دانشکده مهندسی برق
گروه مخابرات

پایان‌نامه

کارشناسی در رشته
مهندسی برق، گرایش مخابرات

عنوان

بررسی و پیاده سازی الگوریتم های بیزوی در مدل سازی سیگنال های چند بعدی دارای همبستگی بین ابعادی

استاد راهنما

دکتر حمید شیخ زاده نجار

پژوهشگر

دانیال خشابی

۸۷۲۳۰۰۱

۱۳۹۱

نام خانوادگی دانشجو: خشابی نام: دانیال

عنوان: بررسی و پیاده سازی الگوریتم های بیزوی در مدل سازی سیگنال های چند بعدی دارای همبستگی بین ابعادی

استاد راهنما: دکتر حمید شیخ زاده نجار

مقطع تحصیلی: کارشناسی رشته: مهندسی برق گرایش: مخابرات

دانشگاه: دانشگاه صنعتی امیرکبیر دانشکده مهندسی برق

تاریخ فارغ‌التحصیلی: ۱۳۹۱ تعداد صفحات: ۸۵

واژگان کلیدی: روش های بیزی، یادگیری ماشین، تقریب توابع، داده های همبسته

چکیده

در این رساله یکی از مهم ترین مباحث اخیر مورد توجه در یادگیری ماشین، یعنی یادگیری بین ابعادی با استفاده از روش های بیزوی را مورد توجه قرار داده ایم. با توجه به اینکه هدف استفاده از روش های بیزوی است، بخش بسیار زیادی از این رساله به معرفی ایده ها و روش های یادگیری و مدل سازی بیزوی اختصاص یافته است. بعد از معرفی مهمترین الگوهای بیزوی شناخته شده، از این روش ها در مدل سازی بین ابعادی شده است. همانطور که در متن رساله معرفی شده، روش های بسیاری برای این هدف معرفی شده اند. اما در اینجا تنها به پیاده سازی عملی و مقایسه ی دو مورد اخیر از این موارد خواهیم پرداخت. در مورد دیگر روش ها، تنها به معرفی کوتاه آنها بسنده کرده ایم.

در فصل اول مفاهیم و ایده های کلی در یادگیری بیزوی، به همراه اهمیت و کاربردهای یادگیری با همبستگی بین ابعادی را معرفی می کنیم. در فصل دوم مهمترین الگوریتم های بیزوی برای یادگیری تک خروجی معرفی شده اند. در فصل سوم نمونه هایی از تعمیم الگوریتم های بیزوی متداول به چند متغیر خروجی را مورد توجه قرار می دهیم. در فصل چهار، دو مورد از مدل های اخیر برای مدل سازی همبستگی بین ابعادی را معرفی کرده و نتیجه ی عملکرد آنها روی چند دسته داده ی آزمایشی بررسی می کنیم. در فصل های ضمیمه، به ترتیب، فرمول های ریاضی پایه برای ساده سازی احتمالی و جبری فرمول ها؛ مهم ترین روش های بیزوی برای استنتاج بیزوی؛ و اثبات های طویل مربوط به برخی از الگوریتم های معرفی شده در متن رساله آورده شده اند.

تقدیم به پدر و مادر مهربان

و برادر عزیزم

خدایا...[1]

به من زیستنی عطا کن که در لحظه مرگ، بر بی‌ثمری لحظه‌ای که برای زیستن گذشته است، حسرت نخورم و مُردنی عطا کن که بر بیهودگیش، سوگوار نباشم. بگذار تا آن را، خود انتخاب کنم، اما آنچنان که تو دوست می‌داری.

تو می‌دانی و همه می‌دانند که شکنجه دیدن بخاطر تو، زندانی کشیدن بخاطر تو و رنج بردن به پای تو تنها لذت بزرگ زندگی من است، از شادی توست که من در دل می‌خندم، از امید رهایی توست که برق امید در چشمان خسته‌ام می‌درخشد و از خوشبختی توست که هوای پاک سعادت را در ریه‌هایم احساس می‌کنم. نمی‌توانم خوب حرف بزنم. نیروی شگفتی را که در زیر کلمات ساده و جمله‌های ضعیف و افتاده، پنهان کرده‌ام دریاب، دریاب.

تو می‌دانی و همه می‌دانند که زندگی از تحمیل لبخندی بر لبان من، از آوردن برق امیدی در نگاه من، از برانگیختن موج شعفی در دل من، عاجز است.

تو، چگونه زیستن را به من بیاموز، چگونه مردن را خود خواهم آموخت.

به من توفیق تلاش در شکست، صبر در نومیدی، رفتن بی‌همراه، جهاد بی‌سلاح، کار بی‌پاداش، فداکاری در سکوت، دین بی‌دنیا، مذهب بی‌عوام، عظمت بی‌نام، خدمت بی‌نان، ایمان بی‌ریا، خوبی بی‌نمود، گستاخی بی‌خامی، قناعت بی‌غرور، عشق بی‌هوس، تنهایی در انبوه جمعیت، و دوست داشتن بی‌آنکه دوست بداند، روزی کن.

اگر تنهاترین تنهاشوم، باز خدا هست

او جانشین همه نداشتن‌هاست...

---

[1] مناجاتی از دکتر علی شریعتی.

سپاس گزاری...

سپاس خداوندگار حکیم را که با لطف بی‌کران خود، آدمی را زیور عقل آراست.

در آغاز وظیفه خود می‌دانم از زحمات بی‌دریغ استاد راهنمای خود، جناب آقای دکتر حمید شیخ زاده نجار، صمیمانه تشکر و قدردانی کنم که قطعاً بدون راهنمایی‌های ارزنده ایشان، این مجموعه به انجام نمی‌رسید. همچنین از دوستان و همکاران گرامی در دانشکده برق، بخصوص دانشجویان فعال در آزمایشگاه پردازش رسانه های دیجیتال (Multimedia Signal Processing Lab) از جمله سرکار خانم نجمه بطحایی، به خاطر همکاری دوستانه و پشتیبانی و راهنمایی های سازندشان در آماده سازی این رساله، تشکر می کنم. از جناب آقای دکتر حسن آقایی نیا که زحمت مطالعه و داوری این رساله را تقبل فرمودند، کمال امتنان را دارم. در پیاده سازی برخی از قسمت برخی از اساتید و دانشجویان به بنده کمک های بسیاری کرده اند که لازم است از آنها در اینجا تشکر کنم:

۱. آقای دکتر Michael Osborne از دانشکده ی مهندسی Oxford به خاطر در اختیارگذاشتن داده های هواشناسی در مقاله ی [۶۰].

۲. آقای دکتر Zoubin Ghahramani از دانشکده ی مهندسی Oxford به خاطر راهنمایی ها در مورد پیاده سازی GP.

۳. آقای دکتر Mauricio Alvarez از دانشکده ی علوم کامپیوتر دانشگاه Manchester به خاطر در اختیار گذاشتن کدشان در مقاله ی [۱] و همچنین کمکشان در پیاده سازی داده ها.

۴. آقای دکتر Michalis Titsias محقق فوق دکتری از مرکز تحقیقات ژنتیک انسانی دانشگاه Oxford به خاطر در اختیار قراردادن کد مقاله ی [۸۰].

همچنین لازم می‌دانم از پدید آورندگان بسته زی‌پرشین، به خصوص جناب آقای وفا خلیقی، که این پایان‌نامه با استفاده از این بسته، آماده شده است، کمال قدردانی را داشته باشم.

در پایان، بوسه می‌زنم بر دستان خداوندگاران مهر و مهربانی، پدر و مادر عزیزم و بعد از خدا، ستایش می‌کنم وجود مقدس‌شان را و تشکر می‌کنم از برادر عزیزم به پاس عاطفه سرشار و گرمای امیدبخش وجودشان، که در این سردترین روزگاران، بهترین پشتیبان من بودند.

دانیال خشابی
۱۳۹۱

فهرست مطالب

لیست تصاویر

فصل ۱

مقدمه: تعریف و اهمیت مساله

۱.۱ مقدمه

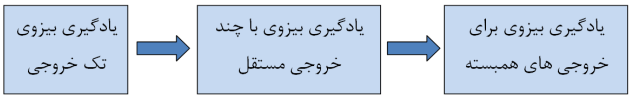
در این فصل انواع مفاهیم و مسائل کلی که در طی این رساله مورد بحث قرار می دهیم را به صورت مقدماتی معرفی خواهیم کرد. شکل ۱.۱ مراحل اصلی مدل سازی مورد نظر این رساله را نشان می دهد. در واقع لازم است ابتدا مهمترین الگوریتم های بیزوی یادگیر را بررسی کنیم. تعمیم الگوریتم های تک خروجی، به الگوریتم های چند خروجی با خروجی های مستقل، معمولا سرراست است. چرا که از کنار هم قرار دادن الگوریتم های تک خروجی بدست می آید. اصلی ترین قسمت مربوط به مدل سازی الگوریتم های چندخروجی با در نظرگرفتن همبستگی آنهاست.

نکته ی دیگری که در ایجا لازم است ذکر کنیم این است که منظور از "الگوریتم های یادگیری" در طول این رساله، الگوریتم های رگرسیون [۱] است. اگرچه ایده های معرفی شده را می توان به روش های مشابه به مسائلی مانندکلاس بندی [۲]، خوشه بندی [۳]، تحلیل عاملی [۴]، یادگیری تقویتی [۵] و غیره تعمیم داد. نکته بعدی این است که اگرچه هدف این رساله مدلسازی الگوریتم های یادگیری ماشین بیزوی بوده، ولی در این رساله تنها الگوریتم بیزوی با ساختارهای غیرگسترده را بررسی می کنیم و وارد مبحث مدل های گرافیکی [۶] به شیوه ای که در [۳۲، ۳۶] مورد بررسی قرار گرفته ایم نمی شویم.

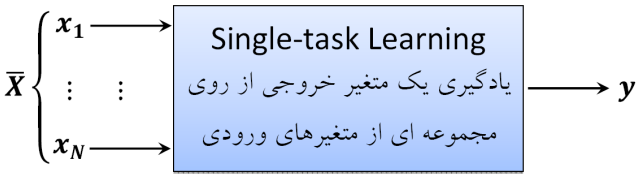
در ادامه ی این فصل، ابتدا از معرفی الگوریتم های یادگیری تک خروجی شروع کرده و مثال هایی از کاربردهای آنها را ارائه خواهیم کرد. در ادامه ی فصل، قبل تعمیم الگوریتم ها به چند بعد، ساختار یادگیری بیزوی [۷] را معرفی و بررسی می کنیم. بسیاری از تعریفات اصلی و روابط بین توزیع های آماری که در تمام طول رساله استفاده خواهیم کرد، دراین فصل معرفی و بررسی می شود. در ادامه وارد بحث یادگیری چند خروجی و کاربردهای آنها خواهیم شد. منظور از "یادگیری" در این رساله "یادگیری ماشین با ناظر" [۸] است. یعنی می خواهیم مدلی طراحی کنیم با گرفتن مجموعه ای از الگوهای (تقریبا) درست، به مدلی نسبتا دقیق از رفتار سیستم برسد. عموما یادگیری با ناظر را به دو قسمت می توان

---

[۱] Regression
[۲] Classification
[۳] Clustering
[۴] Factor Analysis
[۵] Reinforcement Learning
[۶] Graphical models
[۷] Bayesian
[۸] Supervised learning



شکل ۱.۱: مراحل مدل سازی الگوریتم های چند خروجی با همبستگی با ابعادی



شکل ۲.۱: یادگیری تک–خروجی

تقسیم می شود:

۱. رگرسیون؛ که در آن هدف این است که با استفاده از مجموعه ی داده های آموزشی $\mathcal{D} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N}$ الگوی ورودی/خروجی سیستم یادگرفته شود. الگوریتم مورد نظر باید بتواند به ازای داده ی جدید $\mathbf{x}^*$، خروجی متناظر با آن $\mathbf{y}^*$ را با تقریب مناسبی به دست دهد.

۲. کلاس بندی؛ که در آن هدف این است که داده ی ورودی $\mathbf{x}$ را در یکی از از کلاس های $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$ قرار دهیم. در ساده ترین حالت باید داده ورودی را در یکی از دو کلاس $\mathcal{C}_1$ یا $\mathcal{C}_2$ (کلاس بندی دو حالته [۹]) قرار دهیم. به ازای داده ی جدید $\mathbf{x}^*$، باید منطقی ترین کلاس $\mathcal{C}^* \in \mathcal{C}$ را به عنوان خروجی الگوریتم ارائه دهد.

۲.۱ یادگیری با یک خروجی

در آغاز راه بسیاری از الگوریتم ها، برای آنکه از پیچیدگی بی مورد جلوگیری شود، مساله را در حالت ساده تک خروجی در نظر می گیرند. ساختار چنین مساله ای به صورتی است که در شکل ۲.۱ نشان داده شده است. در اینجا تاکید می کنیم که چند خروجی بودن با چندبعدی بودن خروجی ها هرکدام از خروجی ها متفاوت است(تنها در رگرسیون، نه کلاس بندی). در واقع اگرچه در ساختار نشان داده شده در شکل ۲.۱ تنها یک خروجی دیده می شود، خروجی مورد نظر می تواند $\mathbf{y} \in \mathcal{R}^{n \geq 1}$ باشد. معمولا بُعد هرکدام از متغیرهای خروجی مشکلی ایجاد نمی کند. لذا جهت سادگی مضاعف، معمولا خروجی مورد نظر نیز اسکالر در نظر گرفته می شود. در مورد مساله ی رگرسیون، هر مساله ای را که شامل تخمین مقادیر تک متغیره باشد را می توان بوسیله ی چنین الگوریتم هایی یادگرفت. (مثال هایی از یادگیری برای یک بعد خروجی) در مورد الگوریتم های کلاس بندی، هر مساله ی کلاس بندی که شامل دو کلاس باشد را می توان با یادگیری تک خروجی انجام داد. در ادامه مفاهیم اساسی در یادگیری بیزوی را بررسی می کنیم.

[۹] Binary Classification

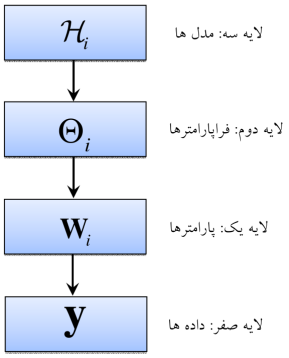
۳.۱ یادگیری بیزوی

۱.۳.۱ چرا یادگیری بیزوی؟

یادگیری بیزوی این امکان را فراهم می آورد اطلاعات اولیه [۱۰] طراح را در مورد داده ها را مدل سازی کنیم و به صورت یکپارچه با اطلاعات حاصل از داده های نمونه گیری شده و ادغام کنیم. همچنین چنین مدل سازی، یکی از اصلی ترین مشکلات در مدل سازی داده ها(برای رگرسیون [۱۱] یا کلاس بندی [۱۲])، یعنی بیش برازش [۱۳] را تا حد زیادی حل می کند.

۲.۳.۱ مدل سازی پارامتری یک ساختار بیزوی

معمولا در لایه صفر مدل، پارامترهایی قرار دارند که مستقیما با خروجی سیستم در ارتباطند. در لایه دوم $\mathbf{\Theta}_i$ یا فراپارامترهای [۱۴] سیستم طبق مدل $\mathcal{H}_i$ قرار دارند که توزیع های روی $\mathbf{W}_i$ پارامترهای مساله(لایه اول) هستند. بالاترین لایه، لایه ی مدل ها یا $\mathcal{H}_i$ است که نوع مدل را مشخص می کند. کارهایی که در رابطه با یک مدل بیزوی مد نظر داریم



شکل ۳.۱: یادگیری ارتباط بین خروجی ها

عبارتند از:

۱. طراحی مدل $\mathcal{H}_i$ شامل توزیع های فراپارامترهای $\mathbf{\Theta}_i$ و پارامترهای $\mathbf{W}_i$ و نحوه ارتباط بین آنها برای ساختن خروجی.

۲. اموزش مدل [۱۵] شامل تخمین پارامترهای بهینه مدل برای مجموعه داده های آموزشی در اختیار $\mathcal{D}$.

۳. اجرای استنتاج روی مجموعه داده های جدید [۱۶] و بررسی عملکرد مدل آموزش دیده و قدرت تعمیم آن برای داده های جدید.

۴. اعمال تقریب های مختلف برای تسریع محاسبات در ضمن نگه داشتن دقت مدل.

[۱۰] Prior information
[۱۱] Regression
[۱۲] Classification
[۱۳] Overfitting
[۱۴] Hyperparameter
[۱۵] Training
[۱۶] Unseen data

استفاده از قانون Bayes ابزاری را فراهم می سازد تا افراد بتوانند مدل های قوی احتمالی برای ۴ عمل فوق را انجام دهند. در ادامه برخی از پایه ای ترین مفاهیم مربوط به مدل سازی بیزوی را بیان می کنیم. بر طبق قانون Bayes می توان رابطه ی زیر را نوشت :

$$p(\mathbf{W}_i|\mathbf{y},\mathbf{X},\mathbf{\Theta}_i,\mathcal{H}_i) = \frac{p(\mathbf{y}|\mathbf{W}_i,\mathbf{X},\mathcal{H}_i)p(\mathbf{W}_i|\mathbf{\Theta}_i,\mathcal{H}_i)}{p(\mathbf{y}|\mathbf{X},\mathbf{\Theta}_i,\mathcal{H}_i)} \quad (۱.۱)$$

مقدار $p(\mathbf{y}|\mathbf{W}_i,\mathbf{X},\mathcal{H}_i)$ مقدار درست نمایی [۱۷] نامیده می شود. مقدار $p(\mathbf{W}_i|\mathbf{\Theta}_i,\mathcal{H}_i)$ توزیع پیشین [۱۸] است که در بردارنده ی اطلاعات/حدس اولیه ما درباره نحوه پراکندگی پارامترهای بهینه است. توزیع پسین [۱۹] $p(\mathbf{W}_i|\mathbf{y},\mathbf{X},\mathbf{\Theta}_i,\mathcal{H}_i)$، توزیعی است روی پارامترهای مساله ($\mathbf{W}_i$) که اطلاعات توزیع پیشین و درست نمایی را ادغام می کند. مخرج رابطه ی Bayes فوق، ضریب نرمالیزاسیون [۲۰]، درست نمایی حاشیه ای [۲۱]، یا گواه [۲۲]، نامیده می شود [۶۶] که می توان آن را به فرم زیر نیز نوشت:

$$p(\mathbf{y}|\mathbf{X},\mathbf{\Theta}_i,\mathcal{H}_i) = \int p(\mathbf{y}|\mathbf{W}_i,\mathbf{X},\mathcal{H}_i)p(\mathbf{W}_i|\mathbf{\Theta}_i,\mathcal{H}_i)d\mathbf{W}_i \quad (۲.۱)$$

به صورتی مشابه می توان فراپارامترها را به عنوان متغیر هدف در نظر گرفت و توزیع پسین را برای آنها نوشت:

$$p(\mathbf{\Theta}_i|\mathbf{y},\mathbf{X},\mathcal{H}_i) = \frac{p(\mathbf{y}|\mathbf{\Theta}_i,\mathbf{X},\mathcal{H}_i)p(\mathbf{\Theta}_i|\mathcal{H}_i)}{p(\mathbf{y}|\mathbf{X},\mathcal{H}_i)} \quad (۳.۱)$$

عبارت $p(\mathbf{\Theta}_i|\mathcal{H}_i)$، توزیع پیشین روی فراپارامترها [۲۳] نامیده می شود. مقدار ثابت نرمالیزه در مخرج را می توان به صورت زیر نیز نوشت:

$$p(\mathbf{y}|\mathbf{X},\mathcal{H}_i) = \int p(\mathbf{y}|\mathbf{\Theta}_i,\mathbf{X},\mathcal{H}_i)p(\mathbf{\Theta}_i|\mathcal{H}_i)d\mathbf{\Theta}_i \quad (۴.۱)$$

در بالاترین لایه می توان قانون Bayes را برای مدل ها نوشت:

$$p(\mathcal{H}_i|\mathbf{y},\mathbf{X}) = \frac{p(\mathbf{y}|\mathbf{X},\mathcal{H}_i)p(\mathcal{H}_i)}{p(\mathbf{y}|\mathbf{X})} \quad (۵.۱)$$

مشابه آنچه در قسمت های فوق انجام دادیم می توان در مورد ثابت نرمالیزه نوشت:

$$p(\mathbf{y}|\mathbf{X}) = \Sigma_i p(\mathbf{y}|\mathbf{X},\mathcal{H}_i)p(\mathcal{H}_i).$$

۳.۳.۱ مدل های Parametric و Nonparametric

مدل های طراحی شده را بر اساس سایز پارامترهای آنها (درجه آزادی آنها) به دو قسمت تقسیم می کنند. در مدل های nonparametric با افزایش تعداد داده های آموزشی؛ میزان پارامترهای مدل نیز متناسب با آن زیاد می شود. به عنوان مثال GP (که در فصل ۲ معرفی خواهد شد.) یک مدل nonparametric است. اگر فرض کنیم متغیر خروجی بدون نویز مدل $\mathbf{f}$ باشد، با افزایش داده های آموزشی، ابعاد $\mathbf{f}$ نیز افزایش می یابد. به عنوان مثال دیگر، در مدل های مبتنی بر توابع پایه، می دانیم وزن توابع پایه، پارامترهای مدل هستند. معمولا (مثلا در RVM و SVM) با افزایش داده های آموزشی، تعداد توابع پایه(و ضرایب آنها) نیز افزایش می یابند. در حالیکه که در ساختارهای متداول شبکه های عصبی(مثلا MLP[۲۴]) ساختار شبکه تماماً توسط طراح می شود و ارتباطی به تعداد داده های آموزشی ندارد.

[۱۷] Likelihood
[۱۸] Prior distribution
[۱۹] Postoterior distribution
[۲۰] Normalization constant
[۲۱] Marginal likelihood
[۲۲] Evidence
[۲۳] Hyper-prior or prior on hyperparameters
[۲۴] Multilayer Perceptron

۴.۳.۱ انتخاب پارامترهای بهینه در یک مدل بیزوی

اجرای بهینه سازی و انتخاب پارامترها، معمولا در مدل های بیزوی نیازمند محاسبه ی انتگرال های پیچیده است. در این میان، ساختار که در آن همه چیز را گوسی در نظر بگیریم (که در فصل بعد تحت عنوان GP معرفی خواهد شد) یک استثناست که می توان در آن انتگرال توابع گوسی را به راحتی حساب کرد. همانطور که گفته شد مدل سازی بر اساس مجموعه ای از پارامترها مانند $\mathbf{W}_i = \{\mathbf{W}_i^1, \ldots, \mathbf{W}_i^n\}$ انجام می گیرد. معمولا در مدل سازی بیزوی علاقمندیم توزیع مربوط به پارامترها، $p(\mathbf{W}|\mathbf{\Theta}, \mathbf{y}, \mathcal{H}_i)$ را داشته باشیم تا بتوانیم برای خروجی الگوریتم، حاشیه های اطمینان بدست آوریم. در صورتی که بخواهیم تنها به مقدار محتمل ترین $\mathbf{W}$ قناعت کنیم، یکی از راه های پیدا کردن بهینه، استفاده از تابع درست نمایی است:

$$\mathbf{W}_i^* = \arg\max_{\mathbf{W}_i} p(\mathbf{y}|\mathbf{W}_i).$$

چنین آموزشی، روش بیشینه درست نمایی [۲۵] نام دارد. در واقع پارامترهای $\mathbf{W}_i$ را به گونه ای انتخاب کرد که میزان درست نمایی داده های آموزشی (احتمال داده های آموزشی، با در دست داشتن مشخصات مدل) را حداکثر کنند.
اگر بخواهیم توزیع روی $\mathbf{W}$ و مقدار محتمل $\mathbf{\Theta}$ را بدست آوریم نیز می توانیم از همان توزیع حاشیه ای روابط ۲.۱ و ۴.۱ استفاده کرده و ابتدا پارامتر بهینه $\mathbf{\Theta}^*$ و $\mathcal{H}^*$ را بدست می آوریم:

$$\mathcal{H}^* = \arg\max_{\mathcal{H}_i} p(\mathbf{y}|\mathbf{X}, \mathcal{H}_i)$$

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}_i} p(\mathbf{y}|\mathbf{X}, \mathbf{\Theta}_i, \mathcal{H}^*)$$

و سپس با استفاده از رابطه ی ۱.۱ توزیع پارامترهای مدل را بدست آوریم:

$$p(\mathbf{W}_i|\mathbf{y}, \mathbf{X}, \mathbf{\Theta}^*, \mathcal{H}^*)$$

اولین ایده ای که برای انجام حداکثر سازی درست نمایی نسبت به متغیرها به نظر می رسد، مشتق گیری از آن نسبت به متغیرهای مورد نظر است:

$$\nabla_{\mathcal{H}} p(\mathbf{y}|\mathbf{X}, \mathcal{H}_i) = 0, \qquad \nabla_{\mathbf{\Theta}_i} p(\mathbf{y}|\mathbf{X}, \mathbf{\Theta}_i, \mathcal{H}^*) = 0.$$

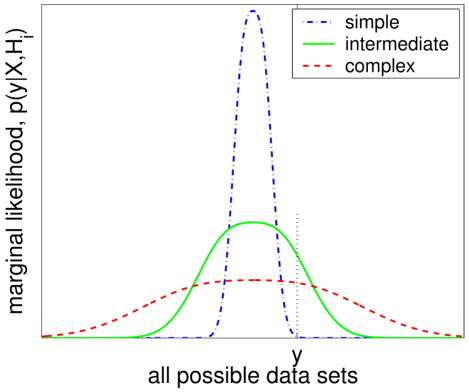
برای یادگیری در یک مدل بیزوی می توان روند های منطقی دیگری را نیز در نظر گرفت. از جمله در [۲۵] مدل های جریمه شده [۲۶] برای درست نمایی در یادگیری منطقی تر بیزوی بررسی شده است.

۵.۳.۱ بهینه سازی تقریبی یک ساختار بیزوی

استفاده از روش های بیزوی نیازمند محاسبه انتگرال هایی است که غالبا محاسبه آن دشوار و گاهی غیرممکن است. به همین دلیل بخش مهمی از توجه محققین این رشته معطوف به روش هایی برای تقریب محاسبات بیزوی بوده است. ضمیمه ی دوم از این رساله اختصاص داده است به معرفی و بررسی روش های آماری مهم برای تقریب ۲.۱ در محاسبات بیزوی که در ادامه فصل ها از آنها استفاده خواهیم کرد. مهمترین این روش ها عبارتند از :

۱. MCMC (Markov Chain Monte Carlo)

[۲۵] Maximum likelihood
[۲۶] Penalized



شکل ۴.۱: یادگیری ارتباط بین خروجی ها (تصویر از [۶۶])

۲. EP (Expectation Propagation)

۳. VB (Variational Bayes)

۴. Laplace Approximation

۶.۳.۱ بحثی بیشتر روی مدل سازی بیزوی

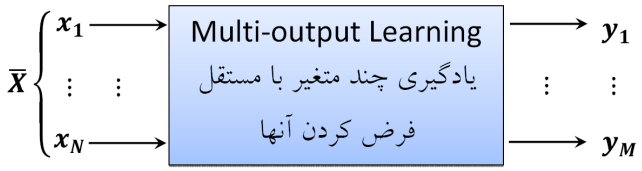
معمولا توزیع پیشین روی مدل ها $\mathcal{H}_i$ یکنواخت است. این به معنی آن است که هیچ مدلی را نسبت به مدل دیگر برتری ندهیم. با این فرض در عمل می توان به جای حساب کردن انتگرال دشوار رابطه ۴.۱ و استفاده از توزیع پسین در رابطه ۳.۱، به عنوان یک تقریب می توان درست نمایی حاشیه ای در رابطه ۲.۱ را نسبت به فراپارامترهای $\boldsymbol{\Theta}_i$ حداکثر کرد. چنین تقریبی Type II maximum likelihood (ML-II) نام دارد [۶۶]. در مرحله حداکثرسازی رابطه ۲.۱ باید دقت کرد در صورتی که مدل دارای فراپارامترهای بسیاری باشد(نسبت به داده های آموزشی مدل)، حداکثرسازی ۲.۱ منجرو به بیش برازش [۲۷] و از بین رفتن قدرت تعمیم [۲۸] مدل خواهد شد.

استفاده از درست نمایی حاشیه ای روی فراپارامترها در ساختار بیزوی این امکان را فراهم می سازد تا به صورت طبیعی تعادلی بین پیچیدگی مدل و قابلیت تعمیم روی داده ها ایجاد کنیم. حضور فراپارامترها با توزیع های مناسب، می توانند تُنُک بودن [۲۹] را به سیستم تحمیل کرده و قابلیت تعمیم آن را افزایش دهد. شکل ۴.۱ سه حالت کلی برای درست نمایی حاشیه ای ۲.۱ را نشان می دهد. همانطور که مشخص است توزیع مورد نظر متغیری از مجموعه داده های ممکن است.

۱. در صورتی که مدل مربوط به مجموعه بسیار محدودی از داده ها باشد، توزیع مورد نظر بسیار باریک و کشیده خواهد بود(نمودار آبی).

۲. در صورتی که مدل مربوط به طیف گسترده ای از مجموعه داده ها باشد، نمودار مورد نظر بسیار پهن خواهدبود(نمودار قرمز).

[۲۷] Overfitting
[۲۸] Generalization power
[۲۹] Sparsity



شکل ۵.۱: یادگیری با چند متغیر خروجی

هیچ کدام از حالت های فوق (نمودار آبی و قرمز) مطلوب نیستند؛ در حالت اول، طراحی بسیار خاص داده های آموزشی است؛ به عبارتی در این آموزش بیش برازش رخ داده است. در حالت دوم، مدل به درستی طراحی نشده است؛ چون مدل بین طیف وسیعی از داده ها تفاوت نمی گذارد. چنین حالاتی مطلوب نیستند؛ چون در اصل مدل برای داده ها معنی خود را از دست می دهد. حالت مطلوب، حالتی میانه (نمودار سبز) است که در آن به مدل سازی مناسب، طیف معقولی از مجموعه داده های ممکن دارای احتمالی مثبت هستند. به مفهوم ایجاد تعادل بین برازش خوب روی داده های آموزشی و قابلیت تعمیم الگوریتم، در اصطلاح محققین این رشته، تیغه ی Occam یا Occam's razor گویند [۶۴] [۳۰]. با توجه کرد که مصالحه بین کیفیت برازش و پیچیدگی مدل به طور طبیعی در رابطه ۲.۱ وجود دارد. در واقع حتی اگر توزیع پیشین مقدار ثابتی داشته باشد، باز هم رابطه ۲.۱ آن مدلی را که قابلیت توصیف بهتری برای مجموعه داده ها ارائه اکند را انتخاب می کند. لذا قابلیت توصیف خوب و پیچیدگی کم را نباید با مساله ی انتخاب توزیع های مناسب روی پارامترها اشتباه گرفت [۶۴].

۴.۱ لزوم الگوریتم های چند خروجی

معمولا در مسائل واقعی، پارامترهایی هایی که به عنوان پارامترهای هدف با آنها سر و کار داریم بیشتر از یک مورد هستند. به عنوان مساله ی کلاس بندی بیش از دو کلاس را در نظر می گیریم. ساختار الگوریتم باید مشابه آنچه در شکل ۵.۱ نشان داده شده است باشد. در واقع اگر به ازای یک داده ی ورودی $\bar{\mathbf{X}}$ اگر

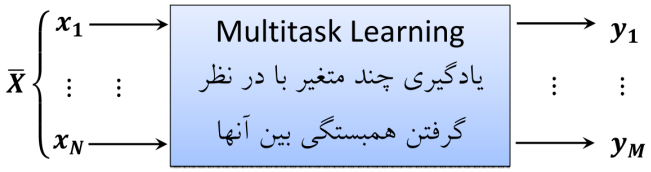
$$\exists 0 \leq i \leq M, s.t. \forall j \in \{1, \ldots, M\}, j \neq i \Rightarrow y_i > y_j$$

داده ی ورودی مورد نظر در کلاس i–ام قرار دارد. مثلا در مساله ی کلاس بندی احساسات [۶۸]، اگر بتوانیم مجموعه از داده های مربوط به ویژگی های [۳۱] صورت را استخراج کنیم، می توانیم با استفاده از مجموعه از داده های آموشی، تشخیص احساسات را یادبگیریم. در صورتی در خروجی M متغیر داشته باشیم، می توان خروجی را حداکثر به 2^M کلاس مجزا تقسیم کرد. معمولا در پیاده سازی های عملی، این تقسیم بندی ها، شامل M کلاس است(داده متعلق به کلاسی است که خروجی متناظر با آن مقدار بیشتری داشته باشد.) نمونه های بی شماری برای کلاس بندی بیش از دو کلاس می توان آورد؛ در مساله ی کلاس بندی تصاویر گل ها [۵۸] در ابتدا تعداد محدودی داده های آموزشی در مورد تعلق هر تصویر به گروهی خاص در اختیار داریم و می خواهیم تعداد بی شماری تصویر دیگر با اندازه ها ویژگی های مختلف را در تعداد مشخصی گروه جای دهیم. مساله ی جالب دیگر می تواند یادگیری دینامیک معکوس یک روبات [۳۲] باشد. چنین مساله ای در [۸۳، ۸۲، ۸۴] بررسی شده است. در مثال مطرح شده در مقالات مذکور لازم است ۲۱

[۳۰] این اصطلاح بعد از اثر Williams نوشته Occam(1285-1349) که در آن سادگی به اندازه در توصیفات و جلوگیری از پیچیدگی بیش از اندازه را نشویق می کرد، عامیانه شد.

[۳۱] Feature

[۳۲] Robot inverse dynamics



شکل ۶.۱: یادگیری ارتباط بین خروجی ها

متغیر ورودی (۷ متغیر مکان، ۷ متغیر سرعت و ۷ متغیر شتاب) را به ۷ گشتاور ناشی از موتور بنگاریم. با داشتن مدلی از دینامیک معکوس یک روبات می توان آن را به صورت سیستم معکوس کنترل کرد.

۵.۱ الگوریتم های چند خروجی با همبستگی بین ابعادی

مساله ی یادگیری همزمان چندین عمل [۳۳] روز به روز توجه بیشتری را در مسائل یادگیری ماشین [۳۴] به خود جلب می کند. در اینگونه مسائل معمولا هدف تخمین با استفاده مجموعه ای از داده های ورودی–خروجی، بدون اطلاعات پیشین یا اطلاعات تقریبی در مورد نحوه ی همبستگی متغیرهای خروجی است.

تا اینجا مدل هایی که معرفی کردیم مخصوص مدل سازی و تحلیل در یک دسته بوده است. فرض کنیم چند دسته از داده ها را بخواهیم مدل سازی کنیم، بطوریکه در این مدل سازی، برخی از پارامترهای دسته داده های جدا از هم، مشترک باشند. چنین دیدگاهی Multitask Learning یا یادگیری داده های چند متغیر با در نظرگرفتن هم بستگی بین آنها نام دارد [۱۲]. در شکل ۶.۱ بلوک چنین الگوریتمی نمایش داده شده است.

مثال دیگر انتخاب بهترین مکان حفاری، برای استخراج منابع زیرزمینی است. واضح است که نمی توان تمام نقاط زمین را تا عمق بسیار سوراخ کرد، تا اینکه در نهایت تشخیص داد بهترین مکان برای حفاری کجاست؛ اما می توان از داده های موجود در اطراف مکان های حفاری شده استفاده کرد و مدلی آماری بر اساس داده های زمین شناسی ساخت و در مورد احتمال موفقیت آمیز بودن حفاری در محل های دیگر استدلال های آماری ارائه کرد. نمونه ای از چنین داده آزمایشی در [۱] بیان شده است. این آزمایش قبلا توسط مقالات بسیاری بررسی شده است. در این آزمایش داده های مربوط به تراکم برخی فلزات در منطقه ای وسعت ۱۴ کیلومتر مربع در سوییس گردآوری شده اند. در برخی مناطق نمونه گیری از برخی از فلزات گرانبها (مانند مس و کادمیوم) به میزان بسیار پایینی انجام شده است. اما داده های مربوط به فلزات کم بهاتر (مانند نیکل، روی و سرب) به صورت فراوان تری در دسترس هستند. هدف آن است که بتوان با استفاده از داده های مربوط به فلزات دیگر، استدلال بهتری در مورد موجودی فلزات گرانبها در مکان های مختلف انجام داد.

با توجه به اینکه در عمل، مدل های طراحی شده، برداشت های ساده شده ای از آنچه در حقیقت رخ می دهند هستند، همواره در مدل سازی سعی می شود درجه آزادی برای متغیرهای الگوریتم در نظر گرفته شود. معمولا چنین درجه آزادی توسط پارامترهایی به نام نویز مدل می شود.

مدل سازی همبستگی خروجی ها در شبکه های حسگر [۳۵] می تواند کاربرد بسیاری داشته باشد. به عنوان نمونه ای دیگر، می توان به آزمایش پیش بینی وضعیت آب و هوایی با داشتن حسگر هایی در شهرهای مجاور و داشتن داده هایی

[۳۳] Multitask learning
[۳۴] Machine learning
[۳۵] Sensor networks

از پارامترهای مختلف از آنها اشاره کرد. نمونه ای از این آزمایش در [۶۰] مطرح شده است. در این مقاله، داده های ۴ شهر مجاور در جنوب انگلستان مورد مطالعه قرار داده شده اند و داده های پارامترهای آنها مانند جهت و مقدار سرعت باد، دمای هوا، دمای دریا و میزان سطح دریا، برای مطالعه گردآوری شده اند [۳۶].

یکی دیگر از کاربردهای یادگیری با همبستگی بین ابعادی، نویز زدایی [۳۷] از تصاویر نویزی شده است. یکی از راه های متداول انتخاب بلوک های کوچک تصاویر، همراه با نواحی مشترک و استفاده از آنها به عنوان متغیرهای خروجی است [۸۰، ۹۰]. مشابه همین عمل را می توان برای مساله ی ترمیم تصاویری [۳۸] است که قسمت هایی از آنها حذف شده است [۸۰، ۹۰]. نمونه ی دیگر، اجرای فیلترینگ مساعدتی [۳۹] برای آنالیز نظرات و رتبه بندی فیلم هاست [۴۵، ۸۰].

[۳۶] داده های مورد نظر به صورت آنلاین در http://www.bramblemet.co.uk قابل دسترسی هستند.

[۳۷] Denoising

[۳۸] Image inpainting

[۳۹] Collaborative filtering

فصل ۲

مهم ترین الگوریتم های آموزش بیزوی

۱.۲ مقدمه

در سال های گذشته شاهد پیشرفت های بسیاری در زمینه ی توسعه ی الگوریتم های یادگیری ماشین [۱] بخصوص در زمینه ی الگوریتم های یادگیری بیزوی [۲] بوده ایم. یادگیری بیزوی ویژگی های اساسی دارد که آن را برای محققان جذاب می کند. از جمله اینکه ساختار استنتاجی که می توان به وسیله ی الگوریتم های بیزوی به وسیله ی ایده ی بیزی بودن به مسائل اضافه کرد بسیار انعطاف پذیر و مشابه روند طبیعی استدلال در انسان هاست. در واقع داشتن پیش فرض هایی در رابطه با یک حقیقت(توزیع پیشین [۳]) و استنتاج به وسیله ی تابعی که امکان پذیربودن رخدادی را به شرط داشتن (رخداد) پارامترهای پیش فرض (تابع یا توزیع درست نمایی [۴]) برای بدست آوردن امکانپذیر بودن رخداد مورد نظر (تابع [۵]) اساس کار استدلال و آموزش بیزوی بوده، که بسیار مشابه روند استدلالی در انسان است.

