتم برای تولید هر نمونه MCMC محاسبه می‌شود. برای محاسبه زمان اجرای الگوریتم، فقط بر زمان مصرف‌شده برای ارزیابی تابع چگالی مدل تمرکز می‌کنیم. بخش‌های دیگری، مانند زمان تکرار هر حلقه، نیز در ثبت زمان لازم برای اجرای الگوریتم دخیل هستند که چون بخش ناچیزی از زمان اجرای کل را شامل می‌شوند، از آن‌ها صرف نظر می‌کنیم. زمان محاسبات برای الگوریتم Ψ را با $C(\Psi)$ نشان می‌دهیم. به‌طور مشابه اگر فقط زمان به‌روزرسانی یک مولفه (پارامتر) مد نظر باشد، از نماد $C(\psi)$ برای هزینه لازم برای نمونه‌گیر تکی ψ استفاده می‌کنیم. بنابراین، برای $\Psi = \{\psi_1, \dots, \psi_k\}$ داریم

$$C(\Psi) = \sum_{i=1}^k C(\psi_i).$$

پس معیار سنجش کارایی محاسباتی یک الگوریتم را با $C(\Psi)$ نشان می‌دهیم که با زمان اجرای به‌روزرسانی همه بلوک‌های آن مرتبط است. بنا به ادعای تورک و همکاران (۲۰۱۷) تا به امروز بر روی کارایی الگوریتم‌های MCMC از منظر محاسباتی و زمان اجرا هیچ تحلیلی صورت نگرفته است و معیار خاصی معرفی نشده است. در متونی که در مورد کارایی الگوریتم‌های MCMC بحث شده است، فقط از وجود ویژگی‌های محاسباتی صحبت شده و بعد به شکلی به معیار $A(\Psi)$ پرداخته‌اند. به‌عنوان نمونه می‌توان رابرتز و ساهو (۱۹۹۷) را نام برد.

اکنون برای هر مدل M سهم $C(\Psi_{scalar})$ و $C(\Psi_{block})$ را محاسبه می‌کنیم. برای این منظور، ابتدا چند نماد را معرفی می‌کنیم. مجموعه همه مولفه‌های ثابت و معلوم مدل M (مثل مشاهدات) را با Y نشان می‌دهیم. این مجموعه از مجموعه پارامترهای نامعلوم مدل، یعنی $\Theta = \{\theta_1, \dots, \theta_d\}$ جدا است. برای هر $\theta_i \in \Theta$ فرض کنید $x_i \in \Theta \cup Y$ مجموعه‌ای از

مولفه‌های مدل باشد که به نوعی به θ_i وابسته هستند یا مجموعه‌ای از اجزای ثابت θ_i در مدل هستند. همچنین هزینه محاسباتی ارزیابی توابع چگالی متناظر با هر زیرمجموعه $x \in \Theta \cup Y$ را با $\text{dens}(x)$ نشان می‌دهیم.

۱.۴.۲ محاسبات روش نمونه‌گیری مولفه به مولفه

در هر تکرار از $\Psi_{\text{scalar}} = \{\psi_1, \dots, \psi_d\}$ ، هزینه محاسباتی هر ψ_i برابر است با

$$C(\psi_i) = \text{dens}(\theta_i) + \text{dens}(x_i) + O(1).$$

مقدار ثابت $O(1)$ بیانگر زمان لازم برای تولید مقدار پیشنهادی و تصمیم‌گیری MH است، که به‌ترتیب شامل تولید نمونه از توزیع‌های نرمال و یکنواخت است. بنابراین، مجموع زمان کل محاسبات مورد نیاز برای الگوریتم Ψ_{scalar} برابر است با

$$C(\Psi_{\text{scalar}}) = \sum_{i=1}^d C(\psi_i) = \sum_{i=1}^d \text{dens}(\theta_i) + \sum_{i=1}^d \text{dens}(x_i) + O(d). \quad (1.2)$$

توجه داشته باشید که تحت Ψ_{scalar} ارزیابی هر چگالی $\text{dens}(\theta_i)$ باید به‌طور مستقل انجام شود. این معادله حتی برای زمانی که محاسبه چگالی θ_i مستلزم محاسبه زیرمجموعه‌ای از یک چگالی چندمتغیره (حتی در حالت فرین $\text{dens}(\Theta)$) باشد، نیز برقرار است. بنابراین، در بدترین حالت، یک تکرار MCMC تکی از Ψ_{scalar} باید d ارزیابی از کل چگالی توام Θ انجام دهد. یک حجم محاسباتی بالای مشابه می‌تواند به واسطه محاسبه هر مولفه $\text{dens}(x_i)$ نتیجه شود.

پس معادله (۱.۲) هم برای زمانی که مولفه‌ها مستقل از هم باشند و هم زمانی که به یک‌دیگر وابسته هستند و هم زمانی که بعضی از آن‌ها به‌صورت بلوکی به هم وابسته هستند، برقرار است.

۲.۴.۲ محاسبات روش نمونه‌گیری بلوکی

اکنون مولفه‌های $C(\Psi_{\text{block}})$ را در نظر می‌گیریم. در هر تکرار از Ψ_{block} ، اجرای نمونه‌گیر $u_{\text{block}}(\Theta)$ نیازمند ارزیابی کامل چگالی‌های پیشین و درست‌نمایی $\text{dens}(\Theta \cup Y)$ است. محاسبه و ارزیابی این چگالی توام کاملاً متفاوت از ارزیابی چگالی‌هایی است که در $C(\Psi_{\text{scalar}})$ ظاهر می‌شوند و تنها یک محاسبه تکی از چگالی توام مدل کامل را شامل می‌شود. علاوه بر این سازواری $u_{\text{block}}(\Theta)$ نیازمند یک تجزیه چولسکی^{۱۶} از ماتریس کوواریانس سازوار است که تعداد عملیات لازم برای این تجزیه برابر $O(d^3)$ است (ترفتن و بائو، ۱۹۹۷). این تجزیه در هر AI تکرار باید انجام شود، به‌طوری که AI فاصله سازوار^{۱۷} u_{block} است و بنابراین هزینه محاسباتی به $O(\frac{d^3}{\text{AI}})$ تبدیل می‌شود. هر تکرار u_{block} نیازمند تولید یک بردار پیشنهادی از نرمال چندمتغیره d

^{۱۶}Cholesky factorization

^{۱۷}Adaptation interval

بعدی است که خود به تعداد $O(d^2)$ عملیات نیاز دارد. همچنین زمان لازم برای تصمیم‌گیری MH قابل اشاره است که می‌توان آن را نادیده گرفت. بنابراین، مجموع محاسبات مورد نیاز Ψ_{block} به صورت زیر است:

$$C(\Psi_{block}) = \text{dens}(\Theta \cup Y) + O\left(\frac{d^3}{AI}\right) + O(d^2).$$

۳.۴.۲ مقایسه زمانی

تا این جا می‌توان گفت هزینه محاسبات برای الگوریتم Ψ_{block} حداقل صورتی درجه دو از d است و به طور نظری $O(d^3)$ است که البته ضریب آن کوچک است. بسته به ساختار توزیعی Θ ، محاسبه چگالی $C(\Psi_{scalar})$ به طور نظری می‌تواند بسیار دشوار باشد و چون هیچ نتیجه نظری در مورد بزرگی نسبی این مولفه‌ها وجود ندارد، برای داشتن یک شهود عملی، در قالب یک مثال شبیه‌سازی، به بررسی زمان لازم برای اجرای Ψ_{block} و Ψ_{scalar} پرداختیم. در مطالعه شبیه‌سازی سه مدل زیر در نظر گرفته شدند که فقط ساختار پیشین بر روی Θ را مشخص می‌کنند:

$$۱) \theta_i \sim N(\mu, \sigma); \quad \theta_i \in \Theta$$

$$۲) \theta_i \sim \text{Gamma}(\alpha, \beta); \quad \theta_i \in \Theta$$

$$۳) \Theta \sim N_d(\mu, \Sigma).$$

شکل ۱.۳ زمان محاسبات را برای هر ۱۰۰۰۰ تکرار بر حسب ثانیه در مقابل بعد d نشان می‌دهد. این محاسبات بدون در نظر گرفتن کارایی الگوریتمی محاسبه شده‌اند. نتایج حاصل می‌توان به اختصار به شرح زیر ارایه کرد:

- زمانی که هر θ_i به طور مستقل از یک توزیع یک متغیره پیروی می‌کند و

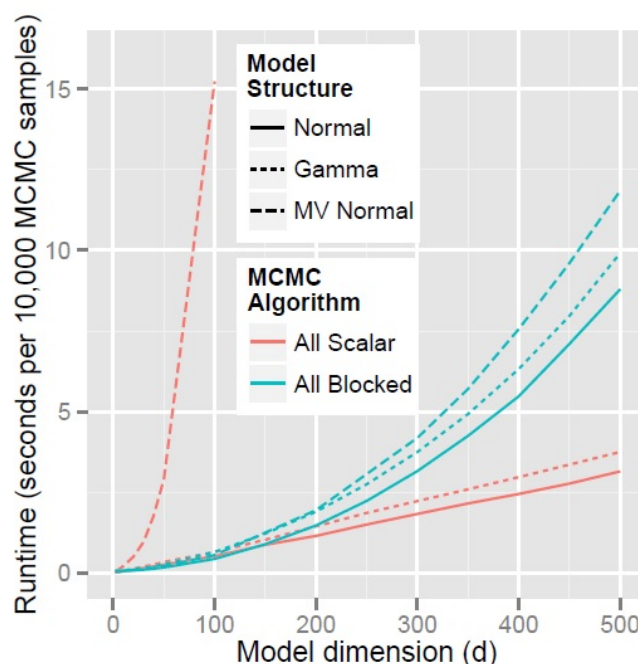
$$\sum_{i=1}^d \text{dens}(\theta_i) = \text{dens}(\Theta) \leq dK; \quad K = \max_{\theta \in \Theta} \text{dens}(\theta)$$

$C(\Psi_{scalar})$ همان $O(d)$ است.

- برای تمام اهداف کاربردی، به نظر می‌رسد $C(\Psi_{block})$ برابر $O(d^2)$ است؛ زیرا ضریب مرتبه d^3 ، برابر $1/AI = 1/2^{10}$ است که نسبتاً عدد کوچکی است.

- هزینه محاسباتی $C(\Psi_{block})$ نسبت به تغییرات توزیع زیربنایی Θ حساس نیست و تغییر چندانی نمی‌کند. اما در مقابل این تغییر توزیع مقدار $C(\Psi_{scalar})$ را به شدت برای حالت چندمتغیره (وابسته) تغییر می‌دهد. در واقع برای زمانی که توزیع زیربنایی Θ چندمتغیر است، $C(\Psi_{scalar})$ برابر $O(d^3)$ است. دلیل آن هم این است که محاسبه چگالی نرمال چندمتغیر دارای هزینه $O(d^2)$ است که d بار برای هر تکرار Ψ_{scalar} رخ می‌دهد. با

این که برای هر دو الگوریتم هزینه محاسباتی برابر $O(d^3)$ است، ولی $C(\Psi_{scalar})$ توسط $C(\Psi_{block})$ مغلوب می‌شود؛ دلیل آن هم اختلاف در ضریب مرتبه سوم برای آن‌ها است. همان‌طور که انتظار داشتیم، محاسبه چگالی‌های یک متغیره گاما نسبت به یک متغیره نرمال کمی هزینه بیشتری دارند. همچنین در این موارد هزینه محاسباتی الگوریتم تک بلوکی بیشتر از الگوریتم مولفه به مولفه است.



شکل ۲.۲: زمان اجرای MCMC برای الگوریتم‌های تک بلوکی و مولفه به مولفه، برای ساختارهای مدل نرمال یک متغیره و گامای یک متغیره و نرمال چندمتغیره.

۵.۲ کارایی کلی

تا این‌جا دو معیار کارایی الگوریتمی و محاسباتی را برای الگوریتم‌های MCMC تعریف و آن‌ها را مورد بررسی قرار دادیم که هر کدام از آن‌ها بر کارایی کل یک الگوریتم MCMC تاثیر دارند. اکنون یک معیار کارایی کل Ψ را می‌توان به‌صورت

$$E(\Psi) = \frac{A(\Psi)}{C(\Psi)} \quad (۲.۲)$$

تعریف کرد. معیار $E(\Psi)$ برای الگوریتم‌های MCMC، اندازه‌ای منطقی از کارایی کل است. زیرا این معیار می‌تواند به‌عنوان تعداد نمونه‌های موثر تولیدشده به ازای هر واحد از زمان اجرای الگوریتم MCMC، برای پارامترهایی از مدل که دارای کندترین آمیختگی هستند، تعبیر شود.

اگر بتوان الگوریتم Ψ ی ساخت که $E(\Psi)$ را ماکسیمم کند، آن‌گاه Ψ کاراترین الگوریتم MCMC بر حسب زمان برای تولید نمونه‌های به‌طور موثر مستقل است که می‌توان از آن‌ها برای تقریب توزیع واقعی توام پسین θ استفاده کرد.

همان‌طور که گفته شد بین دو کران مجموعه الگوریتم‌های MCMC، بی‌شمار الگوریتم دیگر وجود دارند. برای پیدا کردن بهترین الگوریتم به‌جای استفاده از یک مسأله ترکیباتی، در فصل بعدی، یک روش خودکار تکراری معرفی می‌کنیم که با ماکسیمم کردن معیار کارایی کل E همراه است. برای این هدف، بلوک‌بندی بردار پارامترهای θ حائز اهمیت است چرا که با معرفی یک الگوریتم بلوک‌بندی، کارایی کل را ماکسیمم می‌کنیم نه لزوماً به‌صورت تک مولفه‌ای یا تک بلوکی.

فصل ۳

بلوک‌بندی خودکار

۱.۳ مقدمه

در فصل ۲ برای الگوریتم‌های MCMC عمومی دو معیار کارایی الگوریتمی و کارایی محاسباتی را معرفی کردیم. سپس با ترکیب آن‌ها یک معیار کارایی کلی تعریف شد. عملکرد معیارهای پیشنهادی را برای دو الگوریتم MCMC تولید نمونه مولفه به مولفه و همه در یک بلوک (توام) ارزیابی کردیم. این دو الگوریتم دو عضو فرین در خانواده الگوریتم‌های MCMC هستند، در حالی که بین این دو، الگوریتم‌های بلوکی متعدد دیگری وجود دارند.

اگر برای تولید نمونه از توزیع پسین، هدف، انتخاب بهترین الگوریتم در رده الگوریتم‌های MCMC باشد، بهترین الگوریتم، الگوریتمی است که معیار کارایی کلی E در رابطه (۲.۲) را ماکسیمم کند. معیار E برای دو الگوریتم فرین مولفه به مولفه و توام قابل محاسبه است ولی ماکسیمم کردن کارایی کل برای یک الگوریتم مفروض عمومی یک مسأله ترکیباتی است که حل آن بسیار دشوار است. در این فصل یک الگوریتم خودکار بازگو می‌شود به‌طوری که $E(\Psi)$ را برای یک Ψ ایی که در رده همه الگوریتم‌های MCMC قرار می‌گیرد، ماکسیمم کند. این الگوریتم توسط تورک و همکاران (۲۰۱۷) معرفی شده است.

در این فصل، برای تولید یک الگوریتم MCMC کارا روشی خود-تنظیم^۱ برای بلوک‌بندی

^۱Self-tuning

خودکار^۲ پارامترهای مدل‌های سلسله‌مراتبی معرفی می‌کنیم که در این روش از همبستگی پسین تجربی پارامترهای همبسته گروه‌های خوشه‌ای در نمونه‌گیری بلوکی استفاده می‌شود. یک درخت خوشه‌بندی^۳ سلسله‌مراتبی از پارامترهای مدل تشکیل می‌شود که با برش این درخت در هر ارتفاعی، پارامترهای موجود در هر بلوک به‌دست می‌آیند به‌طوری‌که بلوک‌ها دارای کمترین همبستگی هستند. این روش تا زمانی تکرار می‌شود که کارایی الگوریتم MCMC بهبود یابد و همبستگی پسین تجربی دقیق‌تر شود و در نتیجه درخت و پارامترهای گروه‌ها به سطح پایداری برسند. در نتیجه یک افراز از پارامترهای مدل، یک الگوریتم MCMC منحصر به‌فرد مثل $\Psi \in \psi_M$ تولید می‌شود که در نمونه‌گیری بلوکی به‌کار گرفته می‌شود و کارایی کلی $E(\Psi)$ را تا حد امکان افزایش می‌دهد. روشی که در این فصل بازگو می‌کنیم در بعضی از موارد امکان افزایش کارایی را فراهم می‌کند و روشی پایدار را ایجاد می‌نماید (تورک و همکاران، ۲۰۱۷).

با توجه به استفاده از نظریه خوشه‌بندی برای ساخت الگوریتم تکراری پیشنهادی، ابتدا به‌طور خلاصه مباحث مربوط به خوشه‌بندی را مطرح می‌کنیم که عمدتاً برگرفته از اسماعیلی (۱۳۹۳) است.

۲.۳ خوشه‌بندی

تحلیل خوشه‌ای یا به‌صورت ساده خوشه‌بندی، فرآیندی است که در آن مجموعه‌ای از اشیاء (یا مشاهدات) به زیرمجموعه‌هایی تقسیم می‌شوند. هر یک از زیرمجموعه‌ها یک خوشه نامیده می‌شود و اشیاء هر خوشه به یک‌دیگر شبیه و متفاوت با اشیاء سایر خوشه‌ها هستند. البته ممکن است روش‌های خوشه‌بندی متفاوت، خوشه‌های مختلفی را برای داده‌های یکسان تولید کنند.

تحلیل خوشه‌ای به‌صورت گسترده‌ای در بسیاری از حوزه‌ها مانند هوش تجاری، تشخیص الگو، جست‌وجوی وب، و زیست‌شناسی استفاده می‌شود. با کمک خوشه‌بندی در هوش تجاری می‌توان مشتریان را به گروه‌هایی تقسیم کرد که هر گروه دارای مشخصات مشترکی هستند. همچنین خوشه‌بندی در جست‌وجوی وب نیز دارای کاربردهای فراوانی است. برای مثال اغلب جست‌وجو با کمک واژه‌های کلیدی در وب ممکن است نتایج فراوانی را برگرداند. با کمک خوشه‌بندی می‌توان این نتایج را به درون گروه‌هایی سازمان‌دهی کرد و بدین روش نتایج به‌صورت مختصر ارائه می‌شوند و دسترسی به آن‌ها نیز ساده‌تر می‌شود. به‌علاوه با توسعه تکنیک‌های خوشه‌بندی، اسناد را می‌توان بر اساس عناوین آن‌ها دسته‌بندی کرد. خوشه‌بندی اسناد بر اساس موضوع و عنوان در حوزه بازیابی اطلاعات یک کاربرد معمول از روش‌های خوشه‌بندی است.