علیرغم سادگی مدل سازی بیزوی، حل مسائل آموزش بیزوی غالبا بسیار دشوار است. برای بسیاری از مسائل راه حل عددی نسبتا سریعی وجود ندارد و برای حل آنها از روش های عددی بسیار وقت گیر استفاده می شود. به همین دلیل در اغلب روش های فرض گوسی بودن توزیع ها، بخصوص روی توزیع های پیشین، می تواند بسیار ساده ساز باشد.

در این فصل مهم ترین الگوریتم های یادگیری بیزوی را معرفی خواهیم کرد. عمده ی این الگوریتم ها در ۱۰ سال گذشته معرفی شده اند و هم اکنون به علت انتشار مقالات بسیار در این زمینه، ساختار بسیار منسجمی و مرتبی از الگوریتم های معرفی شده، ارتباط بین آنها، و مزایا و معایب هرکدام بر دیگری وجود ندارد. سعی ما در اینجا این است که مهمترین الگوریتم های معرفی شده را اینجا به صورت خلاصه ارائه کنیم. در فصل های بعدی، این الگوریتم های را به چند بعد تعمیم خواهیم داد تا در نهایت بر اساس انها روش هایی برای ایجاد همبستگی بین ابعادی ایجاد کنیم.

---

[۱] Machine Learning
[۲] Bayesian Learning
[۳] Prior distribution
[۴] Likelihood function/distribution
[۵] Postorior

۲.۲ الگوریتم Gaussian Process(GP)

۱.۲.۲ مقدمه

مدل GP بر اساس فرض داشتن یک فرایند گوسی روی داده های ورودی است. در واقع در این مدل به جای اینکه یک فرایند روی زمان بگیریم، روی داده های ورودی در نظر می گیریم. به عبارتی دیگر فهم عملکرد یک GP دشوارتر از درک یک فرایند گوسی چندمتغیره نیست.

اولین بار مدل GP برای رگرسیون در [۸۸] معرفی شد. اگرچه قبل از آن در زمینه های دیگر همچون زمین شناسی آماری از مدلی مشابه آن استفاده می شده.

۲.۲.۲ مدل سازی الگوریتم برای رگرسیون

فرض کنیم که مجموعه ای از زوج مرتب ها از داده های ورودی (برداری) و خروجی (اسکالر یا یک بعدی) نویزی (غیر دقیق) به صورت $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ (معلوم) داشته باشیم و در نظر بگیریم که مجموعه $\{f_i(\mathbf{x}_i)\}_{i=1}^n$ مجموعه بدون نویز خروجی(مجهول) باشد متناظر با مجموعه ورودی–خروجی $\mathcal{D}$ داده باشد که به صورت مستقیم دیده نمی شود، اما در حقیقت وجود دارد. به چنین متغیرهایی که در عمل دیده نمی شوند(یعنی نمی توان از آنها نمونه برداری کرد) متغیر پنهان [۶] می گوییم. یعنی:

$$y_i = f(\mathbf{x}_i) + \epsilon_i \qquad \epsilon \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$$

یا

$$p(\mathbf{y}|\mathbf{f}) \sim \mathcal{N}\left(\mathbf{f}, \sigma^2\right) \tag{۱.۲}$$

تعریف: اگر داشته باشیم $\mathbf{X} = [\mathbf{x}_1 \ ... \ \mathbf{x}_n]^T$ و $\mathbf{f} = [\mathbf{f}_1 \ ... \ \mathbf{f}_n]^T$ در اینصورت یک GP " مجموعه از متغیرهای تصادفی تعریف شده روی مجموعه ورودی ها (محور x) است که دارای توزیع مشترک گوسی روی نقاط متناظرشان روی محور y هستند." [۶۲] یعنی:

$$p(\mathbf{f}|\mathbf{X}) \sim \mathcal{N}\left(0, \mathbf{K}_{\mathbf{f},\mathbf{f}}\right) \tag{۲.۲}$$

در تعریف فوق فرض کرده ایم که میانگین فرایند گوسی توصیف شده صفر است. این فرض با در نظر گرفتن اینکه تابع هدف، با میانگینی غیر مشخص است، قابل توجیه است. $\mathbf{K}$ ماتریس کواریانس بین وروی هاست که به صورت $K_{ij} = k\left(\mathbf{x}_i, \mathbf{x}_j\right)$ تعریف می شوند. نمونه ای از چنین توابعی می تواند به صورت یک تابع نمایی باشد:

$$k\left(\mathbf{x}_i, \mathbf{x}_j\right) = \sigma_f^2 \exp\left[-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2l^2}\right]$$

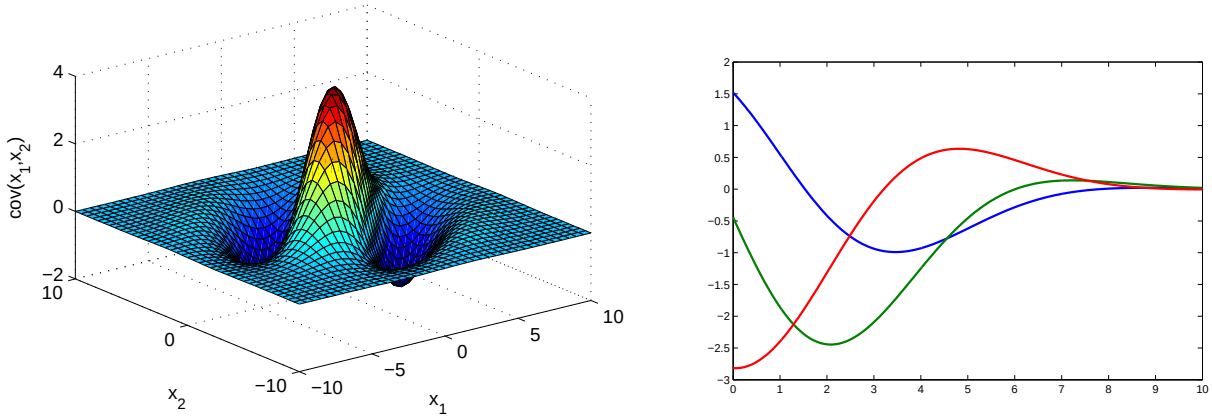
۳.۲.۲ نمونه گیری از توزیع پیشین GP بدون داده های آموزشی و بررسی اثر تابع کواریانس

تعدادی نمونه گیری از GP به ازای یک تابع کواریانس مشخص در شکل ۱.۲آ رسم شده است. نمایش داده شده است.

در شکل نمایش داده شده، نمونه گیری تصادفی از GP روی دو متغیر ورودی و تابع کواریانس SE (Squared Exponential) نمایش داده شده است.

۱. رفتار یکسان در هر دو بعد ($M = (0.3)^{-2} I$)

[۶] Latent variable



(ب) (آ)

شکل ۱.۲: در شکل (الف) نمونه گیری از GP به ازای تابع کواریانس $k(x_i, x_j) = \exp\left\{\frac{-(x_i^2 - x_j^2)}{20}\right\} * 4 * \cos(0.5 * (x_i - x_j))$. در شکل (ب) تابع کواریانس تعیین شده رسم شده است.

۲. در بعد x_1 تغییرات سریع تر و در بعد x_2 تغییرات کندتر است.

$$M_2 = \begin{bmatrix} 1.0278 & -1 \\ -1 & 1.0278 \end{bmatrix}$$

۳. راستای حداکثر تغییرات عمود بر $\hat{x}_1 + \hat{x}_2$ است.

$$M_3 = \begin{bmatrix} 2.278 & 2 \\ 2 & 2.278 \end{bmatrix}$$

همچنین در شکل ۳.۲ ازای پارامترهای مختلفی از تابع کواریانس زیر (معروف به Squared Exponential) نمونه گیری را انجام داده ایم. در هر مورد مشاهده می شود رفتار نمونه برداری، به طور مستقیم با شکل تابع کواریانس ارتباط دارد.

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) + \sigma_n^2 \delta_{i,j}, \qquad \Theta = \left\{l, \sigma_f^2, \sigma_n^2\right\}.$$

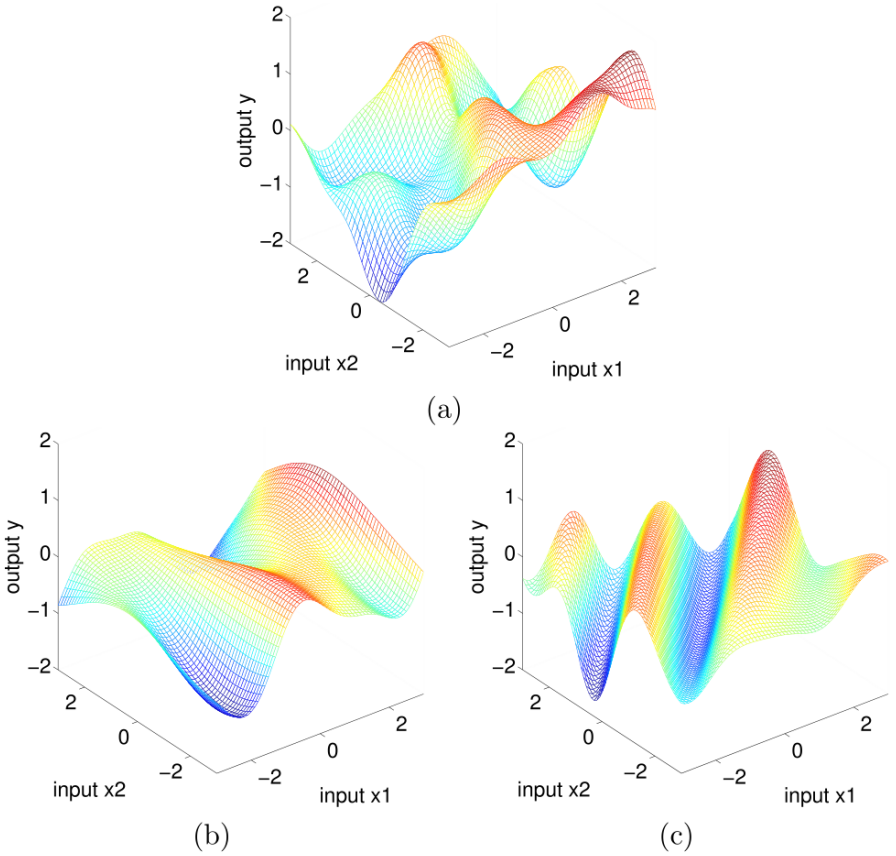
۴.۲.۲ بدست آوردن توزیع خروجی در حضور داده های آموزشی

در این قسمت هدف این است که با داشتن اطلاعات یعنی مجموعه ای از داده های ورودی-خروجی $\mathcal{D}$ و دریافت مجموعه ای داده های ورودی جدید، $\mathbf{X}^* = [\mathbf{x}_1^*, \ldots, \mathbf{x}_m^*]^T$ توزیع خروجی های متناظر یعنی $\mathbf{f}^* = [f_1^*, \ldots, f_m^*]^T$ را بدست آوریم؛ یعنی $p(\mathbf{f}^*|\mathbf{X}^*, \mathcal{D})$ به صورت مشابه با توجه به تعریف GP استدلال فوق می توان فرض کردن که یک فرآیند گوسی بین اجتماع داده های تست و آموزشی وجود داشته باشد؛ یعنی:

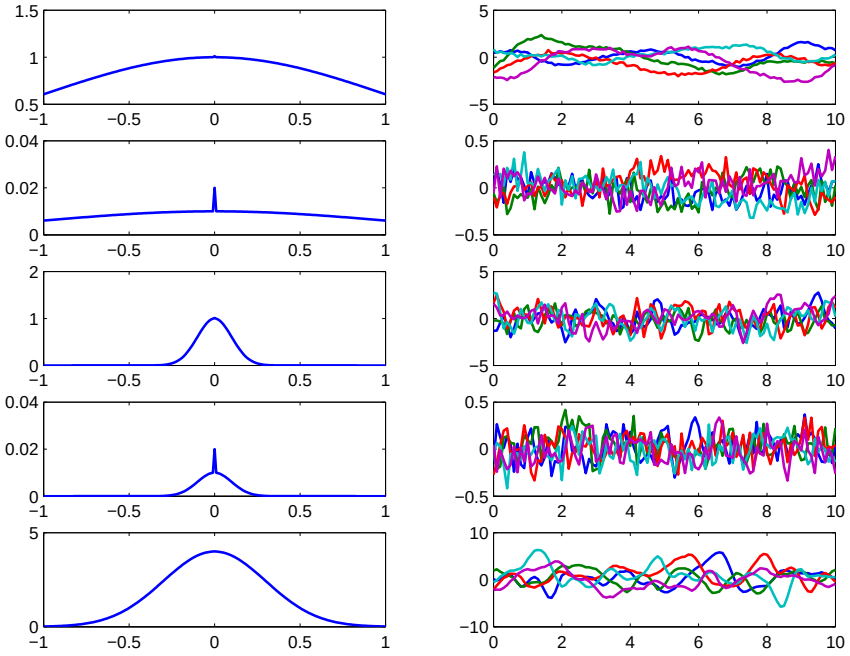
$$p(\mathbf{f}, \mathbf{f}^*|\mathbf{X}) \sim \mathcal{N}\left(0, \begin{bmatrix} \mathbf{K}_{\mathbf{f},\mathbf{f}} & \mathbf{K}_{*,\mathbf{f}} \\ \mathbf{K}_{\mathbf{f},*} & \mathbf{K}_{*,*} \end{bmatrix}\right) \tag{۳.۲}$$

در نمادگذاری فوق ماتریس کواریانس از $*$ به جای $\mathbf{f}^*$ استفاده شده است. با توجه به توضیحات داده شده، می توان عناصر ماتریس کواریانس را به صورت ساده تر زیر نوشت:

$$\mathbf{K}_{\mathbf{f},\mathbf{f}} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & \cdots & k(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$



شکل ۲.۲: نمونه گیری از توزیع های پیشین سه GP با توابع کواریانس متفاوت. در هرکدام از سه توزیع پیشین، تاثیر رفتار تابع کواریانس دیده می شود. (منبع تصویر [۶۶])



شکل ۳.۲: نمونه گیری از چند GP با توابع کواریانس مختلف. مشاهده می شود که شکل تابع کواریانس ارتباط مستقیم با رفتار نمونه ی حاصل دارد.

و به صورتی مشابه :

$$\mathbf{K}_{*,*} = \begin{bmatrix} k(\mathbf{x}_1^*, \mathbf{x}_1^*) & \cdots & k(\mathbf{x}_1^*, \mathbf{x}_m^*) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_m^*, \mathbf{x}_1^*) & \cdots & k(\mathbf{x}_m^*, \mathbf{x}_m^*) \end{bmatrix} \text{ و } \mathbf{K_{f,*}}^T = \mathbf{K_{*,f}} = \begin{bmatrix} k(\mathbf{x}_1^*, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1^*, \mathbf{x}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{x}_m^*, \mathbf{x}_1) & \cdots & k(\mathbf{x}_m^*, \mathbf{x}_n) \end{bmatrix} \text{ با}$$

استفاده از رابطه ۵.۶ در ضمیمه اول می توان نشان داد که با فرض برقرار بودن رابطه ۳.۲ توزیع خروجی داده های جدید(توزیع حاشیه ای) دارای یک توزیع گوسی به صورت زیر است:

$$p(\mathbf{f}^*|\mathbf{f}, \mathbf{X}) = \mathcal{N}\left(\mathbf{K_{*,f}}\mathbf{K_{f,f}^{-1}}\mathbf{y}, \mathbf{K_{*,*}} - \mathbf{K_{*,f}}\mathbf{K_{f,f}^{-1}}\mathbf{K_{f,*}}\right) \quad (۴.۲)$$

معمولا در عمل داده های آموزشی همراه با نویز هستند. لذا فرض می کنیم:

$$y = f(\mathbf{x}) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0, \sigma_n^2).$$

$$\Rightarrow \mathbf{cov}(y_p, y_q) = k(\mathbf{x}_p, \mathbf{x}_q) + \sigma_n^2 \delta_{pq}.$$

$$\Rightarrow \mathbf{K_{y,y}} = \mathbf{K_{f,f}} + \mathbf{\Sigma}, \qquad \mathbf{\Sigma} \stackrel{\triangle}{=} \sigma_n^2 \mathbf{I}.$$

اگر در رابطه ی ۴.۲ جایگزینی $\mathbf{K_{f,f}} \rightarrow \mathbf{K_{f,f}} + \mathbf{\Sigma}$ را انجام دهیم، توزیع جدید با نظر گرفتن نویز مدل بدست می آید. همچنین با فرض اثر نویز روی داده های تست بصورت $y^* = f(x^*) + \epsilon$، کواریانس توزیع حاصل را نیز با $\mathbf{\Sigma}$ جمع می کنیم. در این حالت می توان رابطه توزیع پیش بینی را به صورت زیر نوشت:

$$p(\mathbf{y}^*|\mathbf{y}, \mathbf{X}) = \mathcal{N}\left(\mathbf{K_{*,f}}(\mathbf{K_{f,f}} + \mathbf{\Sigma})^{-1}\mathbf{y}, \mathbf{K_{*,*}} - \mathbf{K_{*,f}}(\mathbf{K_{f,f}} + \mathbf{\Sigma})^{-1}\mathbf{K_{f,*}} + \mathbf{\Sigma}\right)$$

بدین ترتیب می توان مقدار خروجی را همراه با حاشیه های اطمینان [۷] با استفاده از توزیع فوق رسم کرد. منحنی که محتمل ترین رگسیون [۸] را بدهد، توسط پارامتر میانگین بدست آید و پارامتر کواریانس، حاشیه ی اطمینان را برای رگرسیون مورد نظر نشان می دهد. نمونه ای از این رگرسیون در شکل ۴.۲ نشان داده شده است.

می توان گفت کتابخانه ی GPML [۹] [۶۵] تقریبا هر الگوریتم مربوط به GP استاندارد را در خود دارد. همچنین این کتابخانه قابلیت های بسیاری برای بازتولید برنامه های جدید دارد که از آن می توان در ایجاد الگوریتم های مختلف بر اساس GP استفاده کرد.

نکته ی بسیار مهمی که در اینجا باید به آن اشاره کنیم این است که در بدست آوردن جواب رگرسیون توسط رابطه ۲.۴ نیاز به معکوس کردن ماتریس $\mathbf{K_{f,f}}$ است که دارای ابعاد $n \times n$ (n : ابعاد داده های آموزشی) است. در بهترین حالت، می توان عکس ماتریس را پیچیدگی محاسباتی $O(n^3)$ و پیچیدگی حافظه $O(n^2)$ انجام داد(معمولا با استفاده از Cholesky decomposition). ملاحظه می گردد اجرای چنین عملیات محاسباتی برای مسائل واقعی که در آنها با داده ها با ابعاد و نمونه ها با تعداد بسیار روبرو هستیم، بسیار دشوار خواهدبود. لذا لازم است به دنبال روش هایی باشیم تا با استفاده از آنها مرتبه محاسبات GP را کاهش دهیم یا به عبارتی دیگر آن را تُنُک [۱۰] کنیم.

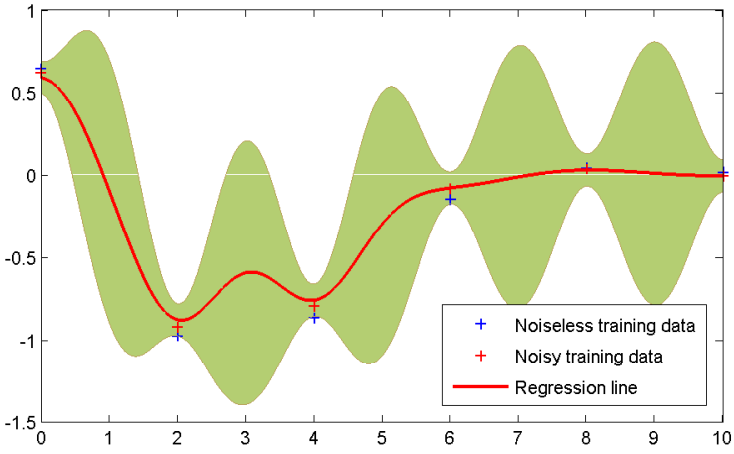
---

[۷] Confidence interval

[۸] در حقیقت به چنین عملی MAP(Maximum a Postoriori) گویند که در GP، به علت گوسی شدن توزیع پسین، مقدار حداکثر آن، به ازای مقدار میانگین بدست می آید.

[۹] قابل دسترسی در لینک مقابل: www.gaussianprocess.org/gpml/

[۱۰] Sparse



شکل ۴.۲: نمونه از رگرسیون بااستفاده از GP. در این شکل داده های اصلی(بدون نویز و پنهان) با + های آبی، داده های حاصل از نمونه گیری (همراه با نویز) با + های قرمز، خط رگرسیون با خط قرمز و حاشیه ی اطمینان به رنگ سبز مشخص شده است.

۵.۲.۲ بررسی اثر تابع کواریانس در آموزش مدل در حضور داده های آموشی

همانطور که می دانیم شکل تابع کواریانس است که تعیین می کند که داده ها، با چه فاصله هایی روی بعد ورودی دارای چه همبستگی باشند. این به نحوه ی پیاده سازی ARD[۱۱] بستگی دارد [۶۶، ۵۶]. معمولا بر اساس میزان ارتباط داده ی موجود با خروجی مطلوب کار می کند. به عنوان مثال تابع کواریانس SE یا Squared Exponential زیر را در نظر می گیریم:

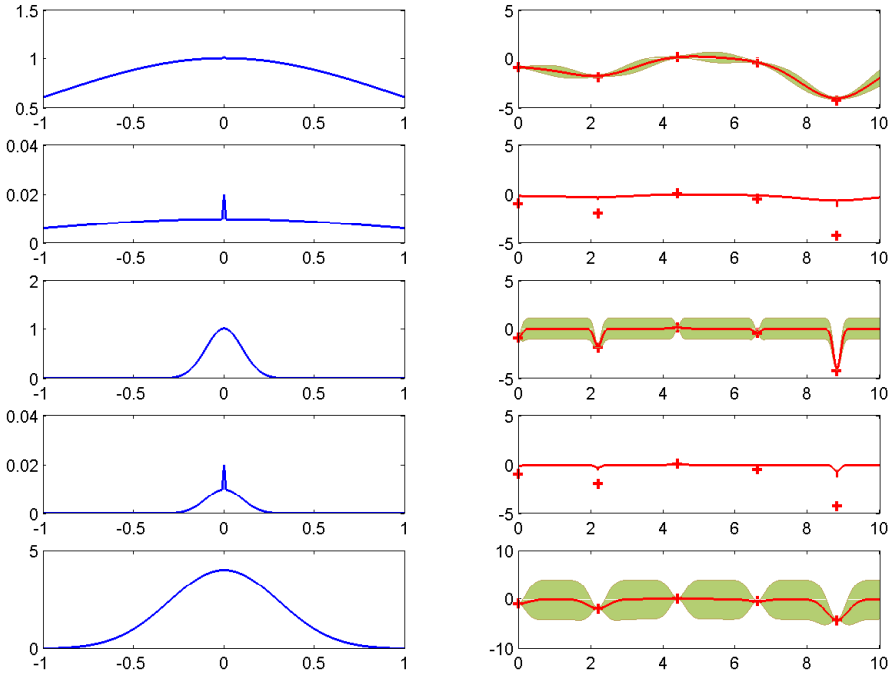
$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T M (\mathbf{x}_i - \mathbf{x}_j)\right\} + \sigma^2 \delta_{i,j}.$$

در تابع کواریانس فوق $\mathbf{\Theta} = \{\sigma_f, M, \sigma\}$ فراپارامترهای مساله هستد که رفتار آنها می تواند چگونگی تقریب را تحت تاثیر قرار دهد. مثال های ارائه شده در این قسمت نشان می دهند دنیایی از انتخاب ها برای ماتریس کواریانس وجود دارد که هر کدام می توانند ویژگی های تغییرات تابع را به شیوه ای عوض کنند. به عنوان نمونه فرض کنیم برای یک GP دلخواه، ماتریس کواریانس را به صورت زیر انتخاب کنیم:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2l^2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) + \sigma_n^2 \delta_{i,j}, \qquad \Theta = \left\{l, \sigma_f^2, \sigma_n^2\right\}.$$

در ادامه می خواهیم اثر هرکدام از فراپارامترها را بر تغییر رفتار توزیع پیش بینی بدست آمده در حضور داده های آموزشی ثابت را بدست می آوریم. فراپارامترها را در هر حالت به ترتیب به صورت های زیر در نظر می گیریم: $(l, \sigma_f, \sigma_n) = (1, 1, 0.1)$، $(l, \sigma_f, \sigma_n) = (0.1, 1, 0.1)$، $(l, \sigma_f, \sigma_n) = (1, 0.1, 0.1)$، $(l, \sigma_f, \sigma_n) = (0.1, 0.1, 0.1)$ و $(l, \sigma_f, \sigma_n) = (2, 0.3, 0.1)$. با در نظر گرفتن داده های آموزشی یکسان آموزش را با مدل های GP با مقادیر فراپارامتراهی مخلف انجام می دهیم. نتایج آزماش در شکل ۵.۲ نشان داده شده است. می توان شکل خروجی را در هر مورد متناسب تابع کواریانس آن توجیه کرد.

[۱۱] Automatic Relevance Determination



شکل ۵.۲: بررسی اثر تغییر تابع کواریانس در رگرسیون روی داده های آزمایشی

۶.۲.۲ کلاس بندی با استفاده از GP

می دانیم در کلاس بندی هدف این است که داده ی ورودی $\mathbf{x}$ را در یکی از از کلاس های $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$ قرار دهیم. در ساده ترین حالت باید داده ورودی را در یکز از دو کلاس $\mathcal{C}_1$ یا $\mathcal{C}_2$ (کلاس بندی دو حالته [۱۲]) قرار دهیم. در عمل کلاس بندی بسیار شبیه به رگرسیون است. اما تفاوت هایی بین آنها وجود دارد. در مدل GP، استفاده از توزیع پیشین گوسی و درست نمایی گوسی، منجرو به توزیع پسین گوسی شد. اگرچه چنین مدلی برای رگرسیون موجب سادگی شد، برای کلاس بندی چنین مدلی مناسب نیست. چرا که خروجی باید میزان عضویت در هر کلاس را مشخص کند. به همین دلیل مدل را باید عوض کرد که نتیجه ی آن پیچیده تر شدن محاسبات است. لذا از راه های تقربی برای حل آن استفاده می شود. یک راه برای استفاده از GP برای کلاس بندی این است که خروجی آن را به تابعی دهیم که مقادیر $(-\infty, \infty)$ را به $[0, 1]$ بنگارد. به عنوان مثال با استفاده از تابع logistic:

$$p(\mathcal{C}_1|\mathbf{y}) = \sigma(\mathbf{y}), \qquad \sigma(y) = \frac{1}{1+\exp(-y)}$$

یا با استفاده از تابع probit (یا توزیع تجمعی نرمال):

$$p(\mathcal{C}_1|y) = \Phi(y), \qquad \Phi(y) = \int_{-\infty}^{y} \mathcal{N}(x|0,1)dx.$$

در این حالت خاص خروجی شامل دو مرحله است:

$$p(f^*|\mathbf{X}, \mathbf{y}, X^*) = \int p(f^*|\mathbf{X}, \mathbf{f}, X^*)p(f|\mathbf{X}, \mathbf{y})d\mathbf{f}.$$

$$\pi^* = p(y^* = +1|\mathbf{X}, \mathbf{y}, X^*) = \int \sigma(f^*)p(f^*|\mathbf{X}, \mathbf{y}, X^*)df^* \qquad (۵.۲)$$

[۱۲] Binary Classification

محاسبه ی دقیق انتگرال ۵.۲ دشوار است. برای حل آن ۲ روش کلی تاکنون به دست آمده است. لازم به توضیح است که π^* یک تابع است، نه یک توزیع احتمالی.

روش اول که منجرو به به جواب های نسبتا دقیق می شود، استفاده از تکنیک های نمونه برداری MCMC (معرفی شده در بخش ۳.۲.۷ ضمیمه ی دوم رساله) است(از جمله در [۵۷]). روش های دیگر شامل تقریب تابع فوق است. ساده ترین و البته یکی از بدترین تقریب ها، تقریب Laplace است که در بخش ۶.۷ از ضمیمه دوم معرفی شده است و در [۲۱] نمونه ای از استفاده آن آورده شده است. در [۵۹] روش دیگری به اسم TAP approximation ارائه شده است. در [۸۷، ۲۲] مدل برای کلاس بندی چند کلاسه نیز شرح داده شده است.

۷.۲.۲ آموزشی پارامترهای GP

تا اینجا مشاهده کردیم چگونه می توان با یک GP با پارامترهای ثابت، عملیاتی مانند رگرسیون روی داده ها انجام داد. در حالیکه در عمل، بخصوص برای مسائل با ابعاد بالا، تنظیم دستی پارامترهای بهینه تقریبا غیر ممکن است. همانطور که معرفی شد ویژگی های رفتاری GP توسط تابع کواریانس آن مشخص می شود. لذا بهینه سازی پارامترهای مساله عبارتست از بهینه سازی روی پارامترهای تابع کوواریانس. به چنین مساله ای در اصطلاح محققان این رشته Model Selection Problem گفته می شود که در اصل مرحله ی آموزش [۱۳] ساختار GP است. در حالت کلی برای بدست آوردن ابزاری برای ارزیابی کیفیت مدل بدست آمده می توان از معیاری های مختلفی استفاده کرد؛ به عنوان نمونه

۱. احتمال مدل به شرط داشتن داده ها [۱۴].

۲. خطای تخمین/تعمیم: خطای تقریب در مشاهده داده های جدید.

۳. حد خطای تخمین/تعمیم.

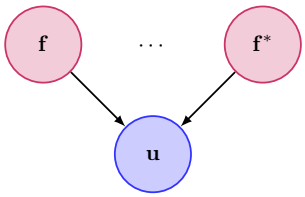
در حالت کلی تابع کواریانس $k(\mathbf{x}, \mathbf{x}')$ توسط مجموعه ای فراپارامترها [۱۵] مانند Θ تعریف می شود. این پارامترها به نحوی تعیین کننده ی رفتار GP در تقریب های مختلف هستند و با تغییر این فراپارامترها، در نتیجه تغییر رفتار تابع $k(\mathbf{x}, \mathbf{x}'; \Theta)$ رفتارهای مختلفی از GP های حاصل خواهیم داشت. معمولا در چارچوب یادگیری بیزوی، آموزش پارامترها از بیشینه کردن درست نمایی حاشیه ای [۱۶] به دست می آید:

$$\log p(\mathbf{y}|\mathbf{X}, \Theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_\mathbf{y}| - \frac{N}{2}\log 2\pi \qquad (۶.۲)$$

که در رابطه فوق $\mathbf{K_y} = \mathbf{K_{f,f}} + \sigma^2\mathbf{I}$ در رابطه درست نمایی حاشیه ای (رابطه ۶.۲) حضور سه عبارت را می توان به این صورت تفسیر کرد: عبارت اول مربوط است به میزان دقت برازش خروجی بر داده های آموزشی. عبارت دوم مقدار منفی است که از افزایش پیچیدگی بیش از اندازه سیستم جلوگیری می کند. عبارت سوم مربوط است به ضریب نرمالیزاسیون و تاثیری در بهینه سازی ندارد.

رابطه ۶.۲ معادله غیرمحدب [۱۷] روی فضای پارامترهای Θ است [۵۰] و لذا نمی توان به آسانی مقدار حداکثر مطلق آن

[۱۳] Training
[۱۴] Training data
[۱۵] Hyperparameter
[۱۶] Maximizing marginal likelihood
[۱۷] Non-convex



شکل ۶.۲: ارتباط بین متغیرهای پنهان و داده های خروجی(خروجی های مشاهده شده و خروجی های مطلوب)

را به آسانی به دست آورد. معمولا در عمل می توان مقدار حداکثر محلی خوبی را با استفاده از گرادیان گیری نسبت به پارامترها بدست آورد. اگر از رابطه ی ۶.۲ نسبت به پارامتر دلخواه θ_j مشتق جزئی بگیریم، داریم:

$$\frac{\partial}{\partial \theta_j}\log p(\mathbf{y}|\mathbf{X},\Theta) = -\frac{1}{2}\mathbf{y}^T\mathbf{K}_y^{-1}\frac{\partial \mathbf{K}_y}{\partial \theta_j}\mathbf{K}_y^{-1}\mathbf{y} - \frac{1}{2}\mathrm{tr}\left(\mathbf{K}^{-1}\frac{\partial \mathbf{K}}{\partial \theta_j}\right) \quad (۷.۲آ)$$

$$= \frac{1}{2}\mathrm{tr}\left(\left(\alpha\alpha^T - \mathbf{K}_y^{-1}\right)\frac{\partial \mathbf{K}_y}{\partial \theta_j}\right), \quad (۷.۲ب)$$

با داشتن مشتق درست نمایی حاشیه ای، می توان مدل GP را نسبت به فراپارامترهای آن بهینه کرد. باید توجه داشت بهینه سازی دوری با گرادیان، الزاما به پارامترهای بهینه ی اصلی منجرو نخواهد شد.

برای بدست آوردن پارامترهای بهینه در کلاس بندی بوسیله ی GP می توان تغییرات جزئی در GP استاندارد(مربوط به رگرسیون) ایجاد کرد و از آنها برای عملیات کلاس بندی مختلف استفاده کرد. جزئیات این تغییرات در [۶۶، ۲۲] آمده است.

۸.۲.۲ روش های مطرح شده برای ایجاد GP تُنُک

ایده ی اولیه این است که متغیرهایی پنهان به اسم $\mathbf{u}$ تعریف کنیم، بطوریکه $\mathbf{u}$ تنها راه ارتباط بین $\mathbf{f}$ و $\mathbf{f}^*$ است(مطابق شکل ۶.۲). در اینصورت چون دانستن $\mathbf{u}$، $\mathbf{f}^*$ اطلاعاتی به $p(\mathbf{f}|\mathbf{u})$ اضافه نمی شود، داریم:

$$p(\mathbf{f}|\mathbf{u},\mathbf{f}^*) = p(\mathbf{f}|\mathbf{u})$$

با توجه به این رابطه داریم:

$$p(\mathbf{f}|\mathbf{u},\mathbf{f}^*) = \frac{p(\mathbf{f},\mathbf{f}^*|\mathbf{u})}{p(\mathbf{f}^*|\mathbf{u})} \approx p(\mathbf{f}|\mathbf{u}) \Rightarrow p(\mathbf{f},\mathbf{f}^*|\mathbf{u}) \approx p(\mathbf{f}^*|\mathbf{u}).p(\mathbf{f}|\mathbf{u})$$

با توجه به این رابطه مشاهده می شود در صورتی که $\mathbf{f}^*$ و $\mathbf{f}$ را روی $\mathbf{u}$ شرطی کنیم، دارای توزیع مستقل از هم خواهندبود. باید توجه شود که این شرطی کردن، یک تقریب است. لذا برای تفکیک توزیع های تقریبی با توزیع های دقیق از q به جای p استفاده می کنیم. لذا داریم:

$$p(\mathbf{f},\mathbf{f}^*) = \int p(\mathbf{f},\mathbf{f}^*|\mathbf{u})d\mathbf{u} \approx q(\mathbf{f},\mathbf{f}^*) = \int q(\mathbf{f}|\mathbf{u})q(\mathbf{f}^*|\mathbf{u})d\mathbf{u}$$

حال توزیع $p(\mathbf{f}|\mathbf{u})$ و $p(\mathbf{f}^*|\mathbf{u})$ را بدست آوریم:

$$p(\mathbf{f},\mathbf{u}) \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}\mathbf{K_{f,f}} & \mathbf{K_{f,u}} \\ \mathbf{K_{u,f}} & \mathbf{K_{u,u}}\end{bmatrix}\right)$$

مشابه رابطه ی ۴.۲ می توان نوشت:

$$p(\mathbf{f}|\mathbf{u}) \sim \mathcal{N}\left(\mathbf{K_{f,u}}\mathbf{K_{u,u}^{-1}}\mathbf{u}, \mathbf{K_{f,f}} - \mathbf{K_{f,u}}\mathbf{K_{u,u}^{-1}}\mathbf{K_{u,f}}\right) \quad (۸.۲)$$

با توجه به رابطه فوق دیده می شود که عملیات عکس کردن در رابطه اولیه GP تبدیل به عکس کردن $\mathbf{K_{u,u}}$ شده است. با انتخاب اندازه ی $\mathbf{u}$ کوچکتر از اندازه $\mathbf{f}$ می توان عملیات را سریع تر انجام داد. حال سوال این است که مجموعه نقاط $\mathbf{u}$ را چگونه باید انتخاب کرد. در ادامه چندین روش مهم مطرح شده برای این منظور را بررسی می کنیم.

روش SoR (Subset of Regressors)

این روش در مقالات [۷۱، ۶۹، ۸۵] مطرح شده است. مدل اصلی به صورت زیر است:

$$\mathbf{f} = \mathbf{K_{f,u}}\mathbf{w_u}, \quad \mathbf{w_u} \sim \mathcal{N}(\mathbf{0}, {\mathbf{K_{u,u}}}^{-1}) \quad \mathbf{u} = \mathbf{K_{u,u}}\mathbf{w_u}.$$

این مدل مشابه روش های مبتنی بر تابع است که در آنها $\mathbf{K_{f,u}}$ ماتریس توابع پایه و $\mathbf{w_u}$ ضرایب آنهاست.

$$\mathbb{E}\mathbf{u} = 0, \quad \mathbb{E}\left[\mathbf{u}\mathbf{u}^T\right] = \mathbb{E}\left[\mathbf{K_{u,u}}\mathbf{w_u}{\mathbf{w_u}}^T\mathbf{K_{u,u}}\right] = \mathbf{K_{u,u}}\mathbb{E}\left[\mathbf{w_u}{\mathbf{w_u}}^T\right]\mathbf{K_{u,u}} = \mathbf{K_{u,u}}.$$

این مدل مشابه روش های مبتنی بر تابع پایه [۱۸] است که در آنها $\mathbf{K_{f,u}}$ ماتریس توابع پایه و $\mathbf{w_u}$ ماتریس ضرایب آنهاست. با جایگزینی $\mathbf{w_u} = \mathbf{K_{u,u}^{-1}}\mathbf{u}$ در مدل اصلی، می توان فرض کرد که مقادیر خروجی ها را روی متغیر های پنهان $\mathbf{u}$ تصویر می کنیم:

$$\mathbf{f}^* = \mathbf{K_{*,u}}\mathbf{K_{u,u}^{-1}}\mathbf{u}, \qquad \mathbf{u} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K_{u,u}}\right)$$

با استفاده از رابطه ی ۸.۲ می توان توزیع متغیر خروجی های داده های آموزشی و داده های جدید مشروط به داده های پنهان را به صورت زیر نوشت:

$$q(\mathbf{f}|\mathbf{u}) = \mathcal{N}(K_{\mathbf{f},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathbf{0}), \quad q(\mathbf{f}^*|\mathbf{u}) = \mathcal{N}(K_{*,\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathbf{0})$$

با استفاده از رابطه ی ۷.۶آ از ضمیمه ی اول می توان توزیع پیشین تقریبی را به صورت زیر نوشت:

$$q(\mathbf{f}, \mathbf{f}^*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\mathbf{f},\mathbf{f}} & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & Q_{*,*} \end{bmatrix}\right)$$

با استفاده از توزیع پیشین تقریبی می توان GP حاصل را با استفاده از روابط GP استاندارد حل کرد. در رابطه فوق $Q_{\mathbf{a},\mathbf{b}} = K_{\mathbf{a},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}K_{\mathbf{u},\mathbf{b}}$. با استفاده از رابطه ی ۴.۲ می توان توزیع پسین را به صورت زیر نوشت:

$$q_{\mathrm{SoR}} = \mathcal{N}\left(Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \mathbf{\Sigma}^2\mathbf{I}\right)^{-1}\mathbf{y}, Q_{*,*} - Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \mathbf{\Sigma}^2\mathbf{I}\right)^{-1}Q_{\mathbf{f},*}\right).$$

نام دیگر برای این روش، تقریب Deterministic Inducing Conditional (DIC) است. این نام گذاری با توجه به رابطه ی قطعی مشخص شده بین $\mathbf{f}$ و $\mathbf{u}$ در صورت مدل است.

روش DTC (Deterministic Training Conditional)

این روش برگرفته از [۱۵] در [۶۷] مطرح شده است. این روش عموما مشهور به Deterministic Training Conditional است. اما در [۶۷] با نام Projected Latent Varaiable(PLV) معرفی شده و در کتاب مشهور [۶۶] به اسم Projected Process Approximation(PPA) آورده شده است.

متغیرهای پنهان $\mathbf{u}$ [۱۹] را در نظر گیریم. می خواهیم تصویر [۲۰] متغیرهای خروجی $\mathbf{f}$ را روی متغیرهای پنهان $\mathbf{u}$ بدست آوریم، به طوریکه لزوما داریم $|\mathbf{u}| < |\mathbf{f}|$. داریم:

$$\mathbf{f} = K_{\mathbf{f},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u} \rightarrow p(\mathbf{f}|\mathbf{u}) = q(\mathbf{f}|\mathbf{u}) = \mathcal{N}(K_{\mathbf{f},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathbf{0})$$

[۱۸] Kernel based methods
[۱۹] Latent variable
[۲۰] Projection

در رابطه فوق، احتمال تقریب زده شده را با q نمایش داده ایم. بدین ترتیب با درنظرگرفتن مدل $\mathbf{y} = \mathbf{f} + \epsilon \quad \epsilon \sim$ $\mathcal{N}(\mathbf{0}, \sigma^{\mathbf{2}}\mathbf{I})$ داریم:

$$p(\mathbf{f}|\mathbf{u}) \approx q(\mathbf{y}|\mathbf{u}) = \mathcal{N}(K_{\mathbf{f},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \sigma^{\mathbf{2}}\mathbf{I})$$

بدیهی است در بدست آوردن توزیع پیش بینی هیچ تقریبی زده نمی شود:

$$q(\mathbf{f}^*|\mathbf{u}) = p(\mathbf{f}^*|\mathbf{u})$$

با استفاده از رابطه ی ۷.۶آ از ضمیمه ی اول می توان توزیع پیشین را به صورت زیر بدست آورد:

$$q(\mathbf{f}, \mathbf{f}^*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\mathbf{f},\mathbf{f}} & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & K_{*,*} \end{bmatrix}\right).$$

که در رابطه فوق $Q_{\mathbf{a},\mathbf{b}} \triangleq K_{\mathbf{a},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}K_{\mathbf{u},\mathbf{b}}$. با کمی عملیات ریاضی مشابه رابطه ی ۴.۲ می توان توزیع پیش بینی را به صورت زیر بدست آورد:

$$q_{\mathrm{DTC}}(\mathbf{f}^*) = \mathcal{N}\left(Q_{*,\mathbf{f}}\left(Q_{*,\mathbf{f}} + \sigma^{\mathbf{2}}\mathbf{I}\right)^{-1}\mathbf{y}, K_{*,*} - Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right)^{-1}Q_{\mathbf{f},*}\right) \qquad (۹.۲)$$

مشاهده می شود که در توزیع پیش بینی فوق، میانگین آن یکسان با تقریب SoR است؛ اما واریانس دو تقریب با هم تفاوت دارند. اگر فرض کنیم $m = |\mathbf{u}|$ و $n = |\mathbf{f}|$، پیچیدگی محاسباتی الگوریتم برابر است با $O(nm^2)$.

روش FITC (Fully Independent Training Conditional)

این روش در [۷۲] مطرح شده است. در مقاله مذکور این روش Sparse Gaussian Process using Pseudo-inputs(SGPP) نامیده شده و در مقالات بعدی به FITC مشهور است. درست نمایی در این روش به صورت زیر تقریب زده می شود:

$$p(\mathbf{y}|\mathbf{f}) \approx q(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(K_{\mathbf{f},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \mathrm{diag}\left[K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}}\right] + \sigma^2\mathbf{I}\right)$$

در توزیع پیش بینی شرطی روی متغیرهای پنهان، هیچ تقریبی نداریم؛ یعنی $q(f^*|\mathbf{u}) = p(f^*|\mathbf{u})$. می توان توزیع پیشین مشترک برای خروجی های جدید و دیده شده را به صورت زیر نوشت:

$$q(\mathbf{f}, f^*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\mathbf{f},\mathbf{f}} - \mathrm{diag}\left[Q_{\mathbf{f},\mathbf{f}} - K_{\mathbf{f},\mathbf{f}}\right] & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & K_{*,*} \end{bmatrix}\right)$$

تفسیر ماتریس کواریانس $K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}}$ ماتریس کواریانس پیشین روی $\mathbf{y}$ منهای ماتریس کواریانس حاصل از اطلاعات $\mathbf{u}$ درباره ی $\mathbf{f}$. در صورتی که ارتباط ریاضی مستقیمی بین $\mathbf{f}$ و $\mathbf{u}$ وجود داشته باشد، $K_{\mathbf{f},\mathbf{f}} = Q_{\mathbf{f},\mathbf{f}}$. می توان مشابه رابطه ی ۴.۲ توزیع پیش بینی را در این تقریب به صورت زیر محاسبه کرد:

$$q_{\mathrm{FITC}}(\mathbf{f}_*|\mathbf{y}) = \mathcal{N}\left(Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \Lambda\right)^{-1}\mathbf{y}, K_{*,*} - Q_{*,\mathbf{f}}\left(Q_{\mathbf{f},\mathbf{f}} + \Lambda\right)^{-1}Q_{\mathbf{f},*}\right), \quad \Lambda = \mathrm{diag}\left[K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right].$$
(۱۰.۲)

پیچیدگی محاسباتی این روش دقیقا مشابه روش های DTC و SoR است.

روش PITC (Partially Independent Training Conditional)

در این تقریب، تقریب FITC بهبود داده می شود؛ به جای اینکه در توزیع PITC از ماتریس قطری استفاده شود، از ماتریس بلوک قطری [۲۱] استفاده می شود. با چنین تقریبی دیگر تمام متغیرهای خروجی دیده شده از هم به طور کامل

[۲۱] Block diagonal matrix

مستقل نیستند؛ بلکه آنهایی که در یک بلوک قرار دارند، وابستگی دارند. مشابه روابط FITC داریم:

$$p(\mathbf{y}|\mathbf{f}) \approx q(\mathbf{y}|\mathbf{f}) = \mathcal{N}\left(K_{\mathbf{f},\mathbf{u}}K_{\mathbf{u},\mathbf{u}}^{-1}\mathbf{u}, \text{blockdiag}\left[K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}}\right] + \sigma^2\mathbf{I}\right), \quad q(f^*|\mathbf{u}) = p(f^*|\mathbf{u})$$

همچنین داریم:

$$q(\mathbf{f}, f^*) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} Q_{\mathbf{f},\mathbf{f}} - \text{blockdiag}\left[Q_{\mathbf{f},\mathbf{f}} - K_{\mathbf{f},\mathbf{f}}\right] & Q_{\mathbf{f},*} \\ Q_{*,\mathbf{f}} & K_{*,*} \end{bmatrix}\right)$$

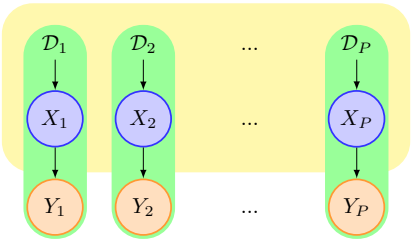
توزیع پیش بینی عینا مطابق رابطه ی ۱۰.۲ بجز اینکه در آن، در تعریف Λ به صورت زیر است:

$$\Lambda = \text{blockdiag}\left[K_{\mathbf{f},\mathbf{f}} - Q_{\mathbf{f},\mathbf{f}} + \sigma^2\mathbf{I}\right]$$

الگوریتم Bayesian Committee Machine

ایده ی اصلی در این تقریب این است که داده های ورودی به P دسته ی مستقل از هم تقسیم کنیم [۸۱، ۶۶]. اگرچه ایده ی Committee Machine تنها مختص GP نیست، اما پیاده سازی آن روی GP آسان تر است. مطابق با شکل ؟؟ داده های ورودی به P قسمت به صورت $\mathcal{D} = \{x_i, y_i\}$ باشد، داریم:

$$p(\mathbf{y}_1, \ldots, \mathbf{y}_P|\mathbf{f}^*, \mathbf{X}) \approx \prod_{i=1}^{P} p(\mathbf{y}_i|\mathbf{f}^*, \mathbf{X})$$



در عمل فرض فوق فرضی بسیار سنگین و ناکارامد است. لذا از آوردن جزئیات عملکرد این روش خودداری می کنیم.

روش Informative Vector Machine

می دانیم که در الگوریتم GP استاندارد، با معکوس ماتریس کواریانس درگیر هستیم که دارای عرض و طول n که باعث می شود عملیات آموزش در زمان $O(n^3)$ انجام گردد. در دیدگاه Informative Vector Machine همه ی داده های آموزشی از لحاظ میزان اطلاعات دارای ارزش یکسان نیستند [۴۳، ۴۰، ۴۱، ۴۴]. بلکه برخی نسبت به برخی دیگر دارای میزان اطلاعات بیشتری هستند. لذا با انتخاب درست زیرمجموعه ای از داده های آموزشی به اندازه d پیچیدگی آموزشی را تا حد بسیاری کاهش داد. با داشتن تابعی که میزان اضافه کردن اطلاعات توسط هر داده آموزشی را به دست دهد، می توان با یک الگوریتم پایین به بالا، مجموعه ی داده های پراطلاعات به طول d را انتخاب کرد. در اینجا تنها به معرفی روابط لازم برای آموزش IVM می پردازیم و جزئیات را در ضمیمه ؟؟ می آوریم. رویکرد کلی که می توان برای ایجاد مدل تُنُک در IVM استفاده شده است، ADF یا Automatic Density Filtering نام دارد که تقریبی گوسی بدست می دهد. فرض کنیم داده های آموزشی به صورت $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ را داشته باشیم. مجموعه داده های پراطلاعات را با I و سایر داده ها را با J نشان می دهیم. در ابتدا که آموزش آغاز می شود، مجموعه I تهی است. در ادامه لازم است پارامتری در اختیار داشته باشیم تا با استفاده از آن، در هر مرحله داده ای را انتخاب

کنیم و تقریب posterior را دقیق تر کند. لازم است تقریبی برای posterior بدست آوریم. در اینجا این تقریب را $Q_i(\mathbf{f})$ را نمایش می دهیم، اگر به مجموعه I، $|I|$ عضو اضافه شده باشد. لذا با دانستن $p(\mathbf{f}|D) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})$ و $p(y_n|f_n) = \mathcal{N}(\mathbf{y_n}|f_n, \beta_n^{-1})$ و رابطه Bayes به صورت $p(\mathbf{f}|\mathcal{D}) = p(\mathbf{f}|\mathcal{D}).\prod_{n=1}^{N} p(y_n|f_n)$ داریم:

$$Q_i(\mathbf{f}) \propto P(\mathbf{f}|\mathcal{D}) \prod_{n=1}^{|I|} \exp\left(-\frac{p_n}{2}(f_n - m_n)^2\right)$$

در ابتدا الگوریتم با تقریب $Q_0(\mathbf{f}) = P(\mathbf{f}|\mathcal{D})$ شروع به کار می کند. در ادامه در هر مرحله با توجه به ADF تقریب گوسی به صورت $Q(\mathbf{f}) = \mathcal{N}(\mathbf{f}|\mathbf{h}, \mathbf{A})$ به دست می آید؛ بطوریکه $\mathbf{A} = \left(\mathbf{K}^{-1} + \mathbf{\Pi}\right)^{-1}$ و $\mathbf{h} = \mathbf{A\Pi m}$ و $\mathbf{\Pi} = diag(\mathbf{p})$. همچنین بردارهای $\mathbf{m} = [m_1...m_N]$ و $\mathbf{p} = [p_1...p_N]$ دارای مقادیر صفر به ازای عناصری که در I نیستند، است. با کمی ساده سازی می توان $\mathbf{A}$ را به صورت به صورت زیر بدست آورد:

$$\mathbf{A} = \mathbf{K} - \mathbf{M}^T\mathbf{M}, \quad \mathbf{M} = \mathbf{L}^{-1}\mathbf{\Pi}_I^{1/2}\mathbf{K}_{I,.}, \quad \mathbf{\Pi} = diag(\mathbf{p})$$

که در آنها ماتریس $\mathbf{L}$ فاکتور پایین مثلث حاصل از تجزیه ی Cholesky ماتریس زیر است(ضمیمه اول):

$$\mathbf{B} = \mathbf{I} + \mathbf{\Pi}_I^{1/2}\mathbf{K}_I\mathbf{\Pi}_I^{1/2}$$

در این مدل از پارامتر آنتروپی به عنوان پارامتری استفاده شده است که نشان می دهد اضافه شدن کدامیک از داده ها به I به صرفه تر است. در واقع آنچه محاسبه می شود، میزان تغییر آنتروپی ناشی از اضافه شدن یک داده ی دلخواه به مجموعه است. داده ی مطلوب داده ای است که مقدار بیشتری به آنتروپی سیستم اضافه کند. برای محاسبه ی میزان تغییر آنتروپی ناشی از اضافه شده داده ای دلخواه لازم است مقادیر h_j و $a_{j,j}$ محاسبه شوند. همچنین لازم است بعد از اضافه کردن داده ای جدید به J، مقادیر $\mathrm{diag}(\mathbf{A})$ و $\mathbf{h}$ را با توجه به داده جدید به روز کنیم. روابط لازم برای به روز رسانی پارامترهای m_i و p_i به صورت زیر هستند:

$$p_i = \frac{\nu_i}{1 - a_{i,i}\nu_i}, \quad m_i = h_i + \frac{\alpha_i}{\nu_i} \tag{۱۱.۲}$$

$$z_i = \frac{y_i.(h_i + b)}{\sqrt{1 + a_{i,i}}}, \quad \alpha_i = \frac{y_i.\mathcal{N}(z_i|0,1)}{\Phi(z_i)\sqrt{1 + a_{i,i}}}, \quad \nu_i = \alpha_i\left(\alpha_i + \frac{h_i + b}{1 + a_{i,i}}\right).$$

هنچنین پارامتر های زیر را در نظر می گیریم: $\mathbf{l} = \sqrt{p_i}\mathbf{M}_{.,i}, \quad l = \sqrt{1 + p_i\mathbf{K}_{i,i} - \mathbf{l}^T\mathbf{l}}, \quad \mu = l^{-1}\left(\sqrt{p_i}\mathbf{K}_{.,i} - \mathbf{M}^T\mathbf{l}\right)$

که در ادامه می توان با استفاده از آنها، بروز رسانی $\mathbf{L} \rightarrow \mathbf{L}^{new}$ را با اضافه کردن ردیف $(\mathbf{l}^T, l)$، بروز رسانی $\mathbf{M} \rightarrow \mathbf{M}^{new}$ را با اضافه کردن ردیف μ انجام داد. در نهایت با بروزرسانی زیر، عملیات بروزرسانی به پایان می رسد:

$$\mathrm{diag}(\mathbf{A}^{new}) \leftarrow \mathrm{diag}(\mathbf{A}) - (\mu_j^2)_j, \quad \mathbf{h}^{new} \leftarrow \mathbf{h} + \alpha_i l p_i^{-1/2}\mu \tag{۱۲.۲}$$

در نهایت می توان داده ی $j \in J$ را که دارای حداکثر تغییرات آنتروپی است را بر اساس رابطه زیر بدست آورد:

$$\Delta_i = \frac{1}{2}\log(1 - a_{j,j}\nu_j) \tag{۱۳.۲}$$

جزئیات مربوط به محاسبات ریاضی روابط فوق در ضمیمه دوم در قسمت توضیحات روش ADF آورده شده است. همچنین در الگوریتم ۱ مراحل اجرای الگوریتم به ترتیب اجرا نشان داده شده است. می توان نشان داد که با داشتن مجموعه داده های $I = \{\mathbf{x_i}, y_i\}_{i=1}^{d}$ می توان به پیچیدگی محاسباتی $O(d^2N)$ (به جای $O(N^3)$) و به پیچیدگی حافظه ای $O(dN)$ (به جای $O(N^2)$) و پیچیدگی محاسباتی در محاسبه داده خروجی به ازای ورودی های جدید $O(d)$ (به جای $O(N)$)دست یافت.

الگوریتم ۱ الگوریتم IVM

Input: Training data $\mathcal{D}$, desired sparsity d.
Output: Set of informative points I.

```
I = ∅, m = 0, Π = diag(0), diag(A) = diag(K), h = 0, J = {1, ..., N}.
repeat
   for j ∈ J do
      Compute Δ_j using equation 2.13.
   end for
   i = arg max_{j∈J} Δ_j
   Update p_i and m_i, using 2.11.
   Update L, M, diag(A), h using equation 2.12 .
   I ← I ∪ {i} , J ← J \ {i}
until |I| = d
```

۹.۲.۲ سایر روش های تقریبی

یکی دیگر از روش های تقریبی در [۲۱] ارائه شده است که در آن عملیات عکس ماتریس با عملیات حداکثرسازی عددی جایگزین شده تا اینکه مرتبه ی محاسباتی الگوریتم به $\mathcal{O}(N^2)$ کاهش پیدا کند. اگرچه مرتبه ی محاسباتی الگوریتم کاهش یافته است، جواب ها نادقیق تر شده اند. به استدلالی دیگر، برای رسیدن به دقت مشابه در حالت بدون تقریب، در روش [۲۱] به زمان بیشتری احتیاج است.

نمونه ای دیگر از بهینه سازی مدل GP برای رگرسیون و کلاس بندی توسط نمونه برداری Monte Carlo معروف به MCMC یا Markov Chain Monte Carlo در [۵۷] انجام شده است. جزئیات مربوط به یادگیری بیزوی با MCMC در ضمیمه ی دوم رساله ارائه شده است.

در [۸۶] از روش Nystrom برای تجزیه ی ماتریس کواریانس $\mathbf{K_{f,f}}$ استفاده شده است. با این تجزیه ادعا شده است که پیچیدگی محاسباتی از $\mathcal{O}(n^3)$ به $\mathcal{O}(nm^2)$ کاهش یافته است.

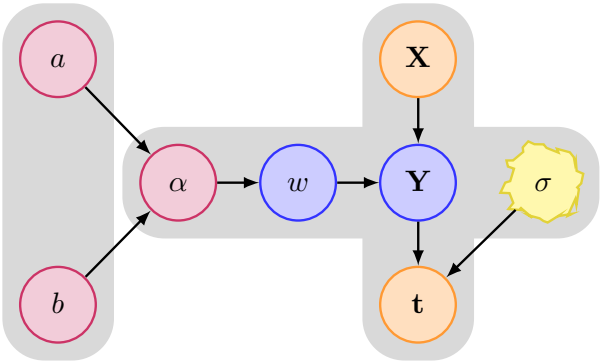
در]۱۴] از تقریب EP (توضیح داده شده در ضمیمه ی ۲ بخش ۵.۷)برای بدست آوردن یک GP سریع استفاده شده است. در]۳۴[مدل معرفی شده در IVM بهبود یافته است.

۳.۲ الگوریتم Relevance Vector Machine

به تدریج که از الگوریتم SVM دور می شویم، کم کم به سمتی حرکت می کنیم که مدل هایی احتمالی برای مدل سازی داده ها بدست آوریم. در حقیقت در مدل های احتمال اگرچه، قضاوت احتمالی است، اما منطقی تر بوده و با ماهیت غیرقطعی [۲۲] داده ها هماهنگ تر است. الگوریتم RVM جزو اولین و مهمترین گام هایی در مدل سازی بر اساس ترکیب خطی توابع پایه [۲۳] است که ساختاری احتمالی برای قضاوت در مورد داده ها را فراهم آورد.

مدل کلی الگوریتم استاندارد RVM به صورت نمایش داده شده در شکل ۱.۸ است [۷۸، ۶، ۷۷، ۷۹، ۷]. در حقیقت مشابه آنچه در مورد الگوریتم SVM معرفی شد، خروجی الگوریتم RVM را می توان به صورت ترکیب خطی از توابع

[۲۲] Uncertain
[۲۳] Kernel methods



شکل ۷.۲: نمایش مدل بین پارامترهای آماری الگوریتم RVM

پایه نوشت.

$$\begin{aligned} \mathbf{t} &= \mathbf{y} + \epsilon \\ &= \sum_{i=1}^{M} w_i K(\mathbf{x}, \mathbf{x}_i) + w_0 + \epsilon \\ &= \mathbf{\Phi w} + \epsilon \end{aligned} \tag{۱۴.۲}$$

پارامترهای مدل نمایش داده شده در شکل ۱.۸ به صورت زیر تعریف می شوند. در این مدل ϵ طبق رابطه ۱۵.۲آ نویز گوسی با میانگین صفر و واریانس σ^2 است و برای مدل سازی عدم اطمینان در مقادیر داده های آموزشی است. طبق رابطه ی ۱۵.۲ب وزن توابع پایه دارای یک توزیع گوسی حول صفر و واریانس α_i^{-1} است. مهم ترین دلیل برای انتخاب این توع توزیع روی ضرایب توابع پایه، بدست آوردن مدلی تنک برای نمایش تابع مورد نظر است.

$$\epsilon \sim \mathcal{N}(0, \sigma^2) \tag{۱۵.۲آ}$$

$$w_i \sim \mathcal{N}(0, \alpha_i^{-1}) \tag{۱۵.۲ب}$$

$$p(\alpha_i) \sim \text{Gamma}(a, b) \tag{۱۵.۲ج}$$

$$p(\sigma^2) \sim \text{Gamma}(c, d) \tag{۱۵.۲د}$$

مراحل لازم برای بدست آوردن پارامترهای مدل از طریق آموزش بوسیله مجموعه داده هایی به صورت $\{\mathbf{x}_n, \mathbf{t}_n\}_{n=1}^N$ انجام خواهد گرفت. با توجه به رابطه ۱۵.۲آ و ۱۴.۲ داریم:

$$p(\mathbf{t}|\mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t} - \mathbf{\Phi w}||^2\right\}. \tag{۱۶.۲}$$

همچنین با فرض مستقل بودن وزن های w_i می توان نوشت:

$$p(\mathbf{w}|\alpha) = \prod_{i=0}^{M} \mathcal{N}(w_i|0, \alpha_i^{-1}). \tag{۱۷.۲}$$

در روابط فوق فرض کرده ایم بردار وزن ها به صورت $\mathbf{w} = [w_i \ ... \ w_M]^T$ است. دلایل مختلفی را می توان برای انتخاب این توزیع برای w_i ها متصور شد:

- با انتخاب توزیعی که دارای حداکثر مقدار حول صفر است، وزن های توابع پایه به سمت صفر میل داده می شوند. زمانی که ضرایب توابع پایه، یعنی w_i ها صفر شوند، در واقع به معنی عدم تاثیر تابع پایه متناظر در خروجی تقریب است. در نهایت، در انتهای آموزش باید انتظار ایجاد مدلی تُنُک [۲۴] از توابع پایه برای تقریب باشیم.

[۲۴] Sparse

الگوریتم ۲ الگوریتم RVM

Input: Training data $\mathcal{D}$.
Output: Regression on output dimensions.

1: Choose a suitable kernel function $\mathbf{\Phi}$, convergence threshold γ_{Th}, pruning threshold α_{Th} and initial values for α and β.
2: **repeat**
3: Compute $\mu = \beta \mathbf{\Sigma} \mathbf{\Phi}^T \mathbf{t}$ and $\mathbf{\Sigma} = \left(\mathbf{A} + \beta \mathbf{\Phi}^T \mathbf{\Phi}\right)^{-1}$.
4: Compute α and β using equations 8.16 and 8.17.
5: Prune the α_i and corresponding kernel function where $\alpha_i > \alpha_{Th}$.
6: Define error convergence rate: $\gamma_i = \Sigma_{ii}\left(\alpha_i^{n+1} - \alpha_i^n\right)$.
7: **until** $\gamma < \gamma_{Th}$

- هر چه تعداد ضرایب w_i کاهش یابد، مدل تنک تر شده و از ایجاد خروجی بیش برازش شده [۲۵] جلوگیری می شود.

جزئیات مربوط به ساده سازی و اثبات فرمول ها در انتهای رساله آورده شده اند.

۴.۲ ارتباط مدل ها

مدل ترکیب خطی زیر بر اساس توابع پایه $\Phi(\mathbf{x})$ را در نظر می گیریم.

$$f(\mathbf{x}) = \Phi(\mathbf{x})\mathbf{w}, \qquad \mathbf{w} \sim (\mathbf{0}, \mathbf{\Sigma}).$$

مدل فوق نمایش مدل سازی RVM است. می توان نشان داد که یک GP معادل با یک RVM است:

$$\begin{cases} \mathbb{E}\left[f(\mathbf{x})\right] = \Phi(\mathbf{x})^T \mathbb{E}\left[\mathbf{w}\right] = 0 \\ \mathbb{E}\left[f(\mathbf{x}) f^T(\mathbf{x}')\right] = \mathbb{E}\left[\Phi(\mathbf{x})^T \mathbf{w}\mathbf{w}^T \Phi(\mathbf{x}')\right] = \Phi(\mathbf{x})^T \mathbb{E}\left[\mathbf{w}\mathbf{w}^T\right] \Phi(\mathbf{x}') = \Phi(\mathbf{x})^T \mathbf{\Sigma} \Phi(\mathbf{x}'). \end{cases}$$

مشاهده می شود یک RVM با مجموعه توابع پایه ی $\Phi(\mathbf{x})$ معادل با یک GP با ماتریس کواریانس $k(\mathbf{x}, \mathbf{x}') = \Phi(\mathbf{x})^T \mathbf{\Sigma} \Phi(\mathbf{x}') = \Sigma_{i=1}^m \alpha_i^{-1} \phi(\mathbf{x}) \phi(\mathbf{x}')$ است. این معادل سازی در عمل درصورتی ممکن است که بتوان ماتریس کواریانس GP را براساس تعداد محدودی از جملات نمایش داد. چنین GP را منحط [۲۶] گویند [۶۲].

به یاد می آوریم که در GP آموزش [۲۷] عبارت بود از بهینه سازی فراپارامترها؛ در حالیکه در RVM این مرحله شامل بهینه سازی فراپارامترها، پیدا کردن Relevance Vector ها برای ایجاد تخمین بهتر است. مرحله ی پیدا کردن Relevance Vector در RVM معادل با استفاده از روش های از پیش فرض شده برای تقریب GP است.

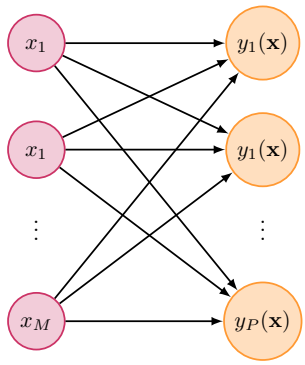
[۲۵] Over-fitted
[۲۶] Degenerate
[۲۷] Training

فصل ۳

تعمیم الگوریتم های آموزش بیزی به چند خروجی

۱.۳ مقدمه

در این قسمت در پی این هدف هستیم که برخی از الگوریتم هایی معرفی شده در فصل ۲ را برای تخمین خروجی چند بعدی تعمیم دهیم. تاکید بر این الگوریتم ها تنها بر جنبه های مختلف یادگیری چند بعد(خروجی)، بدون توجه به رابطه بین خروجی هاست.



شکل ۱.۳: نمایش ساختار بین ورودی ها و خروجی ها؛ هر خروجی مستقل از خروجی های دیگر است و وابسته به تمام ورودی های الگوریتم است.

۲.۳ تعمیم الگوریتم Relevance Vector Machine به چند بعد

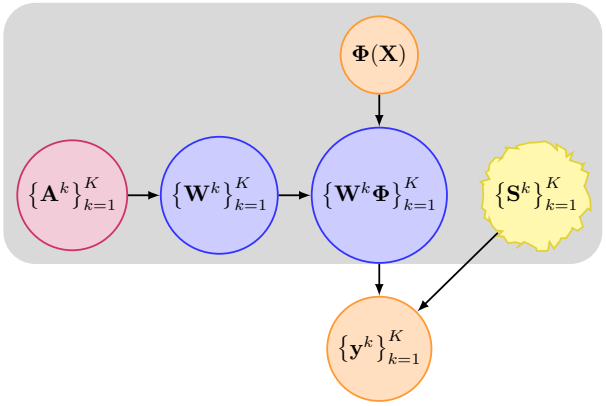
۱.۲.۳ مقدمه

در بخش ۳.۲ الگوریتم استاندارد RVM را توضیح دادیم که برای خروجی اسکالر یک بعدی طراحی شده بود. در این قسمت قصد داریم کارهایی را مرور کنیم که سعی در تعمیم این الگوریتم به چند بعد را داشته اند.

۲.۲.۳ تعمیم ارائه شده توسط [۷۵، ۷۶، ۷۴] (MV-RVM)

در مقالات [۷۵، ۷۶، ۷۴] سعی شده است ایده ی اولیه ی الگوریتم RVM برای رگرسیون چندبعدی (بیش از دو متغیر خروجی) تعمیم داده شود. در حقیقت تعمیم ارائه شده نه تنها چند متغیر خروجی در نظر می گیرد، بلکه هرکدام از

متغیرهای خروجی را یک بردار در نظر می گیرد؛ در حالیکه در الگوریتم RVM استاندارد معرفی شده در بخش ۳.۲ در خروجی تنها یک متغیر اسکالر فرض کرده بودیم. در مقالات مربوطه این الگوریتم Multivariate Relevance Vector Machine یا MV-RVM نامیده شده است؛ لذا ما نیز اینجا آن را به این نام خطاب می کنیم. مشابه الگوریتم



RVM استاندارد که در بخش ۳.۲ معرفی شد، فرض می کنیم داده های ورودی $\mathbf{x} \in \mathbb{R}^Q$ (Q : بعد ورودی) و داده های خروجی $\mathbf{y} \in \mathbb{R}^M$ (M : بعد خروجی) و تعداد متغیر های خروجی، K باشند. برای آموزش الگوریتم، داده های آموزشی $\mathcal{D} = \left\{\mathbf{x}_i, \left\{\mathbf{y}_i^j\right\}_{j=1}^K\right\}_{i=1}^N$ در واقع فرض شده است تعداد N داده ی آموزشی در اختیار ماست. مشابه مدل های دیگر، مدل زیر برای تقریب خروجی انتخاب می شود:

$$\mathbf{y^k} = \mathbf{W}^k\mathbf{\Phi} + \epsilon^k, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{S}^k), \quad 1 \le k \le K$$

در رابطه فوق $\mathbf{S}^k = \text{diag}\left[(\sigma_1^k)^2 \dots (\sigma_M^k)^2\right]$ ماتریس نویز، و $\mathbf{\Phi}$ ماتریس توابع پایه(ماتریس طراحی [۱]) که شامل مقادیری روی تابع پایه [۲] از پیش تعریف شده ی $k(\mathbf{x}_i, \mathbf{x}_j)$ است:

$$\mathbf{\Phi}(\mathbf{x}) = \left[1, \quad k(\mathbf{x}, \mathbf{x}_1), \quad k(\mathbf{x}, \mathbf{x}_2), \quad \dots \quad k(\mathbf{x}, \mathbf{x}_N)\right]^T.$$

ماتریس $\mathbf{W}^k = [\mu_1, \dots, \mu_M] \in \mathbb{R}^{M\times(N+1)}$ ماتریس وزن های توابع پایه است. مشابه الگوریتم RVM ، (معرفی شده در بخش ۳.۲) برای وزن توابع پایه، توزیعی گوسی با میانگین صفر در نظر گرفته می شود:

$$w_{rj}^k \sim \mathcal{N}\left(0, \alpha_j^{k^{-2}}\right), 1 \le r \le M, \quad 1 \le r \le N+1.$$

فرض می کنیم:

$$\mathbf{A} = \text{diag}\left[\alpha_1^{-2} \dots \alpha_{N+1}^{-2}\right].$$

لذا داریم:

$$p(\mathbf{W}^k|\mathbf{A}^k) = \prod_{r=1}^{M}\prod_{j=1}^{N+1}\mathcal{N}\left(w_{rj}^k|0, \alpha_j^{k^{-2}}\right).$$

[۱] Design matrix
[۲] Kernel function

الگوریتم ۳ الگوریتم MV-RVM

Input: Training data $\mathcal{D}$.
Output: Regression(Classification) on output dimensions.

1: $\sigma_r =$ variance of τ_r and $\alpha = \infty$
2: **repeat**
3: Compute $\{\mu_r, \mathbf{\Sigma}_r\}_{r=1}^{M}$ using equations 3. 1 and 3. 1.
4: Compute $\{s_{ri}, q_{ri}\}_{r=1,q=1}^{M,M+1}$ using equations 8.36 and 8. 37.
5: Add theoptimal kernel ϕ_m to the set of optimal kernels, the which most minimizes the log-likelihood based on equation 8.35, or remove the kernel if $\alpha_m = \infty$.
6: Update noise parameters $\left\{\sigma_r^2\right\}_{r=1}^{M+1}$ using equation 8.38.
7: **until MAX-ITERATION**

حال مجهولات مساله عبارتند از $\left\{\mathbf{W}^k, \mathbf{S}^k, \mathbf{A}^k\right\}_{k=1}^{K}$. در نهایت بعد از کمی عملیات ریاضی می توان آموزش الگوریتم را با الگوهای انجام داد:

$$\mu_r = \frac{1}{\sigma_r^2}\mathbf{\Sigma}_r\hat{\Phi}^T\tau_r \qquad (۳.۱آ)$$

$$\mathbf{\Sigma}_r = \left(\frac{1}{\sigma_r^2}\hat{\Phi}^T\hat{\Phi} + \mathbf{A}\right)^{-1} \qquad (۳.۱ب)$$

نمایش دقیق مراحل اجرا در الگوریتم ۳ آمده است. اثبات دقیق مراحل و روابط الگوریتم در ضمیمه ۳.۸ قرار داده شده است.

۳.۲.۳ تعمیم ارائه شده در [۱۶، ۶۱]

در مقالات [۱۶، ۶۱] علاوه بر تعمیم خروجی به چند خروجی، سعی شده است که از چندین نوع تابع پایه [۳] در یک زمان استفاده شود. به استفاده از چندین تابع پایه در آن واحد، Multikernel گفته می شود. در مقالات مورد نظر روش مورد نظر Multi-kernel relevance vector machine یا mRVM نامیده شده است. لذا در ادامه نیز آن را با mRVM خطاب می کنیم.

۳.۳ سایر مدل های چند خروجی

در [۸۸] که اولین بار مدل سازی GP را به جامعه ی یادگیری ماشین معرفی کرده است، مدلی برای چند خروجی نیز ارائه شده است. بر طبق این مدل می توان تابع کواریانس را به صورت $k^{a,b}(\mathbf{x}_i, \mathbf{x}_j) = \delta_{a,b}k(\mathbf{x}_i, \mathbf{x}_j)$ در نظر گرفت که در واقع مستقل سازی رگرسیون بین چند بعد مستقل است. در فصل بعد برای ارزیابی عملکرد الگوریتم ها با همبستگی بین متغیرها، از این مدل استفاده خواهیم کرد. تعمیم مساله ی کلاس بندی از دو کلاس به چند کلاس در [۲۲، ۸۷، ۸۷] انجام شده است.

[۳] Kernel function

فصل ۴

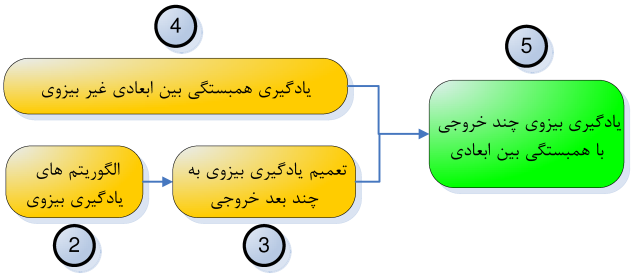
الگوریتم های بیزوی با همبستگی بین ابعادی

۱.۴ مقدمه

تا اینجا در فصل ۲ مهمترین الگوریتم آموزش بیزوی را معرفی کردیم. در فصل ۳ الگوریتم های مطرح شده را به چندین بعد تعمیم دادیم. در این فصل، که در واقع هدف این پژوهش می باشد به معرفی الگوریتم های بیزوی برای آموزش چند خروجی همراه با همبستگی بین ابعادی می پردازیم.

الگوریتم هایی که در این فصل معرفی شده اند، عموما به صورتی ترکیبی از ایده های الگوریتم های بیزوی معرفی شده و االگوریتم های غیر بیزوی یادگیری همبستگی بین ابعادی است. نمایش پیشرفت ایده های پژوهش در تصویر ۱.۴ نمایش داده شده است.

لازم به دکر است با توجه به کثرت ایده های مطرح شده در این زمینه و عدم زمان کافی برای بررسی جزء به جزء تمام ایده ها، تنها برخی از ایده های اخیر را در اینجا بررسی دقیق کرده اید. برای سایر ایده های مطرح شده، تنها به مرور کلیات ایده ها در انتهای فصل بسنده کرده ایم.

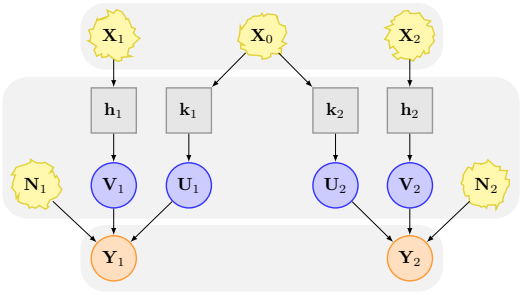


شکل ۱.۴: نمایش روند پیشرفت پژوهش در این رساله. شماره ها نمایش دهنده ی شماره ی فصل هستند.