^۲Automated blocking

^۳Clustering tree

تعریف خوشه‌بندی

یک خوشه به مجموعه‌ای از داده‌ها اطلاق می‌شود که از جهاتی شبیه به هم‌دیگر هستند. الگوریتم‌های خوشه‌بندی به‌طور خودکار ویژگی‌های متمایزکننده زیرگروه‌ها را تعریف و آن‌ها را سازماندهی می‌کنند. آن‌ها مدل را منحصر از روی روابطی که در داده‌ها وجود دارند، تشخیص می‌دهند.

خوشه‌بندی در واقع یافتن ساختار در مجموعه‌ای از داده‌ها است که طبقه‌بندی نشده‌اند. به بیان دیگر می‌توان گفت که خوشه‌بندی قرار دادن داده‌ها در گروه‌هایی است که اعضای هر گروه از زاویه خاصی به یک‌دیگر شباهت دارند و با اعضای سایر خوشه‌ها شباهتی ندارند. معیار شباهت در این‌جا فاصله بوده و نحوه محاسبه این معیار در خوشه‌بندی بسیار مهم است. با محاسبه فاصله بین دو داده می‌توان پی برد تا چه اندازه دو داده به هم نزدیک هستند و بر این اساس می‌توان آن‌ها را در یک خوشه یا خوشه‌های متفاوت قرار داد.

اهداف خوشه‌بندی

خوشه‌بندی را می‌توان به‌عنوان مهم‌ترین روش در یادگیری بدون نظارت^۴ در نظر گرفت. در خوشه‌بندی سعی بر آن است تا داده‌ها به خوشه‌هایی تقسیم شوند که شباهت بین داده‌های درون هر خوشه بیشینه و شباهت بین داده‌ها در خوشه‌های متفاوت کمینه شود. خوشه‌بندی معمولاً با دو هدف انجام می‌شود:

- فهم^۵: بررسی گروه‌های مشابه در مجموعه داده‌ها؛ مانند اسناد مرتبط با هم، گروه‌های ژنی و پروتئینی که دارای کاربرد یکسانی هستند، و گروه سهام با نوسانات قیمت مشابه.
- خلاصه‌سازی^۶: کاهش اندازه مجموعه داده‌های بزرگ.

اگر بخواهیم خوشه‌بندی را در مقابل دسته‌بندی^۷ مورد مقایسه قرار دهیم، در دسته‌بندی هر داده به یک طبقه (دسته) از پیش مشخص تخصیص می‌یابد ولی در خوشه‌بندی هیچ اطلاعی از خوشه‌ها و ساختار آن‌ها وجود ندارد و به عبارتی خود خوشه‌ها نیز از داده‌ها استخراج می‌شوند.

۱.۲.۳ انواع روش‌های خوشه‌بندی

الگوریتم‌های خوشه‌بندی متعددی وجود دارند که تقریباً می‌توان هر کدام از آن‌ها را در یکی از دسته‌های زیر قرار داد:

^۴ Learning without supervisor

^۵ Understanding

^۶ Summarization

^۷ Classification

- روش‌های افراز^۸
- روش‌های خوشه‌بندی سلسله‌مراتبی^۹
- روش‌های خوشه‌بندی مبتنی بر چگالی^{۱۰}
- روش‌های خوشه‌بندی فازی
- روش‌های خوشه‌بندی مبتنی بر مدل^{۱۱}

با توجه به استفاده از خوشه‌بندی سلسله‌مراتبی در این پایان‌نامه، در ادامه، این روش‌های خوشه‌بندی را معرفی می‌کنیم.

۲.۲.۳ روش‌های خوشه‌بندی سلسله‌مراتبی

در یک روش خوشه‌بندی سلسله‌مراتبی یک تجزیه سلسله‌مراتبی از مجموعه اشیاء ایجاد می‌شود. این روش‌ها بر اساس این که این تجزیه سلسله‌مراتبی را چگونه شکل می‌دهند، به دو دسته تجمیعی^{۱۲} و تقسیمی^{۱۳} گروه‌بندی می‌شوند. در نوع تجمیعی که با نام رویکرد پایین به بالا^{۱۴} نیز شناخته می‌شود، این روش هر شی را در یک گروه یا خوشه قرار می‌دهد و کار خود را آغاز می‌کند. در ادامه اشیاء و گروه‌هایی که به یک‌دیگر نزدیک هستند، ادغام می‌شوند و این کار تا جایی که همه اشیاء و گروه‌ها در یک گروه واحد قرار گیرند، ادامه می‌یابد (بالاترین سطح از سلسله‌مراتب). البته در این نوع، می‌توان با کمک یک شرط پایانی نیز الگوریتم را متوقف کرد. از انواع الگوریتم‌های خوشه‌بندی سلسله‌مراتبی تجمیعی رایج می‌توان از الگوریتم‌های پیوند میانگین^{۱۵}، پیوند تکی^{۱۶} و پیوند کامل^{۱۷} نام برد.

در نوع تقسیمی که رویکرد بالا به پایین^{۱۸} نیز خوانده می‌شود، در ابتدا همه اشیاء در یک خوشه قرار می‌گیرند. در هر تکرار، یک خوشه به خوشه‌های کوچک‌تر شکسته می‌شود. این کار تا زمانی که هر شی به تنهایی در یک خوشه قرار بگیرد یا یک شرط پایانی رخ دهد، ادامه پیدا می‌کند. روش‌های خوشه‌بندی سلسله‌مراتبی می‌توانند مبتنی بر فاصله باشند یا بر اساس چگالی و پیوند اشیاء عمل کنند. تفاوت اصلی در بین تمام این روش‌ها به نحوه محاسبه

^۸Partition

^۹Hierarchical clustering

^{۱۰}Density-based clustering

^{۱۱}Model-based clustering

^{۱۲}Agglomerative

^{۱۳}Divisive

^{۱۴}Bottom-up

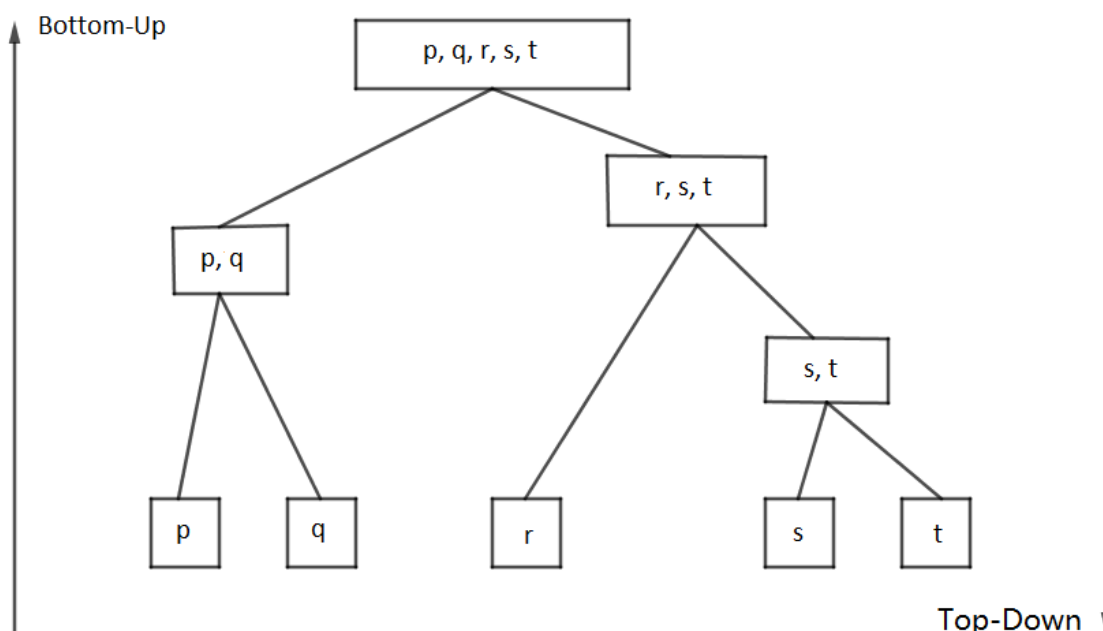
^{۱۵}Average-link

^{۱۶}Single-link

^{۱۷}Complete-link

^{۱۸}Top-down

شباهت بین خوشه‌ها مربوط می‌شود، که در ادامه به تشریح هر یک از آن‌ها می‌پردازیم. شکل ۱.۳ نحوه اجرای روش خوشه‌بندی سلسله‌مراتبی رو با دو نوع پایین به بالا و بالا به پایین، تشریح می‌کند



شکل ۱.۳: نحوه اجرا دو روش خوشه‌بندی سلسله‌مراتبی پایین به بالا و بالا به پایین.

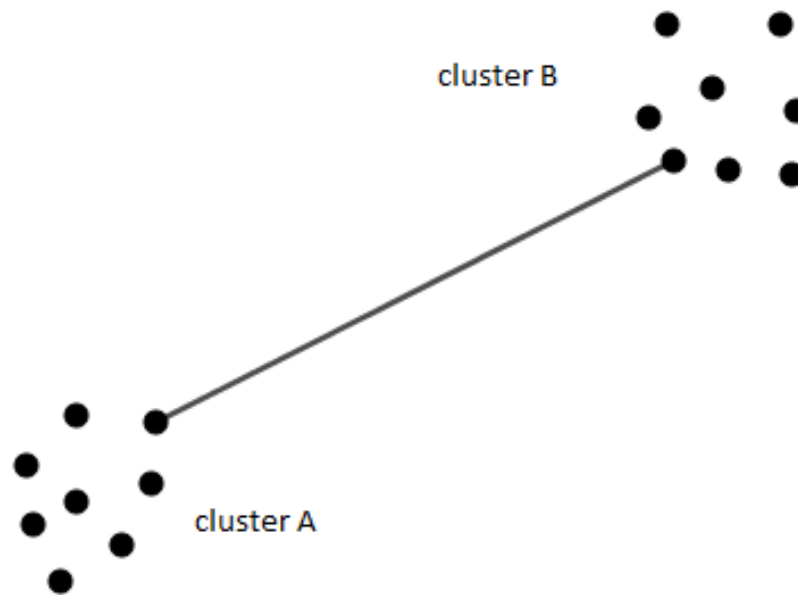
خوشه‌بندی با روش پیوند تکی

این روش یکی از قدیمی‌ترین و ساده‌ترین روش‌های خوشه‌بندی است و جزء روش‌های خوشه‌بندی سلسله‌مراتبی و تجمیعی محسوب می‌شود. به این روش خوشه‌بندی، تکنیک نزدیک‌ترین همسایه^{۱۹} نیز گفته می‌شود. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \min_{i \in A, j \in B} d_{ij}$$

که در آن i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B است. در واقع در این روش شباهت بین دو خوشه، کم‌ترین فاصله بین یک عضو از یکی با یک عضو از دیگری است. در شکل ۲.۳ این مفهوم به‌خوبی نشان داده شده است.

^{۱۹}Nearest neighbour



شکل ۲.۳: شباهت بین دو خوشه در روش پیوند تکی برابر است با کمترین فاصله بین داده‌های دو خوشه

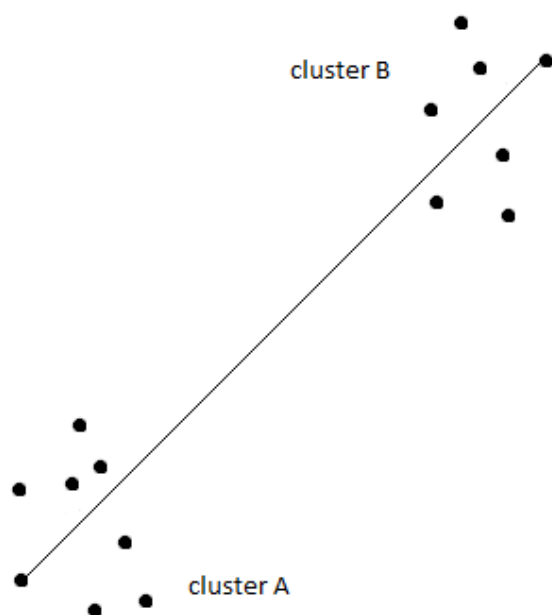
خوشه‌بندی با روش پیوند کامل

این روش همانند پیوند تکی از روش‌های خوشه‌بندی سلسله‌مراتبی و تجمیعی محسوب می‌شود. به این روش خوشه‌بندی، تکنیک دورترین همسایه^۲ نیز گفته می‌شود (وب، ۲۰۰۳). در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \max_{i \in A, j \in B} d_{ij}$$

که i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B است. در واقع در این روش شباهت بین دو خوشه بیشترین فاصله بین یک عضو از یکی با یک عضو از دیگری است. در شکل ۳.۳ این مفهوم توصیف شده است.

^۲Furthest neighbour



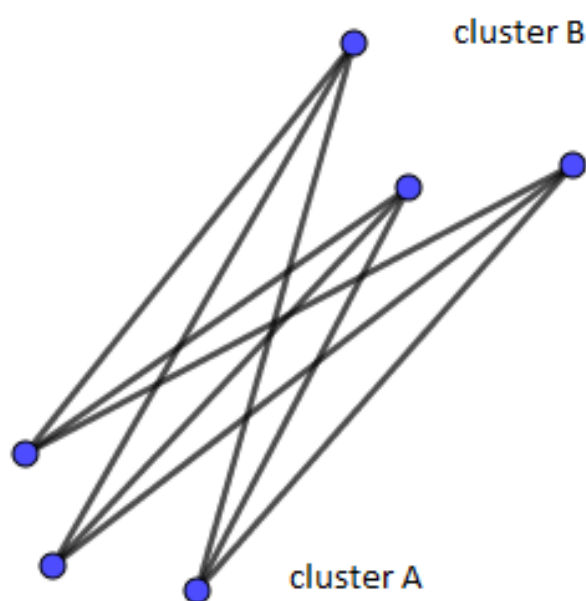
شکل ۳.۳: شباهت بین دو خوشه در روش پیوند کامل برابر است با بیشترین فاصله بین داده‌های دو خوشه

خوشه‌بندی با روش پیوند میانگین

این روش مشابه دو روش قبل از روش‌های خوشه‌بندی سلسله‌مراتبی و تجمیعی است. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \frac{\sum_{i \in A, j \in B} d_{ij}}{N_A N_B}$$

که در آن i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B است. همچنین N_A تعداد اعضای خوشه A و N_B تعداد اعضای خوشه B است. در واقع این روش، شباهت بین دو خوشه را بر اساس میانگین فاصله بین تمام اعضای یکی با تمام اعضای دیگری بیان می‌کند. شکل ۴.۳ عملکرد این روش را نمایش می‌دهد.



شکل ۴.۳: شباهت بین دو خوشه در روش پیوند میانگین برابر است با میانگین فاصله بین تمام اعضای یکی با تمام اعضای دیگری

خوشه‌بندی با روش Ward

این روش از روش‌های خوشه‌بندی سلسله‌مراتبی پایین به بالاست و در صورت وجود داده‌های دورافتاده از معیار Ward برای سنجش شباهت داده‌ها استفاده می‌کند. در روش Ward از مجموع توان‌های دوم تفاضل هر داده از یک خوشه با بردار میانگین آن خوشه، به‌عنوان معیاری برای سنجش شباهت در یک خوشه استفاده می‌شود. الگوریتم زیر را می‌توان برای اجرای روش Ward معرفی کرد (هی، ۱۹۹۹):

۱. ابتدا هر داده به‌عنوان یک خوشه در نظر گرفته می‌شود.
 ۲. به ازای تمام جفت خوشه‌های ممکن از مجموعه خوشه‌ها، آن دو خوشه‌ای که مجموع توان‌های دوم تفاضل داده‌های خوشه حاصل از اجتماع آن‌ها با بردار میانگین خوشه حاصل کمینه باشد، انتخاب می‌شوند.
 ۳. دو خوشه انتخاب‌شده با هم ترکیب می‌شوند.
 ۴. تا زمانی که تعداد خوشه‌ها به تعداد مورد نظر نرسیده است، مراحل ۲ و ۳ تکرار می‌شوند.
- با توجه به این که الگوریتم پیشنهادی در این پایان‌نامه یک الگوریتم MCMC سازوار^{۲۱} نتیجه می‌دهد، در ادامه، الگوریتم‌های سازوار را نیز معرفی می‌کنیم.

^{۲۱} Adaptive MCMC algorithm

۳.۳ الگوریتم‌های MCMC سازوار

الگوریتم‌های MCMC مانند الگوریتم متروپلیس – هستینگز به‌طور گسترده در استنباط آماری بیزی، برای نمونه‌گیری از توزیع‌های پیچیده با ابعاد بالا استفاده می‌شوند. تنظیم پارامترهای توزیع پیشنهادی مانند پارامتر مقیاس تابع چگالی پیشنهادی برای رسیدن به آمیختگی کارا بسیار دشوار است (رابرتز و روزنتال، ۲۰۰۹). الگوریتم‌های MCMC سازوار (هاریو و همکاران، ۲۰۰۱؛ رابرتز و روزنتال، ۲۰۰۵) برای مقابله با این دشواری تلاش می‌کنند تا به‌صورت خودکار مقادیر پارامتر متناظر بهتری از الگوریتم‌های MCMC معمولی، تولید کنند. الگوریتم‌های سازوار، توسعه‌یافته الگوریتم‌های MCMC هستند که در آن هسته انتقال^{۲۲} در هر مرحله مطابق عملکرد مشاهدات تنظیم می‌شود (رابرتز و روزنتال، ۲۰۰۵).

یک مشکل روش‌های سازوار، بیش از حد وابسته بودن آن‌ها به نمونه‌های قبلی است. رابرتز و روزنتال (۲۰۰۵) برای حل این مشکل یک شرط سازواری کاهنده^{۲۳} را پیشنهاد کردند که نشان می‌دهد مجموع فاصله تغییرات بین دو هسته متوالی باید به‌طور یکنواخت به صفر کاهش یابد. این کاهش مجموع فاصله به صفر به معنی همگرایی هسته نیست. برای مثال، فرآیند پیشنهادی قدم زدن تصادفی پیشنهادی که به واریانس تجربی نمونه وابسته شده است چنین ویژگی دارد (هاریو و همکاران، ۱۹۹۳). راه‌حل دیگری که رابرتز و روزنتال پیشنهاد کردند، تنظیم پارامتر به شکلی است که نرخ پذیرش بهینه نزدیک به ۰/۴۴ باشد. این راه‌حل در بسته mcmc (چارلز، ۲۰۰۹) در نرم‌افزار R پیاده شده است. در واقع در این روش برای هر مولفه از پارامتر شبیه‌سازی‌شده یک عامل δ_i مطابق با انحراف معیار فرآیند پیشنهادی قدم زدن تصادفی، هر ۵۰ تکرار را با اضافه کردن یا کم کردن یک عامل مستقل ϵ_t بسته به این که آیا متوسط نرخ پذیرش در آن دسته از ۵۰ تکرار برای این مولفه‌ها بالا یا پایین ۰/۴۴ است، به‌روز رسانی می‌کند. اگر ϵ_t به صفر نزول کند، آن‌گاه شرایط هم‌گرایی صادق است (رابرتز و روزنتال، ۲۰۰۹).