۲.۴ تعمیم الگوریتم GP برای بیش از یک بعد

در مقاله [۱۰، ۹، ۱۱] ایده ای برای مدل سازی همبستگی بین خروجی ها مطرح شده است. می توان گفت ریشه ی این مقالات در مقاله ی [۲۹، ۴۶] برای استفاده از کانولوشن برای همبستگی بین ابعادی است. بر طبق این ایده، می توان همبستگی بین خروجی ها را توسط مجموعه ای از فرایند های پنهان مدل کرد. برای نمونه، مدل سازی ساختار همبستگی برای دو بعد خروجی در شکل ۲.۸، طبق ایده ی [۱۰] نشان داده شده است. در مدل نشان داده شده، فرض شده است،

مجموعه $\mathcal{S}$ ورودی، و مجموعه $\mathcal{Y}$ خروجی های سیستم هستند. به ازای هر خروجی، مجموعه ای از داده های آموزشی به صورت $\mathcal{D}_1 = \{\mathbf{s}_{1,i}, y_{1,i}\}_{i=1}^{N_1}$ و $\mathcal{D}_2 = \{\mathbf{s}_{2,i}, y_{2,i}\}_{i=1}^{N_2}$ در نظر می گیریم. فرایندهای $\{\mathbf{X}(\mathbf{s})_i\}_{i=1}^{3}$ نویز سفید



شکل ۲.۴: مدل سازی دو Gaussian Process توسط فرایندهای پنهان

گوسی هستند که در توابع پایه h_1، h_2، k_1 و k_2 کانواو شده و خروجی را تشکیل می دهند. برای مثال می توان توابع پایه به صورت زیر گوسی در نظر گرفت:

$$\begin{aligned} k_1(\mathbf{s}) &= v_1 \exp\left(-\tfrac{1}{2}\mathbf{s}^\mathbf{T}\mathbf{A_1}\mathbf{s}\right) \\ k_2(\mathbf{s}) &= v_2 \exp\left(-\tfrac{1}{2}(\mathbf{s}-\mu)^\mathbf{T}\,\mathbf{A_2}\,(\mathbf{s}-\mu)\right) \\ h_i(\mathbf{s}) &= w_i \exp\left(-\tfrac{1}{2}\mathbf{s}^\mathbf{T}\mathbf{B_i}\mathbf{s}\right) \end{aligned} \tag{۱.۴}$$

همانطور که در تعریف توابع پایه دیده می شود، ورودی های الگوریتم، به عنوان آرگومان های توابع پایه در نظر گرفته شده اند. در نهایت فرایند های ورودی $\mathbf{X}_i$، بعد از کانوالو با ورودی، خروجی را می سازند:

$$Y_i(\mathbf{s}) = U_i(\mathbf{s}) + V_i(\mathbf{s}) + N_i(\mathbf{s})$$

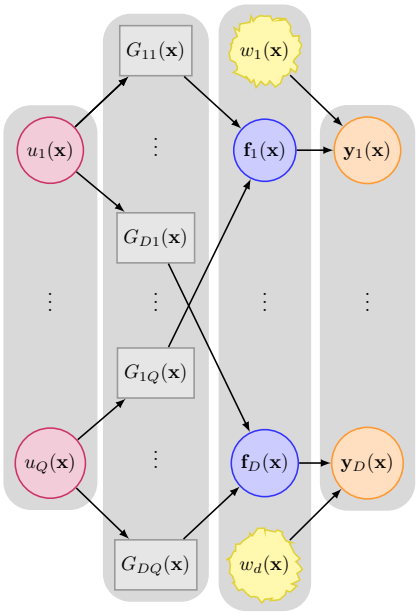
از حقیقت همبستگی بین دو خروجی از طریق فرایند $\mathbf{X}_0$ مدل می شود. همچنین نوآوری های موجود در هریک از دو خروجی از طریق $\mathbf{X}_1$ و $\mathbf{X}_2$ مدل می شود. با توجه به مدل معرفی شده، می توان مشاهده کرد که توزیع خروجی الزاما دارای گوسی است. لذا می توان توزیع مشترک دو خروجی را که یک گوسی است را بدست آورده و برای داده های آموزشی دو بعدی دلخواه مشابه آنچه که در GP استفاده می شود، آموزشی داد. می توان ماتریس کواریانس خروجی را به صورت زیر بدست آورد [۱۰]. جزئیات محاسبه در ضمیمه نیز آورده شده است.

۳.۴ استفاده از فرآیند کانولوشنی برای ایجاد همبستگی بین ابعادی

در این روش فرض شده است خروجی تابع از حاصل کانولوشن بین تعدادی تابع، که نقش هموار سازی [۱] خروجی را دارند، و تعدادی تابع پنهان [۲] بدست آمده است. مدل کلی ساختار معرفی شده در شکل ۳.۴ رسم شده است. با توجه به ساختار معرفی شده، مجموعه توابع $\{u_q(\mathbf{x})\}_{q=1}^{Q}$ توابع پنهان هستند و نقش اضافه کردن همبستگی بین خروجی را دارند. تاثیر مقادیر ورودی در توابع پایه $\{G_{qd}(\mathbf{x})\}_{d=1,q=1}^{D,Q}$ اضافه می شود. در واقع می توان خروجی بدون نویز بعد d–ام را به صورت زیر نوشت:

$$f_d(\mathbf{x}) = \sum_{q=1}^{Q} \int_{\mathcal{X}} G_{d,q}^{i}(\mathbf{x}-\mathbf{z}) u_q^i(\mathbf{z}) d\mathbf{z}. \tag{۲.۴}$$

[۱] Regularization
[۲] Latent function



شکل ۳.۴: نمایش مدل کانولوشنی در ایجاد خروجی های همبسته

پارامترهای مدل نشان داده شده در شکل ۲.۴ را به صورت زیر می توان معرفی کرد:

$$y_d(\mathbf{x}) = f_d(\mathbf{x}) + w_d(\mathbf{x}) \tag{آ۳.۴}$$

$$w_d(\mathbf{x}) \sim \mathcal{N}(\mathbf{x}|0, \sigma^2) \tag{ب۳.۴}$$

$$G_{d,q} = S_{d,q}\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{P}_d^{-1}) \tag{ج۳.۴}$$

$$k_q(\mathbf{x}, \mathbf{x}') = \mathcal{N}(\mathbf{x} - \mathbf{x}'|\mathbf{0}, \mathbf{\Lambda}_q^{-1}) \tag{د۳.۴}$$

$$p(\mathbf{y}|\mathbf{X}, \theta) = \mathcal{N}(0, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \sigma\mathbf{I}) \tag{ه۳.۴}$$

ضریب S_{dq} ضریب کوریانس بین خروجی d–ام و تابع پنهان q–ام است. ماتریس $\mathbf{P}_d$ ماتریس دقت [۳] مربوط به خروجی d–ام است. تابع $k(\mathbf{x}, \mathbf{x}')$ تابع کواریانس برای توابع پنهان است. ماتریس $\mathbf{\Lambda}_q$ ماتریس دقت متغیر پنهان q–ام است. همانطور که در [۲، ۱] توضیح داده شده است، با مدل سازی الگوریتم به صورت شکل ۲.۴ و پارامترهای روابط ۳.۴ می توان خروجی هایی همبسته به دست آورد. لازم به تاکید است که ارتباط خروجی ها با ورودی ها، طبق رابطه ی ۳.۴آ، توسط یک GP مدل می شود. به همین علت است که به مدل فوق یک GP همراه با متغیرهایی پنهان [۴]، گفته می شود.

$$\text{cov}\left[f_d(\mathbf{x}), f_d'(\mathbf{x}')\right] = \sum_{q=1}^{Q} \int_{\mathcal{X}} G_{d,q}^i(\mathbf{x} - \mathbf{z}) \int_{\mathcal{X}} G_{d',q}^i(\mathbf{x}' - \mathbf{z}') k_q(\mathbf{z}, \mathbf{z}') d\mathbf{z}' d\mathbf{z} \tag{۴.۴}$$

$$\text{cov}\left[y_d(\mathbf{x}), y_d'(\mathbf{x}')\right] = \text{cov}\left[f_d(\mathbf{x}), f_d'(\mathbf{x}')\right] + \text{cov}\left[w_d(\mathbf{x}), w_d'(\mathbf{x}')\right] \delta_{d,d'}$$

با فرض کردن توابعی به جای $k_q(\mathbf{z}, \mathbf{z}')$ و $G_{q,d}'(\mathbf{z})$ می توان حاصل کواریانس فوق را ساده کرد. در مدل های قبلی

[۳] Precision matrix
[۴] Latent variable

فرض می شده است که:

$$k_q(\mathbf{z}, \mathbf{z}') = \sigma^2 \delta(\mathbf{z} - \mathbf{z}')$$

در اینصورت می توان رابطه ۴.۴ را به صورت زیر ساده تر کرد:

$$\operatorname{cov}\left[f_d(\mathbf{x}), f_d'(\mathbf{x}')\right] = \sum_{q=1}^{Q} \int_{\mathcal{X}} G_{d,q}^i(\mathbf{x} - \mathbf{z}) G_{d',q}^i(\mathbf{x}' - \mathbf{z}) d\mathbf{z} d\mathbf{z}'$$

در مدل [۲، ۱] فرض شده است که توابع پنهان $u_i(\mathbf{x})$ هر تابعی می توانند باشند. به همین دلیل فرض شده است که روی آنها یک GP با تابع کواریانس $k(\mathbf{x}, \mathbf{x}')$ وجود دارد(رابطه ۳.۴د). صورتی که توابع هموارسازی $G_{d,q}(\mathbf{z})$ را تابعی ضربه در نظر بگیریم، معادله ساده شده و مدل به صورت زیر بدست خواهد آمد:

$$G_{dq}(\mathbf{z}) = \delta(\mathbf{z}) \Rightarrow \operatorname{cov}\left[f_d(\mathbf{x}), f_d'(\mathbf{x}')\right] = \sum_{q=1}^{Q} k_q(\mathbf{z}, \mathbf{z}')$$

همچنین در مورد کواریانس خروجی d–ام با متغیر پنهان q–ام داریم:

$$\operatorname{cov}\left[f_d(\mathbf{x}), u_q^i(\mathbf{z})\right] = \int_{\mathcal{X}} G_{d,q}^i(\mathbf{x} - \mathbf{z}') k_q(\mathbf{z}, \mathbf{z}') d\mathbf{z}'$$

$$k_{f_d, f_d'}(\mathbf{x}, \mathbf{x}') = \Sigma_{q=1}^{Q} S_{dq} S_{d'q} \int_{\mathcal{X}} \mathcal{N}(\mathbf{x} - \mathbf{z} | \mathbf{0}, \mathbf{P}_d^{-1}) \int_{\mathcal{X}} \mathcal{N}(\mathbf{x}' - \mathbf{z}' | \mathbf{0}, \mathbf{P}_d^{-1}) \mathcal{N}(\mathbf{z} - \mathbf{z}' | \mathbf{0}, ^{-1}) d\mathbf{z} d\mathbf{z}'$$

با استفاده از رابطه ۸.۶ از فصل ضمیمه:

$$k_{f_d, f_d'}(\mathbf{x}, \mathbf{x}') = \Sigma_{q=1}^{Q} S_{dq} S_{d'q} \mathcal{N}(\mathbf{x} - \mathbf{x}' | \mathbf{0}, \mathbf{P}_d^{-1} + \mathbf{P}_{d'}^{-1} + \mathbf{\Lambda}_q^{-1}) \tag{۵.۴}$$

نکته عملی که در [۲] آمده این است که در صورتی که تعداد عبارات در رابطه قبل زیاد شود، دقت محاسبه توزیع ها به علت بزرگ شدن ضریب نرمالیزاسیون $\left((2\pi)^{p/2} \left|\mathbf{P}_d^{-1} + \mathbf{P}_{d'}^{-1} + \mathbf{\Lambda}_q^{-1}\right|^{1/2}\right)^{-1}$ به شدت کاهش می یابد. با نرمالیزه کردن خروجی با $\left((2\pi)^{p/4} \left|2\mathbf{P}_{d'}^{-1} + \mathbf{\Lambda}_q^{-1}\right|^{1/4}\right)^{-1}$ و $\left((2\pi)^{p/4} \left|2\mathbf{P}_d^{-1} + \mathbf{\Lambda}_q^{-1}\right|^{1/4}\right)^{-1}$ یعنی اندازه های ماتریس های کواریانس $k_{f_d, f_d}(\mathbf{x}, \mathbf{x}')$ و $k_{f_{d'}, f_{d'}}(\mathbf{x}, \mathbf{x}')$ می توان مشکل را برطرف کرد.

به صورتی مشابه با استفاده از رابطه ۸.۶ از ضمیمه ۶ داریم:

$$\operatorname{cov}\left[f_d, u_q\right] = k_{f_d, f_{d'}} = S_{dq} \mathcal{N}(\mathbf{x} - \mathbf{x}' | \mathbf{0}, \mathbf{P}_d^{-1} + \mathbf{\Lambda}_q^{-1})$$

اکنون روی خروجی ها فرض GP قرار می دهیم. فرض کنیم خروجی ها به صورت $\mathbf{y} = \left[\mathbf{y}_1^T, \ldots, \mathbf{y}_D^T\right]^T$ که هر متغیر خروجی به صورت در تعدادی بعد به صورت $\mathbf{y}_d = \left[y_d(\mathbf{x}_1), \ldots, y_d(\mathbf{x}_N)\right]^T$ تعریف می شود.

$$p(\mathbf{y} | \mathbf{X}, \theta) = \mathcal{N}(\mathbf{y} | \mathbf{0}, \mathbf{K}_{\mathbf{f},\mathbf{f}} + \mathbf{\Sigma}). \tag{۶.۴}$$

در رابطه فوق $\mathbf{K}_{\mathbf{f},\mathbf{f}} \in \mathbb{R}^{DN \times DN}$ که کواریانس بدست آمده توسط فرایند کانولوشنی معرفی شده است که به رابطه ی ۵.۴ سرانجام شده است.

$$\mathbf{K}_{\mathbf{f},\mathbf{f}} = \begin{bmatrix} \mathbf{K}_{\mathbf{f}_1,\mathbf{f}_1} & \ldots & \mathbf{K}_{\mathbf{f}_1,\mathbf{f}_D} \\ \vdots & \ddots & \vdots \\ \mathbf{K}_{\mathbf{f}_D,\mathbf{f}_1} & \ldots & \mathbf{K}_{\mathbf{f}_D,\mathbf{f}_D} \end{bmatrix} \quad \mathbf{K}_{\mathbf{f_i}\mathbf{f_j}} = \begin{bmatrix} k_{f_i,f_j}(\mathbf{x}_1, \mathbf{x}_1) & \ldots & k_{f_i,f_j}(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ k_{f_i,f_j}(\mathbf{x}_N, \mathbf{x}_1) & \ldots & k_{f_i,f_j}(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

در حقیقت $\mathbf{K_{f,f}}$ مقدار کواریانس بین خروجی مربوط به داده ی $\mathbf{x}_i$ مربوط به خروجی d–ام را با خروجی مربوط به داده ی $\mathbf{x}_j$ مربوط به خروجی d'–ام را در بر دارد. ماتریس $\mathbf{\Sigma} \in \mathbb{R}^{ND\times ND}$ اثر نویز در هر بعد را در بر دارد:

$$\mathbf{\Sigma} = \begin{bmatrix} \sigma_1^2\mathbf{I} & 0 & \dots & 0 & 0 \\ 0 & \sigma_2^2\mathbf{I} & \dots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & 0 & \dots & 0 & \sigma_N^2\mathbf{I} \end{bmatrix}_{ND\times ND} \qquad \mathbf{I} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & \ddots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix}_{N\times N}$$

۱.۳.۴ توزیع پیش بینی

مشابه GP استاندارد، می توان به ازای هر داده جدید $\mathbf{X}^*$ توزیع خروجی را از روی رابطه ۴.۲ بدست آورد:

$$p(\mathbf{y}^*|\mathbf{y},\mathbf{X}^*,\mathbf{X},\mathbf{\Theta}) = \mathcal{N}(\mathbf{y}^*|\mathbf{K}_{*,\mathbf{f}}\left(\mathbf{K_{f,f}}+\mathbf{\Sigma}\right)^{-1}\mathbf{y}, \mathbf{K}_{*,\mathbf{f}}\left(\mathbf{K_{f,f}}+\mathbf{\Sigma}\right)^{-1}\mathbf{K_{f,*}}+\Sigma^*) \qquad (۷.۴)$$

با توجه به رابطه ی فوق مشاهده می شود انجام عکس ماتریس در عبارت فوق دارای پیچیدگی $\mathcal{O}(N^3D^3)$ است. می توان پارامترهای مدل فوق را مشابه یک GP معمولی با استفاده از روش هایی بر پایه ی مشتق ساده تر کرد. همچنین می توان مشابه الگوریتم های GP

استاندارد، از روش هایی برای ساده سازی و سرعت بخشیدن به عملیات محاسبه استفاده کرد. در ادامه نمونه ای که در [۱، ۲] مطرح شده است را بررسی می کنیم.

۲.۳.۴ ساده سازی محاسبات خروجی

فرض تقریب بدین صورت است که توزیع خروجی بین ابعاد مختلف با مشاهده ی متغیرهای پنهان مستقل از هم است:

$$p\left(\{f_d(\mathbf{x})\}_{d=1}^D \,|\, \{u_q(\mathbf{x})\}_{d=1}^Q, \mathbf{\Theta}\right) = \prod_{d=1}^D p\left(f_d(\mathbf{x})|\, \{u_q(\mathbf{x})\}_{d=1}^Q, \mathbf{\Theta}\right). \qquad (۸.۴)$$

در عمل چنین فرضی تنها برای شرایطی درست است که تعداد فرایندهای پنهان یا Q قابل مقایسه با تعداد فرایند های خروجی یا D باشد. در رابطه ی فوق فرض شده است Θ پارامترهای مدل است. فرض کنیم M یک عدد از پیش تعیین شده توسط کاربر باشد که $M << N$. مجموعه ی نقاط نمونه گیری شده را $\mathbf{Z}$ می نامیم و به صورت $\mathbf{Z} = \{\mathbf{z}_1, \dots, \mathbf{z}_M\}$ نشان می دهیم. فرض کنیم $\mathbf{u} = [\mathbf{u}_1, \dots, \mathbf{u}_Q]$، که در آن $\mathbf{u}_r = [u_r(\mathbf{z}_1), \dots, u_r(\mathbf{z}_M)]$.

فرض می کنیم روی یک متغیر خروجی خاص و متغیرهای پنهان، یک توزیع مشترک گوسی داشته باشیم:

$$p(\mathbf{f}_q, \mathbf{u}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{\mathbf{f}_q,\mathbf{f}_q} & \mathbf{K}_{\mathbf{f}_q,\mathbf{u}} \\ \mathbf{K}_{\mathbf{u},\mathbf{f}_q} & \mathbf{K}_{\mathbf{u},\mathbf{u}} \end{bmatrix}\right)$$

بدین ترتیب با استفاده از رابطه ی ۵.۶ از ضمیمه ی اول داریم:

$$p(\mathbf{y}_d|\mathbf{u},\mathbf{Z},\mathbf{X},\mathbf{\Theta}) = \mathcal{N}\left(\mathbf{K}_{\mathbf{f}_d,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}\mathbf{u}, \mathbf{K}_{\mathbf{f}_d,\mathbf{f}_d} - \mathbf{K}_{\mathbf{f}_d,\mathbf{u}}{\mathbf{K}_{\mathbf{u},\mathbf{u}}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}_d}\right).$$

با ترکیب رابطه ی فوق در فرض استقلال رابطه ی ۸.۴ خواهیم داشت:

$$p(\mathbf{y}|\mathbf{u},\mathbf{Z},\mathbf{X},\Theta) = \prod_{d=1}^D p(\mathbf{y}_d|\mathbf{u},\mathbf{Z},\mathbf{X},\mathbf{\Theta}) = \prod_{d=1}^D \mathcal{N}\left(\mathbf{K}_{\mathbf{f}_d,\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}\mathbf{u}, \mathbf{K}_{\mathbf{f}_d,\mathbf{f}_d} - \mathbf{K}_{\mathbf{f}_d,\mathbf{u}}{\mathbf{K}_{\mathbf{u},\mathbf{u}}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}_q} + \sigma_d^2\mathbf{I}\right).$$

می توان حاصلضرب فوق را به صورت بسته به صورت بلوک ماتریسی نوشت:

$$p(\mathbf{y}|\mathbf{u},\mathbf{Z},\mathbf{X},\Theta) = \mathcal{N}\left(\mathbf{K}_{\mathbf{f},\mathbf{u}}\mathbf{K}_{\mathbf{u},\mathbf{u}}\mathbf{u}, \mathbf{K_f}, \mathbf{D}+\mathbf{\Sigma}\right),\ \mathbf{D} = \text{blockdiag}\left[\mathbf{K_{f,f}} - \mathbf{K}_{\mathbf{f}_d,\mathbf{u}}{\mathbf{K}_{\mathbf{u},\mathbf{u}}}^{-1}\mathbf{K}_{\mathbf{u},\mathbf{f}}\right].$$

اکنون با فرض اینکه روی مقادیر $u(\mathbf{z})$ یک GP به صورت $\mathbf{u}|\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{u,u}})$ داشته باشیم، روی نمونه های تصادفی مشاهده شده از فرایند $u(\mathbf{z})$ انتگرال گیری می نماییم:

$$p(\mathbf{y}|\mathbf{Z}, \mathbf{X}, \mathbf{\Theta}) = \int p(\mathbf{y}|\mathbf{Z}, \mathbf{X}, \mathbf{\Theta}, \mathbf{u}) p(\mathbf{u}|\mathbf{z}) d\mathbf{u} = \mathcal{N}\left(\mathbf{0}, \mathbf{D} + \mathbf{K_{f,u}} \mathbf{K_{u,u}^{-1}} \mathbf{K_{fu,f}} + \mathbf{\Sigma}\right).$$

رابطه ی فوق توزیع پیشین جدید همراه با تقریب است. اگر این رابطه را با رابطه ی ۶.۴ مقایسه کنیم، رتبه [۵] ی ماریس کواریانس توزیع جدید بدست آمده کمتر شده است که باعث می شود مرتبه ی محاسباتی عکس کردن آن به میزان $\mathcal{O}(N^3D + NDM^2)$ کاهش پیدا کند. در صورتی که $M = N$ باشد، پیچیدگی محاسباتی به میزان $\mathcal{O}(N^3D)$ کاهش پیدا خواهد کرد که معادل با پیچیدگی مدل GP مستقل از هم است. با استفاده از توزیع پسین جدید بدست آمده، می توان این GP جدید را حل کرده و توزیع پیش بینی را به صورت زیر بدست آورد:

$$p(\mathbf{y}^*|\mathbf{y}, \mathbf{X}, \mathbf{X}^*, \mathbf{Z}, \mathbf{\Theta}) = \mathcal{N}\left(\mathbf{K_{f_*,u}} \mathbf{A^{-1}} \mathbf{K_{u,f}} \left(\mathbf{D} + \mathbf{\Sigma}\right)^{-1} \mathbf{y}, \mathbf{D}^* + \mathbf{K_{*,u}} \mathbf{A^{-1}} \mathbf{K_{u,*}} + \mathbf{\Sigma}\right).$$

۴.۴ استفاده از GP به عنوان تابع پایه: مدل Spike and Slab

اخیرا در [۸۰] ایده ای برای استفاده ترکیبی از ویژگی های GP ها در روش های مبتنی بر توابع پایه [۶] معرفی شده است. مدل معرفی شده به این صورت است که فرض می کنیم داده های آموزشی به صورت $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^N$ که در آن $\mathbf{Y} \in \mathbb{R}^{N \times Q}$ و $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N \times D}$ بطوریکه $\mathbf{x}_n$ بردار داده های ورودی باشند.

$$y_{nq} \sim \mathcal{N}(y_{nq}|f_q(\mathbf{x}_n), \sigma_q^2) \quad (۹.۴آ)$$

$$f_q(\mathbf{x}) = \sum_{m=1}^{M} w_{qm} \phi_m(\mathbf{x}) = \mathbf{w}_q^T \phi(\mathbf{x}) \quad (۹.۴ب)$$

$$w_{qm} \sim \pi \mathcal{N}(w_{qm}|0, \sigma_w^2) + (1 - \pi)\delta_0(w_{qm}) \quad (۹.۴ج)$$

$$\phi_m(\mathbf{x}) \sim \mathcal{GP}(\mu_m(\mathbf{x}), k_m(\mathbf{x}_i, \mathbf{x}_j)) \quad (۹.۴د)$$

در مدل فوق، M تعداد پایه ها یا $\mathbf{\Phi} = [\Phi_1(\mathbf{x}) \ldots \Phi_M(\mathbf{x})]$ ها است. همچنین Q تعداد متغیرهای خروجی الگوریتم و هر کدام از متغیرهای خروجی اسکالر فرض شده اند. لذا اگر تعداد داده های آموزشی را با N نشان دهیم، می توان هرکدام از مقادیر خروجی را با y_{nq} نشان داد. همانطور که در [۸۰] ادعا شده است با توجه به اینکه خروجی ترکیب خطی از توابع پنهان [۷] $\{\phi_m(\mathbf{x})\}_{m=1}^M$ است که بین خروجی ابعاد مختلف مشترک هستند، همبستگی بین خروجی های ابعاد مختلف القا [۸] می شود. همچنین برای ابعادی که در مورد داده ها هیچگونه اشتراکی ندارند، کافیست ضرایب w_{nq} مشترکشان صفر باشد. در اینجا یکی از تفاوت های اساسی این مدل با مدل RVM دیده می شود و آن اینکه، بر عکس مدل RVM که در آن مرکز توابع پایه، در ساختار پیش فرض روی داده های آموشی قرار دارند؛ در حالیکه در این مدل، توابع پایه $\{\phi_m(\mathbf{x})\}_{m=1}^M$ غیر پارامتری بوده و توسط یک GP تعریف شده اند(رابطه ی ۹.۴د). این تفاوت می تواند خود را در موارد زیر نشان دهد:

[۵] Rank
[۶] Kernel-based methods
[۷] Latent function
[۸] Induce

۱. اگر در الگوریتم دو بعد خروجی با هم همبستگی داشته باشند، علیرغم نمونه گیری متفاوت در ابعاد مختلف، مدل سازی ارتباط و همبستگی خروجی ها در مدلی مانند RVM استاندارد [۷۹] ، به علت قرار گرفتن مرکز تابع پایه در مکان داده های نمونه، بسیار دشوار است.

۲. همانطور که قبلا نیز اشاره شده است، در بسیاری از الگوریتم ها به دنبال بدست آوردن نمایشی تنک [۹] از توابع پایه هستیم. در الگوریتم RVM استاندارد [۷۹] این نمایش به ازای فرض کردن توابع پایه به مرکزیت داده های آموزشی به دست می آید. در صورتی که ممکن است چینش مناسب تری وجود داشته باشد. برای مثال شکل ۴ .۴ را در نظر گیرید. در شکل ۴.۴آ یک تابع به صورت $\exp\{\frac{(x-0.5)^2}{0.03}\}$ در محدوده $[0,1]$ به همراه نمونه های آن رسم شده است. تابع بازسازی شده توسط RVM استاندارد [۷۹] با توابع پایه مشابه تابع اصلی، در شکل ۴. ۴ب نمایش داده است. اگر چه می توان تابع اصلی را تنها با استفاده از یک تابع پایه در $x = 0.5$ بازسازی کرد، اما الگوریتم RVM استاندارد با استفاده از سه تابع پایه، به تقریبی نه چندان جالب از تابع رسیده است.

۳. تبدیل توابع پایه از حالت قطعی [۱۰] در مدل RVM استاندارد [۷۹] ، به مدلی احتمالی به صورت GP باعث افزایش میزان انعطاف پذیری الگوریتم در ایجاد مناسب توابع پایه خواهدشد. علاوه بر شکل و مکان توابع پایه، تعداد آن ها یعنی M انعطاف پذیر بوده، و از روی داده های آموزشی بدست می آید. به این دلیل است که این مدل را می توان معادل با مدل های چند–پایه ای [۱۱] در نظر گرفت.

تفاوت دیگری که در مدل معرفی شده، نسبت به مدل های قبلی مشابه وجود دارد، نحوه ی قرار داده توزیع پیشین روی ضرایب توابع پایه w_{nq} است. همانطور که در رابطه (۹.۴ ج) مشاهده می شود، عبارتی اضافی به صورت $\pi\delta_0(w_{nq})$ اضافه شده است. این عبارت اضافی احتمال صفر شدن ضرایب را بیشتر می کند. (بر عکس مدل های دیگر که در آن ها، احتمال صفر بودن وزن ها(احتمال نقطه ای در یک توزیع پیوسته)، مقداری صفر است.)

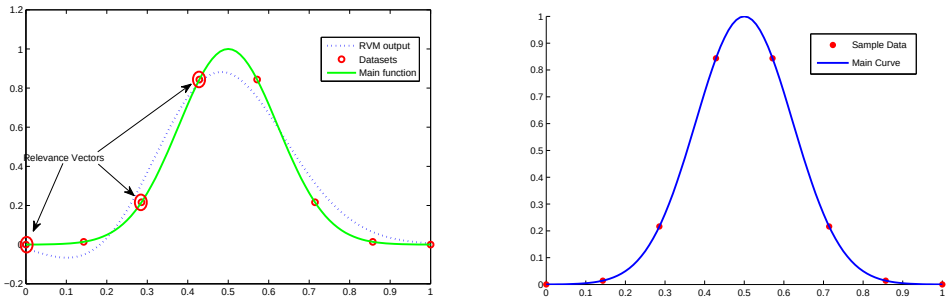
۱.۴.۴ آموزش مدل

با توجه به [۸۰] آموزش مدل معرفی شده در قسمت قبلی با استفاده از روش Variational Bayes انجام می گیرد. توضیحات اولیه مربوط به روش Variational Bayes در قسمت ۳.۷ از ضمیمه ی دوم ارائه شده است؛ لذا در اینجا تنها به معرفی مراحل الگوریتم می پردازیم. یکی از چالش های آموزش این مدل، وجود تابع ضربه در توزیع پیشین روی وزن توابع پایه در رابطه ۹.۴ج است. چرا که بر طبق این روش لازم است توزیع پسین را بتوان به حاصلضرب چند توزیع شناخته شده نوشت. برای حل این مشکل، از توزیع پیشین روی w_{qm} استفاده می کنیم. توزیع های زیر را در نظر می گیریم:

$$\tilde{w}_{qm} \sim \mathcal{N}\left(0, \sigma_w^2\right) \qquad (۱۰.۴آ)$$

$$s_{qm} \sim \pi^{s_{qm}}(1-\pi)^{1-s_{qm}} \qquad (۱۰.۴ب)$$

[۹] Sparse
[۱۰] Deterministic
[۱۱] Multiple kernel learning



(ب) (آ)

شکل ۴.۴: نمایش تاثیر نحوه ی انتخاب توابع در خروجی تقریب، و میزان تُنُک بودن مدل نهایی. در شکل [الف] یک تابع به صورت $\exp\{\frac{(x-0.5)^2}{0.03}\}$ در محدوده $[0,1]$ به همراه نمونه های آن رسم شده است. تابع بازسازی شده توسط RVM استاندارد [۷۹] با توابع پایه مشابه تابع اصلی، در شکل [ب] نمایش داده است. اگر چه می توان تابع اصلی را تنها با استفاده از یک تابع پایه در $x = 0.5$ بازسازی کرد، اما الگوریتم RVM استاندارد با استفاده از سه تابع پایه، به تقریبی نه چندان جالب از شکل اصلی رسیده است.

با توجه به قاعده تبدیل تابع توزیع، با استفاده از دترمینان ژاکوبین، می توان نشان داد که $w_{qm} = \tilde{w}_{qm} s_{qm}$. با فرض استقلال بین توزیع های $\tilde{w}_{qm}$ و s_{qm} می توان گفت که داریم :

$$\Rightarrow \; p(\tilde{w}_{qm}, s_{qm}) = \mathcal{N}(w_{qm}|0,\sigma_w^2)\pi^{s_{qm}}\,(1-\pi)^{1-s_{qm}}$$

توزیع مشترک روی متغیرهای مساله به صورت زیر می باشند:

$$p(\mathbf{Y},\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi}) = \mathcal{N}\left(\mathbf{Y}|\mathbf{\Phi}(\tilde{\mathbf{W}}\circ\mathbf{S})^T,\Sigma\right).\prod_{q,m}\left[\mathcal{N}(\tilde{w}_{qm}|0,\sigma_w^2)\pi^{s_{qm}}(1-\pi)^{1-s_{qm}}.\prod_{m=1}^{M}\mathcal{N}(\phi_m|\mu_m,\mathbf{K}_m)\right]$$

متغیر های $\left\{\mathbf{W},\tilde{\mathbf{S}},\mathbf{\Phi}\right\}$ مجموعه پارامترهای مساله هستند. توزیع پسین به صورت مقابل است:

$$p(\mathbf{Y}) = \sum_{\mathbf{S}}\int_{\tilde{\mathbf{W}},\mathbf{\Phi}}\frac{p(\mathbf{Y},\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi})}{p(\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi})}d\tilde{\mathbf{W}}d\mathbf{\Phi} \qquad (۱۱.۴)$$

توزیع $q(\mathbf{\Phi},\mathbf{S},\tilde{\mathbf{W}})$ توزیعی کمکی است که از آن برای برای تقریب توزیع پسین استفاده می کنیم. با تجزیه ی پارامتر به فاکتور های مستقل از هم داریم:

$$q(\mathbf{\Phi},\mathbf{S},\tilde{\mathbf{W}}) = \prod_{q=1}^{Q}\prod_{m=1}^{M}q(\tilde{w}_{qm},s_{qm})\prod_{m=1}^{M}q(\phi_m)$$

می دانیم $q(\phi_m)$ دارای توزیع گوسی N –متغیره هست. لذا می توان توزیع مشترک چند متغیره کلی را به صورت زیر تجزیه کرد:

$$\begin{aligned} p(\mathbf{Y},\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi}) \;&= \prod_{q=1}^{Q}\mathcal{N}(\mathbf{y}_q|\sum_{m=1}^{M}s_{qm}\tilde{w}_{qm}\phi_m,\sigma_q^2) \\ &\times\prod_{q=1}^{Q}\prod_{m=1}^{M}\left[\mathcal{N}(\tilde{w}_{qm}|0,sigma_w^2)\pi^{s_{qm}}(1-\pi)^{1-s_{qm}}\right]\prod_{m=1}^{M}\mathcal{N}(\phi_m|\mathbf{0},\mathbf{K}_m), \end{aligned} \qquad (۱۲.۴)$$

کران پایین برای $p(\mathbf{Y})$ به صورت زیر است:

$$\mathcal{F} = \sum_{\mathbf{S}}\int_{\tilde{\mathbf{W}},\mathbf{\Phi}}q(\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi})\log\frac{p(\mathbf{Y},\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi})}{q(\tilde{\mathbf{W}},\mathbf{S},\mathbf{\Phi})}d\tilde{\mathbf{W}}d\mathbf{\Phi} \qquad (۱۳.۴)$$

در ادامه، حداکثرسازی کران پایین $\mathcal{F}$ با بروزرسانی دوری انجام می گیرد تا پارامترها و توزیع های بهینه بدست آیند. جزئیات بروزرسانی و نحوه ی بدست آوردن آنها در [۸۰] و توضیحات اضافی آن، به طور کامل آورده شده اند. لذا از آوردن دوباره ی آنها خودداری می کنیم. نکته ای که در بروز رسانی ها وجود دارد، به علت وابستگی توام $q(\phi_m)$، و θ_m فراپارامترهای GP در توابع پایه، استفاده از روش های به روز رسانی متداول، یعنی به روز رسانی پی در پی پارامتر ها، سرعت همگرایی بسیار کمی خواهد بود. به همین دلیل، در مقاله [۸۰] برای آموزش پارامتر های GP از به روز رسانی توام برای پارامترهای $q(\phi_m)$ و θ_m استفاده کرده است. جزئیات این بروزرسانی ها را به مقاله [۸۰] ارجاع می دهیم.

۵.۴ آزمایش مدل ها روی داده های آزمایشی

در این قسمت الگوریتم را روی داده ی مصنوعی ایجاد شده آزمایش می کنیم. برای ایجاد داده های آموزشی مشابه [۱] عمل می کنیم؛ ۴ بار ۲۰۰ داده ی تصادفی از یک مدل کانولوشنی با پارامترهای زیر استخراج می کنیم:

$$S_{11} = S_{12} = S_{21} = S_{22} = 1, \quad P_{11} = P_{21} = 50, P_{31} = 300, P_{41} = 200 \quad \Lambda = 1$$

داده های ایجاد ایجاد شده را به عنوان خروجی های یک مدل ۴ بعدی در نظر می گیرم و سه الگوریتم را روی آن اعمال می کنیم.

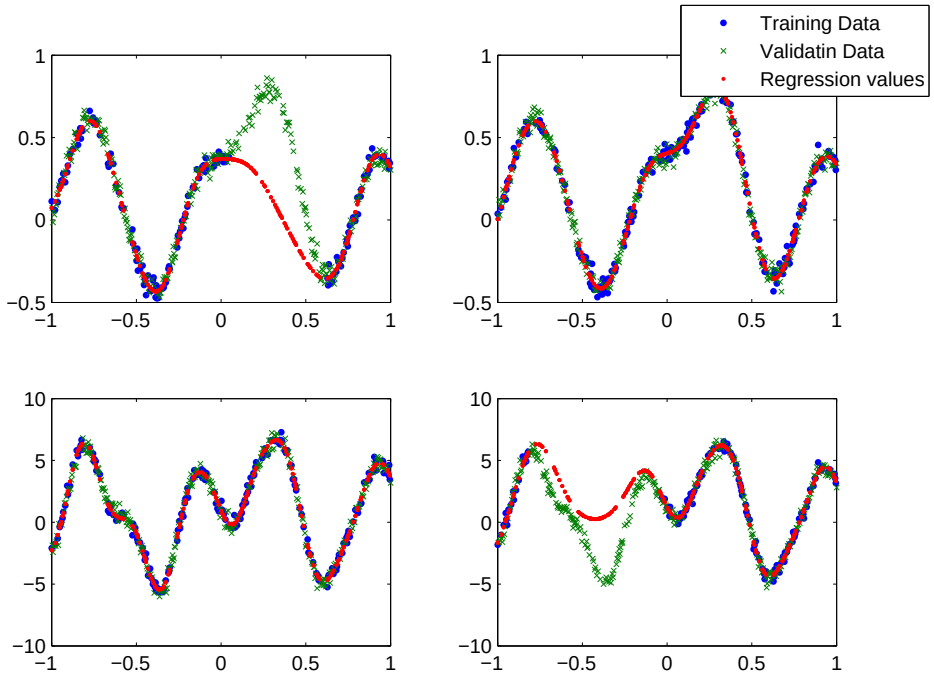
۱. با استفاده از ۴ GP مستقل از هم برای مدل سازی مستقل ۴ بعد؛ نتیجه ی مدل سازی در شکل ۵.۴ نمایش داده شده است.

۲. مدل سازی با استفاده از مدل کانولوشنی؛ نتیجه ی مدل سازی در شکل ۶.۴

۳. مدل سازی با استفاده از مدل Spike and Slab؛ نتیجه ی شکل مدل سازی در شکل ۷.۴ نمایش داده شده است. لازم به ذکر است که در این مدل سازی از ۳ مدل GP به عنوان توابع پایه استفاده شده است. همچنین از تابع مربع نمایی (SE) به عنوان تابع کواریانس مدل استفاده شده است. در واقع فرض شده است که هیچ اطلاعاتی در مورد مدلی که داده های تصادفی از آن به دست آمده اند در دست نیست.

با توجه به شکل های خروجی مشخص است که مدل سازی داده ها با استفاده از الگوریتم ها با همبستگی بین ابعادی بسیار بهتر انجام گرفته است. نکته ای که لازم به ذکر است این است که در مدل کانولوشنی در مدل سازی، محدود به تابع کواریانس گوسی هستیم. در صورتی که در مدل Spike and Slab آزادی عمل بیشتری داریم.

برای ارزیابی بیشتر آزمایشی دیگر ترتیب دادیم. در این آزمایش، از روی یک GP با تابع کواریانس مشخص ۴ سری نمونه ی تصادفی با داده های مستقل از هم با توزیع یکسان ایجاد می کنیم. به صورت تصادفی بازه هایی از داده ها را حذف می کنیم. لازم است الگوریتم بتواند با استفاده از اطلاعات متقاوت بین ابعاد بتواند تقریبی مناسب برای داده های از دست رفته بدست آورد. در شکل ۸.۴ این مساله برای مدل Spike and Slab نمایش داده شده است و با خروجی GP های مستقل مقایسه شده است. نکته ی جالب ای است که به ازای بسیاری از داده های تصادفی ایجاد شده، خروجی مدل کانولوشنی ناپایدار شده و تقریب خوبی از داده ها بدست نمی دهد. در اینجاست که محدویت عملکرد مدل کانولوشنی به دست می آید.



شکل ۵.۴: خروجی مدل GP های مستقل برای داده های آزمایشی؛ در مکان هایی که داده های آموزشی وجود ندارند، مقدار تخمینی از مقدار واقعی فاصله ی بسیاری گرفته است.

۶.۴ مرور کلی سایر ایده های مطرح شده برای ایجاد همبستگی بین ابعادی

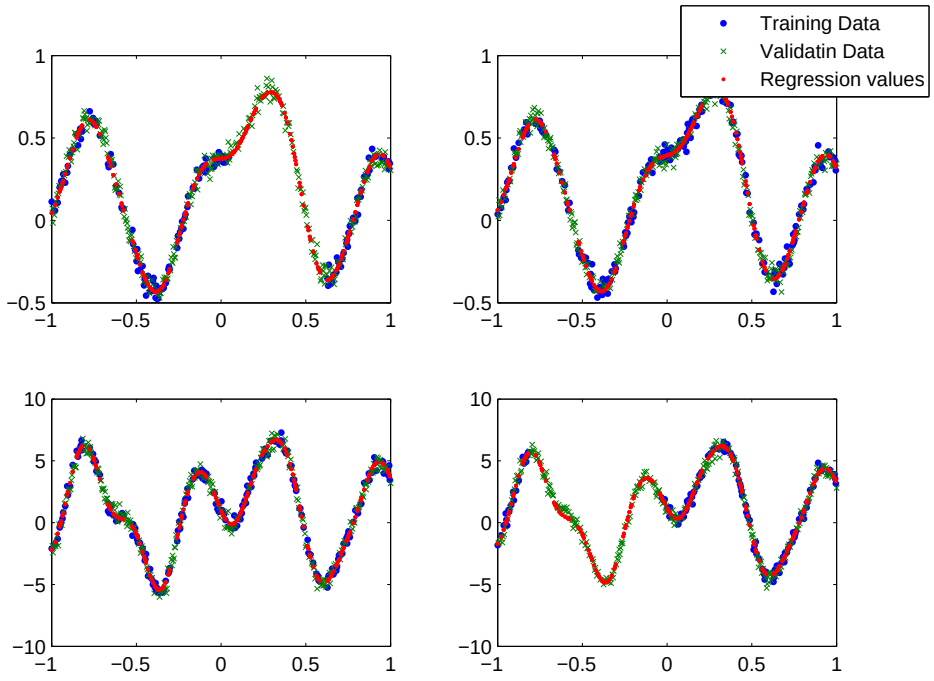
در ابتدای این بخش چند ایده ی کلی برای یادگیری با همبستگی بین متغیرهای خروجی را مطالعه و بررسی کردیم. در ادامه سایر ایده های قدیمی را بررسی خواهیم کرد.

مدلی در [۴۲] ارائه شده است که بر اساس مدل IVM معرفی شده در بخش ۸.۲.۲ است. تعمیم پیچیده تر این مدل در [۷۳] ارائه شده است که ساختار چند GP مستقل به عنوان تابع پایه را به صورت مجموعه ای از توابع پایه به هم متصل کرده است. در [۸۹] مدلی دیگر برای اتحاد چند GP ارائه شده است. در [۸] مدلی معرفی شده است که بسیار شبیه به ساختار معرفی برای Gaussian Process است یک ماتریس کواریانس، علاوه بر ابعاد ورودی، روی ابعاد خروجی در نظر گرفته می شود. در [۸] نشان داده شده است که با داشتن حالت بدون نویز روی داده ها، تقریب خروجی ها تقریبا مستقل از هم انجام خواهد شد. این عملکرد در شرایطی می تواند غیر منطقی باشد. چرا که همبستگی بین خروجی ها، نه تنها می تواند نتیجه ی نویز مشترک بین خروجی ها باشد، می تواند در اثر همبستگی طبیعی بین خروجی ها نیز بدست آید.

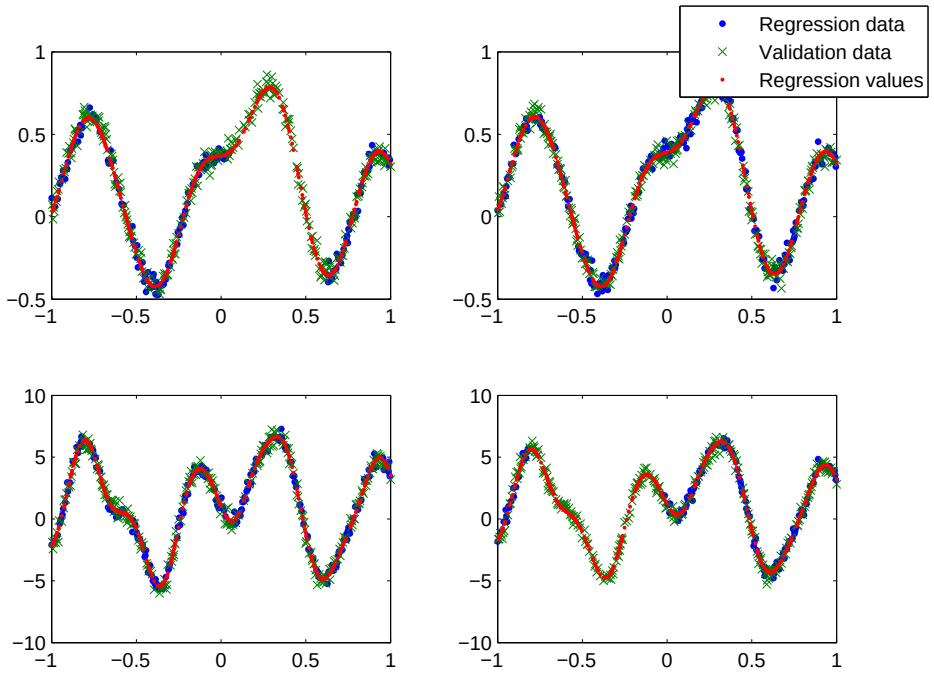
۱.۶.۴ مدل های معرفی شده با ساختار های غیر بیزوی

علاوه بر ساختارهای بیزوی برای ایجاد همبستگی بین ابعادی، ساختارهایی غیر بیزوی نیز برای ایجاد همبستگی بین ابعادی بوجود آمده اند.

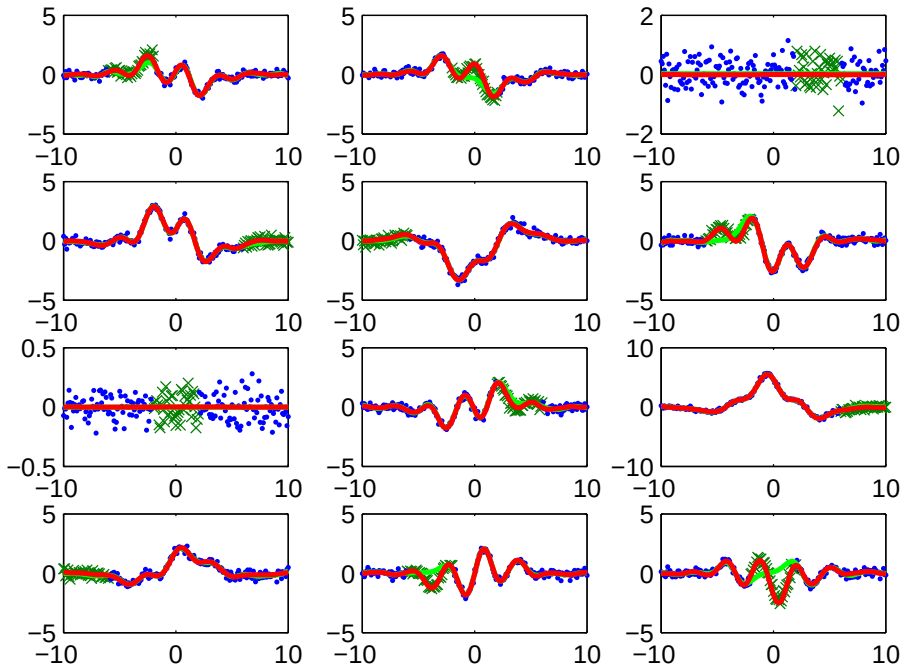
دسته ی بزرگی از این مدل ها بر اساس مدل SVM که ساختاری غیر بیزوی دارد، بنا نهاده شده اند. به عنوان مثال [۳۰] مدلی ساده با چندین تابع پایه برای SVM ارائه داده که ادعا شده است که منجرو به یادگیری بین ابعاد خواهد شد.



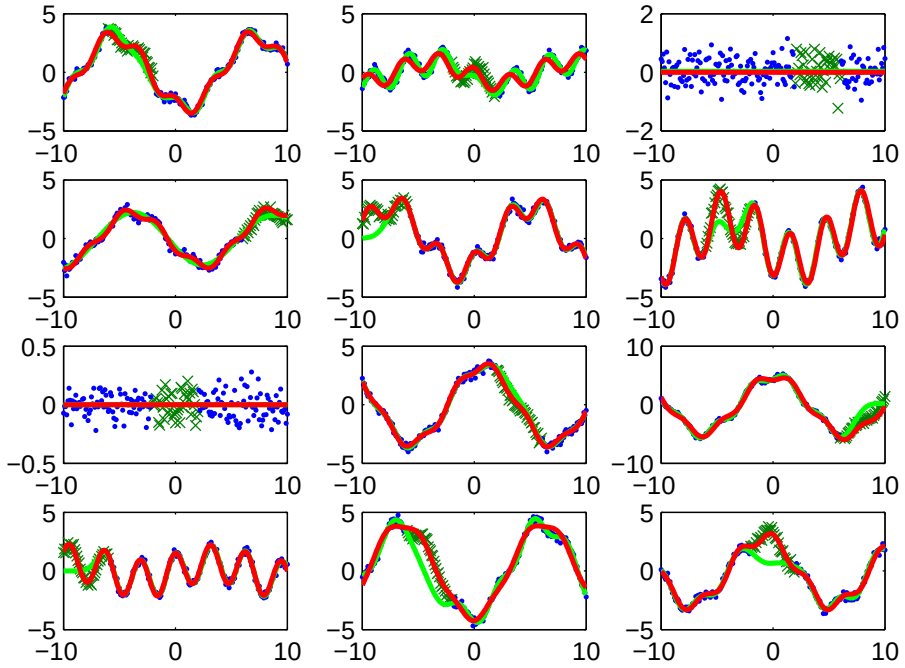
شکل ۶.۴: خروجی مدل Spike and Slab برای داده های آموزشی؛ مدل توانسته است داده های از دست رفته را تا حد بسیار خوبی بازسازی کند.



شکل ۷.۴: خروجی مدل کانولوشنی؛ مدل توانسته است داده های از دست رفته را تا حد بسیار خوبی بازسازی کند.

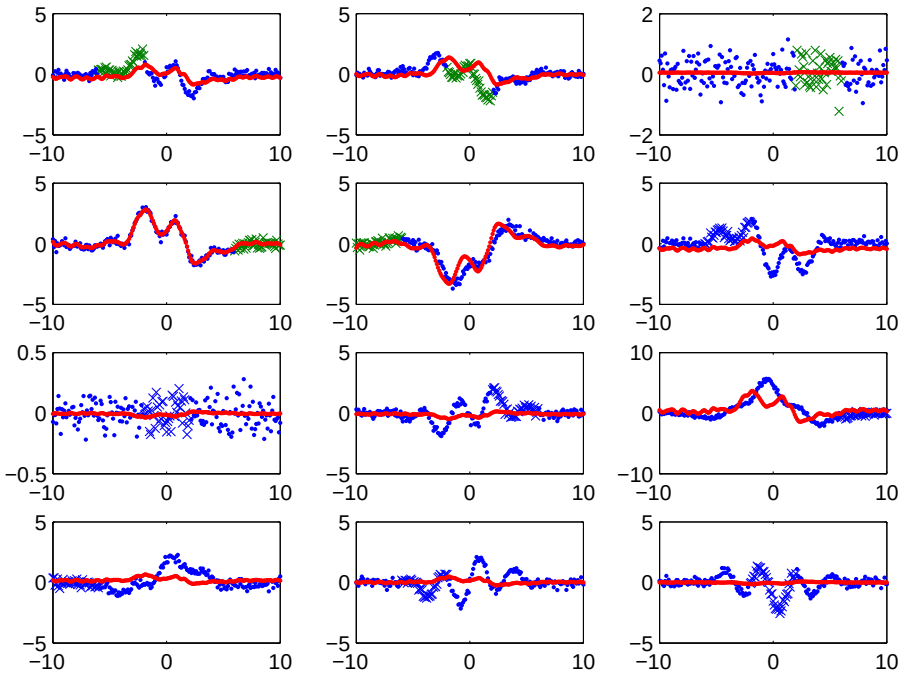


(آ)



(ب)

شکل ۸.۴: در این شکل نتیجه آزمایش مدل ۴.۴ روی مجموعه ای از داده های تصادفی ایجاد شده از یک GP با کواریانس (الف) $k(x_i, x_j) = 4\exp(-x_i^2/20).\cos\left(0.5\left(x_i - x_j\right)\right) + \cos\left(2\left(x_i - x_j\right)\right).\exp\left(-x_j^2/20\right)$ و (ب) $k(x_i, x_j) = 4\cos\left(0.5 \times \left(x_i - x_j\right)\right) + \cos\left(2\left(x_i - x_j\right)\right)$.



شکل ۹.۴: خروجی مدل کانولوشنی به ازای داده های آموزشی؛ مدل به ازای برخی از داده ها ناپایدار شده و نتوانسته است تخمین را انجام دهد.

اگرچه اثباتی برای این ادعا ارائه نشده است. همچنین مدل هایی دیگر برای یادگیری همبستگی چند متغیر در [۱۸، ۵۳] بر اساس مدل SVM ارائه شده است.

در رساله ی [۱۲] قابلیت شبکه ی عصبی برای یادگیری چند متغیرهمبسته به صورت جامع بررسی شده است. در [۲۸، ۲۷] از یادگیری بیزوی برای یادگیری اعمال همبسته در یک شبکه ی دولایه ی پرسپترون استفاده شده است. در [۲۰] شبکه ای گرافیکی مشتمل بر GP مشابه ساختار شبکه عصبی مدل و حل شده است. چنین ساختاری در مقاله ی مذکور Gaussian Process Network نامیده شده است. در [۳] محدودیت های یادگیری بین ابعاد از دیدگاه نظریه ی اطلاعات بررسی شده است. این مقاله با در نظر گرفتن مدلی شامل یک شبکه ی عصبی سعی در بدست آوردن کران های تئوری، در یادگیری بین چند متغیر را مشخص کرده است. در مقاله ی [۵، ۵۲] سعی شده است کران های مربوط به یادگیری چند متغیر توام در حالت کلی تری به دست آید. در واقع هیچ فرضی روی الگوریتم مورد نظر انجام نمی شود.

فصل ۵

نتیجه گیری و کارهای آینده

۱.۵ آنچه در این کار انجام شد

تقریبا تمام ایده هایی که در این رساله بیان شد، مروری بر روش های گذشته است. در فصل اول ابتدا چارچوب بیزوی به صورت کلی معرفی شد و شیوه ی مدل سازی، استنتاج و یادگیری با آن معرفی شد. با توضیحات ارائه شده، مزایا و معایب این چارچوب به صورت کلی ارائه شد. در ادامه، مساله ی اصلی این رساله یعنی یادگیری توام چند متغیر با همبستگی بین ابعادی تعریف و اهمیت آن توضیح داده شد. در فصل دوم یکی از مهمترین الگوریتم های یادگیری بیزوی یعنی Gaussian Process به صورت کامل معرفی شد و اثر انتخاب های مختلف از پارامترهای آن بررسی شد. در ادامه مهمترین الگوریتم های توسعه داده شده برای بدست آوردن یک یادگیری سریع و بهینه بر اساس GP توضیح داده شد. در فصل نسبتا کوتاه سوم، سعی کردیم تعمیم الگوریتم های معرفی شده در فصل دوم برای یادگیری مستقل چند متغیر را شرح دهیم. در ادامه ی بحث، در فصل ۴ دو نمونه از مدل های اخیر معرفی شده برای پیاده سازی یادگیری با همبستگی بین متغیرها معرفی شده و نتایج پیاده سازی آنها روی چند آزمایش نمونه نشان داده شد. مشاهده شد که در نمونه های ایجاد شده، عملکرد الگوریتم ها می توان قابل توجه باشد. اگرچه مواردی از عدم موفقیت آنها در انجام یادگیری مشاهده شد که دلایل آنها از دیدگاه محقق بیان شدند. از پیاده سازی سایر مدل های قدیمی به علت مشابه بودن مدل ها، و در برخی از موارد غیر قابل پیاده سازی بودن ایده، همچنین ضیق وقت خودداری شد. در انتها تنها به سایر ایده های مطرح شده اشاره شد.

۲.۵ ایده ها و کارهای آینده

یادگیری بین ابعادی و یادگیری بیزوی یکی از جدیدترین و داغ ترین زمینه هایی رو به پیشرفت در یادگیری ماشین هستند. به علت وسعت و تنوع ایده های مطرح در زمینه های مختلف هوش محاسباتی، ایده های مختلفی برای کارهای آینده می توان مطرح کرد. یکی از این ایده ها تعمیم ایده ی شبکه عصبی بیزوی گسسته به شبکه ی عصبی پیوسته (مثلا در]۳۵[) است. به عنوان ایده ای دیگر می توان الگوریتم معرفی شده به عنوان مدل کانولوشنی برای یادگیری بین ابعادی را با استفاده از روش IVM برای یادگیری GP انجام داد. ایده ی دیگر، ایده ی تعمیم یادگیری بین ابعادی برای الگوریتم RVM است که تاکنون هیچ مدل چند خروجی با همبستگی بین ابعادی برای آن مطرح نشده است. برای حل این مساله می توان از الگوریتم های مشابه که برای ساختار SVM مطرح شده اند، بهره برد. یکی دیگر از کارهای مهم

آینده می تواند پیاده سازی الگوریتم های موجود روی پدیده های طبیعی باشد. در واقع این الگوریتم ها باید بتوانند برای مسائل واقعی عملکرد جالب توجه داشته باشند. ایده های بسیار دیگری نیز وجود دارند که نیاز به بررسی و کار بیشتر هستندکه در ادامه ی کار اینجانب در آزمایشگاه MSPRL سعی بر توسعه ی آنها خواهم داشت.

فصل ۶

ضمیمه اول: مهم ترین روابط ریاضی استفاده شده به همراه اثبات برخی از آنها

۱.۶ مقدمه

۱.۱.۶ روابط مربوط به جبر ماتریس ها

تساوی ماتریسی Woodbury[۱]:

$$(\mathbf{B}+\mathbf{UCV})^{-1}=\mathbf{B}^{-1}-\mathbf{B}^{-1}\mathbf{U}\left(\mathbf{C}^{-1}+\mathbf{VB}^{-1}\mathbf{U}\right)^{-1}\mathbf{VB}^{-1}$$

لم تساوی دترمینان ها: [۲]

$$\det\left(\mathbf{A}+\mathbf{U}.\mathbf{V}^{\mathbf{T}}\right)=\left(\mathbf{I}+\mathbf{V}^{\mathbf{T}}.\mathbf{A}.\mathbf{U}\right).\det\left(\mathbf{A}\right)$$

مشتق عکس یک ماتریس:

$$\frac{\partial}{\partial\theta}K^{-1}=-K^{-1}\frac{\partial K}{\partial\theta}K^{-1}$$

اگر ماتریس K تقارنی و مثبت قطعی [۳] باشد:

$$\frac{\partial}{\partial\theta}\log|K|=\operatorname{tr}\left(K^{-1}\frac{\partial K}{\partial\theta}\right)$$

همچنین مشابه قسمت قبل، اگر ماتریس K تقارنی و مثبت قطعی باشد [۶۶]:

$$\frac{\partial|K|}{\partial\theta}=|K|\operatorname{tr}\left(K^{-1}\frac{\partial K}{\partial\theta}\right)$$

رابطه ی دیگری که در مشتق گیری از توزیع های چند متغیره (بخصوص در توزیع نُرمال) به صورت مقابل است [۱۷]:

$$\text{if } \mathbf{A}\in\mathbb{R}^{m\times n},\quad \mathbf{x}\in\mathbb{R}^{n\times 1}:$$

$$\Rightarrow\nabla_{\mathbf{x}}\mathbf{Ax}=\mathbf{A}^{T}.$$

[۱] Woodbury Matrix Identity
[۲] Matrix determinant lemma
[۳] Positive definite

و

$$\Rightarrow \nabla_{\mathbf{x}} \mathbf{x}^T \mathbf{A}\mathbf{x} = \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x}. \tag{۱.۶}$$

۲.۱.۶ روابط مهم آماری

اگر $\mathbf{x}$ دارای توزیع دلخواهی باشد، در اینصورت می توان گفت:

$$\mathbb{E}\left[\mathbf{A}\mathbf{x} + \mathbf{y}\right] = \mathbf{A}\mathbb{E}\left[\mathbf{x}\right] + \mathbf{y} \tag{آ۲.۶}$$

$$\mathrm{cov}\left[\mathbf{A}\mathbf{x} + \mathbf{y}\right] = \mathbf{A}\mathrm{cov}\left[\mathbf{x}\right]\mathbf{A}^T \tag{ب۲.۶}$$

اگر فرض کنیم $\mathbf{x}$ دارای توزیع نرمال بصورت $\mathbf{x} \sim \mathcal{N}(\mu, \mathbf{\Sigma})$ باشد، آنگاه حقایق زیر را داریم:

$$\mathbf{A}\mathbf{x} + \mathbf{b} \sim \mathcal{N}\left(\mathbf{A}\mu + \mathbf{y}, \mathbf{A}\mathbf{\Sigma}\mathbf{A}^T\right) \tag{آ۳.۶}$$

$$\mathbf{\Sigma^{-1/2}}\left(\mathbf{x} - \mu\right) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right) \tag{ب۳.۶}$$

$$\left(\mathbf{x} - \mu\right)^{\mathbf{T}} \mathbf{\Sigma^{-1}} \left(\mathbf{x} - \mu\right) \sim \chi^2_n \tag{ج۳.۶}$$

هرگاه داشته باشیم

$$\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix}\mathbf{a}\\ \mathbf{b}\end{bmatrix}, \begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^T & \mathbf{B}\end{bmatrix}\right)$$

که در آنها $\mathbf{C}$ ماتریس Cross-covariance بین $\mathbf{x}$ و $\mathbf{y}$ باشد، در اینصورت توزیع های حاشیه ای $\mathbf{x}$ و $\mathbf{y}$ به ترتیب برابر هستند با :

$$\mathbf{x} \sim \mathcal{N}\left(\mathbf{a}, \mathbf{A}\right) \tag{آ۴.۶}$$

$$\mathbf{y} \sim \mathcal{N}\left(\mathbf{b}, \mathbf{B}\right) \tag{ب۴.۶}$$

همچنین توزیع حاشیه ای متغیر ها عبارتند از :

$$\mathbf{x}|\mathbf{y} \sim \mathcal{N}\left(\mathbf{a} + \mathbf{C}\mathbf{B^{-1}}\left(\mathbf{y} - \mathbf{b}\right), \mathbf{A} - \mathbf{C}\mathbf{B^{-1}}\mathbf{C^T}\right) \tag{آ۵.۶}$$

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}\left(\mathbf{b} + \mathbf{C^T}\mathbf{A^{-1}}\left(\mathbf{x} - \mathbf{a}\right), \mathbf{B} - \mathbf{C^T}\mathbf{A^{-1}}\mathbf{C}\right) \tag{ب۵.۶}$$

در حالت خاص اگر داشته باشیم:

$$\mathbf{z} = \begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix}\mathbf{A} & \mathbf{C}\\ \mathbf{C}^T & \mathbf{B}\end{bmatrix}\right)$$

در اینصورت می توان گفت:

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{C^T}\mathbf{A^{-1}}\mathbf{x}, \mathbf{B} - \mathbf{C^T}\mathbf{A^{-1}}\mathbf{C}) \tag{۶.۶}$$

همچنین حاصلضرب دو تابع گوسی، یک تابع گوسی خواهدشد؛ اگرچه حاصل دارای مساحت زیر نمودار یکه نیست.

$$\mathcal{N}\left(\mu_{\mathbf{1}}, \mathbf{\Sigma_1}\right).\mathcal{N}\left(\mu_{\mathbf{2}}, \mathbf{\Sigma_2}\right) \propto \mathcal{N}\left(\mu_{\mathbf{3}}, \mathbf{\Sigma_3}\right)$$

که در رابطه فوق

$$\mathbf{\Sigma_3} = \left(\mathbf{\Sigma_1^{-1}} + \mathbf{\Sigma_2^{-1}}\right)^{-1} \quad (۶.۷آ)$$

$$\mu_{\mathbf{3}} = \mathbf{\Sigma_3 \Sigma_1^{-1}} \mu_{\mathbf{1}} + \mathbf{\Sigma_3 \Sigma 2^{-1}} \mu_{\mathbf{2}} \quad (۶.۷ب)$$

کانولوشن دو تابع گوسی، یک تابع گوسی است؛ اگرچه حاصل دارای مساحت زیر نمودار یکه نیست.

$$\mathcal{N}\left(\mu_{\mathbf{1}}, \mathbf{\Sigma_1}\right) * \mathcal{N}\left(\mu_{\mathbf{2}}, \mathbf{\Sigma_2}\right) \propto \mathcal{N}\left(\mu_{\mathbf{1}} + \mu_{\mathbf{2}}, \mathbf{\Sigma_1} + \mathbf{\Sigma_2}\right)$$

روابط زیر در ساده سازی توزیع حاصلضرب گوسی ها و حاصل تقسیم آنها مفید هستند:

$$\mathcal{N}\left(x|\mu_1, \mathbf{\Sigma}_1\right) . \mathcal{N}\left(x|\mu_2, \mathbf{\Sigma}_2\right) = \mathcal{N}\left(\mu_1|\mu_2, \mathbf{\Sigma}_1 + \mathbf{\Sigma}_2\right) . \mathcal{N}\left(x|\mu, \mathbf{\Sigma}\right) \quad (۶.۸)$$

$$\mathbf{\Sigma} = \left(\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}\right)^{-1}$$

$$\mu = \mathbf{\Sigma}\left(\mathbf{\Sigma}_1^{-1}\mu_1 + \mathbf{\Sigma}_2^{-1}\mu_2\right)$$

با استفاده از ؟؟؟ تساوی های مفید زیر نیز بدست می آیند:

$$\left(\mathbf{\Sigma}_1^{-1} + \mathbf{\Sigma}_2^{-1}\right)^{-1} = \mathbf{\Sigma}_1 - \mathbf{\Sigma}_1\left(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2\right)^{-1}\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 - \mathbf{\Sigma}_2\left(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2\right)^{-1}\mathbf{\Sigma}_2$$

همچنین برای تقسیم:

$$\mathcal{N}\left(x|\mu_1, \sigma_1^2\right) / \mathcal{N}\left(x|\mu_2, \sigma_2^2\right) = \frac{\sigma_2^2 \mathcal{N}\left(x|\mu, \sigma\right)}{\left(\sigma_2^2 - \sigma_1^2\right)\mathcal{N}\left(\mu_1|\mu_2, \sigma_2^2 - \sigma_1^2\right)}$$

$$\mu = \sigma^2\left(\frac{\mu_1}{\sigma_1^2} - \frac{\mu_2}{\sigma_2^2}\right)$$

$$\sigma^2 = \frac{1}{\frac{1}{\sigma_1^2} - \frac{1}{\sigma_2^2}}$$

فصل ۷

ضمیمه دوم: روش های آماری استنباط پارامترها

۱.۷ مقدمه

در همه الگوریتم های رگرسیون [۱] و کلاس بندی [۲] ، بعد از ایجاد ساختار، به دنبال بدست آوردن پارامتر های بهینه برای مدل، بر اساس داده های آموزشی [۳] هستیم. در تمام مدل های قطعی [۴] با تابعی از پارامتر های مدل موجه هستم که پارامتر های بهینه سیستم به دنبال حداکثر (یا گاهی حداقل) کردن آن به دست خواهند آمد. در مدل های آماری یادگیری، این تابع قطعی تبدیل به یک توزیع احتمالی داده های آموزشی، به شرط پارامترهای مدل تبدیل خواهدشد. همانطور که در فصل اول معرفی شد برای بدست آوردن پارامترهای بهینه ی مدل بیزی، لازم است توزیع درست نمایی حاشیه ای بدست آید. در اغلب موارد نمی توان درست نمایی حاشیه ای را به صورت صریح حساب کرد. به همین دلیل باید به دنبال روش هایی برای تقریب آن بود. بعد از تقریب درست نمایی حاشیه ای، می توان از روی آن، مدل مورد نظر را بهینه کرد.

در کل می توان روش های تقریب درست نمایی جاشیه ای را به دسته تقسیم کرد:

۱. روش های مبتنی بر نمونه برداری(معروف به روش های Monte Carlo) که در آن نیاز به پردازش بالا و زمان بسیار است. چنین روش هایی در نهایت به جواب دقیق همگرا خواهند شد.

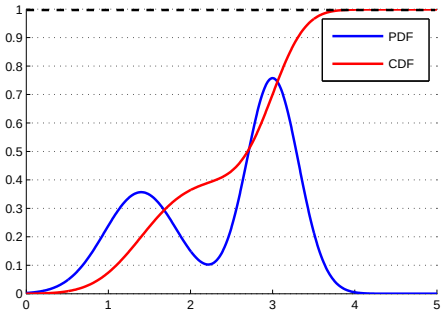
۲. روش های تقریبی که در سعی در تقریب $\mathcal{L}$ دارند؛ اگرچه سریع جواب می دهند، همواره با تقریب همراه اند.

$$\mathcal{L} = \log p(\mathcal{D}|\Theta) \tag{۱.۷}$$

در ادامه می خواهیم انواع روش های متداول برای بدست آوردن پارامترها را از روی توزیع فوق را بررسی کنیم.

برخلاف روش های تقریب با استفاده از نمونه برداری که نیازمند پردازش بی نهایت اند، در ادامه روش هایی از جمله روش Variational Bayes معرفی خواهیم کرد که نیزاز به پردازش نسبی کمتری دارند؛ در عوض روش های

[۱] Regression
[۲] Classification
[۳] Training data-set
[۴] Deterministic



شکل ۱.۷: نمایش یک توزیع دلخواه، توزیع تجمعی مربوطه و استفاده از تابع توزیع تجمعی برای ایجاد نمونه هایی از توریع مورد نظر

مبتنی بر نمونه برداری در حد به جواب تقریب دقیق میل خواهند کرد. در صورتی که روش هایی از جمله VB بر اساس تقریب هایی بنا شده اند که در نهایت، علیرغم سرعت آنها، موجب ایجاد خطا در نتیجه ی نهایی خواهند شد.

۲.۷ روش های مبتنی بر نمونه گیری

در این بخش هدف این است که خلاصه ای از روش هایی از نمونه برداری را معرفی کنیم تا از آنها برای استنباط تقریبی پارامترهای مدل ها استفاده کنیم. اکثر مطالبی که در این بخش آورده شده است از [۷] است. روش های مبتنی بر نمونه برداری، بر اساس ایده س شبیه سازی توزیع های هستند؛ یعنی به جای آنکه به صورت مستقیم صورت ریاضی آنها را آنالیز کنیم، با نمونه های تصادفی ایجاد شده با آن کار می کنیم. در ادامه از ساده ترین ایده های نمونه گیری شروع می کنیم تا به ایده های پیچیده تر برسیم.

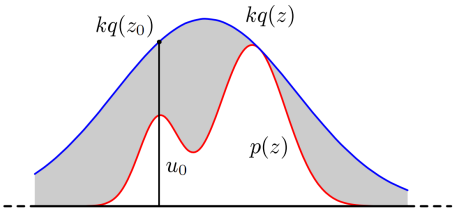
۱.۲.۷ ایجاد نمونه هایی با توزیع مشخص

معمولا می توان توزیع یکنواخت را به وسیله ی الگوریتم های پیاده سازی شده در نرم افزار های محاسباتی ایجاد کرد. با استفاده از این امکان می توان توزیع مشخصی مانند $p(x)$ را از روی توزیع یکنواخت بدست آورد. می دانیم ارتباط بین دو توزیع فرضی به صورت زیر است:

$$p(x) = p(z)\left|\frac{dz}{dx}\right| \ , p(z) = 1 \Rightarrow z = h(x) = P(X \leq x) = \int_{-\infty}^{x} p(x^{'})dx^{'} \Rightarrow x = h^{-1}(z)$$

لذا کافیست توزیع تجمعی مربوط به $p(x)$ را حساب کرده و با استفاده از معکوس آن، توزیع مورد نظر یعنی $p(x)$ را از روی توزیع یکنواخت بسازیم. در شکل ۱.۷ یک توزیع دلخواه و توزیع تجمعی مربوطه نشان داده شده اند. اگر بخواهیم از روی شکل تصور کنیم نمونه های تصادفی از توزیع $p(x)$ چگونه ایجاد می شوند، می توان اینگونه تصور کرد که توزیعی یکنواخت روی محور عمودی قرار داده شده است. به ازای هر نقطه بدست آمده روی محور عمودی، نقطه ی متناظر با آن مقدار را روی توزیع تجمعی بدست آورده و مقدار متناظر را روی محور افقی تصویر می کنیم. با تکرار این عمل، مجموعه نقاط بدست آمده روی محور افقی، دارای توزیع مورد نظر یعنی $p(x)$ خواهند بود.

در بسیاری از مواقع، در مسائل با استنباط آماری، با توزیع هایی پیچیده با ابعاد بسیار مواجه می شویم. لذا باید به دنبال روشی بود که بتوان بدون بدست آوردن توزیع تجمعی و معکوس گیری از آن، نمونه گیری از توزیع مورد نظر را ایجاد



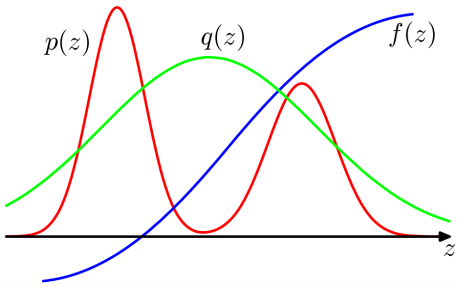
شکل ۲.۷: نمایش عملکرد نمونه گیری ردی. در شکل نمایش داده شده، $p(z)$ توزیعی است که قصد ایجاد نمونه هایی از آن را داریم. همچنین توزیع $q(z)$ توزیع کمکی است که با استفاده از توزیع ها را ایجاد می کنیم. برای اینکه توزیع کمکی تمامی توزیع مورد نظر را بپوشاند، آن را در یک ضریب صحیح ضرب کرده ایم تا $kq(z)$ بدست آید. (تصویر از [۷])

کرد. یکی از این روش های نمونه گیری ردی [۵] است. از شکل ۲.۷ استفاده می کنیم تا عملکرد نمونه گیری ردی را شرح دهیم. در شکل نمایش داده شده، $p(z)$ توزیعی است که قصد ایجاد نمونه هایی از آن را داریم. همچنین توزیع $q(z)$ توزیع کمکی است که از روی یک توزیع شناخته شده ایجاد شده است. برای اینکه توزیع کمکی $q(z)$ تمامی توزیع مورد نظر $p(z)$ را بپوشاند، آن را در یک ضریب صحیح k ضرب کرده ایم تا $kq(z)$ بدست آید. باید توجه شود که توزیع کمکی می تواند هر توزیعی از جمله توزیع نُرمال باشد که با توجه به معلوم بودن صورت آن و سادگی کار با آن، می توانیم نمونه هایی از آن را ایجاد کنیم. مراحل ایجاد یک داده ی تصادفی از یک توزیع پیچیده $p(z)$ به اینصورت است؛ ابتدا از توزیع کمکی $kq(z)$ داده ای تصادفی ایجاد می کنیم. نقطه ی تصادفی ایجاد شده را z_0 می نامیم. بعد از ایجاد نمونه ی تصادفی z_0 از توزیع کمکی $kq(z)$، توزیعی یکنواخت بین صفر و $kq(z_0)$ ایجاد می کنیم و نمونه ای تصادفی از آن را انتخاب می کنیم. در صورتی که نمونه تصادفی زیر توزیع هدف یعنی $p(z)$ باشد، در اینصورت نمونه را قبول می کنیم؛ در غیر اینصورت آن را رَد می کنیم. با توجه به شیوه ی توضیح داده شده در قسمت قبل، بیشتربودن مقدار توزیع کمکی $kq(z)$ از توزیع هدف یعنی $p(z)$ امری ضروری است. همچنین برای اینکه نمونه گیری با کمترین مقدار داده های رَد شده همراه باشد، لازم است فاصله بین $p(z)$ و $kq(z)$ (مشخص شده به رنگ طوسی) تا حد امکان کوچک باشد تا نمونه های رد شده حداقل باشند. با توجه به اینکه توزیع $p(z)$ در عمل می تواند بسیار پیچیده و در فضایی با ابعاد بسیار باشد، انتخاب $q(z)$ و ضریبی مناسب برای آن می تواند عمل بسیار دشواری باشد.