در این‌جا دو مثال از الگوریتم‌های سازوار که دارای بعد بالایی هستند را بررسی می‌کنیم: الگوریتم چندمتغیره متروپلیس سازوار از هاریو و همکاران (۲۰۰۱) و الگوریتم متروپلیس درون گیبز برای مدل‌های سلسله‌مراتبی نامزدوج^{۲۴}. با این دو مثال سعی می‌کنیم نشان دهیم الگوریتم‌های MCMC سازوار در تنظیم پارامتر مقیاس توزیع پیشنهادی بسیار موفق هستند و مداخله کاربر در آن‌ها کم است. خودهمبستگی کامل یک تابع دلخواه را می‌توان به‌عنوان معیاری مناسب برای آمیختگی زنجیر مارکوف معرفی کرد. در روزنتال (۲۰۰۴) و قضیه ۳ رابرتز و روزنتال (۲۰۰۵) می‌توان دید که الگوریتم‌های MCMC همیشه مانایی $\pi(\cdot)$ را حفظ نمی‌کنند. گیلکس و همکاران (۱۹۹۸)، و براکول و کدین (۲۰۰۵) شرایط مختلفی را فراهم کرده‌اند که

^{۲۲} Transition kernel

^{۲۳} Diminishing adaptive conditional

^{۲۴} Nonconjugate hierarchical models

نشان می‌دهد یک الگوریتم MCMC سازوار چه زمانی می‌تواند به توزیع مانای خود همگرا شود. همچنین رابرتز و روزنتال (۲۰۰۵) الگوریتم MCMC سازوار ارگودیک را تحت شرایطی نشان داده‌اند که در آن نیازی به همگرایی پارامترهای سازوار نیست.

الگوریتم متروپلیس سازوار

در این بخش حالتی از الگوریتم متروپلیس سازوار که توسط هاریو و همکاران (۲۰۰۱) معرفی شده است را بررسی می‌کنیم. ابتدا با یک توزیع هدف d بعدی $\pi(\cdot)$ شروع کرده و الگوریتم متروپلیس را با توزیع پیشنهادی Q_n ، n بار تکرار می‌کنیم. ماتریس Σ_n برآورد تجربی ساختار کوواریانس توزیع هدف مبتنی بر اجرای الگوریتم و β یک ثابت کوچک مثبت است که ما آن را برابر $\beta = 0.05$ در نظر می‌گیریم. همان‌طور که در رابرتز و همکاران (۱۹۹۷) و رابرتز و روزنتال (۲۰۰۱) اشاره شده است، توزیع نرمال پیشنهادی $N(x, (\frac{\Sigma_n}{d}))$ در ابعاد بزرگ بهینه است. بنابراین، می‌توان آن را به‌عنوان یک تقریب برای ابعاد بزرگ در نظر گرفت. برای آزمون این الگوریتم، فرض می‌کنیم $\pi(\cdot) = N(0, MM')$ باشد که در آن M یک ماتریس $d \times d$ است و مولفه‌های آن، یعنی $\{M_{ij}\}_{i,j=1}^d$ ، تحقق‌هایی از توزیع نرمال استاندارد هستند. ماتریس کوواریانس هدف به‌صورت $\Sigma_n = MM'$ به‌دست می‌آید که بسیار ناسازگار است، بنابراین اگر بعد بالا باشد نمونه‌گیری از $\pi(\cdot)$ به‌راحتی امکان‌پذیر نیست.

در نتیجه الگوریتم متروپلیس سازوار ماتریس کوواریانس واقعی هدف را تولید می‌کند و به الگوریتمی که دارای نمونه‌های کارایی از $\pi(\cdot)$ باشد همگراست. در واقع تکرارهای زیادی انجام می‌شود تا الگوریتم اطلاعات مورد نظر را تولید کند. به عبارت دیگر آن‌چه که این الگوریتم تولید می‌کند، یک ماتریس کوواریانس $d \times d$ با پارامترهای زیاد است که این نشان‌دهنده عملکرد بسیار موثر الگوریتم سازوار در ابعاد بالا است.

الگوریتم متروپلیس درون گیبز سازوار

فرض کنید برای $i = 1, \dots, r_i$ و $j = 1, \dots, k$ مشاهدات y_{ij} از توزیع نرمال $N(\theta_i, \nu)$ پیروی کنند، به‌طوری‌که هر θ_i از توزیع کوشی $C(\mu, A)$ آمده باشد و $\mu \sim N(0, 1)$. همچنین فرض کنید A و ν هر دو دارای توزیع گامای معکوس $IG(1, 1)$ هستند. اکنون با قرار دادن $\varsigma = \{\theta_1, \theta_2, \dots, \theta_k, A, \nu, \mu\}$ ، توزیع پسین مدل را $\pi(\varsigma | \{y_{ij}\})$ قرار می‌دهیم. فرض کنید $k = 500$ باشد، بنابراین بعد ς برابر ۵۰۳ است. استنباط در این مدل بر اساس روش‌های انتگرال‌گیری عددی در بعد بالا بسیار پیچیده است و چون صورت توزیع پسین بسته نیست، استفاده از روش نمونه‌گیری گیبز یک انتخاب معمول است. با توجه به این‌که شکل توزیع‌های شرطی کامل برای همه پارامترها بسته نیست، از یک الگوریتم متروپلیس هستینگز درون گیبز (تیرنی، ۱۹۹۴) استفاده می‌کنیم.

در الگوریتم مذکور مقادیر پیشنهادی ۵۰۳ متغیر مورد نظر را می‌توان از توزیع پیشنهادی

$N(\circ, \sigma^2)$ تولید کرد. مطابق با نسبت احتمال پذیرش در الگوریتم MH، مقدار پیشنهادی رد یا پذیرفته می‌شود. با تکرار به اندازه کافی بزرگ این فرآیند، نمونه‌های تولیدشده در توزیع پسین به $\pi(\varsigma|\{y_{ij}\})$ همگرا می‌شوند. برای انتخاب پارامتر مقیاس σ^2 ، می‌توان از یک الگوریتم سازوار استفاده کرد. برای هر متغیر ς_i ، $i = 1, \dots, k+3$ ، مقدار لگاریتم پارامتر مقیاس برابر $\log S_i$ قرار داده می‌شود که لگاریتم انحراف معیار نمونه‌های تولیدشده قبلی است. برای شروع $\log S_1 = \circ$ برای به‌روزرسانی همه متغیرها در نظر گرفته می‌شود. بعد از 5° تکرار نمونه‌گیری، هر $\log S_i$ با اضافه یا کم کردن یک مقدار $\delta(n)$ به‌روزرسانی می‌شود. اگر نرخ پذیرش پیشنهادی برای هر متغیر نزدیک به 0.44 باشد، آن‌گاه الگوریتم سازوار خواهد بود.

۴.۳ الگوریتم بلوک‌بندی خودکار

فرض کنید $\mathcal{M}$ یک مدل سلسله‌مراتبی مشخص با پارامترهای $\Theta = \{\theta_1, \dots, \theta_d\}$ باشد. الگوریتمی که در این بخش معرفی می‌کنیم، فرآیند بلوک‌بندی خودکار را نشان می‌دهد. نتیجه اجرای این فرآیند، الگوریتم $\Psi_{AutoBlock} \in \Psi_{\mathcal{M}}$ را تولید می‌کند. در حین اجرای این روش نمونه‌های تولیدشده به‌عنوان نمونه‌های پسین حفظ نمی‌شوند، اما الگوریتم MCMC بلوکی کارای نتیجه‌شده برای تولید نمونه‌های معتبر از توزیع پسین هدف قابل به‌کارگیری است. لازم به ذکر است که اندیس‌های j و k شامل مقادیر $1, \dots, d$ هستند.

جدول ۱.۳: الگوریتم بلوک‌بندی خودکار

$\circ \rightarrow i$	: ۱
$\Psi_{Scalar} \rightarrow \Psi_\circ$	: ۲
$i + 1 \rightarrow i$	: ۳
از Θ تحت Ψ_{i-1} نمونه تولید کن به‌طوری که X_j نشان‌دهنده زنجیر نمونه‌ای از θ_j باشد	: ۴
5° درصد اول زنجیر X_j را حذف کن	: ۵
$cor(X_j, X_k) \rightarrow \rho_{jk}$	: ۶
$1 - \rho_{jk} \rightarrow d_{jk}$	: ۷
ماتریس فاصله D که عناصر آن d_{jk} ‌ها هستند را تشکیل بده	: ۸
درخت خوشه‌بندی سلسله‌مراتبی T که از D ساخته می‌شود را تشکیل بده	: ۹
$\{\Psi(T_{cut=\circ}), \Psi(T_{cut=\circ/\wedge}), \Psi(T_{cut=\circ/\vee}), \dots, \Psi(T_{cut=1})\} \rightarrow \Psi_{cand}$	: ۱۰
$\arg \max_{\Psi \in \Psi_{cand}} E(\Psi) \rightarrow \Psi_i$	: ۱۱
اگر $E(\Psi_i) > E(\Psi_{i-1})$ و $\Psi_i \neq \Psi_{i-1}$ آن‌گاه به مرحله ۳ برو	: ۱۲
$\Psi_i \rightarrow \Psi_{AutoBlock}$	: ۱۳
$\Psi_{AutoBlock}$ را برگردان	: ۱۴

فرآیند با الگوریتم MCMC اولیه $\Psi = \Psi_{Scalar}$ که همان الگوریتم نمونه‌گیری مولفه به مولفه شامل تمام پارامترهای مدل است، آغاز می‌شود. در واقع زمانی که هیچ اطلاع پیشینی در مورد بلوک‌های بالقوه در بین پارامترهای مدل نداشته باشیم، الگوریتم Ψ_{Scalar} به‌عنوان نقطه شروع برای دستیابی به ساختار وابستگی پسین انتخاب مناسبی است. تکرارهای بعدی بر پایه همبستگی پسین تجربی تکرار قبلی ساخته می‌شوند، به‌طوری‌که پارامترهای درون هر بلوک در هر تکرار همبستگی بالای درون‌گروهی را نشان می‌دهند.

قبل از محاسبه همبستگی تجربی ρ_{jk} بین پارامترهای j و k ، به دلخواه مثلاً ۵۰ درصد نمونه‌های اولیه از کل نمونه‌های متوالی را حذف می‌کنیم. این روش را با سوزاندن نباید اشتباه گرفت. هدف از سوزاندن، فراموش کردن حالت اولیه و اطمینان از همگرایی به توزیع هدف است. اما در این جا با دور انداختن نمونه‌های اولیه، اجازه داده می‌شود تا تمام نمونه‌گیری‌های بلوکی و مولفه به مولفه سازوار زمان خود-تنظیم بیشتری داشته باشند و در نتیجه کارایی الگوریتمی بهینه نظری خود را به‌دست آورند. انتخاب ۵۰ درصد کاملاً دلخواه است و در بیشتر موارد بیش از اندازه است و بدون داشتن تاثیری بر روی عملکرد الگوریتم، می‌توان از آن صرف‌نظر کرد.

ماتریس فاصله D که مولفه‌های آن از رابطه $d_{jk} = 1 - |\rho_{jk}|$ به‌دست می‌آیند را تشکیل می‌دهیم. قطر اصلی این ماتریس صفر است. اگر $d_{ij} \approx 0$ همبستگی قوی و اگر $d_{ij} \approx 1$ همبستگی ضعیف را می‌توان نتیجه گرفت و همچنین همبستگی ρ و $-\rho$ فاصله یکسان را نتیجه می‌دهد.

در نرم‌افزار R از تابع `hclust` می‌توان برای تشکیل درخت سلسله‌مراتبی T با استفاده از ماتریس فاصله D استفاده کرد. روش خوشه‌بندی پیش‌فرض تابع `hclust` در نرم‌افزار R، روش پیوند کامل^{۲۵} است (ایورت، ۲۰۱۱؛ فصل ۴) که برای کار ما مناسب است، زیرا این اطمینان را خواهیم داشت که تمام پارامترهای درون هر خوشه دارای یک می‌نیم قدر مطلق همبستگی جفتی هستند. اگر h را در بازه $[0, 1]$ در درخت T انتخاب کنیم، آن‌گاه قدر مطلق همبستگی که بین جفت پارامترها وجود دارد، حداقل $1 - h$ است.

در نرم‌افزار R می‌توان از تابع `cutree` برای برش درخت خوشه‌بندی سلسله‌مراتبی T در ارتفاع خاصی مثل $h \in [0, 1]$ استفاده کرد. از برش درخت در ارتفاع مشخص h برای تعیین تعداد بلوک‌ها و پارامترهای هر بلوک در نمونه‌گیری MCMC استفاده می‌شود. در واقع هدف اصلی، بلوک‌بندی پارامترهاست تا کارایی الگوریتمی را افزایش دهد.

الگوریتم $\Psi(T_{cut=h}) \in \Psi_M$ را به‌عنوان الگوریتم MCMC منحصر به فردی که توسط نمونه‌گیری مولفه به مولفه یا بلوکی اعمال شده بر روی پارامترهای بلوک‌ها از برش T در ارتفاع h نتیجه می‌شود، تعریف می‌کنیم. برای همه T ‌ها؛ $\Psi(T_{cut=0}) = \Psi_{scalar}$ و $\Psi(T_{cut=1}) = \Psi_{block}$ که همان دو عضو فرین الگوریتم‌های مولفه به مولفه و تک بلوکی را در رده همه الگوریتم‌های MCMC بلوکی نشان می‌دهند. عموماً هیچ مقدار بهینه‌ای از مقدار h وجود ندارد. همان‌طور

^{۲۵}Complete linkage

که در فصل ۲ اشاره شده است، هر $h \in [0, 1]$ می‌تواند کارایی $E(\Psi(T_{cut=h}))$ را برای یک مدل خاص $\mathcal{M}$ ماکسیمم کند.

به‌جای تلاش برای استنباط این که چه ارتفاعی می‌تواند یک ارتفاع مناسب برای مدل $\mathcal{M}$ باشد، در بازه $[0, 1]$ نقاط برش مختلفی را در نظر می‌گیریم و الگوریتم MCMC حاصل از آن نقاط را به‌دست می‌آوریم. خروجی الگوریتم، Ψ_{cand} ، مجموعه‌ای از الگوریتم‌های MCMC برای یک تکرار خاص است. این روش اجازه می‌دهد که فرآیند بلوک‌بندی مطابق مدل، و ساختار همبستگی پسین انجام شود. الگوریتم MCMC برای هر تکرار ($i \geq 1$) که از میان Ψ_{cand} انتخاب شده است، بیشترین کارایی کلی را تولید می‌کند.

برای برآورد زمان خودهمبستگی تجمیع‌شده، و در نتیجه محاسبه معیار کارایی الگوریتمی $A(\Psi)$ ، می‌توان از تابع `effectiveSize` در بسته‌افزار `coda` در نرم‌افزار R استفاده کرد. بنا به تامپسون (۲۰۱۰) روشی که برای برآورد زمان خودهمبستگی تجمیع‌شده در این تابع پیاده‌سازی شده است، در بین چندین روش برآورد سریع‌ترین روش همگرا به مقدار واقعی است. با افزایش $E(\Psi_i)$ در تکرارهای متوالی، برآورد بهتری از همبستگی پسینی نتیجه می‌شود و بنابراین بلوک‌بندی پارامترها با دقت بیشتری اجرا می‌شود. این نتیجه باعث افزایش بیشتر معیار کارایی کل $E(\Psi_i)$ در تکرارهای بعدی می‌شود. این فرآیند تکراری را تا زمانی ادامه می‌دهیم که

$$\Psi_i = \Psi_{i-1} \quad (1)$$

یعنی در دو تکرار متوالی الگوریتم‌های یکسانی نتیجه شوند. در چنین حالتی، از آن مرحله به بعد الگوریتم MCMC Ψ_i برای تولید نمونه انتخاب می‌شود. یا اگر

$$E(\Psi_i) < E(\Psi_{i-1}) \quad (2)$$

که به معنی کاهش کارایی بین تکرارها است و نشان می‌دهد که این فرآیند نمی‌تواند به الگوریتم MCMC کارآمدتری از وضعیت فعلی آن تبدیل شود. در واقع این فرآیند با شرط (۱) متوقف می‌شود و محاسبه متغیر تصادفی در نمونه‌گیری و زمان اجرا ممکن است هم‌زمان با شرط (۲) پایان یابد.

خروجی فرآیند بلوک‌بندی خودکار به‌صورت $\Psi_{AutoBlock}$ است که همان الگوریتم MCMC انتخاب‌شده در آخرین تکرار می‌باشد. معمولاً، الگوریتم $\Psi_{AutoBlock}$ از تکرار دوم به بعد یکسان باقی می‌ماند، یعنی فرآیند بلوک‌بندی به حالت مانا همگرا می‌شود. مراحل اجرای این الگوریتم در جدول ۱.۳ فهرست شده‌اند.

فصل ۴

ارزیابی عملکرد بلوک‌بندی خودکار

۱.۴ مقدمه

در این فصل، فرآیند بلوک‌بندی خودکار را بر روی پنج مثال که شامل داده‌های واقعی و مدل‌های شبیه‌سازی هستند، ارزیابی کرده و نتایج را با الگوریتم‌های MCMC استاندارد مقایسه می‌کنیم. مدل‌های مختلفی شامل مدل‌های اثرات تصادفی، اتو رگرسیون، فضای حالت گاوسی، فضایی و آمیخته خطی تعمیم‌یافته را مورد بررسی قرار می‌دهیم. افزون بر این، یک مثال بومی را نیز ارزیابی می‌کنیم که مربوط به تعداد دفعات مراجعه بیماران سالخورده در شهر اهواز به بیمارستان آریا این شهر است.

همان‌طور که در فصل ۲ مطرح شد، طراحی بهینه الگوریتم‌های MCMC به محیط محاسباتی و سیستم عامل رایانه‌ای که اجرای الگوریتم روی آن انجام می‌شود، وابسته است. به همین دلیل اطلاعات مربوط به ابزارهای نرم‌افزاری استفاده‌شده و مشخصات سیستم عامل مورد نظر را فهرست می‌کنیم.