معمولا در مسائل استنباط بیزوی، با نسبتی از $p(z)$ رو برو هستیم. یعنی اگر داشته باشیم $p(z) = \frac{1}{Z_p}\tilde{p}(z)$، که در آن Z_p ضریبی نرمالیزه و نامعلوم است؛ لذا $p(z)$ در دسترس نیست. اما در عوض $\tilde{p}(z)$ معلوم است. لذا به دنبال روشی هستیم که مستقل از ضریب نرمالیزه نمونه های تصادفی را ایجاد کند. اگر دوباره به مراحل روش نمونه برداری ردی برگردیم، می توانیم مشاهده کنیم که این روش مستقل از ضریب نرمالیزه می تواند نمونه های تصادفی را ایجاد کند.

متاسفانه در کاربردهای عملی، با افزایش بعد z، میزان نمونه های ردی، به صورت نمایی افزایش می یابند [۴]. این نکته باعث می شود که نمونه برداری ردی در کاربردهای عملی چندان محبوب نباشد. برای حل این مشکل می توان ایده هایی را مطرح کرد که در آن $\log q(z)$ مقعر [۶] باشد. چنین روشی تحت نام "نمونه برداری ردی پویا [۷]" مطرح می شود. در [۲۴] مطرح شده اند. در این روش $kq(z)$ تکه ای نمایی [۸] است. بعد از هر نمونه برداری به گونه ای بروزرسانی

[۵] Rejection sampling
[۶] Log-concave
[۷] Adaptive rejection sampling
[۸] Peacewise exponential



شکل ۳.۷: نمایش استفاده از نمونه گیری برای تقریب یک انتگرال. (تصویر از [۷])

می شود که به $p(z)$ نزدیک تر شود. نمونه ای مشابه در [۲۳] نیز ارائه شده است.

۲.۲.۷ روش های مبتنی بر نمونه گیری برای تقریب مقدار انتگرال ها

فرض کنیم بخواهیم انتگرال $\int f(z)p(z)dz$ را که معادل با امید ریاضی [۹] $\mathbb{E}_p[f]$ حساب کنیم. با توجه به قضیه ی همگرایی میانگین آماری به میانگین احتمالی، عبارت زیر تخمینی برای امید ریاضی اشاره شده است. این تخمین گر، تخمین گری ناااریب [۱۰] از مقدار واقعی است. توجه شود که داده های نمونه z^i از توزیع $p(z)$ نمونه برداری می شوند.

$$\mathbb{E}_p[f] \approx \frac{1}{L}\sum_{i=1}^{L} f(z^i), \quad z^i \sim p(z)$$

در صورت یکه نتوان نمونه برداری را از روی $p(z)$ انجام داد، می توان مشکل را با کمی تغییر در راه حل برطرف کرد.

$$\mathbb{E}_p[f] = \int f(z)p(z)dz = \int f(z)\frac{p(z)}{q(z)}q(z)dz \approx \frac{1}{L}\sum_{i=1}^{L} f(z^i)\frac{p(z^i)}{q(z^i)}, \quad z^i \sim q(z)$$

مطابق شکل ۲.۲.۷ چون نمی توان از روی $p(z)$ نمونه برداری کرد، به ناچار از روی $q(z)$ نمونه برداری کرده و مقدار انتگرال را با استفاده از نمونه های بدست آمده حساب می کنیم. چنین روشی Importance Sampling نام دارد. این روش یکی از کارامدترین روش های نمونه برداری و تقریب انتگرال در بسیاری از کاربردهای عملی است. با استفاده از این ایده ساده می توان احتمالات بسیار کوچک کوچک یا بسیار نزدیک به یک، یا هر نسبتی که مربوط به رخداد یک اتفاق نادر است را با دقت بسیار بالایی اندازه گیری کرد. یکی از چنین کاربردهایی، اندازه گیری Bit error rate با شبیه سازی سیستم هایی است که در آنها احتمال رخداد خطا بسیار کم است [۳۱].

در صورتی که حالت دیگری را در نظر گیریم که در آن مقدار دقیق $p(z)$ را نداریم اما نسبتی از آن را یعنی $\tilde{p}(z)$ توسط ضریب نرمالیزه از رابطه $p(z) = \frac{1}{Z_p}\tilde{p}(z)$ داشته باشیم، می توان محاسبات را به صورت زیر تغییر داد:

$$\mathbb{E}_p[f] = \int f(z)p(z)dz = \frac{1}{Z_p}\int f(z)\tilde{p}(z)dz = \frac{1}{Z_p}\int f(z)\frac{\tilde{p}(z)}{q(z)}q(z)dz = \frac{1}{Z_p}\sum_{i=1}^{L} f(z^i)\frac{\tilde{p}(z^i)}{q(z^i)}, \quad z^i \sim q(z)$$

که می توان ضریب نرمالیزه را از روی رابطه زیر بدست آورد:

$$1 = \mathbb{E}_p[1] = \int p(z)dz = \frac{1}{Z_p}\int \tilde{p}(z)dz = \frac{1}{Z_p}\int \frac{\tilde{p}(z)}{q(z)}q(z)dz = \frac{1}{Z_p}\sum_{i=1}^{L}\frac{\tilde{p}(z^i)}{q(z^i)}, \quad z^i \sim q(z)$$

$$\Rightarrow Z_p = \sum_{i=1}^{L}\frac{\tilde{p}(z^i)}{q(z^i)}, \quad z^i \sim q(z)$$

[۹] Expectation
[۱۰] Unbiased

الگوریتم ۴ الگوریتم نمونه برداری Metropolis-Hastings

1: Start with random samples z_0, s.t. $p(z_0) > 0$.
2: **repeat**
3: genrate random sample, z^* from proposal distribution, $z_* \sim q(Z, z_t)$, given the sandom sample of the previous iteration z_t.
4: Calculate: $\alpha = \min\left\{1, \frac{\tilde{p}(z_*)q(z_*,q_{t-1})}{\tilde{p}(z_{t-1})q(z_{t-1},q_*)}\right\}$.
5: $\alpha = \begin{cases} \geq 1 & : \text{Accept the sample: } x_t = z_* \\ < 1 & : \text{Accept the sample with probability of } \alpha. \end{cases}$
6: **until** TERMINATION-CONDITION

$$\Rightarrow \mathbb{E}_p[f] = \frac{\sum_{i=1}^{L} f(z^i)\frac{\tilde{p}(z^i)}{q(z^i)}}{\sum_{i=1}^{L}\frac{\tilde{p}(z^i)}{q(z^i)}}, \quad z^i \sim q(z)$$

تخمین گر مذکور، یک تخمین گر اریب [۱۱] است [۴]. در حالت کلی همواره نسبت دو تخمین گر نااریب، یک تخمین گر نااریب نیست.

با استفاده از ایده ی انتگرال گیری عددی، می توان انتگرال موجود در توزیع درست نمایی حاشیه ای را از میادن برداشت. هرچه میزان نمونه برداری بیشتر شود، جواب نهایی دقیق تر خواهد بود.

۳.۲.۷ روش Markov Chain Monte Carlo

روش های MCMC تعمیم یافته ی ایده های نمونه برداری از یک توزیع دلخواه با استفاده از مفهوم زنجیره های مارکوف اند؛ بدین ترتیب که الگوریتم در هر مرحله داده ی تصادفی جدیدی همچون $x^{(i)}$ ایجاد می کند که احتمال آن توسط تابع گذار [۱۲] $\mathcal{P}(x, x')$ تعیین می شود.

$$x^{(i-1)} \xrightarrow{\mathcal{P}} x^{(i)}$$

یکی از مشهورترین و مهم ترین الگوریتم های MCMC روش Metropolis-Hastings است [۵۱، ۲۶]. فرض کنیم بخواهیم از توزیع $p(z) = \frac{1}{Z_p}\tilde{p}(z)$ نمونه برداری کنیم. فرضکنیم ضریب نرمالیزه را نداشته باشیم و بخواهیم از $\tilde{p}(z)$ نمونه برداری کنیم. یک توزیع کمکی برای مشخص کردن احتمال گذار نمونه ها تعریف می کنیم:

$$q(z_1, z_2) = \Pr(z_1 \to z_2).$$

این تابع به اسامی مختلف دیگری از جمله Jumping Distribution، Proposal Distribution و Candidate Generating Distribution نیز نامیده می شود. نحوه ی اجرای مراحل نمونه گیری در الگوریتم ۴ نمایش داده شده است. در چنین روش هایی نمونه برداری تصادفی از توزیع های آماری، در صورتی که قدرت محاسباتی بی نهایت داشته باشیم، دقت محاسباتی تا حد بسیار بالایی خواهد بود. یکی از مهم ترین مسائلی که در طراحی الگوریتم های MCMC، بررسی عملکرد آنها از دید زمان رسیدن به توزیع پایای [۱۳] زنجیره است. لازم است تعداد نمونه هایی مانند $z_0, \ldots, z_k$ گرفته و بیرون ریخته شود تا اینکه بعد از نمونه ی k–ام مطمئن باشیم نمونه های $z_{k+1}, z_{k+2}, \ldots$ تا حد زیادی از توزیع مورد نظر پیروی خواهند کرد. چنین زمانی Burn-in period گفته می شود. اثبات اینکه الگوریتم

[۱۱] Biased
[۱۲] Transition function
[۱۳] Stationary distribution

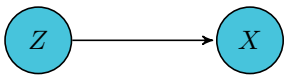
Metropolis-Hastings به یک زنجیره ی مارکوف با توزیع پایای $q(z_1, z_2)$ همگرا خواهد شد در [۲۶] آمده است. یکی دیگر از الگوریتم های مهم خانواده MCMC، الگوریتم نمونه برداری Gibbs است که حالت خاصی از الگوریتم Metropolis-Hastings است که در آن نمونه ها همواره پذیرفته می شوند($\alpha = 1$). این الگوریتم مخصوص نمونه گیری از توزیع های چند متغیره است. به عنوان مثال فرض کنیم بخواهیم از توزیع دو متغیره ی $p(x, y)$ نمونه برداری کنیم. کافیست به صورت دوری، از توزیع های شرطی تک متغیره هر کدام از پارامترها استفاده کنیم.

$$\begin{cases} x_t \sim p(x|y_{t-1}) \\ y_t \sim p(y|x_t) \end{cases}$$

این ایده در حالت کلی برای هر توزیع n–متغیره دیگر درست است. جزئیات بیشتر در مورد مگرایی و اثبات آن در [۷۰، ۶۳] آمده است.

۳.۷ روش Variational Bayes

در ادامه اساس تقریب بر اساس روش Variational Bayes را معرفی خواهیم کرد. توضیحات آمده در ادامه از ترکیبی از مراجع [۱۳، ۷] هستند. شکل ۴.۷ را در نظر می گیریم. در این شکل $\mathbf{X} \in \mathbb{R}^n$ و $\mathbf{Z} \in \mathbb{R}^m$ دو بردار از متغیرهای تصادفی هستند. فرض می کنیم که مشاهداتی از $\mathbf{X}$ داشته ایم و قصد داریم از روی آن ها در مورد متغیرهای پنهان [۱۴] $\mathbf{Z}$ استنباط کنیم. در دنیای احتمال این استباط را به صورت $p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})$ نمایش می دهیم و آن را توزیع پسین می نامیم؛ یعنی هدف بدست آوردن توزیع $\mathbf{Z}$ است، به شرط داشتن اطلاعاتی از روی $\mathbf{X}$. با استفاده از این توزیع می توان تخمین خروجی انجام داد. همچنین علاقمندیم $p_{\mathbf{X}}(\mathbf{x})$ درست نمایی حاشیه ای را بدست آوریم. با استفاده از درست نمایی حاشیه می توان مدل را بهینه کرد.



شکل ۴.۷: مدل گرافیکی برای ارتباط سلسله مراتبی دو متغیر تصادفی

در مورد توزیع مشترک این دو متغیر می توان نوشت:

$$p_{\mathbf{X},\mathbf{Z}}(\mathbf{x}, \mathbf{z}) = p_{\mathbf{X}|\mathbf{Z}}(\mathbf{x}|\mathbf{z})p_{\mathbf{X}}(\mathbf{x}). \tag{۲.۷}$$

در روش VB متداول است از ابزاری به نام Kullback-Leibler divergence یا دیورژان KL برای سنجش میزان درسی تقریب یک توزیع پارامتری به یک توزیع مشخص استفاده شود. در ادامه این پارامتر را بدست می آوریم و عمکرد آن برای تقریب فاصله را نشان می دهیم. از رابطه ۲.۷ داریم:

$$\log p_{\mathbf{X}}(\mathbf{x}) = \log p_{\mathbf{X},\mathbf{Z}}(\mathbf{x}, \mathbf{z}) - \log p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x}). \tag{۳.۷}$$

و داریم:

$$\log p_{\mathbf{X}}(\mathbf{x}) = \log \frac{p_{\mathbf{X},\mathbf{Z}}(\mathbf{x}, \mathbf{z})}{q_{\mathbf{Z}}(\mathbf{z})} - \log \frac{p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})}{q_{\mathbf{Z}}(\mathbf{z})}. \tag{۴.۷}$$

با ضرب دو طرف در $q_{\mathbf{Z}}(z)$ داریم:

$$q_{\mathbf{Z}}(z) \log p_{\mathbf{X}}(\mathbf{x}) = q_{\mathbf{Z}}(z) \log \frac{p_{\mathbf{X},\mathbf{Z}}(\mathbf{x}, \mathbf{z})}{q_{\mathbf{Z}}(\mathbf{z})} - q_{\mathbf{Z}}(z) \log \frac{p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})}{q_{\mathbf{Z}}(\mathbf{z})}. \tag{۵.۷}$$

[۱۴] Latent variables

با مشتق گیری نسبت به $\mathbf{z}$ داریم:

$$\int_{\mathbb{R}^m} q_{\mathbf{Z}}(z) \log p_{\mathbf{X}}(\mathbf{x}) d\mathbf{z} = \int_{\mathbb{R}^m} q_{\mathbf{Z}}(z) \log \frac{p_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z})}{q_{\mathbf{Z}}(\mathbf{z})} d\mathbf{z} - \int_{\mathbb{R}^m} q_{\mathbf{Z}}(z) \log \frac{p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})}{q_{\mathbf{Z}}(\mathbf{z})} d\mathbf{z}. \quad (۶.۷)$$

تعریف می کنیم:

$$\mathcal{L}(q_{\mathbf{z}}) \triangleq \int_{\mathbb{R}^m} q_{\mathbf{Z}}(z) \log \frac{p_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z})}{q_{\mathbf{Z}}(\mathbf{z})} d\mathbf{z} \quad (۷.۷)$$

$$\begin{aligned} KL(q_{\mathbf{Z}}||p_{\mathbf{X}|\mathbf{Z}}) &\triangleq -\int_{\mathbb{R}^m} q_{\mathbf{Z}}(z) \log \frac{p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})}{q_{\mathbf{Z}}(\mathbf{z})} d\mathbf{z} \\ &= \int_{\mathbb{R}^m} q_{\mathbf{Z}}(z) \log \frac{q_{\mathbf{Z}}(\mathbf{z})}{p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\ &= \mathbb{E}_{q_{\mathbf{Z}}} \left[\log \frac{q_{\mathbf{Z}}(\mathbf{z})}{p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})} \right] \end{aligned} \quad (۸.۷)$$

لذا بر اساس تعاریف فوق داریم:

$$\log p_{\mathbf{X}}(\mathbf{x}) = \mathcal{L}(q_{\mathbf{Z}}) + KL(q_{\mathbf{Z}}||p_{\mathbf{Z}|\mathbf{X}}) \quad (۹.۷)$$

تعریف $KL\left(q_{\mathbf{Z}}||p_{\mathbf{Z}|\mathbf{X}}\right)$ را دیورژانس KL می نامیم. این پارامتر همان کمیتی است که قصد داریم از آن برای اندازه گیری فاصله تا مقدار مطلوب استفاده کنیم. از این پس این پارامتر را دیورژانس KL صدا می زنیم. این کمیت دارای ویژگی های زیر است. اثبات این دو رابطه در مقاله [۳۸] ارائه شده است.

$$KL(q_{\mathbf{Z}}||p_{\mathbf{Z}|\mathbf{X}}) \geq 0 \quad (آ۱۰.۷)$$

$$KL(q_{\mathbf{Z}}||p_{\mathbf{Z}|\mathbf{X}}) = 0 \iff q_{\mathbf{Z}} = p_{\mathbf{Z}|\mathbf{X}} \quad (ب۱۰.۷)$$

با توجه به ویژگی دوم می توان دید هرچه دو توزیع به هم نزدیک تر باشند، مقدار دیورژانس KL به صفر نزدیک تر است . اگرچه با توجه به تعریف باید یادآور شد که این دیورژانس دارای خاصیت تقارنی بین مولفه هایش نیست؛ یعنی:

$$KL(p||q) \neq KL(q||p).$$

از این روی کاملا به منزله ی "فاصله" نیست. ایده ی کلی در VB این است که با استفاده از دیورژانس KL مقداری از $q_{\mathbf{Z}}(z)$ را بدست آوریم که مقدار دیورژانس KL را حداقل کند. مقدار بدست آمده برای $q_{\mathbf{Z}}(z)$ تقریبی از توزیع مجهول $p_{\mathbf{Z}|\mathbf{X}}$ است که به آن معمولا توزیع پسین [۱۵] خوانده می شود. به معادله ۹.۷ دوباره توجه می کنیم که در آن مقدار توزیع $q_{\mathbf{Z}}(\mathbf{z})$ را اختیاری انتخاب کرده ایم. متوجه می شویم که سمت چپ معادله، توزیع $q_{\mathbf{Z}}(\mathbf{z})$ نیست. لذا حداقل کردن دیورژانس KL معادل با حداکثرکردن سمت دیگر معادله است؛ یعنی:

$$q^*_{\mathbf{Z}} = \arg\min_{q_{\mathbf{Z}} \in Q} KL(q_{\mathbf{Z}}(z)||p_{\mathbf{Z}|\mathbf{X}}) = \arg\max_{q_{\mathbf{Z}} \in Q} \mathcal{L}(q_{\mathbf{Z}}). \quad (۱۱.۷)$$

که در تعریف فوق، Q مجموعه تمام حالت هایی است که می توان $q_{\mathbf{Z}}$ را از میان آن ها انتخاب کرد. در اینجا لازم به تذکر نیز هست که با توجه به معادله ۱۰.۷آ مشاهده می شود که:

$$\log p_{\mathbf{X}}(\mathbf{x}) \geq \mathcal{L}(q_{\mathbf{Z}}) \Longrightarrow p_{\mathbf{X}}(\mathbf{x}) \geq \exp\left\{\mathcal{L}(q_{\mathbf{Z}})\right\}. \quad (۱۲.۷)$$

[۱۵] Posterior

راه دیگر بدست آوردن این کران از طریق نامساوی [۱۶] است:

$$\ln \int q(\mathbf{z}) \frac{p(\mathbf{z},\mathbf{x})}{q(\mathbf{z})} d\mathbf{z} \geq \int q(\mathbf{z}) \ln \frac{p(\mathbf{z},\mathbf{x})}{q(\mathbf{z})} d\mathbf{z}$$

$$\Rightarrow p(\mathbf{x}) \geq \int q(\mathbf{z}) \ln \frac{p(\mathbf{z},\mathbf{x})}{q(\mathbf{z})} d\mathbf{z} = \exp\left(\mathcal{L}(q(\mathbf{z}))\right)$$

به عبارت دیگر مقدار $\exp\{\mathcal{L}(q_{\mathbf{Z}})\}$ یک کران پایین برای توزیع $\mathbf{X}$ یا درست نمایی حاشیه ای است.

در طی روش برای انجام محاسبات لازم است تقریب هایی بر روی توزیع ها در نظر بگیریم تا حل مساله ممکن شود. اینکه از کدام تقریب استفاده شود و به چه صورتی بستگی به نوع مساله دارد. در ادامه برخی از مهم ترین تقریب هایی را که در زمینه ی این رساله استفاده خواهیم کرد را معرفی می کنیم. یکی از مرسوم ترین روش ها این است که فرض کنیم بین توزیع المان های توزیع روی $\mathbf{Z}$ استقلال وجود دارد:

$$q_{\mathbf{Z}}(\mathbf{z}) = \prod_{i=1}^{m} q_{Z_i(Z_i)} \tag{۱۳.۷}$$

انجام فرض استقلال متغیرها تنها برای آن است که بتوانیم محاسبات را ساده کنیم. در عمل باید بررسی شود که آیا این تقریب عملی است یا خیر. به غیر از فرض استقلال می توان راه های دیگری را یافت که قابلیت انعطاف بیشتری به $q_{\mathbf{Z}(\mathbf{z})}$ برای تقریب $p_{\mathbf{Z}|\mathbf{X}}(\mathbf{z}|\mathbf{x})$ دهد. یک راه دیگر استفاده از یک توزیع پارامتری به صورت $q_{\mathbf{Z}}(\mathbf{z};\omega)$ است که در آن ω پارامترهای توزیع اند. لذا دیورژانس KL به صورت تابعی از ω در خواهدآمد که به دنبال حداقل کردن آن هستیم. می توان پارامترهای بهینه را با بهینه سازی نسبت به ω بدست آورد. در ادامه فرض می کنیم بتوانیم توزیع تقریب زننده $q_{\mathbf{Z}}(\mathbf{z})$ را به تعدادی فاکتور مستقل از هم جدا کرد. باید توجه کرد که در اینجا روی هرکدام از فاکتورها هیچ شرطی نگذاشته ایم. برای انجام آنالیز ابتدا لازم است هرکدام از فاکتورها را از سایر عبارات جدا کنیم تا بتوانیم بصورت جداگانه روی آنها عملیات انجام دهیم؛ توجه به تقریب فوق، روابط معرفی شده بازنویسی می کنیم تا برای المانی دلخواه مانند

[۱۶] Jensen's inequality

Z_j از $\mathbf{Z}$ صورتی بدست آوریم که تنها به Z_j وابسته باشد:

$$
\begin{aligned}
\mathcal{L}(q_{\mathbf{Z}}) &= \mathbb{E}_{q_{\mathbf{Z}}}\left[\log \frac{p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})}{q_{\mathbf{Z}}(\mathbf{Z})}\right] \\
&= \mathbb{E}_{q_{\mathbf{Z}}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z}) - \log q_{\mathbf{Z}}(\mathbf{Z})\right] \\
&= \mathbb{E}_{q_{\mathbf{Z}}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z}) - \log \prod_{k=1}^{m} q_{Z_k}(Z_k)\right] \\
&= \mathbb{E}_{q_{\mathbf{Z}}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right] - \sum_{k=1}^{m} \mathbb{E}_{q_{Z_j}}\left[\log q_{Z_k}(Z_k)\right] \\
&= \mathbb{E}_{q_{Z_j}}\left[\mathbb{E}_{\prod_{\substack{i=1\\ i\neq j}}^{m} q_{Z_i}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right]\right] - \sum_{k=1}^{m} \mathbb{E}_{q_{Z_j}}\left[\log q_{Z_k}(Z_k)\right] \\
&= \mathbb{E}_{q_{Z_j}}\left[\mathbb{E}_{\prod_{\substack{i=1\\ i\neq j}}^{m} q_{Z_i}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right]\right] - \mathbb{E}_{q_{Z_j}}\left[\log q_{Z_j}(Z_j)\right] - \sum_{\substack{k=1\\ k\neq j}}^{m} \mathbb{E}_{q_{Z_k}}\left[\log q_{Z_k}(Z_k)\right] \\
&= \mathbb{E}_{q_{Z_j}}\left[\log\left(\exp\left\{\mathbb{E}_{\prod_{\substack{i=1\\ i\neq j}}^{m} q_{Z_i}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right]\right\}\right)\right] - \mathbb{E}_{q_{Z_j}}\left[\log q_{Z_j}(Z_j)\right] - \sum_{\substack{k=1\\ k\neq j}}^{m} \mathbb{E}_{q_{Z_k}}\left[\log q_{Z_k}(Z_k)\right] \\
&= \mathbb{E}_{q_{Z_j}}\left[\log\left(\exp\left\{\mathbb{E}_{\prod_{\substack{i=1\\ i\neq j}}^{m} q_{Z_i}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right]\right\}\right) - \log q_{Z_j}(Z_j)\right] - \sum_{\substack{k=1\\ k\neq j}}^{m} \mathbb{E}_{q_{Z_k}}\left[\log q_{Z_k}(Z_k)\right] \\
&= \mathbb{E}_{q_{Z_j}}\left[\log \frac{\exp\left\{\mathbb{E}_{\prod_{\substack{i=1\\ i\neq j}}^{m} q_{Z_i}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right]\right\}}{q_{Z_j}(Z_j)}\right] - \sum_{\substack{k=1\\ k\neq j}}^{m} \mathbb{E}_{q_{Z_k}}\left[\log q_{Z_k}(Z_k)\right] \\
&= KL\left(q_{Z_j}(Z_j) || \exp\left\{\mathbb{E}_{\prod_{\substack{i=1\\ i\neq j}}^{m} q_{Z_i}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{X},\mathbf{Z})\right]\right\}\right) - \sum_{\substack{k=1\\ k\neq j}}^{m} \mathbb{E}_{q_{Z_k}}\left[\log q_{Z_k}(Z_k)\right]
\end{aligned}
\tag{۱۴.۷}
$$

حال تمام که فاکتور $q_{Z_j}(Z_j)$ را از سایر فاکتورها جدا کرده ایم، تمامی دیگر فاکتور ها، یعنی $\{q_{Z_i}(Z_i)\}_{i\neq j}$ ثابت می کنیم و تنها نسبت به $q_{Z_j}(Z_j)$ بهینه سازی انجام می دهیم. حداقل مقدار زمانی رخ می دهد که:

$$
\ln q^*_{Z_j}(Z_j) = \mathbb{E}_{\prod_{i=1,i\neq j}^{m}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z})\right] + \text{ cte }, \quad 1 \leq j \leq m \tag{۱۵.۷}
$$

با توجه به خاصیت نرمالیزه بودن توزیع $q^*_{Z_j}$ می توان گفت:

$$
q^*_{Z_j}(Z_j) = \frac{exp\left\{\mathbb{E}_{\prod_{i=1,i\neq j}^{m}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z})\right]\right\}}{\int exp\left\{\mathbb{E}_{\prod_{i=1,i\neq j}^{m}}\left[\log p_{\mathbf{X},\mathbf{Z}}(\mathbf{x},\mathbf{z})\right]\right\} dZ_j}, \quad 1 \leq j \leq m
$$

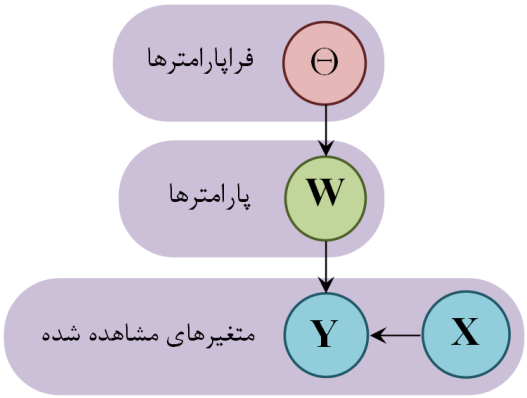
مشاهده می شود که با داشتن $\{q_{Z_i}\}_{i\neq j}$ می توان مقدار $q_{Z_j}(Z_j)$ را بدست آورد. معمولا کار با رابطه ی ۱۵.۷ در عمل راحت تر است. معادلات بدست آمده در رابطه ۱۵.۷ قیود مساله برای حداکثرسازی درست نمایی اند. اگرچه روابط نهایی بدست آمده جواب صریح را بدست نمی دهند؛ چراکه در آنها، مقدار توزیع بهینه هر فاکتور بر اساس فاکتورهای دیگر بدست می آید. لذا جواب های مساله را با مقدار دهی اولیه $\{q_{Z_i}\}_i$ و بروزرسانی دوری آنها، حل می کنیم. در چنین مساله ای، همگرایی بروزرسانی ها، به علت محدب بودن [۱۷] تابع هدف نسبت به هرکدام از فاکتور ها، تضمین شده است [۷].

اکنون مدلی پیچیده تر بصورت شکل ؟؟ در نظر بگیریم.

در مورد توزیع پسین داریم:

$$
p(\mathbf{W}|\mathbf{\Theta},\mathbf{Y},\mathbf{X}) = \frac{p(\mathbf{Y}|\mathbf{\Theta},\mathbf{X},\mathbf{W})p(\mathbf{W}|\mathbf{\Theta})}{p(\mathbf{Y}|\mathbf{\Theta},\mathbf{X})} \tag{۱۶.۷}
$$

[۱۷] Convexity



در مورد توزیع درست نمایی حاشیه ای داریم:

$$p(\mathbf{Y}|\mathbf{\Theta},\mathbf{X}) = \int p(\mathbf{Y}|\mathbf{\Theta},\mathbf{X},\mathbf{W})p(\mathbf{W}|\mathbf{\Theta})d\mathbf{W}. \tag{۱۷.۷}$$

هدف آن است که پارامتر

$$\mathbf{\Theta}^*$$

(مقدار بهینه برای $\mathbf{\Theta}$) و توزیع پسین بهینه $p(\mathbf{W}|\mathbf{\Theta}^*,\mathbf{Y},\mathbf{X})$ را بدست آوریم: می دانیم از رابطه ی ۱۷.۷ می توان مقدار بهینه $\mathbf{\Theta}^*$ را بدست آورد:

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} p(\mathbf{Y}|\mathbf{X},\mathbf{\Theta}) \tag{۱۸.۷}$$

در نتیجه توزیع بهینه نیز نتیجه می شود:

$$\Rightarrow p(\mathbf{W}|\mathbf{\Theta}^*,\mathbf{Y},\mathbf{X}) \tag{۱۹.۷}$$

طبق آنچه در قسمت قبل توضیح داده شد کران پایین برای توزیع درست نمایی حاشیه ای به صورت زیر بدست می آید:

$$p(\mathbf{Y}|\mathbf{\Theta},\mathbf{X}) \geq \exp \mathcal{L}\left(q(\mathbf{W}),\mathbf{\Theta}\right) = \exp \int q(\mathbf{W}) \log \frac{p(\mathbf{Y},\mathbf{\Theta},\mathbf{W})}{q(\mathbf{W})} d\mathbf{W}. \tag{۲۰.۷}$$

با حداکثر سازی کران پایین $\exp \mathcal{L}$ نسبت به پارامترهای $q(\mathbf{W})$ و $\mathbf{\Theta}$، به تقریبی بسیار مشابهی از درست نمایی حاشیه ای واقعی میل کرده و مدل بهینه بدست می آید. همانطور که معرفی شد، حداکثر سازی کران پایین $\exp \mathcal{L}$ معادل با حداقل سازی دیورژانس KL زیر است:

$$KL(q(\mathbf{W})||p(\mathbf{W}|\mathbf{\Theta},\mathbf{Y},\mathbf{X})) = \int q(\mathbf{W}) \frac{p(\mathbf{W}|\mathbf{\Theta},\mathbf{Y},\mathbf{X})}{q(\mathbf{W})} d\mathbf{W}.$$

همطانطور که گفته شد پاسخ بهینه برای حداقل کردن دیورژانس KL به صورت زیر است:

$$q(W_j) \propto \exp \int \left(\prod_{i \neq j}^{n} q(W_i) \right) p(\mathbf{W},\mathbf{\Theta},\mathbf{Y}|\mathbf{X}) d\mathbf{W}_{i \neq j}, \qquad 1 \geq j \geq n. \tag{۲۱.۷}$$

همانطور که قبلا نشان داده شد اگر $KL(q||p) = 0 \Rightarrow q = p$ در صورتی که از بروزرسانی ۲۱.۷ استفاده کنیم، داریم:

$$q(\mathbf{W}) = \prod_{i=1}^{n} q(W_i) \approx p(\mathbf{W}|\mathbf{\Theta},\mathbf{X},\mathbf{Y}).$$

لذا توزیعی تقریبی برای $\mathbf{W}$ بدست می آید. برای بدست آوردن $\mathbf{\Theta} = \mathbf{\Theta}^*$ به جای آنکه بهینه سازی رابطه ی ۱۸.۷ را انجام دهیم، از کران پایین آن در رابطه ی ۲۰.۷ استفاده می کنیم:

$$\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} \exp \mathcal{L}\left(q(\mathbf{W}), \mathbf{\Theta}\right) = \arg\max_{\mathbf{\Theta}} \mathcal{L}\left(q(\mathbf{W}), \mathbf{\Theta}\right)$$

بدین صورت قانون بروزرسانی $\mathbf{\Theta}^{t+1}$ به ازای مرحله ی $t+1$–ام بدست می آید. در نهایت معادلات بروزرسانی دوری به صورت زیر بدست می آیند:

$$\begin{cases} \mathbf{Theta}^{t+1} = \mathcal{L}\left(q^t(\mathbf{W}), \mathbf{\Theta}^t\right) \\ q^{t+1}(W_j) \propto \exp \int \left(\prod_{i \neq j}^{n} q(W_i)\right) p(\mathbf{W}, \mathbf{\Theta^{t+1}}, \mathbf{Y}|\mathbf{X}) d\mathbf{W}_{i \neq j}, \qquad 1 \geq j \geq n. \end{cases}$$

نشان داده می شود با قوانین بروز رسانی فوق در هر دور تقریب بهتری بدست می آید و مقدار کران پایین در هر دور الزاما بیشتر می شود [۷].

۴.۷ روش Automatic Density Filtering (ADF)

روش ADF به طور مستقل در زمینه های مختلف برای تقریب توزیع های آماری معرفی شده است؛ از جمله در [۳۹، ۳۷، ۴۹] با نام های دیگری مثل Moment matching و Weak marginalization معرفی شده است. می توان گفت ADF به نحوی تعمیم یافته فیلتر کالمن [۱۸] است [۵۵]. مطالبی که در اینجا بیان می کنیم از [۵۴، ۵۵] هستند. شکل ۴.۷ را در نظر گیرید. متغیر $\mathbf{X}$، متغیر مشاهده شده و متغیر $\mathbf{Z}$، متغیر پنهان هستند. هدف بدست آوردن $p(\mathbf{Z}|\mathbf{X})$ (توزیع پسین روی $\mathbf{Z}$)، برای انجام تخمین [۱۹] و توزیع درست نمایی حاشیه ای (توزیع داده ها $p(\mathbf{X})$)، برای انتخاب پارامترهای بهینه است. فرض کنیم توزیعی که می خواهیم از آن برای تقریب استفاده کنیم، یک توزیع از خانواده توزیع های نمایی به صورت زیر باشد [۴۴]:

$$q_\theta^{new}(\mathbf{z}) = \frac{1}{Z(\theta)} \exp\left(\theta^T \Phi(\mathbf{z})\right), \quad Z(\theta) = \int \exp\left(\theta^T \Phi(z)\right) dz$$

$\Phi(z)$ آماره طبیعیِ [۲۰] z نامیده می شود. ساده ترین نمونه این آماره ها، میانگین و واریانس در توزیع گوسی است. برای حداقل کردن برای حداقل کردن اختلاف در $q_\theta(z)$ و $p(z,x)$ داریم:

$$f(\theta) = KL(p||q) = \langle \log \frac{p}{q_\theta} \rangle_p = \langle \log(p) \rangle_p + \langle \log\left(Z(\theta)\right) \rangle_p - \langle \theta^T \Phi(\mathbf{z}) \rangle_p$$

$$\nabla_\theta f(\theta) = 0 \Rightarrow \nabla_\theta f(\theta) = \nabla_\theta \log Z(\theta) - \langle \Phi(\mathbf{z}) \rangle = 0 \qquad (۲۲.۷)$$

همچنین داریم:

$$\nabla_\theta \log Z(\theta) = \frac{\nabla_\theta Z(\theta)}{Z(\theta)} = \frac{\nabla_\theta \int \exp\left(\theta^T \Phi(\mathbf{z})\right)}{Z(\theta)} = \frac{\int \nabla_\theta \exp\left(\theta^T \Phi(\mathbf{z})\right)}{Z(\theta)} = \langle \Phi(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})} \qquad (۲۳.۷)$$

با ادغام نتایج فوق از روابط ؟؟ داریم:

$$\nabla_\theta f(\theta) = 0 \Rightarrow \langle \Phi(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})} = \langle \Phi(\mathbf{z}) \rangle_p \qquad (۲۴.۷)$$

با محاسبه ی گرادیان دوم $f(\theta)$ می توان نشان داد که پاسخ فوق یک حداقل از $f(\theta)$ را بدست می دهد:

$$[\nabla \nabla_\theta f(\theta)]_{ij} = \frac{\partial^2 \log Z(\theta)}{\partial \theta_j \partial \theta_i} = \frac{\partial}{\partial \theta_j} \frac{\int \Phi_i(\mathbf{z}) \exp\left(\theta^T \Phi(\mathbf{z})\right) d\mathbf{z}}{Z(\theta)} = \langle \Phi_i(\mathbf{z}), \Phi_j(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})} - \langle \Phi_i(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})} \cdot \langle \Phi_j(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})}$$

(۲۵.۷)

[۱۸] Kalman filter
[۱۹] Estimation
[۲۰] Natural statistic

از روابط فوق می توان نتیجه گیری کرد برای بدست آوردن بهترین تخمین از یک توزیع دلخوه با استفاده اا‌ز یک توزیع از خانواده ی نمایی با استفاده از معیار KL دیورژانس، کافیست آماره های آنها یکسان قرار گیرند. به عنوان یک حالت خاص اگر فرض کنیم $q_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ نتایج روابط [؟، ؟] به صورت زیر بدست می آیند:

$$\begin{cases} \boldsymbol{\mu}^* = \langle z \rangle_p \\ \boldsymbol{\Sigma}^* = \langle z z^T \rangle_p - \langle z \rangle_p \langle z \rangle_p^T \end{cases}$$

فرض کنیم بتوانیم توزیع مشترک را به صورت فاکتورهایی تجزیه کنیم: $p(\mathbf{X}, \mathbf{Z}) = \prod_i t_i(z)$ قاعده کلی در مورد تجزیه ی توزیع به حاصلضرب ها وجود ندارد؛ اما در کل باید به این نکته توجه کرد که بهتر است عبارات کم باشد تا میزان محاسبات لازم کمتر باشد. ضمن اینکه این را نیز باید مد نظر داشت که هر عبارت باید به اندازه ی کافی ساده باشد تا بتوان آن را با یک توزیع شناخته شده (از جمله توزیع های خانواده ی نمایی) تخمین زد. دلیل دیگر برای استفاده از توزیع های نمایی، کافی بودن تعداد محدودی از گشتاور های آنها برای شناخت کامل توزیع است. فرض کنیم $q_\theta(\mathbf{z})$ تخمین مورد نظر برای توزیع پسین باشد که دارای فرم شناخته شده ای است(مثلا گوسی). در ابتدا داریم $q_\theta(\mathbf{z}) = 1$. در ادامه، در هر مرحله فاکتور t_i را به تقریب اضافه کرده و تقریب را بهتر می کنیم. توزیع $\hat{p}(\mathbf{z})$ توزیعی کمکی است که از آن در شناسایی استفاده می کنیم:

$$\hat{p}(\mathbf{z}) = \frac{1}{\tilde{Z}(\theta)} t_i(\mathbf{z}) q_\theta^{old}(\mathbf{z}), \tilde{Z}(\theta) = \int t_i(\mathbf{z}) q_\theta^{old}(\mathbf{z}) d\mathbf{z}. \quad (۲۶.۷)$$

توزیع $q_\theta^{old}(\mathbf{z})$، توزیع پسین در تقریب مرحله ی قبل است. می توان با استفاده از $KL\left(\hat{p}(\mathbf{z}) || q_\theta^{new}(\mathbf{z})\right)$، با این فرض که $q_\theta^{new}(\mathbf{z})$ یک توزیع از خانواده توزیع های نمایی است، با صفر قراردادن مشتقات دیورژانس KL می توان گشتاورهای لازم برای شناسایی کامل $q_\theta^{new}(\mathbf{z})$ را بدست آورد. در ادامه جهت سادگی اندیس old را از $q_\theta^{old}(\mathbf{z})$ می اندازیم. با توجه به تعریف رابطه ی ۲۶.۷ داریم:

$$\nabla_\theta q_\theta(\mathbf{z}) = \nabla_\theta \frac{1}{Z(\theta)} \int \exp() \, \theta^T \Phi(z) d\mathbf{z} = \nabla_\theta \left[\frac{1}{Z(\theta)}\right] \exp\left(\theta^T \Phi(z)\right) + \frac{1}{Z(\theta)} . \nabla_\theta \exp\left(\theta^T \Phi(z)\right)$$

$$\Rightarrow \nabla_\theta q_\theta(\mathbf{z}) = -\frac{\nabla_\theta Z(\theta)}{Z(\theta)} q_\theta(z) + \Phi(z) q_\theta(z) = -\langle \Phi(z) \rangle_{q_\theta(z)} + \Phi(z) q_\theta(z).$$

با ضرب رابطه ی فوق در $\frac{1}{\tilde{Z}(\theta)}(\theta) . t_i(\mathbf{z})$ و انتگرال گیری نسبت به $\mathbf{z}$ داریم:

$$\nabla_\theta \frac{t_i(\mathbf{z} q_\theta(\mathbf{z}))}{\tilde{Z}(\theta)} = -\langle \Phi(\mathbf{z}) \rangle_{q_\theta(\mathbf{zs})} . \frac{1}{\tilde{Z}(\theta)} t_i(\mathbf{z}) q_\theta(\mathbf{z})) + \Phi(\mathbf{z}) . \frac{1}{\tilde{Z}(\theta)} t_i(\mathbf{z}) q_\theta(\mathbf{z})).$$

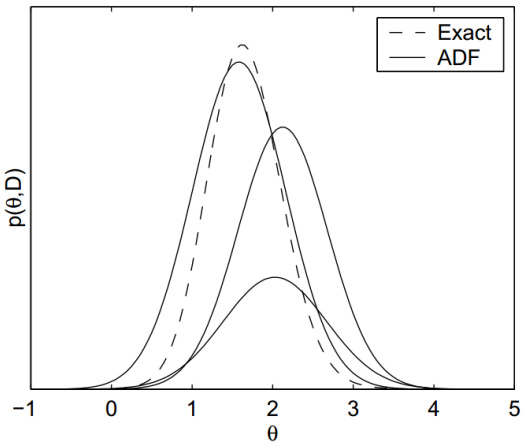
$$\Rightarrow \frac{1}{\tilde{Z}(\theta)} \nabla_\theta \nabla_\theta \tilde{Z}(\theta) = -\langle \Phi(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})} + \langle \Phi(\mathbf{z}) \rangle_{\hat{p}(\mathbf{z})}.$$

$$\Rightarrow \langle \Phi(\mathbf{z}) \rangle_{p(\mathbf{z})} = \nabla_\theta \log\left(\tilde{Z}(\theta)\right) + \langle \Phi(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})}.$$

یادآوری می شود در تمام مراحل قبل منظور از $q_\theta(\mathbf{z})$، $q_\theta^{old}(\mathbf{z})$ بوده است. برای محاسبه ی $q_\theta^{new}(\mathbf{z})$ از روی $q_\theta^{new}(\mathbf{z})$ می توان دوباره از روابط ؟؟ استفاده کرد؛ یعنی کافیست آماره های مختلف روی توزیع $KL\left(q_\theta^{new}(\mathbf{z}), \hat{p}(\mathbf{z})\right)$ و $\hat{p}(\mathbf{z})$ را با هم یکسان قرار دهیم:

$$\langle \Phi(\mathbf{z}) \rangle_{q_\theta^{new}(\mathbf{z})} = \langle \Phi(\mathbf{z}) \rangle_{\hat{p}(\mathbf{z})}.$$

$$\Rightarrow \langle \Phi(\mathbf{z}) \rangle_{q_\theta^{new}(\mathbf{z})} = \nabla_\theta \log\left(\tilde{Z}(\theta)\right) + \langle \Phi(\mathbf{z}) \rangle_{q_\theta(\mathbf{z})}.$$



شکل ۵.۷: نمایش اثر نحوه ی انتخاب فاکتور در نتیجه ی نهایی در تقریب ADF. تصویر از [۵۵].

برای اینکه معادله ی فوق به ازای یک توزیع از خانواده ی نمایی حل کنیم، لازم است بتوانیم $\nabla_\theta \log\left(\tilde{Z}(\theta)\right)$ و $\langle\Phi(\mathbf{z})\rangle_{q_\theta(\mathbf{z})}$ را به صورت به ازای آ» به صورت بسته به دست آوریم. اگر فرض کنیم $q_\theta(\mathbf{z}) = q(z;\boldsymbol{\mu},\boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma})$ و $\tilde{Z}(\theta) = \tilde{Z}(\boldsymbol{\mu},\boldsymbol{\Sigma}) = \int t(\mathbf{z}).q(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma})\,d\mathbf{z}$ با استفاده از قانون ۱.۶ از ضمیمه اول، داریم :

$$\nabla_\mu q(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\,q(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma})$$

$$\Rightarrow \mathbf{z}.q(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \boldsymbol{\mu}.q(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma}) + \boldsymbol{\Sigma}.\nabla_{\boldsymbol{\mu}} q(\mathbf{z};\boldsymbol{\mu},\boldsymbol{\Sigma})$$

با ضرب دو طرف در $\frac{1}{\tilde{Z}}t_i(\mathbf{z})$ و انتگرال گیری نسبت به $\mathbf{z}$ داریم:

$$\boldsymbol{\mu}^* = \boldsymbol{\mu} + \frac{1}{\tilde{Z}}\boldsymbol{\Sigma}.\nabla_{\boldsymbol{\mu}}\int t(\mathbf{z})q(\mathbf{z})d\mathbf{z} \qquad (۲۷.۷آ)$$

$$= \boldsymbol{\mu} + \frac{1}{\tilde{Z}}\boldsymbol{\Sigma}.\nabla_{\boldsymbol{\mu}}\tilde{Z}(\boldsymbol{\mu},\boldsymbol{\Sigma}) \qquad (۲۷.۷ب)$$

$$= \boldsymbol{\mu} + \boldsymbol{\Sigma}.\nabla_{\boldsymbol{\mu}}\log\left(\tilde{Z}(\boldsymbol{\mu},\boldsymbol{\Sigma})\right) \qquad (۲۷.۷ج)$$

$$= \boldsymbol{\mu} + \boldsymbol{\Sigma}.\mathbf{g}, \qquad \stackrel{\triangle}{=} \nabla_{\boldsymbol{\mu}}\log\left(\tilde{Z}(\boldsymbol{\mu},\boldsymbol{\Sigma})\right) \qquad (۲۷.۷د)$$

در مورد آماره ی واریانس داریم:

$$\nabla_{\boldsymbol{\Sigma}} q(\mathbf{z}) = \frac{1}{2}\left(-\boldsymbol{\Sigma}^{-1} + \boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\right)q(\mathbf{z})$$

$$\Rightarrow \mathbf{z}\mathbf{z}^T q(\mathbf{z}) = 2\boldsymbol{\Sigma}.\left[\nabla_{\boldsymbol{\Sigma}} q(\mathbf{z})\right]\boldsymbol{\Sigma} + \left(\boldsymbol{\Sigma} + \mathbf{z}\boldsymbol{\mu} + \boldsymbol{\mu}\mathbf{z}^T - \boldsymbol{\mu}\boldsymbol{\mu}^T\right)q(\mathbf{z})$$

$$\Rightarrow \langle\mathbf{z}\mathbf{z}^T\rangle = \boldsymbol{\Sigma} + 2\boldsymbol{\Sigma}\mathbf{G}\boldsymbol{\Sigma}\langle\mathbf{z}\rangle_{\hat{p}(\mathbf{z})}\boldsymbol{\mu}^T + \boldsymbol{\mu}\langle\mathbf{z}\rangle^T_{\hat{p}(\mathbf{z})} - \boldsymbol{\mu}\boldsymbol{\mu}^T, \qquad \mathbf{G} = \nabla_{\boldsymbol{\Sigma}}\log\left(\tilde{Z}(\boldsymbol{\mu},\boldsymbol{\Sigma})\right)$$

$$\Rightarrow \boldsymbol{\Sigma}^* = \langle\mathbf{z}\mathbf{z}^T\rangle_{\hat{p}(\mathbf{z})} - \langle\mathbf{z}\rangle_{\hat{p}(\mathbf{z})}\langle\mathbf{z}\rangle^T_{\hat{p}(\mathbf{z})} = \boldsymbol{\Sigma} - \boldsymbol{\Sigma}\left(\mathbf{g}\mathbf{g}^T - 2\mathbf{G}\right)\boldsymbol{\Sigma}.$$

با توجه به قوانین بروز رسانی فوق مشاهده می شود در صورتی که ترتیب اضافه کردن مجموعه فاکتورهای $\{t_i\}_i$ عوض شود، تقریب نهایی عوض خواهد شد. این مساله ی در آزمایشی در شکل ۵.۷ نمایش داده شده است.

۵.۷ روش Expectation Propagation (EP)

می توان گفت این روش به گونه ای در خانواده ی روش VB است؛ چون همانگونه که در ادامه خواهیم گفت تقریب هایی بسیار مشابه آن دارد. اما به دلیل اهمیت و کاربرد آن، جداگانه به معرفی آن می پردازیم. روش EP اولین بار در [۵۵] معرفی شد. روش EP مشابه روش ARD است. بر طبق ادعای [۵۴، ۵۵] روش EP، روش های ADF و "انتشار باور دوری" [۲۱] را ادغام می کند. در حقیقت EP تقریب های مربوط به ADF را به صورت دوری [۲۲] استفاده می کند تا اینکه در نهایت به بهترین تقریب ممکن همگرا شود.

اشکال ADF در آن است که از روی فاکتورهای توزیع هدف، تنها یک بار عبور می کند. در نهایت این نکته، باعث ایجاد این اشکال می شود که ممکن است با عبور از ترتیب های متفاوت از فاکتورها، به نتایج متفاوتی برسد(شکل ۵.۷). در حالیکه در EP آنقدر از روی فاکتورها عبور می کنیم تا اینکه مطمئن شویم به شرط همگرایی رسیده ایم. به همین علت پیچیدگی EP به اندازه ی یک ضریب ثابت از ADF بزرگتر است. در EP به جای اینکه مثل ADF تخمین KL را به $q_\theta(\mathbf{z})$ اعمال کنیم به هرکدام از فاکتورهای $t_i(\mathbf{z})$ اعمال می کنیم و سپس تخمین $p(\mathbf{z}|\mathbf{x})$ را بروز رسانی می کنیم. بدین صورت، ترتیب انتخاب فاکتورها از مجموعه ی $\{t_i\}_i$ تفاوتی در تقریب نهایی ایجاد نخواهد کرد. چون ممکن است چندین بار از روی فاکتور ها عبور کنیم. الگوریتمی که در EP استاندارد استفاده می شود در الگوریتم ۵ نشان داده شده است. نکته ی مهمی که در ضمن اجرای مراحل الگوریتم از آن بهره بده می شود، این است که حاصل ضرب دو نمایی، یک نمایی خواهد بود. برای محاسبه ی توزیع درست نمایی حاشیه ای کافیست:

الگوریتم ۵ الگوریتم Expectation Propagation

1: Initialze $\{\tilde{t}_i\}$
2: $q_\theta(\mathbf{z}) = \frac{\prod_i \tilde{t}_i}{\int \prod_i \tilde{t}_i}$
3: **repeat**
4: Choose a $\tilde{t}_i$ and $q_\theta^{\setminus i}(\mathbf{z}) \propto \frac{q_\theta(\mathbf{z})}{\tilde{t}_i(\mathbf{z})}$.
5: Use ADF to compute $\theta^* = \arg\min_\theta KL\left(q_\theta^{\setminus i}(\mathbf{z}) t_i || q_\theta(\mathbf{z})\right)$.
6: $= \tilde{t}_i(\mathbf{z}) \frac{q_{\theta^*}(\mathbf{z})}{q_\theta^{\setminus i}(\mathbf{z})}$.
7: **until** all $\tilde{t}_i$s converge

$$p(\mathbf{x}) \approx \int p(\mathbf{x}|\mathbf{z}) p(\mathbf{z}) d\mathbf{z} d\mathbf{z} = \int \prod_i \tilde{t}_i(\mathbf{z}) d\mathbf{z}.$$

۶.۷ روش Laplace Approximation

رابطه ی ؟؟؟ را می توان حول حداکثر آن تقریب زد. هرچه توزیع پسین برای فراپارامترهای دارای قله ی تیزتری باشد، این تقریب دقیق تر است. معمولا توزیع های پسین برای فراپارامترهای اینگونه اند. اما توزیع پسین حول پارامترها، معمولا شکلی پیچیده دارند [۴۸].

فرض کنیم در مدل بیزوی پارامترهای $\mathbf{\Theta}$ و مدل $\mathbf{m}$ را داریم؛ بر طبق قانون Bayes داریم:

$$p(\mathbf{\Theta}|\mathbf{y}, \mathbf{m}) = \frac{p(\mathbf{y}|\mathbf{\Theta}, \mathbf{m}) p(\mathbf{\Theta}|\mathbf{m})}{p(\mathbf{y}|\mathbf{m})}.$$

[۲۱] Loopy belief propagation
[۲۲] Iterative

لگاریتم صورت را به صورت مقابل زیر تعریف می کنیم:

$$t(\mathbf{\Theta}) = \ln\left[p(\mathbf{\Theta}|\mathbf{m})p(\mathbf{y}|\mathbf{\Theta},\mathbf{m})\right] = \ln p(\mathbf{\Theta}|\mathbf{m}) + \sum_{i=1}^{n} \ln p(y_i|\mathbf{\Theta},\mathbf{m})$$

تقریب Laplace [۳۳، ۴۷] می فرض کنیم داشته باشیم: $\mathbf{\Theta}^* = \arg\max_{\mathbf{\Theta}} p(\mathbf{\Theta}|\mathbf{y},\mathbf{m})$

اگر بسط تیلور $t(\mathbf{\Theta})$ را حول $\mathbf{\Theta} = \mathbf{\Theta}^*$ بنویسیم:

$$t(\mathbf{\Theta}) = t(\mathbf{\Theta}^*) + (\mathbf{\Theta}-\mathbf{\Theta}^*)^T \frac{\partial t(\mathbf{\Theta})}{\partial \mathbf{\Theta}}|_{\mathbf{\Theta}=\mathbf{\Theta}^*} + \frac{1}{2!}(\mathbf{\Theta}-\mathbf{\Theta}^*)^T \frac{\partial^2 t(\mathbf{\Theta})}{\partial \mathbf{\Theta}^2}|_{\mathbf{\Theta}=\mathbf{\Theta}^*}(\mathbf{\Theta}-\mathbf{\Theta}^*) + \dots$$

به علت اینکه $\mathbf{\Theta} \approx \mathbf{\Theta}^*$ نقطه ی حداکثر محلی است $\frac{\partial t(\mathbf{\Theta})}{\partial \mathbf{\Theta}}|_{\mathbf{\Theta}=\mathbf{\Theta}^*} \approx 0$. لذا تقریب، فقط مشتقدوم را در نظر می گیریم. ماتریس $\mathbf{H}(\mathbf{\Theta})$، ماتریس Hessian یا ماتریس مشتق های دوم از $t(\mathbf{\Theta})$ نسبت به $\mathbf{\Theta}$ است:

$$\mathbf{H}(\mathbf{\Theta}^*) = -\nabla\nabla \ln p(\mathbf{\Theta}|\mathbf{y},\mathbf{m})|_{\mathbf{\Theta}=\mathbf{\Theta}^*} = \frac{\partial^2 \ln p(\mathbf{\Theta}|\mathbf{y},\mathbf{m})}{\partial \mathbf{\Theta}\mathbf{\Theta}^T}|_{\mathbf{\Theta}=\mathbf{\Theta}^*} = \frac{\partial^2 t(\mathbf{\Theta})}{\partial \mathbf{\Theta}\mathbf{\Theta}^T}|_{\mathbf{\Theta}=\mathbf{\Theta}^*}.$$

$$\Rightarrow t(\mathbf{\Theta}) \approx t(\mathbf{\Theta}^*) + \frac{1}{2}(\mathbf{\Theta}-\mathbf{\Theta}^*)^T \mathbf{H}(\mathbf{\Theta}^*)(\mathbf{\Theta}-\mathbf{\Theta}^*)$$

آنگاه در مورد درست نمایی حاشیه ای داریم:

$$p(\mathbf{y}|\mathbf{m}) = \int p(\mathbf{y}|\mathbf{\Theta},\mathbf{m})p(\mathbf{\Theta}|\mathbf{m})d\mathbf{\Theta}$$

$$\Rightarrow \ln p(\mathbf{y}|\mathbf{m}) = \ln \int p(\mathbf{y}|\mathbf{\Theta},\mathbf{m})p(\mathbf{\Theta}|\mathbf{m})d\mathbf{\Theta}$$

$$\approx \ln \int \exp\left(t(\mathbf{\Theta})\right) d\mathbf{\Theta} = t(\mathbf{\Theta}^*) + \frac{1}{2}\ln\left|2\pi\mathbf{H}^{-1}\right| = \ln p(\mathbf{\Theta}^*|\mathbf{m}) + \ln p(\mathbf{y}|\mathbf{\Theta}^*,\mathbf{m}) + \frac{1}{2}\ln|2\pi| - \frac{1}{2}\ln|\mathbf{H}|$$

در نهایت تقریب Laplace مربوط به درست نمایی حاشیه ای به صورت زیر بدست می آید:

$$p(\mathbf{y}|\mathbf{m})_{\text{Laplace}} = p(\mathbf{\Theta}^*|\mathbf{m}).p(\mathbf{y}|\mathbf{\Theta}^*,\mathbf{m}).|2\pi| - \frac{1}{2}\ln\left|\mathbf{H}^{-1}\right|^{\frac{1}{2}}$$

در تقریب دو فاکتور اول مربوط به معیار MAP[۲۳] است. فاکتور آخر مربوط به شکل منحنی $t(\mathbf{\Theta})$ در همسایگی نقطه ی $\mathbf{\Theta} = \mathbf{\Theta}^*$ است.

تقریب فوق دارای اشکالاتی است؛ اول اینکه فرض می کند منحنی $t(\mathbf{\Theta})$ دارای شکلی شبیه به گوسی در همسایگی نقطه ی بهینه است؛ چنین فرضی معمولا در داده های آموزشی زیاد برقرار است. اما در بسیازی ار مواقع ممکن است $t(\mathbf{\Theta})$ دارای چندین حداکثر باشد که در اینصورت تقریب مذکور می تواند خطای بسیار زیادی در بر داشته باشد. گاهی حتی ممکن است در $\mathbf{H}$ غیرقابل محاسبه باشد. علاوه بر مساله ی دقت تقریب، پیچیدگی محاسباتی تقریب Lalace می تواند زیاد باشد. در محاسبه ی ماتریس Hessian پیچیدگی محاسباتی $\mathcal{O}(nd^2)$ به خاطر محاسبه ی مشتقات، و پیچیدگی محاسباتی $\mathcal{O}(d^3)$ برای محاسبه ی دترمینان، تحمیل می شود.

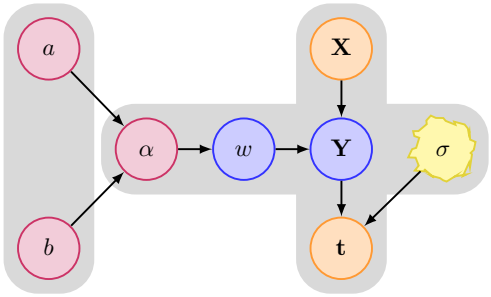
[۲۳] Maximum a postoriori

فصل ۸

ضمیمه سوم: محاسبات اضافی مربوط به الگوریتم ها

۱.۸ محاسبات اضافی مربوط به آموزش در الگوریتم RVM

شکل ۱.۸ را دوباره برای یادآوری روابط بین پارامترهای مدل می آوریم. لازم به یادآوری است که مدل معرفی شده به عنوان RVM استاندارد، بر اساس [۷۸، ۶، ۷۷، ۷۹، ۷، ۱۹] می باشد.



شکل ۱.۸: نمایش مدل بین پارامترهای آماری الگوریتم RVM

با توجه به آنچه در معرفی ساختار RVM داشتیم:

$$\begin{aligned} \mathbf{t} &= \mathbf{y} + \epsilon \\ &= \sum_{m=1}^{M} w_i K(\mathbf{x}, \mathbf{x}_m) + w_0 + \epsilon \\ &= \mathbf{\Phi}(\mathbf{x})\mathbf{w} + \epsilon \end{aligned} \tag{۱.۸}$$

در واقع با داشتن بردار مشاهدات ورودی $\mathbf{x}$ ، خروجیِ همراه با خطای $\mathbf{t}$ به دست آمده است. فرض کرده ایم بردار وزن ها به صورت $\mathbf{w} = [w_i \ ... \ w_M]^T$ است. همچنین ماتریس طراحی [۱] یا ماتریس توابع پایه [۲] است که دارای المان هایی از پیش طراحی شده به صورت $k(\mathbf{x_n}, \mathbf{x_m})$ و دارای ابعاد $N \times (M+1)$ است. از این به بعد جهت سادگی، $\mathbf{\Phi}(\mathbf{x})$ را با $\mathbf{\Phi}$ نمایش می دهیم.

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \sigma^2) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t} - \mathbf{\Phi}\mathbf{w}||^2\right\}. \tag{۲.۸}$$

[۱] Design matrix
[۲] Kernel function matrix

$$\begin{aligned} p(\mathbf{w}|\alpha) &= \prod_{i=0}^{M} \mathcal{N}(w_i|0,\alpha_i^{-1}) \\ &= (2\pi)^{-M/2} \prod_{j=0}^{M} \alpha_j^{1/2} \exp\left\{-\frac{\alpha_j||w_i||^2}{2}\right\}. \end{aligned} \tag{۳.۸}$$

بر اساس قانون Bayes داریم:

$$p(\mathbf{w}|\mathbf{t},\mathbf{x},\alpha,\sigma^{\mathbf{2}}) = \frac{p(\mathbf{w}|\mathbf{x},\alpha,\sigma^{\mathbf{2}})p(\mathbf{t}|\mathbf{w},\mathbf{x},\alpha,\sigma^{\mathbf{2}})}{p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}})}. \tag{۴.۸}$$

با توجه به ارتباط متغیرهای آماری در ساختار RVM می توان برخی از شرط های موجود در فرمول ۵.۸ را به علت استقلال آماری آنها حذف کرد:

$$p(\mathbf{w}|\mathbf{t},\mathbf{x},\alpha,\sigma^{\mathbf{2}}) = \frac{p(\mathbf{w}|\cancel{\mathbf{x}},\cancel{\alpha},\sigma^{\mathbf{2}})p(\mathbf{t}|\mathbf{w},\mathbf{x},\alpha,\cancel{\sigma^{\mathbf{2}}})}{p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}})}.$$

$$p(\mathbf{w}|\mathbf{t},\mathbf{x},\alpha,\sigma^{2}) = \frac{p(\mathbf{w}|\sigma^{\mathbf{2}})p(\mathbf{t}|\mathbf{w},\mathbf{x},\alpha)}{p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}})}. \tag{۵.۸}$$

از طرفی دیگر می توان مخرج رابطه فوق را به صورت زیر نوشت:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}}) &= \int_{\mathbf{w}\in\mathbb{R}^+} p(\mathbf{t},\mathbf{w}|\alpha,\sigma^{\mathbf{2}})d\mathbf{w} \\ &= \int_{\mathbf{w}} p(\mathbf{t}|\mathbf{w},\alpha,\sigma^{\mathbf{2}})p(\mathbf{w}|\alpha,\sigma^{\mathbf{2}})d\mathbf{w} \\ &= \int_{\mathbf{w}} p(\mathbf{t}|\mathbf{w},\cancel{\alpha},\sigma^{\mathbf{2}})p(\mathbf{w}|\alpha,\cancel{\sigma^{\mathbf{2}}})d\mathbf{w} \\ &= \int_{\mathbf{w}} p(\mathbf{t}|\mathbf{w},\sigma^{\mathbf{2}})p(\mathbf{w}|\alpha)d\mathbf{w} \end{aligned} \tag{۶.۸}$$

با توجه به روابط ۲.۸ و ۳.۸ و ۶.۸ داریم:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}}) &= \int_{\mathbf{w}} p(\mathbf{t}|\mathbf{w},\sigma^{\mathbf{2}})p(\mathbf{w}|\alpha)d\mathbf{w} \\ &= \int_{\mathbf{w}} (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi}\mathbf{w}||^2\right\} \times \\ &\qquad (2\pi)^{-M/2} \prod_{j=0}^{M} \alpha_j^{1/2} \exp\left\{-\frac{\alpha_j||w_j||^2}{2}\right\} d\mathbf{w} \\ &= (2\pi\sigma^2)^{-N/2}(2\pi)^{-M/2} \prod_{j=0}^{M} \alpha_j^{1/2} \int_{\mathbf{w}} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi}\mathbf{w}||^2 - \frac{\alpha_j||w_j||^2}{2}\right\} d\mathbf{w}. \end{aligned} \tag{۷.۸}$$

فرض کنیم داشته باشیم $\mathbf{A} = \mathrm{diag}(\alpha_i)$. در اینصورت می توان نوشت $\mathbf{w}^T\mathbf{A}\mathbf{w} = \sum_{i=0}^{M} \alpha_i||w_i||^2$. همچنین داریم: $|\mathbf{A}|^{1/2} = \prod_{j=0}^{M} \alpha_j^{1/2}$. لذا می توان رابطه ۷.۸ را به اینصورت ساده تر کرد:

$$\begin{aligned} p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}}) &= (2\pi\sigma^2)^{-N/2}(2\pi)^{-M/2}|\mathbf{A}|^{1/2} \int_{\mathbf{w}} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi}\mathbf{w}||^2 - \sum_{i=0}^{M}\frac{\alpha_i||w_i||^2}{2}\right\} d\mathbf{w} \\ &= (2\pi\sigma^2)^{-N/2}(2\pi)^{-M/2}|\mathbf{A}|^{1/2} \int_{\mathbf{w}} \exp\left\{-\frac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi}\mathbf{w}||^2 - \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w}\right\} d\mathbf{w}. \end{aligned} \tag{۸.۸}$$

در ادامه عبارت داخل نمای فرمول ۸.۸ را ساده می کنیم:

$$\begin{aligned}\frac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi w}||^2+\frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} &= \frac{1}{2\sigma^2}(\mathbf{t}-\mathbf{\Phi w})^T.(\mathbf{t}-\mathbf{\Phi w})+\frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w}\\ &= \frac{1}{2\sigma^2}\left\{\mathbf{t^T}.\mathbf{t}-\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}-\mathbf{w^T}.\mathbf{\Phi^T}.\mathbf{t}+\mathbf{w^T}.\mathbf{\Phi^T}.\mathbf{\Phi}.\mathbf{w}\right\}+\mathbf{w}^T\frac{\mathbf{A}}{2}\mathbf{w}\\ &= \frac{1}{2\sigma^2}\left\{\mathbf{t^T}.\mathbf{t}-2\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}\right\}+\mathbf{w}^T\left\{\frac{1}{2\sigma^2}\mathbf{\Phi^T}.\mathbf{\Phi}+\frac{\mathbf{A}}{2}\right\}\mathbf{w}\end{aligned} \tag{۹.۸}$$

در رابطه قبل از این استفاده شده است که ماتریس $\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}$ اسکالر است و ترانهاده آن با خودش برابر خواهدبود؛ یعنی: $\left(\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}\right)^T=\mathbf{w^T}.\mathbf{\Phi^T}.\mathbf{t}$.

با توجه به اینکه می دانیم در حالت کلی در یک توزیع گوسی n -متغیره داریم:

$$\int_{\mathbf{x}}(2\pi)^{-k/2}|\mathbf{\Sigma}|^{-1/2}\exp\left\{-\frac{1}{2}\left(\mathbf{x}-\mu\right)^T\mathbf{\Sigma}^{-1}\left(\mathbf{x}-\mu\right)\right\}d\mathbf{x}=1$$

سعی می کنیم نمای رابطه ۹.۸ را به صورت نمای یک گوسی، با میانگین و واریانس مشخص در آوریم:

$$\left(\mathbf{x}-\mu\right)^T\mathbf{\Sigma}^{-1}\left(\mathbf{x}-\mu\right)=\mathbf{x^T\Sigma^{-1}x}-\mu^\mathbf{T\Sigma^{-1}x}-\mathbf{x^T\Sigma^{-1}}\mu+\mu^\mathbf{T\Sigma^{-1}}\mu$$

با تناظر دادن نمای گوسی فوق با نمای رابطه ۹.۸ می توان واریانس توزیع گوسی مورد نظر را به صورت $\mathbf{\Sigma^{-1}}=\frac{1}{\sigma^2}\mathbf{\Phi^T}.\mathbf{\Phi}+\mathbf{A}$ تعریف کرد. البته برای اینکه توزیع گوسی مورد نظر برای اعداد حقیقی برقرار باشد لازم است ماتریس کواریانس آن positive-semidefinite و تقارنی باشد. در ادامه باید عبارات را طوری جابجا کنیم که تا کاملا به صورتی نمای گوسی با میانگین مشخص در آیند. با تناظر عبارات، باید عبارت زیر برقرار باشد:

$$\mu^\mathbf{T\Sigma^{-1}w}=\frac{1}{\sigma^2}\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}=\left(\frac{1}{\sigma^2}\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}\right)^T$$

با ساده کردن تساوی فوق می توان بدست آورد:

$$\mu=\frac{\mathbf{1}}{\sigma^\mathbf{2}}\mathbf{\Sigma^T\Phi^T t}=\frac{\mathbf{1}}{\sigma^\mathbf{2}}\mathbf{\Sigma\Phi^T t}$$

لذا اکنون می توان نمای رابطه ۹.۸ را به صورت یک نمای یک گوسی و تعدادی عبارات اضافه، که تابع $\mathbf{w}$ نیستند، تجزیه کرد:

$$\begin{aligned}\tfrac{1}{2\sigma^2}\left\{\mathbf{t^T}.\mathbf{t}-2\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}\right\}+\mathbf{w}^T\left\{\tfrac{1}{2\sigma^2}\mathbf{\Phi^T}.\mathbf{\Phi}+\tfrac{\mathbf{A}}{2}\right\}\mathbf{w} &= \tfrac{1}{2}\left\{\tfrac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\tfrac{2}{\sigma^2}\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}+\mathbf{w}^T\left\{\tfrac{1}{\sigma^2}\mathbf{\Phi^T}.\mathbf{\Phi}+\tfrac{\mathbf{A}}{2}\right\}\mathbf{w}\right\}\\ &= \tfrac{1}{2}\left\{\tfrac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\tfrac{1}{\sigma^2}\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}-\tfrac{1}{\sigma^2}\mathbf{t^T}.\mathbf{\Phi}.\mathbf{w}+\mathbf{w}^T\mathbf{\Sigma}^{-1}\mathbf{w}\right\}\\ &= \tfrac{1}{2}\left\{\tfrac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\mathbf{w^T\Sigma^{-1}}\mu-\mu^\mathbf{T\Sigma^{-1}w}+\mathbf{w}^T\mathbf{\Sigma}^{-1}\mathbf{w}\right\}\\ &= \tfrac{1}{2}\left\{\tfrac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}+\left(\mathbf{w}-\mu\right)^T\mathbf{\Sigma}^{-1}\left(\mathbf{w}-\mu\right)-\mu^\mathbf{T\Sigma^{-1}}\mu\right\}.\end{aligned} \tag{۱۰.۸}$$

با توجه به رابطه ۸.۸ و ۹.۸ و ۱۱.۸ می توان نوشت:

$$\begin{aligned}p(\mathbf{t}|\mathbf{x},\alpha,\sigma^\mathbf{2}) &= (2\pi\sigma^2)^{-N/2}(2\pi)^{-M/2}|\mathbf{A}|^{1/2}\exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\mu^\mathbf{T}.\mathbf{\Sigma^{-1}}.\mu\right)\right\}\\ &\int_{\mathbf{w}}\exp\left\{-\tfrac{1}{2}\left(\mathbf{w}-\mu\right)^T\mathbf{\Sigma}^{-1}\left(\mathbf{w}-\mu\right)\right\}d\mathbf{w}.\end{aligned} \tag{۱۱.۸}$$

با استفاده از ویژگی یکه بودن مساحت زیر نمودار یک تابع گوسی می توان انتگرال را از بین برد:

$$\begin{aligned}p(\mathbf{t}|\mathbf{x},\alpha,\sigma^\mathbf{2}) &= (2\pi\sigma^2)^{-N/2}(2\pi)^{-M/2}|\mathbf{A}|^{1/2}\exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\mu^\mathbf{T}.\mathbf{\Sigma^{-1}}.\mu\right)\right\}(2\pi)^{M/2}\left|\mathbf{\Sigma}\right|^{1/2}\\ &= (2\pi\sigma^2)^{-N/2}|\mathbf{A}|^{1/2}\exp\left\{-\frac{1}{2}\left(\frac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\mu^\mathbf{T}.\mathbf{\Sigma^{-1}}.\mu\right)\right\}|\mathbf{\Sigma}|^{1/2}.\end{aligned} \tag{۱۲.۸}$$

حال نمای رابطه ۱۲.۸ را با استفاده از تساوی ماتریسی Woodbury

باز کرده و ساده می کنیم:

$$\begin{aligned}\frac{1}{\sigma^2}\mathbf{t^T}.\mathbf{t}-\mu^{\mathbf{T}}.\mathbf{\Sigma^{-1}}.\mu &= \frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{t^T}.\mathbf{t}-\left(\frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{t^T\Phi\Sigma^T}\right)\mathbf{\Sigma^{-1}}\left(\frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{\Sigma\Phi^T t}\right)\\ &= \frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{t^T}.\mathbf{t}-\frac{\mathbf{1}}{\sigma^{\mathbf{4}}}\mathbf{t^T\Phi\Sigma\Phi^T t}\\ &= \mathbf{t^T}\left(\frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{I}-\frac{\mathbf{1}}{\sigma^{\mathbf{4}}}\mathbf{\Phi\Sigma\Phi^T}\right)\mathbf{t}\\ &= \mathbf{t^T}\left(\frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{I}-\frac{\mathbf{1}}{\sigma^{\mathbf{4}}}\mathbf{\Phi}\left(\frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{\Phi^T\Phi}+\mathbf{A}\right)^{\mathbf{-1}}\mathbf{\Phi^T}\right)\mathbf{t}\\ &= \mathbf{t^T}\left(\sigma^{\mathbf{2}}\mathbf{I}-\mathbf{\Phi A^{-1}\Phi^T}\right)^{\mathbf{-1}}\mathbf{t}.\end{aligned} \tag{۱۳.۸}$$

همچنین می توان دترمینان $|\mathbf{\Sigma}|^{1/2}$ در رابطه ۱۲.۸ را با استفاده از لِم دترمینان ها (رابطه ؟؟ از ضمیمه ؟؟) ساده تر کرد:

$$|\mathbf{\Sigma}| = |\frac{\mathbf{1}}{\sigma^{\mathbf{2}}}\mathbf{\Phi^T\Phi}+\mathbf{A}|^{-1} = |\mathbf{A}|.\left(\mathbf{I}+\mathbf{\Phi^T}\frac{\mathbf{A^{-1}}}{\sigma^{\mathbf{2}}}\mathbf{\Phi}\right)$$

لذا می توان رابطه ۱۲.۸ را به صورت ساده تر زیر نوشت:

$$p(\mathbf{t}|\mathbf{x},\alpha,\sigma^{\mathbf{2}}) = (2\pi\sigma^2)^{-N/2}\left(\mathbf{I}+\mathbf{\Phi^T}\frac{\mathbf{A^{-1}}}{\sigma^{\mathbf{2}}}\mathbf{\Phi}\right)^{-1/2}\exp\left\{-\frac{1}{2}\left(\mathbf{t^T}\left(\sigma^{\mathbf{2}}\mathbf{I}+\mathbf{\Phi A^{-1}\Phi^T}\right)^{\mathbf{-1}}\mathbf{t}\right)\right\}. \tag{۱۴.۸}$$

که یک توزیع گوسی با میانگین صفر و واریانس $\mathbf{I}+\dfrac{\mathbf{\Phi A^{-1}\Phi^T}}{\sigma^{\mathbf{2}}}$ است.