همه مدل‌های آماری و الگوریتم‌های MCMC ارائه‌شده در این فصل، شامل روش بلوک‌بندی خودکار، توسط بسته‌افزار NIMBLE (تیم توسعه نیمبل، ۲۰۱۸؛ دی‌والپین و همکاران، ۲۰۱۷) در نرم‌افزار R ساخته شده‌اند. بسته NIMBLE این امکان را فراهم می‌کند تا بتوان مدل‌های سلسله‌مراتبی را درون محیط R با استفاده از ساختاری شبیه زبان BUGS (لون و همکاران، ۲۰۰۰؛ ۲۰۱۲) تعریف کرد. الگوریتم‌های MCMC در NIMBLE نیز با استفاده از زبان مخصوص

این بسته نوشته شده‌اند. با نوشتن مدل در این بسته و اجرای آن از طریق توابعی مشخص، مدل به یک کد زبان C++ تبدیل شده و الگوریتم روی آن اجرا می‌شود. در نهایت خروجی به محیط R برمی‌گردد. راهنمای مختصری درباره نحوه استفاده از NIMBLE را در پیوست پایان‌نامه آورده‌ایم.

همه مثال‌های این فصل بر روی نسخه 1.5.3 زبان R تحت سیستم عامل ویندوز با پردازنده Core i5 اجرا شده‌اند.

۲.۴ مثال‌های کاربردی

ابتدا مثال‌ها و مدل‌های به کار گرفته‌شده برای تحلیل آن‌ها را معرفی می‌کنیم. تحلیل نتایج را در بخش بعدی ارائه خواهیم کرد.

۱.۲.۴ مدل اثرات تصادفی

مجموعه داده‌های زاد و ولد خوک‌ها (معروف به litters) و مدل مورد استفاده برای تحلیل آن‌ها، از مجموعه مثال‌های راهنمای WinBUGS (اسپیگل‌هالتر و همکاران، ۱۹۹۶) انتخاب شده است. این مجموعه داده شامل دو گروه از ۱۶ خوک هستند که متغیر پاسخ در آن تعداد مرگ و میر ناشی از زاد و ولد خوک‌ها را نشان می‌دهد. جدول ۱.۴ داده‌های این مجموعه را نشان می‌دهد. در هر گروه نسبت زنده ماندن کم و بیش مشابه هستند ولی یکسان نیستند. برای این داده‌ها فرض می‌کنیم مشاهدات دارای توزیع دوجمله‌ای هستند که در هر گروه احتمال‌های دوجمله‌ای به‌صورت اثرات تصادفی حاصل از یک توزیع بتا مدل‌سازی می‌شوند.

جدول ۱.۴: نسبت زنده ماندن نوزادان دو گروه از ۱۶ خوک

خوک	۱	۲	۳	۴	۵	۶	۷	۸
مجموعه اول	$\frac{13}{13}$	$\frac{12}{12}$	$\frac{9}{9}$	$\frac{9}{9}$	$\frac{8}{8}$	$\frac{8}{8}$	$\frac{12}{13}$	$\frac{11}{12}$
مجموعه دوم	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{8}{9}$	$\frac{11}{13}$	$\frac{4}{5}$	$\frac{5}{7}$	$\frac{7}{10}$	$\frac{7}{10}$
خوک	۹	۱۰	۱۱	۱۲	۱۳	۱۴	۱۵	۱۶
مجموعه اول	$\frac{8}{9}$	$\frac{4}{5}$	$\frac{7}{9}$	$\frac{4}{7}$	$\frac{5}{10}$	$\frac{3}{6}$	$\frac{3}{10}$	$\frac{0}{7}$
مجموعه دوم	$\frac{12}{12}$	$\frac{11}{11}$	$\frac{10}{10}$	$\frac{9}{9}$	$\frac{10}{11}$	$\frac{9}{10}$	$\frac{9}{10}$	$\frac{8}{9}$

فرض کنید r_{ij} تعداد مرگ و میر مشاهده‌شده باشد که دارای توزیع دوجمله‌ای با اندازه

نمونه n_{ij} و احتمال p_{ij} است که p_{ij} از توزیع بتا پیروی می‌کند. بنابراین

$$r_{ij} \sim \text{Bin}(n_{ij}, p_{ij}) \quad i = 1, \dots, 16$$

$$p_{ij} \sim \text{Beta}(a_j, b_j) \quad j = 1, 2$$

که در آن پارامترهای a_j و b_j دارای توزیع گاما $\text{gamma}(1, 0.001)$ هستند. این مدل توسط جورج و همکاران (۱۹۹۳) تحلیل شده است. پارامترهای توزیع بتا (بر حسب a و b) سبب ایجاد همبستگی قوی بین هر a_i و b_i می‌شود. با در نظر گرفتن این همبستگی، انتخاب یک الگوریتم MCMC که هر زوج a_i و b_i را در یک بلوک قرار دهد، می‌تواند یک الگوریتم کارا نتیجه دهد.

۲.۲.۴ مدل اتورگرسیو

داده‌های این مثال نیز از مجموعه مثال‌های WinBUGS گرفته شده است. این مجموعه داده مربوط به تعداد افراد مبتلا به سرطان پستان در ۷۷ گروه مختلف در کشور ایسلند است که توسط یک مدل رگرسیون پواسون تحلیل می‌شود. این مجموعه در مثال‌های راهنمای WinBUGS با نام ice مشخص شده است. متغیرهای تبیینی در این مجموعه داده، گروه‌های سنی (شامل ۱۳ گروه از ۲۰- تا ۸۴-۸۰ ساله؛ age)، سال تولد (که ۱۱ گروه کوهورت^۱ از سال ۱۸۴۰ تا ۱۹۴۹ را تشکیل می‌دهد؛ years)، و تعداد کل افراد حاضر در هر گروه کوهورت (person-years)، هستند.

این داده‌ها توسط بریسلو و کلیتون (۱۹۹۳) تحلیل شده‌اند. ساختار این مجموعه داده در جدول ۲.۴ نمایش داده شده است. مدلی که برای تحلیل این داده‌ها در نظر می‌گیریم،

جدول ۲.۴: مجموعه داده ice

شماره	گروه سنی	سال تولد	تعداد افراد مبتلا	تعداد کل افراد
۱	۱	۶	۲	۴۱۳۸۰
۲	۱	۷	۰	۴۳۶۵۰
⋮				
۷۷	۱۳	۵	۳۱	۱۳۶۰۰

یک مدل اتورگرسیو مرتبه دوم هموار برای اثرات گروه‌های کوهورت سال تولد است. به‌طور

^۱Cohort

دقیق‌تر، مدل پیشنهادی برای تحلیل این داده‌ها به صورت زیر است:

$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\lambda_i = \exp(\eta_i)$$

$$\eta_i = \log(\text{person} - \text{years}_i) + \alpha_{\text{age}_i} + \beta_{\text{years}_i}$$

به‌طوری که β ها، اثر تصادفی گروه‌های کوهورت با پارامتر دقت مشترک τ هستند، که در آن

$$\beta_k = 2\beta_{k-1} - \beta_{k-2} + \epsilon_k$$

و

$$\epsilon_k \sim N(0, \tau^{-1}).$$

در این مدل از توزیع‌های پیشین زیر برای ساخت توزیع پسین استفاده می‌کنیم:

$$\alpha_{\text{age}_i} \sim N(0, 10^6)$$

$$\tau = \frac{1}{\sigma^2}$$

$$\sigma \sim \text{Unif}(0, 1).$$

۳.۲.۴ مدل فضای حالت گاوسی

در این مثال، داده‌ها را از مدل فضای حالت گاوسی زیر شبیه‌سازی می‌کنیم:

$$y_t = X_t + \epsilon_t \quad \epsilon_t \sim N(0, \sigma_\epsilon^2)$$

$$X_t = b + aX_{t-1} + \nu_t \quad \nu_t \sim N(0, \sigma_\nu^2) \quad t = 2, \dots, n.$$

برای مطالعه وجود وابستگی پسینی بین پارامترها بر عملکرد الگوریتم‌های MCMC، دو نسخه پارامتری متفاوت برای معادله حالت در نظر می‌گیریم:

(الف) مدل با پارامترهای عرض از مبدا b و میانگین μ به‌طوری که $a = 1 - \frac{b}{\mu}$. در این حالت دو پارامتر عرض از مبدا b و میانگین μ از هم مستقل هستند. در این حالت یک الگوریتم MCMC کارا تر الگوریتمی است که دو پارامتر را در دو بلوک مجزا به‌روز کند.

(ب) مدل با پارامترهای خودهمبستگی a و عرض از مبدا b . در این نسخه، دو پارامتر مذکور همبسته هستند و بهتر است از یک الگوریتم MCMC با یک بلوک شامل هر دو پارامتر استفاده شود.

برای بررسی عملکرد الگوریتم‌های با بلوک‌های تک متغیره و یک بلوک توام، برای هر دو مدل، ۱۰۰ مشاهده و حالت پنهان تولید کردیم. در مدل اول (یعنی مدل با پارامترهای مستقل)، از توزیع‌های پیشین زیر برای ساخت توزیع پسین استفاده کردیم:

$$\begin{aligned} X_1 | \mu, \sigma_\epsilon^2, \sigma_\nu^2 &\sim N(\mu, \sigma_\epsilon^2 + \sigma_\nu^2) & y_1 | \sigma_\epsilon^2 &\sim N(X_1, \sigma_\epsilon^2) \\ \mu &\sim N(0, 1000^2) & b &\sim N(0, 1000^2) \\ \sigma_\epsilon^2 &\sim \text{Unif}(0.0001, 1) & \sigma_\nu^2 &\sim \text{Unif}(0.0001, 1). \end{aligned}$$

به‌طور مشابه، برای مدل دوم (یعنی مدل با پارامترهای وابسته) از توزیع‌های پیشین زیر استفاده کردیم:

$$\begin{aligned} X_1 | a, b, \sigma_\epsilon^2, \sigma_\nu^2 &\sim N\left(\frac{b}{1-a}, \sigma_\epsilon^2 + \sigma_\nu^2\right) & y_1 | \sigma_\nu^2 &\sim N(X_1, \sigma_\nu^2) \\ a &\sim \text{Unif}(-0.9999, 0.9999) & b &\sim N(0, 1000) \\ \sigma_\epsilon^2 &\sim \text{Unif}(0.0001, 1) & \sigma_\nu^2 &\sim \text{Unif}(0.0001, 1). \end{aligned}$$

۴.۲.۴ مدل فضایی

داده‌های این مثال شامل تعداد گونه‌های گوش ماهی در ۱۴۸ مکان مختلف از سواحل نیویورک و نیوجرسی است که توسط مرکز علمی شیلات شمال شرقی آمریکا در سال ۱۹۹۳ گردآوری شده‌اند. این مجموعه داده در آدرس

<http://www.biostat.umn.edu/~brad/data/myscallops.txt>

به‌طور رایگان در دسترس است. این داده‌ها توسط بنرجی و همکاران (۲۰۰۳) تحلیل شده است. بر اساس کار بنرجی و همکارانش، میانگین لگاریتم تعداد گونه‌ها را با یک میدان تصادفی گاوسی مدل‌بندی می‌کنیم. ساختار کوواریانس مدل را نیز یک مدل نمایی در نظر می‌گیریم. یعنی فرض می‌کنیم

$$\begin{aligned} (g_1, \dots, g_{148})' &\sim N_{148}(\mu \mathbf{1}, \Sigma) \\ \Sigma = (\sigma_{ij}) &= \sigma^2 e^{-\frac{d_{ij}}{\rho}} \end{aligned}$$

که در آن d_{ij} فاصله مکانی بین مشاهدات و g_i اثرات فضایی مدل هستند. همچنین ۱ برداری از ۱ ها با بعد ۱۴۸ است و پارامترهای σ و ρ پارامترهای ساختار وابستگی میدان تصادفی است. با توجه به آن که اطلاعاتی از متغیرهای تبیینی در دسترس نداریم، تابع میانگین را برابر مقدار ثابت μ در نظر می‌گیریم.

با این توضیحات، مدل سلسله‌مراتبی به صورت زیر تعریف می‌شود:

$$\begin{aligned} y_i &\sim \text{Poisson}(\lambda_i), \quad i = 1, \dots, 148 \\ \lambda_i &= \exp(g_i). \end{aligned}$$

از آن‌جا که ساختار کوواریانس نمایی یک داد و ستد بین پارامترهای σ و ρ القا می‌کند، انتظار داریم که این دو پارامتر در توزیع پسین همبسته باشند. بنابراین قرار دادن آن‌ها در یک بلوک می‌تواند به الگوریتمی کارا بیانجامد.

۵.۲.۴ مدل آمیخته خطی تعمیم‌یافته

مجموعه داده این مثال، شامل ۹۶۸ داده از تعداد دفعات مراجعه یا تماس شهروندان سالخورده به یک کلینیک در چهار دوره شش ماهه است که مربوط به ۱۲۱ نفر شهروند است که در یک برنامه پزشکی شرکت کرده‌اند (والر و زلترمن، ۱۹۹۷). متغیر پاسخ در این مثال را با یک رگرسیون پواسون مدل‌بندی می‌کنیم. پیشگوی خطی مدل شامل چندین اثر تصادفی و ثابت است، که اثرات متقابل متغیرهای تبیینی را نیز در بر دارد. برای مشاهده جزئیات داده‌ها و اثرات مدل به والر و زلترمن (۱۹۹۷) مراجعه کنید. مدل مورد نظر برای این داده‌ها به صورت زیر است:

$$\begin{aligned} y_{rst} &\sim \text{Poisson}(m_{rst}) \\ m_{rst} &= \exp(\eta_{rst}) \\ \eta_{rst} &= \mu + \beta_s + \gamma_t + \alpha_r + \theta_{rt} + \psi_{rs}; \end{aligned} \quad \begin{cases} r = 1, \dots, 121 \\ s = 1, 2 \\ t = 1, \dots, 4 \end{cases}$$

که در آن r تعداد افراد شرکت‌کننده در برنامه پزشکی است، $s = 1$ برای تماس‌ها و $s = 2$ برای مراجعه‌کنندگان است، و $t = 1, \dots, 4$ تعداد دوره‌های شش ماهه است. همچنین β_s ، α_r و γ_t به ترتیب اثرات مربوط به افراد، نوع مراجعه، و زمان است. علاوه بر این، θ_{rt} و ψ_{rs} اثرات متقابل افراد شرکت‌کننده و زمان، و افراد و نوع مراجعه است. توزیع‌های پیشین برای اثرات نیز عبارتند از

$$\begin{aligned} \mu &\sim N(0, 1000) & \beta_1 &\sim N(0, 1000), & \beta_2 &= 0 \\ \gamma_1, \gamma_2, \gamma_3 &\stackrel{iid}{\sim} N(0, 1000), & \gamma_4 &= 0 \\ \alpha_r | \tau_r &\sim N(0, \tau_r) & \tau_r^{-1} &\sim \text{Unif}(0, 10) \\ \psi_{rs} | \tau_{rs} &\sim N(0, \tau_{rs}) & \tau_{rs}^{-1} &\sim \text{Unif}(0, 10) \\ \theta_{rs} | \tau_{rs} &\sim N(0, \tau_{rs}) & \tau_{rs}^{-1} &\sim \text{Unif}(0, 10). \end{aligned}$$

۳.۴ تحلیل مثال‌های کاربردی

برای ارزیابی عملکرد الگوریتم‌های MCMC مختلف در مثال‌های ذکر شده، سه معیار کارایی را در نظر گرفتیم. همان‌طور که در فصل ۲ مطرح شد، سه نوع کارایی زیر را مد نظر قرار دادیم:

۱. کارایی الگوریتمی

۲. کارایی محاسباتی

۳. کارایی کل

برای محاسبه کارایی الگوریتمی، برای داشتن تعبیر مناسب‌تر، به جای معیار $A(\Psi)$ از رابطه $ESS = 10000 A(\Psi)$ استفاده کردیم، که در آن ESS اندازه نمونه موثر است. این مقیاس‌بندی معیار $A(\Psi)$ یک تعبیر طبیعی به همراه دارد که عبارتست از تعداد نمونه‌های موثر (برای پارامتر با کندترین آمیختگی) که از یک زنجیر MCMC به طول ۱۰۰۰۰ نتیجه می‌شود.

به‌طور مشابه، به جای گزارش معیار کارایی محاسباتی $C(\Psi)$ ، از معیار $Runtime = 10000 C(\Psi)$ استفاده کردیم که به‌عنوان زمان مورد نیاز (بر حسب ثانیه) برای تولید ۱۰۰۰۰ نمونه MCMC تعبیر می‌شود. اما معیار کارایی کل را به صورت همان نسخه اصلی

$$Efficiency = \frac{ESS}{Runtime} = \frac{A(\Psi)}{C(\Psi)} = E(\Psi) \quad (1.4)$$

استفاده کردیم، زیرا این معیار مستقل از هر نوع مقیاسی است و همان تعبیر شهودی تعداد نمونه‌های موثر تولیدشده در هر ثانیه از زمان اجرای الگوریتم (برای پارامتر با کندترین آمیختگی) را شامل می‌شود.

نمونه‌گیری MCMC با استفاده از یک هسته تولید عدد تصادفی ثابت و مقادیر اولیه یکسان برای هر مدل اجرا شد. با این تنظیمات، الگوریتم‌های MCMC یکسان، زنجیرهای نمونه یکسان و در نتیجه ESS یکسان را تولید می‌کنند؛ اما لزوماً زمان اجرا و در نتیجه کارایی کل مشابهی نخواهند داشت.

به‌طور کلی، شش نوع الگوریتم MCMC متفاوت را مورد بررسی قرار دادیم.

- الگوریتم تک بلوکی، Ψ_{block} ، که برای معرفی آن از نماد AB^۲ استفاده شده است.
- الگوریتم مولفه به مولفه، Ψ_{scalar} ، که برای آن از نماد AS^۳ استفاده شده است.
- الگوریتم بلوک‌بندی خودکار پیشنهادی در این پایان‌نامه، یعنی $\Psi_{AutoBlock}$ ، که به صورت AUB^۴ نمایش داده می‌شود.

^۲ All blocked

^۳ All scalar

^۴ Auto Blocking

- الگوریتمی که از امکان نمونه‌گیری مزدوج استفاده می‌کند. در این نوع الگوریتم، از روش نمونه‌گیری بلوکی برای گروهی از پارامترها که از توزیع‌های چندمتغیره آمده باشند و از روش نمونه‌گیری مولفه به مولفه برای پارامترهایی که از توزیع‌های یک متغیره پیروی کنند، استفاده می‌شود. در چنین الگوریتمی اگر ساختار مدل اجازه دهد، می‌توان از روش‌های نمونه‌گیری مزدوج نیز استفاده کرد. به چنین الگوریتمی، پیش‌فرض^۵ می‌گوییم و از نماد Def برای معرفی آن استفاده می‌کنیم. برای درک چنین الگوریتمی به مثال زیر توجه کنید.

مثال ۱.۴. مدلی با چهار پارامتر $\theta_1, \theta_2, \theta_3, \theta_4$ را در نظر بگیرید که توزیع‌های پیشین تعریف‌شده بر روی پارامترها منجر به چگالی‌های شرطی کامل زیر شوند:

$$۱. (\theta_1, \theta_2) \quad \pi(\theta_1, \theta_2 | \theta_3, \theta_4)$$

$$۲. \theta_3 \quad \pi(\theta_3 | \theta_1, \theta_2, \theta_4)$$

$$۳. \theta_4 \quad \pi(\theta_4 | \theta_1, \theta_2, \theta_3)$$

به‌طوری که چگالی‌های شرطی کامل دو بلوک اول و دوم صورت بسته ندارند ولی چگالی شرطی کامل بلوک سوم، بر اساس ویژگی مزدوج، هم‌توزیع با توزیع پیشین تعریف‌شده روی θ_4 است. بنابراین، در این حالت، برای تولید نمونه از بلوک اول از یک روش نمونه‌گیری متروپلیس-هستینگز دو متغیره و برای تولید نمونه از بلوک دوم از یک روش نمونه‌گیری متروپلیس-هستینگز یک متغیره می‌توان استفاده کرد. برای تولید نمونه از بلوک سوم به‌سادگی می‌توان از یک روش نمونه‌گیری مزدوج استفاده کرد و نیازی به معرفی زنجیر مارکوف نیست. الگوریتم MCMC این مثال یک الگوریتم Def است که اکثر نرم‌افزارهای مجری روش‌های MCMC، مثل WinBUGS، به‌صورت پیش‌فرض، از این نوع الگوریتم برای به‌روزرسانی پارامترهای مدل استفاده می‌کنند.

- نوع دیگر الگوریتم، الگوریتم IB^۶ است که در آن بلوک‌بندی بر اساس اطلاعات اولیه از پارامترهای مدل توسط محقق و به صورت ثابت انجام می‌شود. به‌عنوان مثال، اگر مدل رگرسیون خطی زیر را در نظر بگیریم

$$y = x'\beta + \epsilon \quad \epsilon \sim N(0, \sigma^2)$$

که در آن $\beta = (\beta_0, \dots, \beta_p)'$ ، برآوردگرهای کمترین توان‌های دوم (که معادل برآوردگرهای ماکسیمم درست‌نمایی نیز هستند)، وابسته هستند. به عبارت دقیق‌تر

$$\text{cov}(\hat{\beta}) = \sigma^2 (X'X)^{-1}$$

^۵Default

^۶Informed blocking

که در آن X ماتریس طرح حاصل از متغیرهای تبیینی x برای همه مشاهدات است. از طرفی برآوردگر σ^2 مستقل از $\hat{\beta}$ است. بنابراین انتظار داریم مشابه این رفتار در توزیع پسین این پارامترها دیده شود. پس با اطلاع قبلی می‌توان پارامترهای β را در یک بلوک و σ^2 را در بلوکی جدا قرار داد.

- زمانی که در مدل سلسله‌مراتبی M اثر تصادفی وجود داشته باشد، از الگوریتم ^{V}ICL استفاده می‌کنیم و بردار اثرات تصادفی را در یک بلوک قرار خواهیم داد.

۱.۳.۴ مدل اثرات تصادفی

همان‌طور که در مثال اول در تعریف مدل اثرات تصادفی بیان شد، چون پارامترهای توزیع بتا موجب همبستگی قوی بین هر a_i و b_i می‌شوند، پس انتظار داریم که الگوریتم MCMC تولیدشده توسط الگوریتم IB عملکرد خوبی داشته باشد و معادل الگوریتم بلوک‌بندی خودکار باشد. ارتفاع برش $h = 0.1$ نشان می‌دهد که در هر زوج (a_i, b_i) همبستگی پسین بالای ۰/۹ وجود دارد. بهترین آمیختگی زمانی حاصل می‌شود که مقدار ESS مربوط به الگوریتم بزرگ باشد. برای این مثال، الگوریتم ICL بیشترین ESS را تولید کرده است. در نتیجه از آمیختگی خوبی برخوردار است. جدول ۳.۴ نتایج معیارهای کارایی محاسبه‌شده برای الگوریتم‌های مختلف را نشان می‌دهد. همان‌طور که از مقادیر جدول می‌توان دید، کارایی کل الگوریتم‌های IB و ICL مشابه الگوریتم بلوک‌بندی خودکار است.

جدول ۳.۴: معیارهای کارایی برای مثال اثرات تصادفی در مجموعه داده litters

مدل	الگوریتم MCMC	ESS	Runtime	Efficiency
اثرات تصادفی	AB	۰/۴	۰/۲۹	۱/۳
	Def	۱/۱	۱/۱۹	۱/۰
	AS	۲/۱	۰/۵۱	۴/۲
	IB	۱۹/۰	۵۰/۰	۳۸/۲
	ICL	۱۰۱/۳	۲/۶۴	۳۸/۵
	AUB	۱۹/۰	۰/۴۸	۳۹/۲

۲.۳.۴ مدل اتورگرسیو

در مثال داده‌های سرطان پستان در کشور ایسلند، برای مدل اتورگرسیو، کمترین آمیختگی مربوط به نمونه‌گیری AS است. زمانی که تمام ۲۴ پارامتر مدل (شامل متغیرهای پنهان فرآیند

^VInfomed cross-level

اتورگرسیو، اثرات ثابت و ابرپارامتر) را در یک بلوک قرار دهیم، زمان اجرای الگوریتم AB نسبت به AS نصف خواهد شد. مطابق ساختار فرآیند اتورگرسیو، چون میان پارامترها همبستگی وجود دارد، زمانی که تمام پارامترهای مدل را در یک بلوک قرار دهیم، زمان اجرای الگوریتم AB نسبت به نمونه‌گیری AS کمتر و کارایی بیشتری خواهد داشت. همچنین زمان اجرای بلوک‌بندی خودکار مشابه نمونه‌گیری AB است. در بلوک‌بندی خودکار کارایی الگوریتمی و کارایی کل ماکسیمم مقدار خود را دارند.

جدول ۴.۴: معیارهای کارایی برای مثال اتورگرسیو در مجموعه داده سرطان پستان ایسلند

مدل	الگوریتم MCMC	ESS	Runtime	Efficiency
اتو رگرسیو	AB	۸/۹	۰/۳	۲۷/۳
	AS	۶/۵	۰/۶	۱۱/۵
	AUB	۱۲/۷	۰/۳	۳۷/۵

۳.۳.۴ مدل فضای حالت گاوسی

جدول ۵.۴ نتایج حاصل از دو نوع بلوک‌بندی به ازای هر دو نسخه پارامتری مدل فضای حالت را نشان می‌دهد. در حالت مستقل، الگوریتمی که تمام پارامترها را در یک بلوک در نظر می‌گیرد، به سرعت اجرا می‌شود اما کارایی پارامتر عرض از مبدا فرآیند اتورگرسیو بسیار پایین است. در این حالت چون پارامترهای عرض از مبدا و میانگین مستقل از هم هستند، اگر آن‌ها را در بلوک‌های جدا از هم قرار دهیم، آنگاه کارایی الگوریتمی بلوک‌های تک پارامتری بیشتر از حالتی است که تمام پارامترها در یک بلوک باشند. الگوریتم Def به هر فضای حالت، نمونه‌گیری نرمال مزدوج اختصاص می‌دهد. در نتیجه کارایی الگوریتمی زیاد و زمان اجرا طولانی است. بنابراین، طبق رابطه (۱.۴) مقدار کارایی کل کاهش می‌یابد. در بلوک‌بندی خودکار اگر ارتفاع برش $h = 0/8$ باشد، یک بلوک از شش پارامتر شامل پنج تا فضای حالت و خطای مشاهده‌شده و یک بلوک مجزا از دو پارامتر فرآیند اتورگرسیو ایجاد می‌شود. این ترکیب، موجب افزایش ۴۰ درصدی کارایی تحت نمونه‌گیری AS می‌شود که بیشترین کارایی مربوط به الگوریتم‌های غیرخودکار MCMC است.

در حالت پارامتری کردن وابسته، اگر تمام پارامترها در یک بلوک باشند دارای آمیختگی ضعیف‌تری نسبت به حالتی است که هر پارامتر را در یک بلوک جدا در نظر بگیریم. برای بررسی وجود همبستگی بین پارامترهای عرض از مبدا و ضریب همبستگی فرآیند اتورگرسیو در مدل فضای حالت، ابتدا این دو پارامتر را در بلوک‌های جدا از هم قرار داده و الگوریتم را اجرا می‌کنیم. نتیجه اجرا سریع، اما ESS عرض از مبدا بسیار ضعیف است. بنابراین بین عرض از مبدا و ضریب همبستگی، وابستگی ضعیفی وجود دارد. در الگوریتم Def نمونه‌گیری

مزدوج صورت می‌گیرد اما مشابه الگوریتم AS کارایی الگوریتمی پایینی از پارامترهای همبسته فرآیند اتورگرسیو تولید می‌کند. اگر ارتفاع برش را $h = 0.1$ در نظر بگیریم، الگوریتم بلوک‌بندی خودکار پارامتر بلوکی یکسانی شامل عرض از مبدا فرآیند اتورگرسیو و ضریب همبستگی همانند الگوریتم IB، انتخاب می‌کند. زمان اجرای هر دو الگوریتم برابر هستند. کارایی تولیدشده توسط الگوریتم بلوک‌بندی خودکار ۲۰ برابر الگوریتم AB است که بیشترین کارایی را در بین الگوریتم‌های غیرخودکار دارد و ۲۵ درصد بیشتر از کارایی الگوریتم IB است. بنابراین در این مثال، فرآیند بلوک‌بندی خودکار بدون نیاز به مداخله کاربر بیشترین کارایی را تولید می‌کند.

جدول ۵.۴: معیارهای کارایی برای مثال فضای حالت شبیه‌سازی شده

مدل	الگوریتم MCMC	ESS	Runtime	Efficiency
فضای حالت (مستقل)	AB	۰/۳	۰/۸	۰/۴
	Def	۲۷/۶	۴/۶	۶/۰
	AS	۲۰/۲	۱/۳	۱۵/۷
	AUB	۲۹/۱	۱/۳	۲۲/۴
فضای حالت (همبسته)	AB	۰/۶	۰/۷	۰/۸
	Def	۱/۷	۴/۹	۰/۴
	AS	۱/۱	۱/۳	۰/۸
	IB	۱۸/۴	۱/۲	۱۵/۶
	AUB	۲۶/۱	۱/۲	۲۰/۹

۴.۳.۴ مدل فضایی

جدول ۶.۴ مربوط به نتایج عملکرد MCMC برای مدل فضایی است که روابط جالبی را در کارایی MCMC نشان می‌دهد. مدل فضایی شامل ۱۴۸ پارامتر پنهان با توزیع توام نرمال چند متغیره است و سه پارامتر توزیعی (μ, σ, ρ) که در ساختار توزیع نرمال تعریف شده‌اند. از آن‌جا که این پارامترها رابطه‌ای با هم ندارند، تنها تفاوت بین الگوریتم AB و الگوریتم Def این است که در الگوریتم AB سه پارامتر توزیعی در همان بلوک بزرگ پارامترهای پنهان (اثرات فضایی) قرار می‌گیرند. بنابراین اختلاف زمان ۵ ثانیه‌ای در اجرای الگوریتم (با زمان طولانی‌تر برای الگوریتم Def) به سه بار محاسبه کمتر تابع چگالی چندمتغیره (به ازای هر تکرار MCMC) تحت الگوریتم AB برمی‌گردد. از طرفی، در الگوریتم‌های Def و AB، چون پارامترها از هم مستقل‌اند، پس انتظار داریم که بلوک‌بندی بر روی آن‌ها اثر منفی داشته باشد. به همین دلیل این دو دارای ESS کوچکتری هستند. با حذف یکی از سه پارامتر در AB مقدار می‌نیم ESS دو برابر می‌شود. الگوریتم Def آمیختگی بهتری نسبت به AB دارد. دلیل آن هم حضور سه

پارامتر مستقل در یک بلوک است که به عملکرد خوبی منتهی نمی‌شود.

در نمونه‌گیری AS تمام مولفه‌ها مستقل از هم هستند و هر نمونه از یک چگالی چند متغیره استخراج می‌شود. بنابراین زمان اجرای این الگوریتم افزایش می‌یابد. کمترین مقدار ESS برای اثرات فضایی مربوط به الگوریتم‌های AB و Def است. پارامتر ρ نیز کمترین آمیختگی را تحت الگوریتم AS داراست که مقدار ESS آن از $۱۳۹/۶$ به $۱۷۱/۳$ (تحت الگوریتم Def) افزایش یافته است. این مثالی از تاثیر پارامتر با کمترین آمیختگی بر روی کارایی الگوریتمی سایر پارامترهای مدل است.

اگر در روش بلوک‌بندی خودکار ارتفاع برش $h = ۰/۱$ انتخاب شود، آن‌گاه یک بلوک شامل ρ و σ خواهیم داشت که نشان می‌دهد همبستگی پسین تجربی بین این دو پارامتر حداقل $۰/۹$ است. یک پارامتر پنهان کمترین آمیختگی را با ESS ای برابر ۱۲۰۸ تولید می‌کند. همچنین کارایی بلوک‌بندی خودکار تقریباً ۱۰ برابر کارایی الگوریتم AS است. زمان اجرای این الگوریتم در مقایسه با الگوریتم AS کاهش یافته است، زیرا تولید نمونه‌های یک بلوک از چگالی چند متغیره پنهان به زمان کمتری نیاز دارد.

جدول ۶.۴: معیارهای کارایی برای مثال مدل فضایی برای گونه‌های گوش ماهی

مدل	طرح MCMC	ESS	Runtime	Efficiency
فضایی	AB	$۰/۲$	$۵/۷۱$	$۰/۰۴$
	Def	$۰/۴$	$۱۰/۸۶$	$۰/۰۴$
	AS	$۱۷۱/۳$	$۸۳/۸۷$	$۲/۰$
	AUB	$۱۲۰۸/۰$	$۷۸/۶۲$	$۱۵/۴$

۵.۳.۴ مدل آمیخته خطی تعمیم یافته

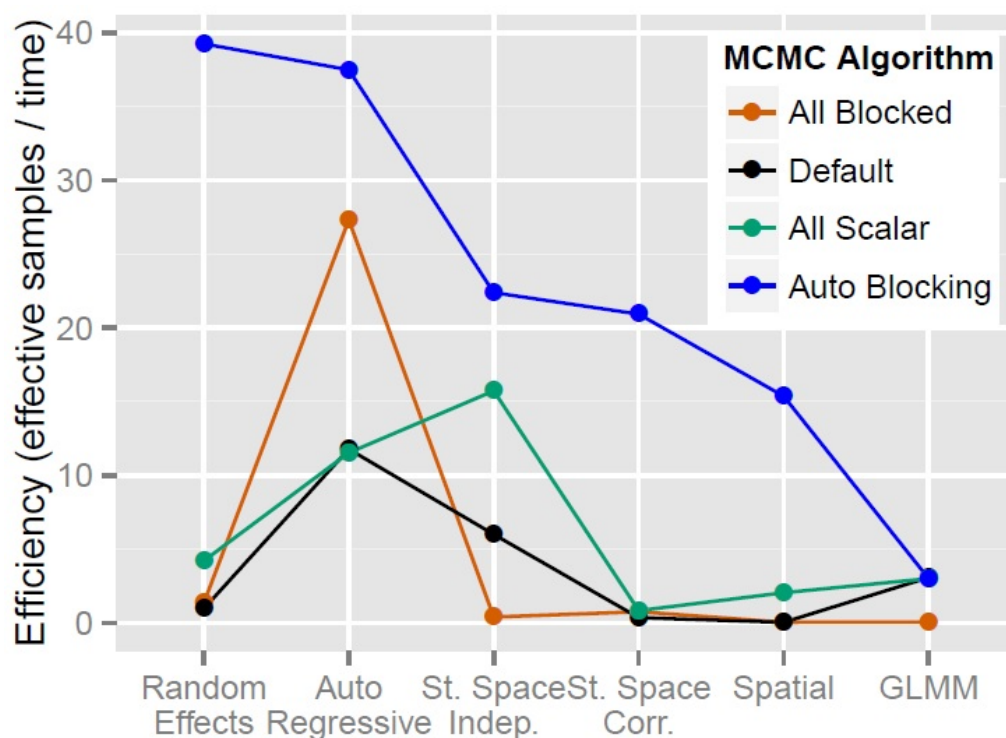
مدل GLMM شامل ۲۰۰۰ مولفه تصادفی است. در این مدل فرآیند بلوک‌بندی خودکار به الگوریتم AS که همان حالت اولیه فرآیند بلوک‌بندی خودکار است، همگراست و کارایی کل هر دو الگوریتم حدود ۳ است. اثرات تصادفی در این مدل سبب همبستگی میان پارامترهای نمونه‌ای در مجموعه داده‌ها نمی‌شود. به همین منظور، برای تعداد زیادی از اثرات تصادفی ناهمبسته، در غیاب یک توزیع چند متغیره، نمونه‌گیری تک متغیره بیشترین کارایی را تولید می‌کند. الگوریتم AB شامل یک تک بلوک با ۸۵۸ بعد است که زمان اجرای آن تقریباً دو برابر زمان اجرای الگوریتم AS است و کارایی کل حدود $۰/۰۵$ تولید می‌کند.

جدول ۷.۴: معیارهای کارایی برای مثال مدل آمیخته خطی تعمیم‌یافته برای داده‌های مراجعه‌کنندگان سالخورده

مدل	طرح MCMC	ESS	Runtime	Efficiency
GLMM	AB	۲/۲	۴۴/۳	۰/۰۵
	AS	۶۰/۹	۲۲/۶	۳/۰
	AUB	۶۰/۹	۲۲/۶	۳/۰

۴.۴ کارایی حاصل از بلوک‌بندی خودکار

شکل ۱.۴، کارایی کل به‌دست آمده برای هر پنج مثال را با هم نشان می‌دهد. در هر مثال کارایی الگوریتم بلوک‌بندی خودکار نسبت به هر یک از الگوریتم‌های غیرخودکار حدود ۵۰ درصد افزایش یافته است، به‌جز مثال GLMM که الگوریتم‌های بلوک‌بندی خودکار منطبق بر الگوریتم AS است. تغییراتی که در کارایی نسبی الگوریتم‌های غیرخودکار در این مثال‌ها مشاهده می‌شود، این تصور که کارایی کل الگوریتم‌های MCMC به شدت به ساختار مدل سلسله‌مراتبی وابسته است را تقویت می‌کند.

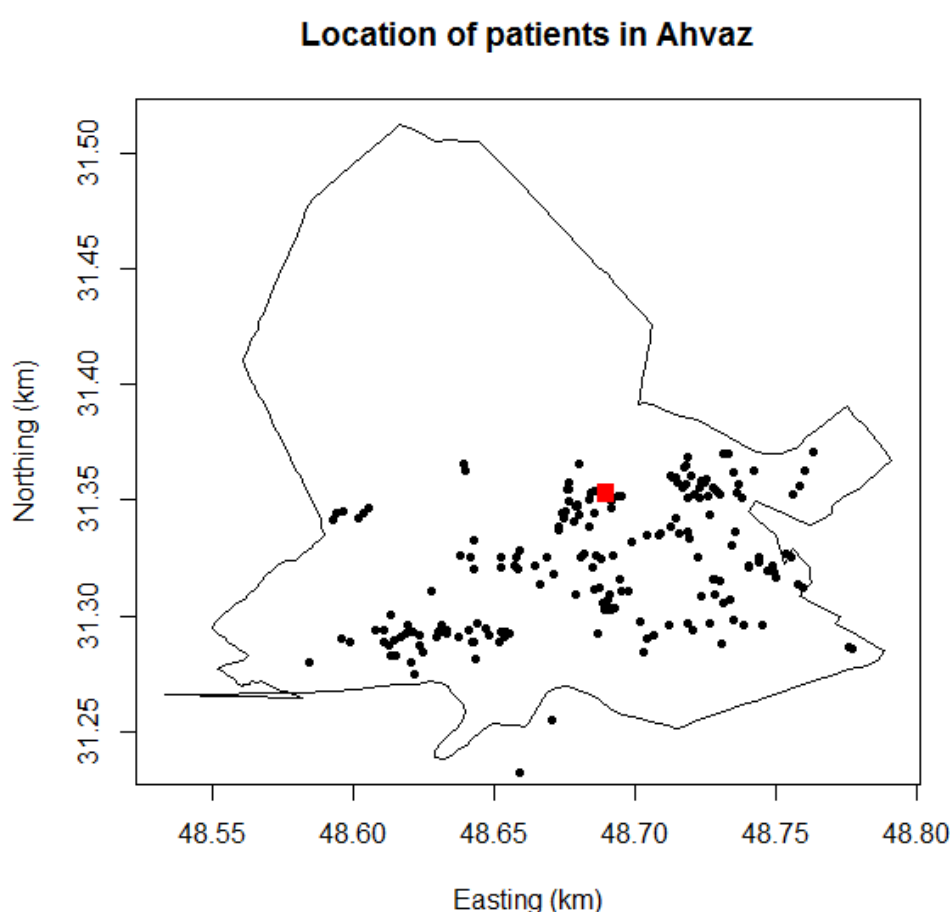


شکل ۱.۴: کارایی الگوریتم‌های MCMC برای مجموعه مثال‌ها

به‌طور کلی استنباط در مورد یک الگوریتم MCMC کارا برای یک مسئله خاص بسیار دشوار است.

۵.۴ تعداد دفعات مراجعه بیماران شهر اهواز

در این بخش عملکرد الگوریتم بلوک‌بندی خودکار را بر روی یک مجموعه داده واقعی بومی ایران تشریح می‌کنیم. این مجموعه داده مربوط به اطلاعات ثبت‌شده بخشی از بیماران مراجعه‌کننده به بیمارستان آریا (با حجم ۲۰۰) در شهر اهواز است.



شکل ۲.۴: محل سکونت بیماران مراجعه‌کننده به بیمارستان آریا اهواز. مربع بزرگ قرمز محل بیمارستان آریا را نشان می‌دهد.

شکل ۲.۴ مکان‌های سکونت این بیماران را در بخش‌های جنوبی و مرکزی شهر اهواز نشان می‌دهد. نقطه‌ای که با یک مربع بزرگ قرمز رنگ نمایش داده شده است، موقعیت جغرافیایی بیمارستان آریا در شهر اهواز است. متغیر پاسخ در این مجموعه داده، تعداد دفعات مراجعه به

بیمارستان و متغیرهای تبیینی شامل جنسیت، سن، بخش پذیرش و وضعیت تاهل هستند. دو متغیر تبیینی جنسیت و وضعیت تاهل دودویی، سن پیوسته و بخش پذیرش یک متغیر رسته‌ای با شش دسته جراحی، مادران و نوزادان، CCU، شیمی درمانی، اورژانس و داخلی هستند. اطلاعات متغیرهای مختلف در جدول ۸.۴ فهرست شده‌اند.

جدول ۸.۴: متغیرهای مختلف مجموعه داده بیماران مراجعه‌کننده به بیمارستان آریا

اسم متغیر	محتوی
age	سن بیمار
sex	جنسیت: (۰) زن (۱) مرد
married	وضعیت تاهل: (۰) مجرد (۱) متاهل
section	بخش پذیرش: (۰) جراحی (۱) زنان و نوزادان (۲) CCU (۳) شیمی درمانی (۴) اورژانس (۵) داخلی

با توجه به ماهیت فضایی داده‌ها و شمارشی بودن متغیر پاسخ، از یک مدل فضایی مبتنی بر توزیع پواسون برای تحلیل آن‌ها استفاده می‌کنیم. برای وارد کردن متغیر بخش پذیرش که یک متغیر رسته‌ای با ۶ سطح است، از متغیرهای ظاهری^۸ استفاده کردیم که با توجه به تعداد سطوح متغیر section، پنج متغیر ظاهری (با اسم‌های section_۱ تا section_۵) تعریف شدند. لازم به ذکر است که برای تعریف متغیرهای ظاهری، بخش جراحی به عنوان بخش مبنا انتخاب شد. یعنی تاثیر سایر بخش‌های پذیرش بر پاسخ نسبت به این بخش سنجیده می‌شود.

اثر فضایی حاکم بر داده‌ها را با یک میدان تصادفی گاوسی مدل‌بندی کردیم که برای ساختار وابستگی آن یک مدل نمایی را در نظر گرفتیم. با توجه به یک تحلیل اولیه، با فرض استقلال بین مشاهدات، توسط یک مدل رگرسیون پواسون کلاسیک، به این نتیجه رسیدیم که متغیر تبیینی سن و متغیر ظاهری (ساختگی) section_۴ کاملاً بی‌تاثیر هستند. بنابراین آن‌ها را از

^۸Dummy variables

مدل خارج کردیم. در نهایت مدل سلسله‌مراتبی زیر را برای برازش داده‌ها به کار گرفتیم:

$$y_i \sim \text{Poisson}(\lambda_i), \quad i = 1, \dots, 200$$

$$\log(\lambda_i) = \beta_0 + \beta_1 \text{sex}_i + \beta_2 \text{married}_i + \beta_3 \text{section}_1 i$$

$$+ \beta_4 \text{section}_2 i + \beta_5 \text{section}_3 i + \beta_6 \text{section}_4 i + \delta_i$$

که در آن $y_i = y(s_i)$ متغیر پاسخ برای بیمار i ام با محل سکونت s_i است که تعداد دفعات مراجعه به بیمارستان آریا است، $(\beta_0, \beta_1, \dots, \beta_6)'$ ضرایب رگرسیونی و $\delta_i = \delta(s_i)$ میدان تصادفی فضایی با ساختار زیر است:

$$\delta(s_i) \sim \text{GRF}(\circ, \Sigma)$$

$$\Sigma = (\sigma_{ij}) = \sigma^2 e^{-\frac{\|s_i - s_j\|}{\rho}}.$$

برای همه پارامترهای رگرسیونی از توزیع پیشین نرمال با میانگین صفر و واریانس ۱۰۰ استفاده کردیم. همچنین برای پارامترهای ساختار وابستگی فضایی مدل توزیع‌های پیشین $\sigma \sim \text{Unif}(0.001, 0.5)$ و $\rho \sim \text{Unif}(0.01, 1.0)$ را در نظر گرفتیم. برای این مجموعه داده، چهار الگوریتم MCMC را برای تولید نمونه از توزیع پسین حاصل، در نظر گرفتیم: (۱) الگوریتم تک بلوکی، (۲) الگوریتم مولفه به مولفه، (۳) الگوریتم پیش‌فرض (که در آن سه بلوک برای به ترتیب اثرات فضایی، پارامترهای رگرسیونی، و پارامترهای وابستگی در نظر گرفته می‌شود) و (۴) الگوریتم بلوک‌بندی خودکار. نتایج معیارهای کارایی برای این چهار الگوریتم در جدول ۹.۴ گزارش شده‌اند.

جدول ۹.۴: معیارهای کارایی برای داده‌های تعداد مراجعه بیماران شهر اهواز

الگوریتم MCMC	ESS	Runtime	Efficiency
AB	۱/۷۰	۳۹/۲۷	۰/۰۴
Def	۸/۷۲	۷۲/۵۹	۰/۱۲
AS	۱۹/۳۱	۸۸۳/۲۵	۰/۰۲
AUB	۸۷/۳۹	۶۴۵/۴۲	۰/۱۴

با توجه به نتایج جدول می‌توان گفت

۱. کاراترین الگوریتم از منظر معیار کارایی کل، الگوریتم بلوک‌بندی خودکار هست.
۲. الگوریتم پیش‌فرض عملکردی نزدیک به الگوریتم AUB دارد، ولی از نظر معیار کارایی الگوریتمی، روش AUB حداقل ۴ برابر کاراتر از سایر روش‌های نمونه‌گیری MCMC است.
۳. در مقابل، دو الگوریتم دیگر کارایی کل بسیار ضعیفی دارند.

با اجرای الگوریتم نمونه‌گیری خودکار، زنجیری به طول ۲۲ هزار و دوره سوزاندن ۲۰۰۰، و انتخاب طول گام ۱۰، اجرا شد. بنابراین نتایج استنباط بر اساس یک نمونه نهایی ۲۰۰۰ تایی به‌دست آمدند. جدول ۱۰.۴ نتایج استنباط پسینی حاصل از نمونه‌های نهایی را نشان می‌دهد. در این جدول، هر دو معیار میانگین و میانه توزیع پسین را به‌عنوان برآورد پارامترهای

جدول ۱۰.۴: نتایج استنباط پسینی حاصل از اجرای الگوریتم نمونه‌گیری خودکار برای داده‌های تعداد مراجعه بیماران شهر اهواز

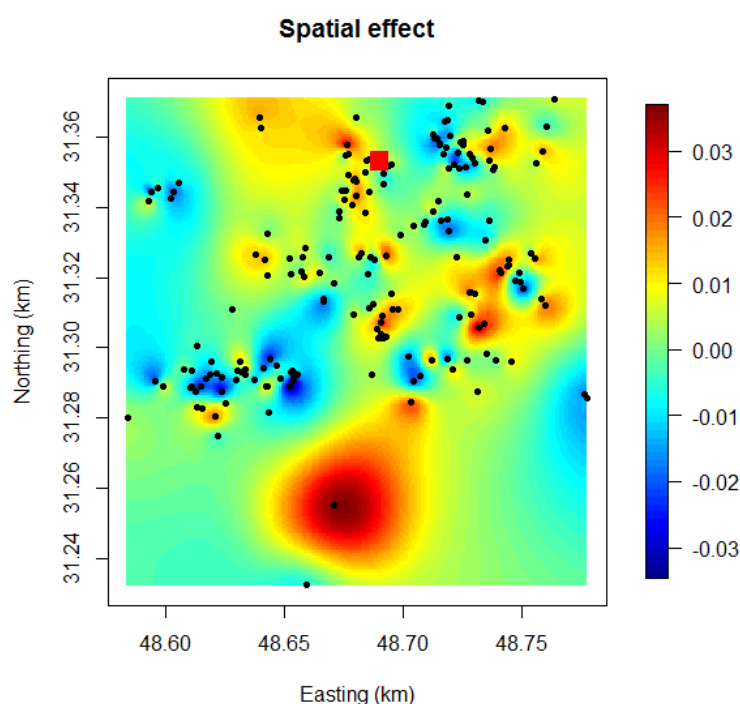
پارامترها	برآورد			فاصله اعتبار ۹۵٪ (L, U)
	میانگین پسینی	میانه پسینی	SE	
β_0	۰/۱۱۵	۰/۱۱۸	۰/۲۱۴	(-۰/۳۲۵, ۰/۵۳۳)
β_1	-۰/۰۶۰	-۰/۰۶۰	۰/۱۵۳	(-۰/۳۵۸, ۰/۲۲۲)
β_2	۰/۱۶۰	۰/۱۶۱	۰/۲۱۰	(-۰/۲۴۵, ۰/۵۶۸)
β_3	-۰/۱۹۸	-۰/۱۹۳	۰/۱۷۰	(-۰/۵۳۸, ۰/۱۳۵)
β_4	۰/۴۳۴	۰/۴۳۹	۰/۲۵۷	(-۰/۰۶۱, ۰/۹۳۲)
β_5	۲/۱۲۱	۲/۱۲۵	۰/۲۲۶	(۱/۶۶۵, ۲/۵۵۶)
β_6	-۰/۲۵۰	-۰/۲۲۹	۰/۳۱۵	(-۰/۸۹۳, ۰/۳۳۲)
ρ	۰/۳۷۵	۰/۲۱۶	۰/۴۸۲	(۰/۱۰۶, ۲/۳۱۹)
σ	۰/۳۹۵	۰/۳۸۸	۰/۰۶۷	(۰/۲۸۱, ۰/۵۵۰)

مدل گزارش کرده‌ایم. فاصله‌های اعتبار ۹۵٪ را نیز محاسبه و گزارش کرده‌ایم. با توجه به نتایج جدول، تنها متغیر تبیینی بستری در بخش شیمی درمانی (نسبت به سایر بخش‌ها) بر متوسط تعداد دفعات مراجعه بیماران معنی‌دار است و اثری مثبت دارد. این نتیجه به دلیل عدم معنی‌داری سایر بخش‌ها (نسبت به بخش جراحی) معتبر است. زیرا سایر بخش‌ها نسبت به بخش جراحی تفاوتی معنی‌دار در متوسط تعداد مراجعه ایجاد نمی‌کنند. پس اختلاف معنی‌دار بخش شیمی درمانی با بخش جراحی به سایر بخش‌ها نیز توسعه می‌یابد. این نتیجه کاملاً هم منطقی است؛ زیرا بیمارانی که نیازمند شیمی درمانی هستند، به‌طور منظم باید در زمان‌هایی معین به بیمارستان مراجعه کنند. ولی نتیجه غافلگیرکننده آن است که سایر بخش‌های بستری بیمارستان به نظر تأثیری بر تعداد دفعات مراجعه ندارند و بیماران فقط بر حسب اضطرار به بیمارستان مراجعه می‌کنند. کما این که ممکن است به بیمارستان‌های دیگر نیز مراجعه کنند و الگوی خاصی برای بیمارستان آریا وجود ندارد. البته این نتیجه بر حسب یک نمونه ۲۰۰ تایی از همه بیماران مراجعه‌کننده است و با در نظر گرفتن همه بیماران، نتیجه‌گیری نهایی می‌تواند تعدیل شود.

از طرفی با توجه به مقدار برآوردشده برای پارامتر دامنه ρ ، می‌توان تقریباً وابستگی فضایی ضعیف مراجعه‌کنندگان به این بیمارستان را نتیجه گرفت. مقدار برآوردشده پارامتر واریانس

نیز بزرگ است که یک دلیل منطقی برای آن نزدیک بودن محل‌های سکونت این نمونه ۲۰۰ تایی به همدیگر است و بیانگر آن است که مقداری از آن ناشی از تغییرات پاسخ در مکان‌های نزدیک به هم است.

شکل ۳.۴ نقشه پهنه‌بندی اثر فضایی برآوردشده داده‌ها به همراه مکان‌های سکونت بیماران را نشان می‌دهد. از روی این شکل کاملاً واضح است که اثر فضایی برای این داده‌ها معنی‌دار است و محل سکونت افراد و به‌ویژه فاصله تا محل بیمارستان می‌تواند نقش موثری بر تعداد دفعات مراجعه داشته باشد. البته اثر فضایی در بخش جنوبی ناحیه تحت مطالعه به‌صورت فعال بزرگ است و به اصطلاح یک نقطه داغ^۹ را نشان می‌دهد. به نظر می‌رسد که شاخص فاصله محل سکونت تا بیمارستان، می‌تواند عاملی تاثیرگذار بر تعداد مراجعه بیماران باشد. پس به‌عنوان مرحله بعدی تحلیل دقیق‌تر داده‌ها، شاید وارد کردن فاصله محل سکونت بیماران تا بیمارستان، به‌عنوان یک متغیر تبیینی، کاملاً تاثیرگذار باشد.



شکل ۳.۴: اثر فضایی بیماران مراجعه‌کننده به بیمارستان آریا اهواز

^۹Hot spot

۶.۴ نتیجه‌گیری و آینده تحقیق

در این پایان‌نامه یک الگوریتم MCMC کارا برای بلوک‌بندی خودکار پارامترهای مدل‌های سلسله‌مراتبی معرفی کردیم که در این روش از همبستگی پسین تجربی پارامترهای گروه‌های خوشه‌ای در نمونه‌گیری بلوکی استفاده شده است. یک درخت خوشه‌بندی سلسله‌مراتبی از پارامترهای مدل تشکیل می‌شود که با برش این درخت در هر ارتفاعی، پارامترهای موجود هر بلوک به دست می‌آیند، به طوری که بلوک‌ها دارای کمترین همبستگی پسینی هستند.

با استفاده از مثال‌های متنوع از مدل‌های پیچیده آماری، نشان دادیم کارایی الگوریتم پیشنهادی در این پایان‌نامه از تمام الگوریتم‌های MCMC استاندارد بالاتر است. می‌توان عملکرد الگوریتم پیشنهادی را با استفاده از سایر روش‌های خوشه‌بندی سلسله‌مراتبی نیز ارزیابی کرد. انواع روش‌های خوشه‌بندی شامل پیوند کامل، پیوند میانگین، پیوند تکی و معیار Ward (ایورت، ۲۰۱۱) است. در این پایان‌نامه، به صورت پیش‌فرض، روش خوشه‌بندی پیوند کامل در نظر گرفته شده است. برای آینده تحقیق، در تشکیل درخت سلسله‌مراتبی از سایر روش‌های خوشه‌ای نام برده شده می‌توان استفاده کرد.

پیوست آ

راهنمای استفاده از NIMBLE

۱. آ مقدمه

کلمه NIMBLE شامل حروف اختصاری کلمه‌های

Numerical Inference for Statistical Models using Bayesian and Likelihood Estimation

است، همچنین NIMBLE محیطی برای تحلیل مدل‌های آماری R، به‌ویژه مدل‌های سلسله‌مراتبی و روش‌های فشرده محاسباتی است. با وجود این که NIMBLE در R قرار دارد، اما با برنامه‌نویسی جداگانه مدل‌ها و الگوریتم‌ها، همراه با کامپایل کردن آن‌ها برای اجرای سریع، نسبت به R دارای عملکرد بهتری است. همچنین، از NIMBLE برای استنباط‌های عددی برآوردهای درست‌نمایی و بیزی، در مدل‌های آماری استفاده می‌شود. NIMBLE متن‌باز^۱ است و یک زبان برای نوشتن الگوریتم‌های مدل ارائه می‌دهد که از چهار جزء تشکیل شده است.

۱. محیطی برای نوشتن مدل‌های آماری، که توسعه‌یافته زبان BUGS است.

۲. شامل الگوریتم‌های فراوانی از جمله الگوریتم MCMC است.

۳. زبان NIMBLE مانند R است، و برای نوشتن الگوریتم مدل‌هایی که در BUGS عمل می‌کنند، به کار می‌رود.

^۱Open source

۴. یک کامپایلر است که ++C را برای مدل‌ها و الگوریتم‌ها تولید و کامپایل می‌کند، و به شما اجازه استفاده متوالی از R را بدون آن که اطلاعی از ++C داشته‌باشد می‌دهد.

۲.۱ مراحل ساخت مدل در محیط NIMBLE

برای تشریح مراحل ساخت و اجرای یک مدل در محیط NIMBLE از مجموعه داده‌های Pump موجود در مجموعه مثال‌های winBUGS (اسپیگل‌هالتر و همکاران، ۱۹۹۶) استفاده می‌کنیم. این مجموعه داده مربوط به ۱۰ پمپ کارخانه برق است. جدول ۱.۱ داده‌های این مجموعه را نشان می‌دهد. برای این داده‌ها فرض می‌کنیم مشاهدات، که تعداد پمپ‌های معیوب هستند از توزیع پواسون پیروی می‌کنند. همچنین، فرض کنید θ_i نرخ خرابی و i طول عمر پمپ i ام (در ۱۰۰۰ ثانیه از ساعت) باشد. بنابراین

$$x_i \sim \text{Poisson}(\theta_i t_i)$$

$$\theta_i \sim \text{Gamma}(\alpha, \beta) \quad i = 1, \dots, 10$$

که در آن پارامترهای α و β به ترتیب دارای توزیع $\text{Exp}(1)$ و $\text{Gamma}(0.1, 1)$ هستند. این مدل توسط جورج و همکاران (۱۹۳۱) تحلیل شده است.

جدول ۱.۱: مجموعه داده‌های Pump

پمپ	۱	۲	۳	۴	۵	۶	۷	۸	۹	۱۰
t_i	۹۴/۵	۱۵/۷	۶۲/۹	۱۲۶	۵/۲۴	۳۱/۴	۱/۰۵	۱/۰۵	۲/۱	۱۰/۵
θ_i	۵	۱	۵	۱۴	۱۳	۱۹	۱	۱	۴	۲۲

برای ساختن مجموعه داده‌های Pump ابتدا کد مدل که شامل ثابت‌ها، داده‌ها و مقدار اولیه هستند را می‌نویسیم. تعریف تابع بیز سلسله‌مراتبی و توابع پیشین و درست‌نمایی در nimbleCode صورت می‌گیرد.

```
pumpCode <- nimbleCode({
  for (i in 1:N){
    theta[i] ~ dgamma(alpha,beta)
    lambda[i] <- theta[i]*t[i]
    x[i] ~ dpois(lambda[i])
  }
  alpha ~ dexp(1.0)
  beta ~ dgamma(0.1,1.0)
})
```

در این جا $x[i]$ تعداد پمپ‌های معیوب ثبت شده در طول زمان $t[i]$ برای i امین پمپ و $\theta[i]$ نرخ خرابی پمپ i ام است. در این مثال هدف، برآورد پارامترهای α و β است.

برای ایجاد یک مدل NIMBLE از کد BUGS از تابع `nimbleModel` استفاده می‌کنیم

`nimbleModel (code, constants=list(), data=list(), inits=list(), name=NULL)`

آرگومان‌های این تابع به شرح زیر می‌باشد :

- `code`: همان کد مدلی است که در قسمت `nimbleCode` تعریف کرده‌ایم.
- `constants`: فهرستی از مقادیر ثابت در مدل هستند که آن‌ها را نمی‌توان تغییر داد. برای سازگاری بیشتر با BUGS و JAGS می‌توان مقادیر ثابت را در بخش معرفی داده‌ها مشخص کرد. به عنوان مثال مقادیر متغیرهای تبیینی در یک مدل رگرسیونی که از قبل ثابت‌های مشخصی هستند.
- `data`: داده‌ها به صورت `list` نمایش داده می‌شود. نام همه‌ی داده‌های (متغیرهای) معرفی شده در NIMBLE باید در بخش مدل حضور داشته باشد. همچنین در NIMBLE داده‌های گمشده با NA معرفی می‌شود.
- `inits` شامل مقادیر اولیه برای شروع نمونه‌گیری است، که نحوه معرفی آن به شکل لیست می‌باشد. مقادیر اولیه برای تمام گره‌های تصادفی (به جز متغیرها) در نظر گرفته می‌شوند (در واقع برای همه پارامترها لازم است).

```
pumpConsts <- list(N = 10,
t = c(94.3, 15.7, 62.9, 126, 5.24,
31.4, 1.05, 1.05, 2.1, 10.5))
pumpData <- list(x = c(5, 1, 5, 14, 3, 19, 1, 1, 4, 22))
pumpInits <- list(alpha = 1, beta = 1,
theta = rep(0.1, pumpConsts$N))
pump <- nimbleModel(code = pumpCode, name = "pump", constants = pumpConsts,
data = pumpData, inits = pumpInits)
```

برای مشاهده گره‌ها داریم:

```
pump$getNodeNames()
[1] "alpha"                "beta"                  "lifted_d1_over_beta"
[4] "theta[1]"              "theta[2]"              "theta[3]"
[7] "theta[4]"              "theta[5]"              "theta[6]"
[10] "theta[7]"              "theta[8]"              "theta[9]"
```

[13] "theta[10]"	"lambda[1]"	"lambda[2]"
[16] "lambda[3]"	"lambda[4]"	"lambda[5]"
[19] "lambda[6]"	"lambda[7]"	"lambda[8]"
[22] "lambda[9]"	"lambda[10]"	"x[1]"
[25] "x[2]"	"x[3]"	"x[4]"
[28] "x[5]"	"x[6]"	"x[7]"
[31] "x[8]"	"x[9]"	"x[10]"

مقادیر لگاریتم احتمال درستنمایی x برابر است با:

```
pump$logProb_x
[1] -2.998011 -1.118924 -1.882686 -2.319466 -4.254550 -20.739651
[7] -2.358795 -2.358795 -9.630645 -48.447798

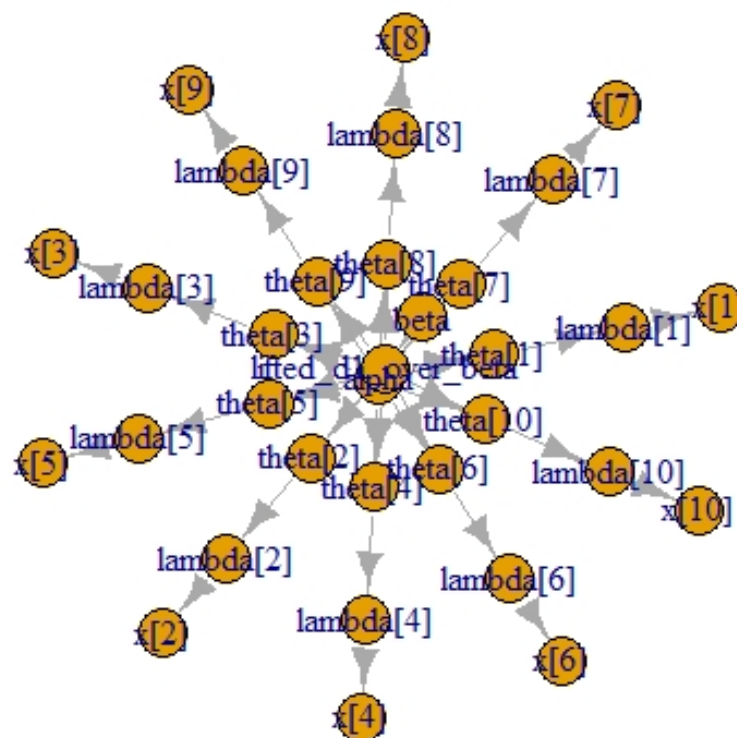
pump$alpha
[1] 1

pump$theta
[1] 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1

pump$lambda
[1] 9.430 1.570 6.290 12.600 0.524 3.140 0.105 0.105 0.210 1.050
```

از بسته igraph برای رسم نمودار در NIMBLE استفاده می‌شود که خروجی آن مانند شکل زیر یک گراف بی‌دور جهتداری است.

```
pump$plotGraph()
```



قبل از اجرای الگوریتم MCMC بر روی مدل، ابتدا مدل را کامپایل می‌کنیم، یعنی تولید کد ++C، کامپایل کردن کد، بارگذاری و بازگرداندن آن به R با یک شی که می‌تواند مانند مدل غیرکامپایل استفاده شود. برای کامپایل کردن مدل از تابع compileNimble به صورت زیر استفاده می‌کنیم:

```
Cpump <- compileNimble(pump)
Cpump$theta
[1] 0.15514136 1.88240160 1.80451250 0.83617765 1.22254365 1.15835525
[7] 0.99001994 0.30737332 0.09461909 0.15720154
```

الگوریتم MCMC با استفاده از تابع nimbleMCMC اجرا می‌شود. ساختار این تابع به شکل زیر است:

```
nimbleMCMC (code, constants=list(), data=list(), inits, nchains=1, niter=10000, summary=FALSE, WAIC=TRUE)
```

● `nchain`: تعداد زنجیرهای MCMC برای اجرا که به‌طور پیش‌فرض یک در نظر گرفته می‌شود.

● `niter`: تعداد تکرارهای MCMC که به‌طور پیش‌فرض ۱۰۰۰۰ بار تکرار می‌شود.

● `WAIC`: اگر برابر با `TRUE` باشد، معیار `WAIC` مدل محاسبه می‌شود. اگر زنجیر چندگانه اجرا شود آن‌گاه معیار `WAIC` را با استفاده از نمونه‌های پسین زنجیرها محاسبه می‌کند.

● `summary`: اگر مقدار آن `TRUE` باشد در هر زنجیر MCMC خلاصه‌ای از نمونه‌های پسین هر پارامتر را نشان می‌دهد، اما به‌طور پیش‌فرض مقدار این آرگومان `FALSE` است.

این تابع قابلیت اجرای چندین زنجیر MCMC را دارد و تعداد تکرارهای MCMC، تاخیرها و سوزاندن را مشخص می‌کند. همچنین نمونه‌های پسین، خلاصه‌ای از نمونه‌های پسین و معیار `WAIC` را به‌عنوان خروجی نشان می‌دهد. برای مثال دو زنجیر MCMC هر کدام با ۱۰۰۰۰ نمونه را در نظر بگیرید.

```
mcmc.out <- nimbleMCMC(code = pumpCode, constants = pumpConsts,
data = pumpData, inits = pumpInits,
nchains = 2, niter = 10000,
summary = TRUE, WAIC = TRUE)
```

```
names(mcmc.out)
```

```
[1] "samples" "summary" "WAIC"
```

```
mcmc.out$summary
```

```
$`chain1`
```

	Mean	Median	St.Dev.	95%CI_low	95%CI_upp
alpha	0.6980435	0.6583506	0.2703768	0.2878982	1.314046
beta	0.9286260	0.8215685	0.5496913	0.1836991	2.287270

```
$chain2
```

	Mean	Median	St.Dev.	95%CI_low	95%CI_upp
alpha	0.6910196	0.6580365	0.2654838	0.2771956	1.285815
beta	0.9162727	0.8143443	0.5375082	0.1857723	2.270243

```
$all.chains
Mean      Median    St.Dev. 95%CI_low 95%CI_upp
alpha 0.6945316 0.6580365 0.2679578 0.2832985 1.299932
beta 0.9224494 0.8182816 0.5436554 0.1854908 2.278544
mcmc.out$waic
NULL
```

برای ایجاد یک الگوریتم MCMC بر روی مدل از تابع `configureMCMC` استفاده می‌کنیم که یکی از آرگومان‌های تابع `buildMCMC` است.

```
configureMCMC (model, thin=1, print= FALSE, autoBlock= FALSE, ...)
```

- `model`: مدلی که در `nimbleModel` تعریف شده است.
- `thin`: طول گام برای انتخاب نمونه MCMC است.
- `monitors`: در این قسمت مشخص می‌شود که از کدام پارامتر، نمونه تولید خواهد شد.
- `autoBlock`: اگر TRUE باشد، نمونه‌گیری بلوک‌بندی خودکار صورت می‌گیرد اما به‌طور پیش‌فرض مقدار این آرگومان FALSE است.

برای ایجاد یک تابع MCMC از یک مدل NIMBLE از تابع `buildMCMC` استفاده می‌شود. ساختار این تابع به‌صورت زیر است؛

```
buildMCMC (conf, ...)
```

برای اجرای یک یا چند زنجیر از الگوریتم MCMC و بازگرداندن نمونه‌ها، خلاصه‌ای از مقادیر توزیع پسین موردنظر و WAIC از تابع `runMCMC` استفاده می‌کنیم.

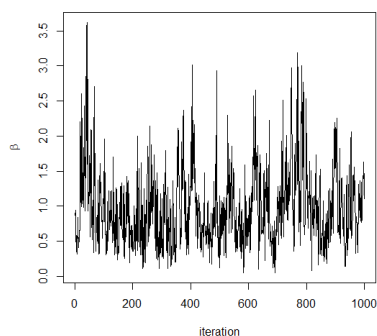
```
runMCMC (mcmc, niter= 10000, nburnin=0, thin, nchain=1, ...)
```

- `niter`: تعداد تکرارها برای اجرای هر زنجیر MCMC است که به‌طور پیش‌فرض ۱۰۰۰۰ در نظر گرفته شده است.
- `nburnin`: سوزاندن چند نمونه اول برای کاهش وابستگی بین نمونه‌ها در یک زنجیر MCMC.
- `nchain`: تعداد زنجیرهای MCMC برای اجرا، که به‌طور پیش‌فرض یک زنجیر در نظر گرفته شده است.

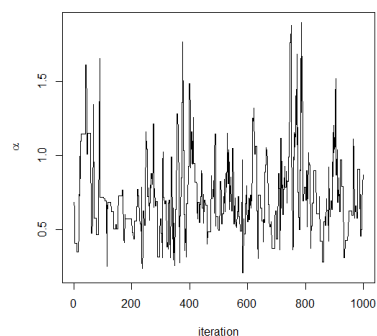
حال کاربرد این سه تابع را در مدل Pump نشان می‌دهیم.

```
> pumpConf <- configureMCMC(pump, print = TRUE)
[1] RW sampler: alpha
[2] conjugate_dgamma_dgamma sampler: beta
[3] conjugate_dgamma_dpois sampler: theta[1]
[4] conjugate_dgamma_dpois sampler: theta[2]
[5] conjugate_dgamma_dpois sampler: theta[3]
[6] conjugate_dgamma_dpois sampler: theta[4]
[7] conjugate_dgamma_dpois sampler: theta[5]
[8] conjugate_dgamma_dpois sampler: theta[6]
[9] conjugate_dgamma_dpois sampler: theta[7]
[10] conjugate_dgamma_dpois sampler: theta[8]
[11] conjugate_dgamma_dpois sampler: theta[9]
[12] conjugate_dgamma_dpois sampler: theta[10]
> pumpConf$addMonitors(c("alpha", "beta", "theta"))
thin = 1: alpha, beta, theta
> pumpMCMC <- buildMCMC(pumpConf)
> CpumpMCMC <- compileNimble(pumpMCMC, project = pump)
> niter <- 1000
> set.seed(1)
> samples <- runMCMC(CpumpMCMC, niter = niter)

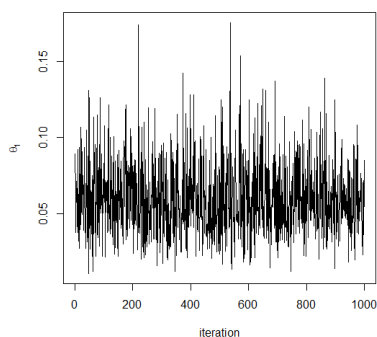
> par(mfrow = c(1, 4), mai = c(.6, .4, .1, .2))
> plot(samples[, "alpha"], type = "l", xlab = "iteration",
+ ylab = expression(alpha))
> plot(samples[, "beta"], type = "l", xlab = "iteration",
+ ylab = expression(beta))
> plot(samples[, "alpha"], samples[, "beta"], xlab = expression(alpha),
+ ylab = expression(beta))
> plot(samples[, "theta[1]"], type = "l", xlab = "iteration",
+ ylab = expression(theta[1]))
```



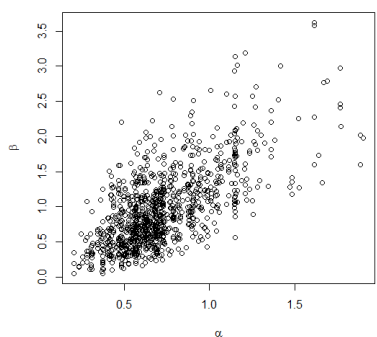
(ب)



(ا)



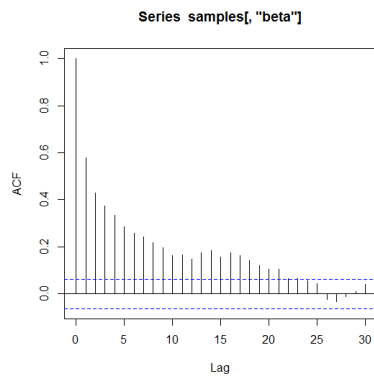
(د)



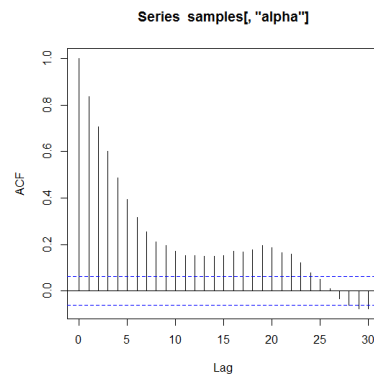
(ج)

تابع خودهمبستگی یک ابزار استاندارد در سری زمانی است که استقلال (همبستگی) مشاهدات را حول تغییرات زمانی اندازه می گیرد. همچنین یک معیار آمیختگی برای خودهمبستگی هر پارامتر توسط نمودارهای acf نشان داده می شود.

```
> acf(samples[, "alpha"])
> acf(samples[, "beta"])
```



(و)



(ه)

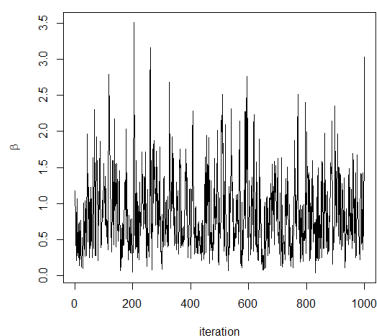
فرض کنید یک نمونه‌گیری بلوکی سازواری به α و β اضافه کنیم، می‌خواهیم بررسی کنیم که آیا نمونه‌گیری سازوار باعث بهبود آمیختگی می‌شود یا خیر.

```
> pumpConf$addSampler(target = c("alpha", "beta"), type = "RW_block",
+ control = list(adaptInterval = 100))
```

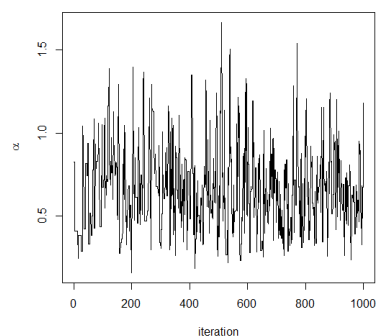
```
> pumpMCMC2 <- buildMCMC(pumpConf)
> CpumpNewMCMC <- compileNimble(pumpMCMC2, project = pump,
+ resetFunctions = TRUE)
```

```
> set.seed(1)
> CpumpNewMCMC$run(niter)
NULL
```

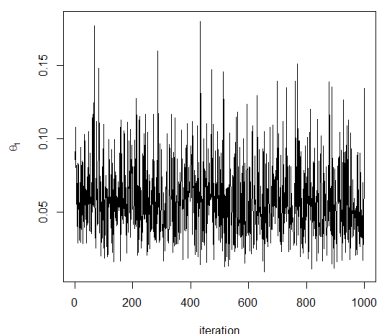
```
> samplesNew <- as.matrix(CpumpNewMCMC$mvSamples)
> par(mfrow = c(1, 4), mai = c(.6, .4, .1, .2))
> plot(samplesNew[, "alpha"], type = "l", xlab = "iteration",
+ ylab = expression(alpha))
> plot(samplesNew[, "beta"], type = "l", xlab = "iteration",
+ ylab = expression(beta))
> plot(samplesNew[, "alpha"], samplesNew[, "beta"], xlab = expression(alpha),
+ ylab = expression(beta))
> plot(samplesNew[, "theta[1]", type = "l", xlab = "iteration",
+ ylab = expression(theta[1]))
```



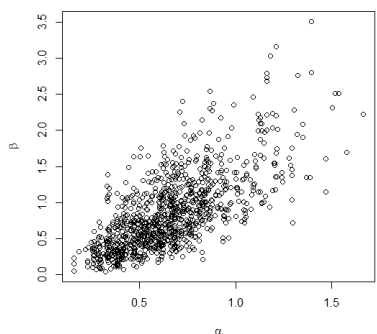
(ح)



(ز)



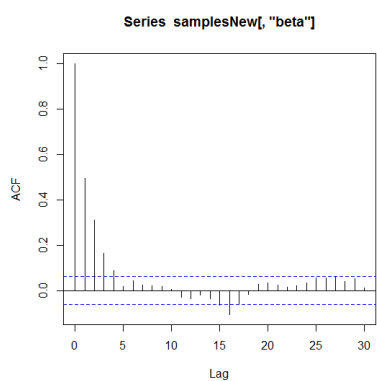
(ی)



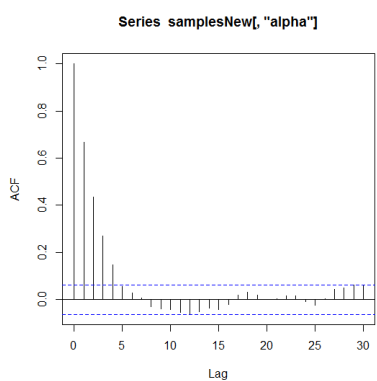
(ط)

```
> acf(samplesNew[, "alpha"])
```

```
> acf(samplesNew[, "beta"])
```



(ل)



(ک)

همان طور که مشاهده می شود نمونه گیری بلوکی خودهمبستگی α و β را کاهش می دهد.

مجموعه توزیع‌هایی که برای معرفی تابع درستنمایی و توزیع پیشین در NIMBLE وجود دارند، در راهنمای نرم‌افزار موجود هستند.

توزیع	نحوه استفاده	تابع چگالی
برنولی	<code>dbern(prob=p)</code>	$p^x(1-p)^{1-x}$
بتا	<code>dbeta(shape1=a, shape2=b)</code>	$\frac{x^{a-1}(1-x)^{b-1}}{\beta(a,b)}$
کای-دو	<code>dchisq(df=k)</code>	$\frac{x^{\frac{k}{2}-1} \exp(-x/2)}{2^{k/2} \Gamma(k/2)}$
نمایی	<code>dexp(rate=λ)</code>	$\lambda \exp(-\lambda x)$
گاما	<code>dgamma(shape=r, scale=λ)</code>	$\frac{\lambda^r x^{r-1} \exp(-\lambda x)}{\Gamma(r)}$
نرمال	<code>dnorm(mean=μ, tau=τ)</code>	$\left(\frac{\tau}{\pi}\right)^{\frac{1}{2}} \exp \frac{-\tau(x-\mu)^2}{2}$
پواسون	<code>dpois(lambda=λ)</code>	$\frac{\exp(-\lambda) \lambda^x}{x!}$

اما برای نوشتن یک تابع جدید در NIMBLE از `nimbleFunction` استفاده می‌شود. تعریف یک `nimbleFunction` ساده همانند تعریف یک تابع در R است. خروجی `nimbleFunction` یک تابع است و می‌توان آن را کامپایل کرد. ساختار این تابع به شکل زیر می‌باشد.

`nimbleFunction (setup= NULL, run= function= ()){} )`

یک مثال ساده برای محاسبه حداقل مربعات از پارامترهای رگرسیون خطی را بررسی می‌کنیم.

```
> solveLeastSquares <- nimbleFunction(
+ run = function(X = double(2), y = double(1)) {
+ ans <- inverse(t(X) %*% X) %*% (t(X) %*% y)
+ return(ans)
+ returnType(double(2))
+ } )
```

```
> X <- matrix(rnorm(400), nrow = 100)
> y <- rnorm(100)
> solveLeastSquares(X, y)
[,1]
[1,] -0.04923065
[2,] -0.01581343
[3,] 0.08329691
[4,] -0.11434964
```

```
> CsolveLeastSquares <- compileNimble(solveLeastSquares)
> CsolveLeastSquares(X, y)
[,1]
[1,] -0.04923065
[2,] -0.01581343
[3,]  0.08329691
[4,] -0.11434964
```

در این مثال یک مدل خطی به 100 متغیر پاسخ (y) برای تولید چهار متغیر تبیینی تصادفی (x) برازش می‌دهیم.

مراجع

- [۱] اسماعیلی، م. (۱۳۹۳). داده‌کاوی (مفاهیم و تکنیک‌ها)، چاپ سوم، واحد تولید انتشارات نیاز دانش.
- [۲] پارسیان، ا. (۱۳۹۱). مبانی آمار ریاضی، چاپ دهم، مرکز نشر دانشگاه صنعتی اصفهان.
- [۳] محمدزاده، م. (۱۳۹۱). آمار فضایی و کاربردهای آن، چاپ اول، انتشارات دانشگاه تربیت مدرس.
- [4] Abramaowitz, M. and Stegun, I. (1972). *Handbook of Mathematical Functions*, Courier Dover Publications, New York.
- [5] Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2003). *Hierarchical Modeling and Analysis for Spatial Data*, CRC Press, London.
- [6] Bayes, T. (1763). An essay toward solving a problem in a doctrine of chances, *In Philosophical Transactions of the Royal Society of London*, 53. 370-418.
- [7] Birkhoff, G. D. (1942). What is the ergodic theorem?, *American Mathematical Monthly*, 49(4), 222-226.
- [8] Breslow, N. E. and Clayton, D. G. (1993). Approximate inference in generallized linear mixed models, *Journal of the American Statistical Association*, 88, 9-25.
- [9] Brockwell, A.E., and Kadane, J.B. (2005). Identification of regeneration times in MCMC simulation, with application to adaptive schemes, *Journal of Computational and Graphical Statistics*, 14, 436-458.
- [10] Caffo, B. S., Jank, W., and Jones, G. L. (2005). Ascent-based Monte Carlo expectation-maximization, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, 67(2): 235-251.

-
- [11] Casella, G. and Berger, R. (2002). *Statistical Inference*, Thomson Learning, Duxbury.
 - [12] David, M. (1977). *Geostatistical Ore Reserve Estimation*, Elsevier, Amsterdam.
 - [13] De Valpine, P., Turek, D., Paciorek, C. J., Anderson-Bergman, C., Temple Lang, D. and Bodik, R. (2017). Programming with models: writing statistical algorithms for general model structures with NIMBLE, *Journal of Computational and Graphical Statistics*, 26(2), 403-413.
 - [14] Durbin, J. and Koopman, S. (2001). Time series analysis by state space methods, *Journal of the Royal Statistical Society (Series C)*, 47, 299-350.
 - [15] Everitt, B. (2011). *Cluster Analysis*, 5th Edition, Wiley Series in Probability and Statistics, Wiley, New York.
 - [16] Fisher, R. A. (1935). *The Design of Experiments*, Oliver And Boyd; Edinburgh, London.
 - [17] Gauss, C. F. (1821). *Theoria Combinationis Observationum Erroribus Minimis Obnoxiae*, Werke 4, Gottingen, New York.
 - [18] Gelfand, A. and Smith, A. (1990). Sampling based approaches to calculating marginal densities, *Journal of the American Statistical Association*, 85, 398-409.
 - [19] Gelman, A., Kass, R. E., Carlin, B. P., Neal, R. (1998). Markov chain Monte Carlo in practice: a roundtable discussion, *The American Statistician*, 52, 93-100.
 - [20] Gelman, A., Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences, *Statistical Science*, 7, 457-511.
 - [21] Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of image, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721-742.
 - [22] Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation, *Journal of the American Statistical Association*, 102(477), 359-378.
 - [23] George, E. I., Markov, U. E., and Smith, A. F. M. (1993). Conjugate likelihood distributions, *Scandinavian Journal of Statistics*, 20, 147-56.

- [24] Geweke, J. (1992). Evaluating the accuracy of sampling-based approaches to calculating posterior moments, *In Bayesian Statistics*, (eds JM Bernardo, JO Berger, AP Dawid and AFM Smith), 4, 169-193.
- [25] Gilks, W. R. (2005). *Markov Chain Monte Carlo*, In Encyclopedia of Biostatistics, John Wiley, New York.
- [26] Gilks, W. R., Roberts, G. O., and Sahu, S. K. (1998). Adaptive Markov chain Monte Carlo, *Journal of the American Statistical Association*, 93, 1045-1054.
- [27] Haario, H., Saksman, E., and Tamminen, J. (1999). Adaptive proposal distribution for random walk Metropolis algorithm, *Computational Statistics*, 14(3), 375-369.
- [28] Haario, H., Saksman, E., and Tamminen, J. (2001). An adaptive Metropolis algorithm, *Bernoulli Society for Mathematical Statistics and Probability*, 7, 223-242.
- [29] Harvey, A. (1989). *Forecasting, Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge.
- [30] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chain and their applications, *Biometrika*, 57(1), 97-109.
- [31] He, Q. (1999). A review of clustering algorithms as applied in IR, *Journal of the Graduate School of Library and Information Science University of illinois at Urbana-Campaign*, 6.
- [32] Heidelberger, P., Welch, P. D. (1983). Simulation run length control in the presence of an initial transient, *Operations Research*, 31, 1109-1144.
- [33] Hesterberg, T. C. (1987). Importance sampling in multivariate problems, *Proceedings of the Statistical Computing Section of the American Statistical Association*, 412-417.
- [34] Hill, B. M. (1975). A simple general approach to inference about the tail of a distribution, *The Annals of Statistics*, 3(5), 1163-1174.
- [35] Hjort, N. L., Dahl, F. A., and Steinbakk, G. H. (2006). Post-processing posterior predictive p-Values, *Journal of the American Statistical Association*, 101(475), 1157-1174.

- [36] Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems, *Proceedings of the Royal Society*, 186, 453-461.
- [37] Journel, A. G. and Huijbregts, C. J. (1978). *Mining Geostatistics*, Academic Press, London.
- [38] Lele, S. R., Dennis, B., and Lutscher, F. (2007). Data cloning: easy maximum likelihood estimation for complex ecological models using Bayesian Markov chain Monte Carlo methods, *Ecology Letters*, 10(7), 551–563.
- [39] Liu, J. S., Wong, W. H., and Kong, A. (1994). Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes, *Biometrika*, 81(1), 27–40.
- [40] Lunn, D., Jackson, C., Best, N., Thomas, A., and Spiegelhalter, D. (2012). *The BUGS Book: A Practical Introduction to Bayesian Analysis*, CRC Press, New York.
- [41] Lunn, D. J., Thomas, A., Best, N., and Spiegelhalter, D. (2000). WinBUGS - A Bayesian modelling framework: concepts, structure, and extensibility, *Statistics and Computing*, 10(4), 325–337.
- [42] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculation by fast computing machines, *The Journal of Chemical Physics*, 21(6), 1087-1092.
- [43] Mengersen, K. L. and Tweedie, R. L. (1996). Rates of convergence of the Hastings and Metropolis algorithm, *The Annals of the Statistics*, 24(1), 101-121.
- [44] Migon, H., Gamerman, D., Lopez, H., and Ferreira, M. (2005). *Bayesian Dynamic Models*, In D. Day and C. Rao (eds), Handbook of Statistics, 25, Elsevier B. V., Chapter 19, 553-588.
- [45] Mukerjee, R. and Ghosh, M. (1997). Second-order probability matching priors, *Biometrika*, 84(4), 970-975.
- [46] Neal, P. and Roberts, G. (2006). Optimal scaling for partially updating MCMC algorithms, *The Annals of Applied Probability*, 16, 475-515.
- [47] Nelder, J. and Wedderburn, R. (1972). Generalized linrear models, *Journal of the Royal Statistical Society, Series A*, 135, 370-384.

- [48] NIMBLE Development Team. (2018). NIMBLE: MCMC, particle filtering, and programmable hierarchical modeling, version 0.6-12. <https://cran.r-project.org/package=nimble>
- [49] Ntzoufras, I. (2011). *Bayesian Modeling Using WinBUGS*, John Wiley and Sons, Hoboken.
- [50] Plummer, N., Best, N., Cowles, K., Vines, K. (2006). The coda package, *R Project*, <http://cran.r-project.org/doc/packages/coda.pdf>.
- [51] Robert, C., Casella, G. (2010). *Introducing Monte Carlo Methods with R*, Springer, New York.
- [52] Robert, C. P., Casella, G. (2004). *Mont Carlo Statistical Methods*, Springer, New York.
- [53] Roberts, G. O., Gelman, A., and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis-Hastings algorithms, *The Annals of Applied Probability*, 7(1), 110-120.
- [54] Roberts, G. O. and Rosenthal, J. S. (2001). Optimal scaling for various Metropolis-Hastings algorithm, *Statistical Science*, 16(4), 351-367.
- [55] Roberts, G. O. and Rosenthal, J. S. (2005). Coupling and ergodicity of adaptive MCMC, *Journal of Applied Probability*, 44(2), 458-472.
- [56] Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive MCMC, *Journal of Computational and Graphical Statistics*, 18(2), 349-367.
- [57] Roberts, G. O. and Sahu, S. K. (1997). Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler, *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(2), 291-317.
- [58] Roberts, G. O. and Tweedi, R. L. (1996). Geometric convergence and central limit theorems for multidimensional Hastings and Metropolis algorithms, *Biometrika*, 83(1), 95-110.
- [59] Rosenthal, J.S. (2004). *Adaptive MCMC Java Applet*, New York.
- [60] Sargent, D. J., Hodges, J. S., and Carlin, B. P. (2000). Structured Markov chain Monte Carlo, *Journal of Computational and Graphical Statistics*, 9(2), 217-234.

-
- [61] Spiegelhalter, D., Thomas, A., Best, N., and Gilks, N. (1996). BUGS 0.5: Bayesian inference using Gibbs sampling manual (version ii), *Journal of MRC Biostatistics Unit, Institute of Public Health, Cambridge, UK*, 1-59.
 - [62] Straatsma, T., Berendsen, H., and Stam, A. (1986). Estimation of statistical errors in molecular simulation calculations, *Molecular Physics*, 57(1), 89-95.
 - [63] Student (1907). On the error of counting with a haemocytometer, *Biometrika*, 5, 351-360.
 - [64] Thompson, M. B. (2010). A comparison of methods for computing autocorrelation time, *arXiv preprint arXiv:1011.0175*.
 - [65] Tierney, L. (1994). Markov chains for exploring posterior distributions (with discussion), *Annals of Statistics*, 22, 1701-1762.
 - [66] Tierney, L. and Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities, *Journal of the American Statistical Association*, 81, 82-86.
 - [67] Trefethen, L. N. and Bau, D. (1997). *Numerical Linear Algebra*, 50, SIAM.
 - [68] Turek, D., de Valpine, P., Paciorek, C. J., and Anderson-Bergman, C. (2017). Automated parameter blocking for efficient Markov chain Monte Carlo sampling, *Bayesian Analysis*, 12, 465-490.
 - [69] Von Mises, R. (1931). *Wahrscheinlichkeitsrechnung und ihre Anwendungen in Der Statistik und Theoretischen Physik*, Deuticke, Vienna.
 - [70] Waller, L. A. and Zelterman, D. (1997). Log-linear modeling with the negative multinomial distribution, *Biometrics*, 53(3), 971-982.
 - [71] West, M., and Harrison J. (1997). *Bayesian Forecasting and Dynamic Models*, 2nd Edition, Springer, New York.
 - [72] Yang, R., and Berger, J. O. (1996). *A Catalog of Noninformative Priors*, Tech. Rep. Purdue University, USA.
 - [73] Web, A. R. (2003). *Statistical Pattern Recognition*, John Wiley and Sons.
 - [74] Wei, W. W. S. (2006), *Time Series Analysis: Univariate and Multivariate Methods*, Second Edition. Pearson/Addison Wesley.

Abstract

Markov chain Monte Carlo (MCMC) algorithms, such as Metropolis-Hastings algorithm and Gibbs sampling, are methods for producing samples from complex probability distributions. If the speed of convergence is slowed down in the MCMC methods, the algorithm runs for a long time. One of the reasons that make the convergence of the MCMC algorithms to be slow is separately updating the components of the distribution that are dependent. Group blocking of elements that are dependent is one of the conventional approaches to avoid such a problem. In this thesis, we represent an automated blocking method with a good performance to prevent subjective blocking. Finally, we evaluate the automated blocking process by using both real data and simulated examples and compare the results with those from standard MCMC algorithms.

Keywords: Metropolis-Hastings algorithm, Gibbs sampler, Automated blocking sampling, Dependent parameters, Efficiency of MCMC sampling methods.



Faculty of Mathematical Sciences

MSc Thesis in Statistics

Automated Parameter Blocking for Markov Chain Monte Carlo Sampling

By: Faezeh Kojoori

Supervisor:

Dr. Hossein Baghishani

January 2019