با نگاهی دوباره به رابطه ۵.۸ یادآوری می کنیم که تابحال توانسته ایم توزیع مخرج رابطه بیزی مورد نظر را حساب کنیم. همچنین توزیع اجزای صورت را طبق روابط ۲.۸ و ۳.۸ که هر دو گوسی هستند، در اختیار داریم. به این ترتیب، می توانیم توزیع خروجی را نیز که با توجه به نسبت دو گوسی، یک گوسی است را بدست آوریم:

$$\begin{aligned}p(\mathbf{w}|\mathbf{t},\mathbf{x},\alpha,\sigma^{\mathbf{2}}) &= \frac{p(\mathbf{w}|\alpha)p(\mathbf{t}|\mathbf{w},\sigma^{\mathbf{2}})}{p(\mathbf{t}|\alpha,\sigma^{\mathbf{2}})}\\ &= \frac{(2\pi)^{-M/2}|\mathbf{A}|^{1/2}\exp\left\{-\dfrac{\alpha_j||w_i||^2}{2}\right\}.(2\pi\sigma^2)^{-N/2}\exp\left\{-\dfrac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi w}||^2\right\}}{(2\pi\sigma^2)^{-N/2}\left(\mathbf{I}+\mathbf{\Phi^T}\dfrac{\mathbf{A^{-1}}}{\sigma^{\mathbf{2}}}\mathbf{\Phi}\right)^{-1/2}\exp\left\{-\dfrac{1}{2}\left(\mathbf{t^T}\left(\sigma^{\mathbf{2}}\mathbf{I}+\mathbf{\Phi A^{-1}\Phi^T}\right)^{\mathbf{-1}}\mathbf{t}\right)\right\}}\\ &= (2\pi)^{-M/2}|\mathbf{A}|^{1/2}\left(\mathbf{I}+\mathbf{\Phi^T}\tfrac{\mathbf{A^{-1}}}{\sigma^{\mathbf{2}}}\mathbf{\Phi}\right)^{1/2}\\ &+\exp\left\{-\tfrac{\alpha_j||w_i||^2}{2}-\tfrac{1}{2\sigma^2}||\mathbf{t}-\mathbf{\Phi w}||^2+\tfrac{1}{2}\left(\mathbf{t^T}\left(\sigma^{\mathbf{2}}\mathbf{I}+\mathbf{\Phi A^{-1}\Phi^T}\right)^{\mathbf{-1}}\mathbf{t}\right)\right\}\end{aligned} \tag{۱۵.۸}$$

با استفاه از رابطه ی ۱۴.۸ می توان توزیع حاشیه ای را نسبت هرکدام پارامترهای سیستم بهینه کرد:

$$\frac{\partial}{\partial\alpha_i}p(\mathbf{t}|\mathbf{x},\alpha,\sigma^2)=0\Rightarrow\alpha_i=\frac{1-\alpha_{\mathbf{i}}\Sigma_{ii}}{\mu_{ii}}. \tag{۱۶.۸}$$

تعریف می کنیم:

$$\gamma_i = 1-\alpha_i\Sigma_{ii}$$

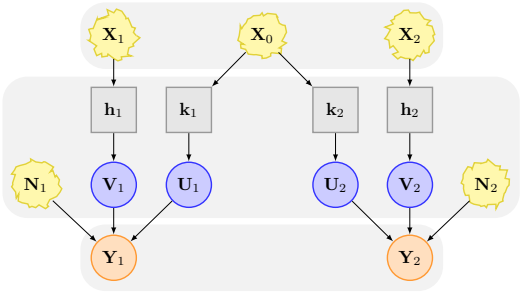
،

$$\frac{\partial}{\partial\beta}p(\mathbf{t}|\mathbf{x},\alpha,\sigma^2)=0\Rightarrow\beta=\frac{N-\Sigma_{ii}\gamma_i}{||\mathbf{t}-\mathbf{\Phi}\mu||^2}. \tag{۱۷.۸}$$

الگوریتم ۶ الگوریتم RVM

Input: Training data $\mathcal{D}$.
Output: Regression on output dimensions.

1: Choose a suitable kernel function $\mathbf{\Phi}$, convergence threshold γ_{Th}, pruning threshold α_{Th} and initial values for α and β.
2: **repeat**
3: Compute $\mu = \beta \mathbf{\Sigma} \mathbf{\Phi}^T \mathbf{t}$ and $\mathbf{\Sigma} = \left(\mathbf{A} + \beta \mathbf{\Phi}^T \mathbf{\Phi}\right)^{-1}$.
4: Compute α and β using equations 8.16 and 8.17.
5: Prune the α_i and corresponding kernel function where $\alpha_i > \alpha_{Th}$.
6: Define error convergence rate: $\gamma_i = \Sigma_{ii} \left(\alpha_i^{n+1} - \alpha_i^n\right)$.
7: **until** $\gamma < \gamma_{Th}$



شکل ۲.۸: مدل سازی دو Gaussian Process همبسته توسط فرایندهای پنهان

با بدست آوردن پارامترهای بهینه برای مدل می توان خروجی الگوریتم به ازای داده ی جدید بدست آورد:

$$p(t^*|\mathbf{x}^*, \alpha, \beta) = \mathcal{N}\left(\mu^T \mathbf{\Phi}, \sigma^2 + \mathbf{\Phi^T \Sigma \Phi}\right)$$

.

۲.۸ اثبات روابط مربوط به مدل Dependent Gaussian Process

در اینجا دوباره شکل مدل را برای یادآوری می آوریم. قسمتی از اثبات در مقاله های اصلی یعنی [۱۰، ۹، ۱۱] آورده شده است. با اینحال به علت بسیار بودن جزئیات لازم برای پیاده سازی، مراحل اثبات رو دوباره در اینجا می آوریم. در قسمت معرفی مدل، مطابق با شکل ۲.۸ تنها برای دو خروجی نمایش داده شد. اکنون در اینجا مدل را برای P ورودی و N خروجی بدست می آوریم. مدل کلی در شکل ۳.۸ نمایش داده شده است.

$$\begin{aligned} U_n(\mathbf{s}) &= \sum_{m=1}^{M} h_{mn}(\mathbf{s}) \star X_m(\mathbf{s}) \\ &= \sum_{m=1}^{M} \int_{\Re^P} h_{mn}(\alpha) X_m(\mathbf{s} - \alpha) d\alpha \end{aligned} \tag{۱۸۸}$$

با توجه به آنچه که از مدل معرفی شد فرض شده است، مجموعه $\mathcal{S}$ ورودی، و مجموعه $\mathcal{Y}$ خروجی های سیستم هستند. به ازای هر خروجی، مجموعه ای از داده های آموزشی به صورت $\mathcal{D}_1 = \{\mathbf{s}_{1,i}, y_{1,i}\}_{i=1}^{L_1}$ الی $\mathcal{D}_N = \{\mathbf{s}_{N,i}, y_{N,i}\}_{i=1}^{L_N}$ در نظر می گیریم. لذا مجموعه $\mathcal{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$ را به عنوان داده های آموزشی سیستم در نظر می گیریم.
با توجه به شکل، مجموعه $\{X_i(\mathbf{s})\}_{i=1}^{M}$ فرایندهای تصادفی مستقل، ایستا بصورت نویز سفید گوسی هستند. با توجه

به شکل مدل داریم:

$$Y_i(\mathbf{s}) = U_i(\mathbf{s}) + V_i(\mathbf{s}) + N_i(\mathbf{s})$$

اگر توان نویز سفید i –ام یعنی $N(\mathbf{s})_i$ برابر با σ_i^2 باشد، به ازای دو ورودی دلخواه $\mathbf{s}_a$ و $\mathbf{s}_b$ برای همبستگی بین خروجی ها داریم:

$$C_{ij}^Y(\mathbf{s}_a, \mathbf{s}_b) = C_{ij}^U(\mathbf{s}_a, \mathbf{s}_b) + \sigma^2 \delta_{ij}\delta_{ab}$$

نشان می دهیم $U_i(\mathbf{s})$ فرایندهایی تصادفی با میانگین صفر هستند:

$$\begin{aligned} \mathbb{E}U_i(\mathbf{s}) &= \mathbb{E}\left\{\sum_{m=1}^{M}\int_{\Re^P} h_{mi}(\alpha)X_m(\mathbf{s}-\alpha)d\alpha\right\} \\ &= \sum_{m=1}^{M}\int_{\Re^P} h_{mi}(\alpha)\mathbb{E}\left\{X_m(\mathbf{s}-\alpha)\right\}d\alpha \\ &= 0 \end{aligned} \tag{۱۹.۸}$$

و

$$\begin{aligned} \mathbb{E}\left\{U_i(\mathbf{s}_a)U_j(\mathbf{s}_b)\right\} &= \mathbb{E}\left\{\sum_{m=1}^{M}\int_{\Re^P} h_{mi}(\alpha)X_m(\mathbf{s}_a-\alpha)d\alpha \sum_{n=1}^{M}\int_{\Re^P} h_{nj}(\beta)X_n(\mathbf{s}_b-\beta)d\beta\right\} \\ &= \sum_{m=1}^{M}\sum_{n=1}^{M}\int_{\Re^P}\int_{\Re^P} h_{mi}(\alpha)h_{nj}(\beta)\mathbb{E}\left\{X_m(\mathbf{s}_a-\alpha)X_n(\mathbf{s}_b-\beta)\right\}d\alpha d\beta \end{aligned} \tag{۲۰.۸}$$

می دانیم $X_m(\mathbf{s}_a-\alpha)$ و $X_n(\mathbf{s}_b-\beta)$ دو فرآیند نویز سفید گوسی مستقل و ایستا هستند. لذا تنها زمان دارای همبستگی هستند که:

۱. اولا $m=n$ باشد (با توجه به مستقل بودن منابع نویز).

۲. ورودی دو منبع نویز، یکسان باشد. یعنی $\mathbf{s}_a-\alpha = \mathbf{s}_b-\beta$.

لذا می توان رابطه ۲۰.۸ را به اینصورت ساده تر کرد:

$$\begin{aligned} \mathbb{E}\left\{U_i(\mathbf{s}_a)U_j(\mathbf{s}_b)\right\} &= \sum_{m=1}^{M}\int_{\Re^P}\int_{\Re^P} h_{mi}(\alpha)h_{mj}(\beta)\delta\left(\alpha-(\beta+\mathbf{s}_a-\mathbf{s}_b)\right)d\alpha d\beta \\ &= \sum_{m=1}^{M}\int_{\Re^P} h_{mi}(\beta+\mathbf{s_a}-\mathbf{s_b})h_{mj}(\beta)d\beta \end{aligned} \tag{۲۱.۸}$$

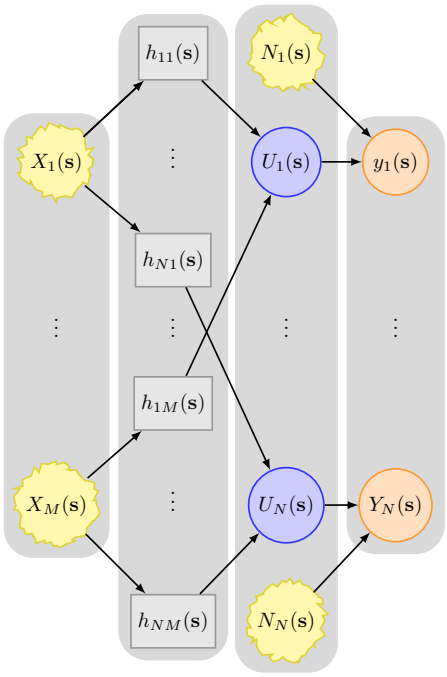
لذا دیده می شود همبستگی محاسبه شده تنها به اختلاف دو ورودی بستگی دارد. در اینجا توابع پایه را توابعی گوسی در نظر گرفته و محاسبه را ادامه می دهیم:

$$h_{mn}(\mathbf{s}) = v_{mn}\exp\left(-\frac{1}{2}(\mathbf{s}-\mu_{\mathbf{mn}})^{\mathbf{T}}\mathbf{A_{mn}}(\mathbf{s}-\mu_{\mathbf{mn}})\right) \tag{۲۲.۸}$$

که در آن $v_{mn}\in\Re$ و $\mu_{mn}\in\Re^P$ و $\mathbf{A}$ یک ماتریس در ابعاد $P\times P$ است. در اینصورت داریم:

$$h_{mn}(\mathbf{s}) = v_{mn}(2\pi)^{P/2}|\mathbf{A_{mn}}|^{-1/2}\mathcal{N}\left(\mathbf{s}|\mu_{\mathbf{mn}}, \mathbf{A_{mn}^{-1}}\right)$$

$$\begin{aligned} h_{mi}(\beta+\mathbf{s_a}-\mathbf{s_b})h_{mj}(\beta) &= v_{mi}v_{mj}(2\pi)^P|\mathbf{A_{mi}}|^{-1/2}|\mathbf{A_{mn}}|^{-1/2}\mathcal{N}\left(\beta+(\mathbf{s_a}-\mathbf{s_b})|\mu_{\mathbf{mi}}, \mathbf{A_{mi}^{-1}}\right)\mathcal{N}\left(\beta|\mu_{\mathbf{mj}}, \mathbf{A_{mj}^{-1}}\right) \\ &= v_{mi}v_{mj}(2\pi)^P|\mathbf{A_{mi}}|^{-1/2}|\mathbf{A_{mn}}|^{-1/2}\mathcal{N}\left(\beta|\mu_{\mathbf{mi}}-(\mathbf{s_a}-\mathbf{s_b}), \mathbf{A_{mi}^{-1}}\right)\mathcal{N}\left(\beta|\mu_{\mathbf{mj}}, \mathbf{A_{mj}^{-1}}\right) \end{aligned} \tag{۲۳.۸}$$



شکل ۳.۸: مدل سازی مجموعه ای از Gaussian Process های همبسته توسط مجموعه ای از فرایندهای پنهان

با توجه به رابطه ؟؟؟ از ضمیمه ؟؟ داریم:

$$h_{mi}(\beta+\mathbf{s_a}-\mathbf{s_b})h_{mj}(\beta)=v_{mi}v_{mj}(2\pi)^P|\mathbf{A_{mi}}|^{-1/2}|\mathbf{A_{mn}}|^{-1/2}\mathcal{N}\left(\mu_{\mathbf{mj}}|\mu_{\mathbf{mi}}-(\mathbf{s_a}-\mathbf{s_b}),\mathbf{A_{mi}^{-1}}+\mathbf{A_{mj}^{-1}}\right)\mathcal{N}\left(\beta|\mu,\mathbf{A}^{-}\right.$$

که در آن mu و $\mathbf{A}$ مقادیر ثابتی بر حسب μ_1، μ_2، $\mathbf{\Sigma}_1$ و $\mathbf{\Sigma}_2$ هستند. با بازگشت به رابطه انتگرالی در فرمول ؟؟ و اینکه مساحت زیر نمودار هر توزیع احتمالی، برابر واحد است می توان نوشت:

$$\begin{aligned}\mathbb{E}\left\{U_i(\mathbf{s}_a)U_j(\mathbf{s}_b)\right\} &= \sum_{m=1}^{M}\int_{\Re^P} h_{mi}(\beta+\mathbf{s_a}-\mathbf{s_b})h_{mj}(\beta)d\beta\\ &= \sum_{m=1}^{M}\int_{\Re^P} v_{mi}v_{mj}(2\pi)^P|\mathbf{A_{mi}}|^{-1/2}|\mathbf{A_{mj}}|^{-1/2}\mathcal{N}\left(\mu_{\mathbf{mj}}|\mu_{\mathbf{mi}}-(\mathbf{s_a}-\mathbf{s_b}),\mathbf{A_{mi}^{-1}}+\mathbf{A_{mj}^{-1}}\right)\mathcal{N}\left(\beta|\mu,\right.\\ &= \sum_{m=1}^{M} v_{mi}v_{mj}(2\pi)^P|\mathbf{A_{mi}}|^{-1/2}|\mathbf{A_{mj}}|^{-1/2}\mathcal{N}\left(\mu_{\mathbf{mj}}|\mu_{\mathbf{mi}}-(\mathbf{s_a}-\mathbf{s_b}),\mathbf{A_{mi}^{-1}}+\mathbf{A_{mj}^{-1}}\right)\\ &= \sum_{m=1}^{M}\frac{v_{mi}v_{mj}(2\pi)^{P/2}|\mathbf{A_{mi}}|^{-1/2}|\mathbf{A_{mj}}|^{-1/2}}{|\mathbf{A_{mi}}+\mathbf{A_{mj}}|^{-1/2}}\exp\left(-\frac{1}{2}\mu_{\mathbf{0}}^{\mathbf{T}}\mathbf{\Sigma_0^{-1}}\mu_{\mathbf{0}}\right)\end{aligned} \tag{۲۴.۸}$$

که در رابطه فوق $\mathbf{\Sigma}_0=\left(A_{mi}^{-1}+A_{mj}^{-1}\right)$ و $\mu_0=\mu_{mj}-\mu_{mi}+(\mathbf{s}_a-\mathbf{s}_b)$.

با تعریف $\mathbf{d}_{ab}=\mathbf{s}_a-\mathbf{s}_b$ به ازای دو ورودی دلخواه داریم:

$$\begin{aligned}C_{ii}^{Y}(\mathbf{d}_{ab}) &= C_{ii}^{U}(\mathbf{d}_{ab})+\delta_{ab}\sigma_i^2\\ C_{ij}^{Y}(\mathbf{d}_{ab}) &= C_{ij}^{U}(\mathbf{d}_{ab})\end{aligned} \tag{۲۵.۸}$$

کواریانس بین خروجی i -ام و خروجی j -ام به ازای داده های آموزشی D_i و D_j. عبارت است از:

$$\mathbf{C}_{ij}^{Y}=\begin{bmatrix}C_{ij}^{Y}(\mathbf{s}_{i,1}-\mathbf{s}_{j,1}) & \cdots & C_{ij}^{Y}(\mathbf{s}_{i,1}-\mathbf{s}_{j,L_j})\\ \vdots & \ddots & \vdots\\ C_{ij}^{Y}(\mathbf{s}_{i,L_i}-\mathbf{s}_{j,1}) & \cdots & C_{ij}^{Y}(\mathbf{s}_{i,L_i}-\mathbf{s}_{j,L_j})\end{bmatrix}$$

لذا می توان ماتریس کواریانس را برای خروجی های مدل به صورت زیر ساخت:

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_{11}^{Y} & \cdots & \mathbf{C}_{1N}^{Y} \\ \vdots & \ddots & \vdots \\ \mathbf{C}_{N1}^{Y} & \cdots & \mathbf{C}_{NN}^{Y} \end{bmatrix}$$

با توجه به تعریف فوق می دانیم $\mathbf{C}_{ij}$ دارای ابعاد $L_i \times L_j$ و ماتریس $\mathbf{C}$ دارای ابعاد $\left(\sum_{i=1}^{N} L_i\right) \times \left(\sum_{i=1}^{N} L_i\right)$ است.

می توان فرض کرد توزیع خروجی یک GP با میانگین صفر و واریانس $\mathbf{C}$ باشد:

$$p\left(\mathbf{y}|\Theta, \mathcal{S}\right) = \mathcal{GP}\left(0, \mathbf{C}\right) \Rightarrow \mathcal{L} = -\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{y}^{T}\mathbf{C}^{-1}\mathbf{y} - \frac{1}{2}\sum_{i=1}^{N} N_i \log 2\pi$$

که Θ پارامترهای مجهول مدل می باشند. حال می خواهیم توزیع خروخی را به ازای داده های تست بدست آوریم. فرض کنیم بخواهیم خروجی y_i را در به ازای داده های ورودی $\mathbf{s}_i^*$ به طول N_i^* بدست آوریم. در اینصورت ماتریس کواریانس به اینصورت خواهد بود:

$$\mathbf{C}^{predictive} = \begin{bmatrix} \mathbf{C} & \mathbf{C}_*^i \\ \left(\mathbf{C}_*^i\right)^T & \mathbf{C}_{**} \end{bmatrix}$$

لذا مشابه تعاریف GP می توان نوشت:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{y}_i^* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}| \begin{bmatrix} \mathbf{C} & \mathbf{C}_*^i \\ \left(\mathbf{C}_*^i\right)^T & \mathbf{C}_{**} \end{bmatrix}\right)$$

که در تعریف فوق داریم:

$$\left(\mathbf{C}_*^i\right)^T = \left[\mathbf{C}_{i^*1}^{Y}, \mathbf{C}_{i^*2}^{Y}, \ldots, \mathbf{C}_{i^*N}^{Y}\right] \quad \mathbf{C}_{i^*j}^{Y} = \begin{bmatrix} C_{ij}^{Y}(\mathbf{s}_{i,1}^* - \mathbf{s}_{j,1}) & \cdots & C_{ij}^{Y}(\mathbf{s}_{i,1}^* - \mathbf{s}_{j,L_j}) \\ \vdots & \ddots & \vdots \\ C_{ij}^{Y}(\mathbf{s}_{i,L_i^*}^* - \mathbf{s}_{j,1}) & \cdots & C_{ij}^{Y}(\mathbf{s}_{i,L_i^*}^* - \mathbf{s}_{j,L_j}) \end{bmatrix}$$

با استفاده از رابطه ؟؟ از ضمیمه ؟؟ می توان نوشت:

$$\mathbf{y}|\mathbf{y}_i^* \sim \mathcal{N}\left(\mu_p, \mathbf{\Sigma}_p\right)$$

$$\mu_p = \mathbf{C}_*^i\mathbf{C}^{-1}\mathbf{y} \quad \mathbf{\Sigma}_p = \left(\mathbf{C}_{**}\right)^T\mathbf{C}^{-1}\left(\mathbf{C}_*^i\right)^T$$

با توجه به تعاریف فوق می دانیم ماتریس $\mathbf{C}^{predictive}$ دارای ابعاد $\left(N_i^* + \sum_{l=1}^{N} L_l\right) \times \left(N_i^* + \sum_{l=1}^{N} L_l\right)$ است.

۳.۸ اثبات روابط مربوط به تعمیم RVM به چند بعد در [۷۵، ۷۶، ۷۴]

این روش در بخش ۲.۲.۳ معرفی شد. همانطور که گفته شد، فرض می کنیم داده های ورودی $\mathbf{x} \in \mathbb{R}^Q$ (Q : بعد ورودی) و داده های خروجی $\mathbf{y} \in \mathbb{R}^M$ (M : بعد خروجی) و تعداد خروجی ها، K باشند. برای آموزش الگوریتم، داده های آموزشی $\mathcal{D} = \left\{\mathbf{x}_i, \left\{\mathbf{y}_i^j\right\}_{j=1}^{K}\right\}_{i=1}^{N}$. در واقع فرض شده است تعداد N داده ی آموزشی در اختیار ماست. مشابه مدل اولیه RVM (معرفی شده در بخش ۳.۲) ، مدل زیر برای تقریب خروجی انتخاب می شود:

$$\mathbf{y^k} = \mathbf{W}^k\mathbf{\Phi} + \epsilon^k, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{S}^k), \quad 1 \leq k \leq K$$

در رابطه فوق $\mathbf{S}^k = \text{diag}\left[(\sigma_1^k)^2 \ldots (\sigma_M^k)^2\right]$ ماتریس نویز، و $\mathbf{\Phi} = [1, \phi(\mathbf{x}, \mathbf{x}_1), \phi(\mathbf{x}, \mathbf{x}_2), ..., \phi(\mathbf{x}, \mathbf{x}_N)]^T$ ماتریس توابع پایه(ماتریس طراحی [۳])، $\mathbf{W}^k \in \mathbb{R}^{M\times(N+1)}$ ماتریس وزن های توابع پایه. در حقیقت در مدل فوق،

[۳] Design matrix

K رابطه مستقل برای K خروجی در نظر گرفته ایم. مشابه الگوریتم استاندارد RVM برای ضرایب وزن، توزیعی گوسی با میانگین صفر در نظر گرفت:

$$w_{rj}^k \sim \mathcal{N}\left(0, \alpha_j^{k^{-2}}\right), \qquad 1 \leq r \leq M, \quad 1 \leq r \leq N+1.$$

فرض می کنیم: $\mathbf{A} = \text{diag}\left[\alpha_1^{-2}....\alpha_{N+1}^{-2}\right]$. لذا داریم:

$$\begin{aligned} p(\mathbf{W}^k|\mathbf{A}^k) &= \prod_{r=1}^{M}\prod_{j=1}^{N+1}\mathcal{N}\left(w_{rj}^k|0,\alpha_j^{k^{-2}}\right) \\ &= \prod_{r=1}^{M}\mathcal{N}\left(\mathbf{W}^k(r,:)|0,\mathbf{A}^k\right) \end{aligned} \tag{۲۶.۸}$$

$$\begin{aligned} p(\mathbf{y}^k|\mathbf{W}^k,\mathbf{S}^k) &= \prod_{r=1}^{M} p(y_r^k|\mathbf{W}^k,\mathbf{S}^k) \\ &= \prod_{r=1}^{M}\mathcal{N}(y_r^k|\mathbf{W}^k(r,:)\mathbf{\Phi},\sigma_r^2\mathbf{I}) \end{aligned} \tag{۲۷.۸}$$

می توان قانون Bayes را به صورت زیر نوشت که در آن به دلیل استقلال برخی از متغیرها، از شروط احتمال حذف می شوند:

$$p(\mathbf{W}^k|\mathbf{y}^k,\mathbf{S}^k,\mathbf{A}^k) = \frac{p(\mathbf{y}^k|\mathbf{W}^k,\mathbf{S}^k,\not{\mathbf{A}}^k).p(\mathbf{W}^k|\not{\mathbf{S}}^k,\mathbf{A}^k)}{p(\mathbf{y}^k|\mathbf{S}^k,\mathbf{A}^k)}.$$

بعد از ساده سازی قانون به صورت زیر به دست می آید:

$$p(\mathbf{W}^k|\mathbf{y}^k,\mathbf{S}^k,\mathbf{A}^k) = \frac{p(\mathbf{y}^k|\mathbf{W}^k,\mathbf{S}^k).p(\mathbf{W}^k|\mathbf{A}^k)}{p(\mathbf{y}^k|\mathbf{S}^k,\mathbf{A}^k)}. \tag{۲۸.۸}$$

می توان مخرج قانون فوق را به صورت زیر نوشت :

$$p(\mathbf{y}^k|\mathbf{S}^k,\mathbf{A}^k) = \int_{\mathbf{W}^k} p(\mathbf{y}^k|\mathbf{S}^k,\mathbf{W}^k)p(\mathbf{W}^k|\mathbf{A}^k)d\mathbf{W}^k$$

همچنین می توان توزیع پسین را به صورت زیر بدست آوردن:

$$p(\mathbf{W}^k|\mathbf{y}^k,\mathbf{S}^k,\mathbf{A}^k) \propto \prod_{r=1}^{M}\mathcal{N}(\mathbf{w}_r|\mu_r^k,\mathbf{\Sigma}_r^k).$$

به طوریکه :

$$\mu_r = \sigma_r^{-2}\mathbf{\Sigma}_r\mathbf{\Phi}^T\tau_r \quad \mathbf{\Sigma}_r = \left(\sigma^{-2}\mathbf{\Phi}^T\mathbf{\Phi}+\mathbf{A}\right)^{-1}. \tag{۲۹.۸}$$

با داشتن توزیع پیشین روی $\mathbf{W}$ (رابطه ی ۲۶.۸) می توان مقدار بهینه برای ماتریس وزن ها را با استفاده از روش بیشینه درست نمایی روی فراپارامترها بدست آورد. با استفاده از گوسی بودن توریع در صورت رابطه Bayes (رابطه ؟؟) به راحتی می توان از آن انتگرال گرفت و عبارت ۲۸.۸ را ساده کرد:

$$\begin{aligned} p(\mathbf{y}^k|\mathbf{S}^k,\mathbf{A}^k) &= \int_{\mathbf{W}^k} p(\mathbf{y}^k|\mathbf{S}^k,\mathbf{W}^k)p(\mathbf{W}^k|\mathbf{A}^k)d\mathbf{W}^k \\ &= \prod_{r=1}^{M}\int_{\mathbf{W}^k}\mathcal{N}(y_r|\mathbf{\Phi}\mathbf{W}^k(r,:),\sigma_r^2\mathbf{I}).\mathcal{N}\left(\mathbf{W}^k(r,:)|0,\mathbf{A}^k\right)d\mathbf{W}^k \\ &= \prod_{r=1}^{M}|\mathbf{H}_r|^{-1/2}\exp\left(-\frac{1}{2}y_r^T\mathbf{H}_r^{-1}y_r\right). \end{aligned} \tag{۳۰.۸}$$

در رابطه ی فوق حاصل به صورت حاصلضرب مجموعه ای از توزیع های گوسی چند متغیره در آمده است که در آن $\mathbf{H}_r = \sigma^2\mathbf{I}+\mathbf{\Phi}\mathbf{A}\mathbf{\Phi}^T$. حال مجهولات مساله عبارتند از $\left\{\mathbf{W}^k,\mathbf{S}^k,\mathbf{A}^k\right\}_{k=1}^K$. باید با انجام درست نمایی حداکثر

روی پارامترها و فراپارامترهای مدل، مقادیر بهینه ر ا برای آنها بدست بیاوریم. همانطور که در معرفی RVM استاندارد توضیح داده شد، به طور کلی دو روش برای بهینه سازی فراپارامترهای مدل وجود دارد:

- اولی که در [۷۸] معرفی شد ساختاری از بالا به پایین دارد؛ یعنی در ابتدا مقداری اولیه برای تمام پارامترها ($\mathbf{A}, \mathbf{W}, \mathbf{S}$) در نظر می گیرید و به تدریج مقادیر را طوری به روز رسانی می کنه که به تقریبی دلخواه برسد. همچنین توابع پایه ای که در تقریب (رگرسیون یا کلاس بندی) تقشی ندارند را حذف می کند.

- در دومی، که ساختاری از پایین به بالا دارد؛ یعنی به تدریج عناصر و مقادیر پارامترهای $\mathbf{A}, \mathbf{W}, \mathbf{S}$ را به مدل اضافه می کند. همانطور که اشاره شد در مقالات مربوط به این ساختار نشان داده شده است که مدل دوم (پایین به بالا) الگوریتم سریع تر از مدل بالا به پایین به عملکرد مطلوب می رسد [۷۹].

برای سادگی میتوان با لگاریتم رابطه ۳۰.۸ کار کرد:

$$\mathcal{L}(\alpha) = -\Sigma_{r=1}^{M} \left[\log |\mathbf{H}_r| - \tau_r^T \mathbf{H}_r^{-1} \tau_r\right] \tag{۳۱.۸}$$

ابتدا ماتریس $\mathbf{H}$ را ساده تر می کنیم. فرض کنیم ϕ_j، j –امین تابع پایه در ماتریس طراحی $\mathbf{\Phi}$ باشد:

$$\begin{aligned} \mathbf{H}_r &= \sigma_r \mathbf{I} + \Sigma_{j \neq i} \alpha_j^{-1} \phi_j \phi_j^T + \alpha_i^{-1} \phi_i \phi_i^T \\ &= \mathbf{H}_{r(-1)} + \alpha_i^{-1} \phi_i \phi_i^T \end{aligned} \tag{۳۲.۸}$$

که در رابطه فوق ماتریس $\mathbf{H}_{r(-i)}$ ماتریس طراحی بدست آمده بعد از حذف اثر تابع پایه i –ام از $\mathbf{H}_r$ است. با استفاده از روابطه ۱.۱.۶ و ۱.۱.۶ در ضمیمه ۱.۱.۶ می توان نوشت:

$$|\mathbf{H}_r| = \left|\mathbf{H}_{r(-i)}\right| \left|1 + \alpha_i^{-1} \phi_i^T \mathbf{H}_{r(-i)}^{-1} \phi_i \mathbf{H}_r\right| \tag{۳۳.۸آ}$$

$$\mathbf{H}_r^{-1} = \mathbf{H}_{r(-i)}^{-1} - \frac{\mathbf{H}_{r(-i)}^{-1} \phi \phi^T \mathbf{H}_{r(-i)}^{-1}}{\alpha_i + \phi_i^T \mathbf{H}_{r(-i)}^{-1} \phi}. \tag{۳۳.۸ب}$$

با قرار دادن روابط ۳۳.۸آ و ۳۳.۸ب در رابطه ۳۱.۸ می توان نوشت:

$$\begin{aligned} \mathcal{L}(\alpha) &= -\Sigma_{r=1}^{M} \left[\log |\mathbf{H}_r| - \tau_r^T \mathbf{H}_r^{-1} \tau_r\right] \\ &= -\Sigma_{r=1}^{M} \left[\log \left|\mathbf{H}_{r(-i)}\right| - \tau_r^T \mathbf{H}_{r(-i)}^{-1} \tau_r\right] \\ &= \Sigma_{r=1}^{M} \left\{\log \alpha_i + \log\left(\alpha_i + \phi_i^T \mathbf{H}_{r(-i)}^{-1} \phi\right) - \frac{(\phi^T \mathbf{H}_{r(-i)}^{-1} \tau_r)^2}{\alpha_i + \phi_i^T \mathbf{H}_{r(-i)}^{-1} \phi}\right\} \\ &= \mathcal{L}(\alpha_{-\mathbf{i}}) + \Sigma_{r=1}^{M} \left\{\log \alpha_i - \log(\alpha_i + s_{ri}) + \frac{q_{ri}^2}{\alpha_i + s_{ri}}\right\} \\ &= \mathcal{L}(\alpha_{-i}) + l(\alpha_i) \end{aligned} \tag{۳۴.۸}$$

$$l(\alpha_i) = \Sigma_{r=1}^{M} \left\{\log \alpha_i - \log(\alpha_i + s_{ri}) + \frac{q_{ri}^2}{\alpha_i + s_{ri}}\right\} \tag{۳۵.۸}$$

در رابطه فوق $s_{ri} = \phi_i^T \mathbf{H}_{r(-i)}^{-1} \phi$ و $q_{ri} = \phi^T \mathbf{H}_{r(-i)}^{-1} \tau_r$. همچنین α_{-i} وکتور فراپارامترهای $[\alpha_1, \ldots, \alpha_M]$ است که α_i از آن حذف شده است. می خواهیم $\mathcal{L}(\alpha)$ را نسبت به پارامتر α_i مشتق بگیریم:

$$\frac{\partial \mathcal{L}(\alpha)}{\partial \alpha_i} = \Sigma_{r=1}^{M} \left\{\frac{1}{\alpha_i} - \frac{1}{\alpha_i + s_{ri}} - \frac{q_{ri}^2}{(\alpha_i + s_{ri})^2}\right\} = 0$$

می توان نشان داد که :

$$s_{ri} = \frac{\alpha_i S_{ri}}{\alpha_i - S_{ri}} \quad q_{ri} = \frac{\alpha_i Q_{ri}}{\alpha_i - S_{ri}} \tag{۳۶.۸}$$

که در رابطه فوق:

$$S_{ri} = \phi_i^T \mathbf{H}_r^{-1} \phi_i = \sigma_r^{-2} \phi_i^T \phi_i - \sigma_r^{-4} \phi_i^T \mathbf{\Phi} \mathbf{\Sigma}_r \mathbf{\Phi}^T \phi_i, \tag{آ۳۷.۸}$$

$$Q_{ri} = \phi_i^T \mathbf{H}_r^{-1} \tau_t = \sigma_r^{-2} \phi_i^T \tau_r - \sigma_r^{-4} \phi_i^T \mathbf{\Phi} \mathbf{\Sigma}_r \mathbf{\Phi}^T \tau_t. \tag{ب۳۷.۸}$$

با استفاده از رابطه فوق می توان عبارتی برای بروزرسانی α_i بدست آورد. الگوریتم MVRVM بدون هیچ تابع پایه ای شروع به کار می کند و به تدریج توابع پایه را اضافه می کند. لذا در اینجا باید به دنبال این نیز باشیم که در هر مرحله کدام تابع پایه ای را اضافه کنیم که بیشتر موجب افزایش میزان لگاریتم درست نمایی شود. با مراجعه به رابطه ۸ .۳۴ دیده می شود بهترین گزینه آن مورد است که برای $l(\alpha_i)$ مقدار کمتری به دست دهد:

$$\alpha_i^{opt} = \arg\min_{\alpha_i} l(\alpha_i)$$

بر طبق استدلالاتی که در [۷۹] انجام شده است، می توان با توجه به حالت های زیر، هرکدام از توابع پایه را به مدل اضافه کرد یا آن را بیرون انداخت. فرض کنیم α_m^{old} مقدار قبل از بروز رسانی و مقدار α_m^{opt} مقدار آن بعد از بروز رسانی باشد:

- اگر $\alpha_m^{old} = \infty$ و $\alpha_m^{opt} < \infty$. لذا ϕ_m انتخاب نشده است؛ اکنون در این مرحله آن را انتخاب می کنیم.
- اگر $\alpha_m^{old} < \infty$ و $\alpha_m^{opt} = \infty$. لذا ϕ_m قبلا در مدل بوده است اما آن را بیرون می اندازیم.
- اگر $\alpha_m^{old} < \infty$ و $\alpha_m^{opt} < \infty$. لذا ϕ_m قبلا انتخاب شده است و آن را همچنان در مدل نگه می داریم.

نشان داده شده است که می توان مقادیر بروزرسانی سایر پارامتر ها، یعنی σ_r^2, γ_i را به صورت های زیر با مشتق گیری از $\mathcal{L}$ نسبت به آن متغیر بدست آورد:

$$\sigma_r^2 = \frac{||\tau_r - \mathbf{\Phi}\mu_r||^2}{M - \Sigma_i^M \gamma_i}, \; r = 1, \ldots, P \tag{۳۸.۸}$$

$$\gamma_i = P - \alpha_i \Sigma_{r=1}^P \tag{۳۹.۸}$$

که در رابطه فوق $(\Sigma_r)_{ii}$، i –امین عضو قطری از $\mathbf{\Sigma_r}$ است. در نهایت کافیست مراحل اجرای MVRVM در الگوریتم ۷ خلاصه شده است.

الگوریتم ۷ الگوریتم MV-RVM

Input: Training data $\mathcal{D}$.
Output: Regression(Classification) on output dimensions.

1: σ_r = variance of τ_r and $\alpha = \infty$
2: **repeat**
3: Compute $\{\mu_r, \mathbf{\Sigma}_r\}_{r=1}^{M}$ using equations 3. 1 and 3. 1.
4: Compute $\{s_{ri}, q_{ri}\}_{r=1,q=1}^{M,M+1}$ using equations 8.36 and 8. 37.
5: Add theoptimal kernel ϕ_m to the set of optimal kernels, the which most minimizes the log-likelihood based on equation 8.35, or remove the kernel if $\alpha_m = \infty$.
6: Update noise parameters $\left\{\sigma_r^2\right\}_{r=1}^{M+1}$ using equation 8.38.
7: **until MAX-ITERATION**

مراجع

[1] Alvarez, M. and Lawrence, N. Sparse convolved gaussian processes for multi-output regression. 2008.

[2] Álvarez, M.A. and Lawrence, N.D. Computationally efficient convolved multiple output gaussian processes. *Journal of Machine Learning Research*, 12:1425–1466, 2011.

[3] Baxter, J. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, 28(1):7–39, 1997.

[4] Beal, M.J. *Variational algorithms for approximate Bayesian inference*. Ph.D. thesis, PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[5] Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. *Learning Theory and Kernel Machines*, pp. 567–580, 2003.

[6] Bishop, C.M. The relevance vector machine. in *Advances in Neural Information Processing Systems 12*, 2000.

[7] Bishop, C.M. *Pattern recognition and machine learning*, vol. 4. springer New York, 2006.

[8] Bonilla, E., Chai, K.M., and Williams, C. Multi-task gaussian process prediction. *Advances in Neural Information Processing Systems*, 20(October), 2008.

[9] Boyle, P. Gaussian processes for regression and optimisation. 2008.

[10] Boyle, P. and Frean, M. Dependent gaussian processes. *Advances in Neural Information Processing Systems*, 17:217–224, 2005.

[11] Boyle, P. and Frean, M. Multiple output gaussian process regression. 2005.

[12] Caruana, R. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[13] Cevher, Volkan. *Variational Bayes Approximation*. Rice University, 2008. Lecture notes of Graphical Models course.

[14] Csató, L. Gaussian processes: iterative sparse approximations. 2002.

[15] Csató, L. and Opper, M. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668, 2002.

[16] Damoulas, T. and Girolami, M.A. Probabilistic multi-class multi-kernel learning: On protein fold recognition and remote homology detection. *Bioinformatics*, 24(10):1264, 2008.

[17] Duchi, J. Properties of the trace and matrix derivatives. *Tutorial*, 2007.

[18] Evgeniou, T. and Pontil, M. Regularized multi–task learning. in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 109–117. ACM, 2004.

[19] Fletcher, T. Relevance vector machines explained, 2008.

[20] Friedman, N. and Nachman, I. Gaussian process networks. in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 211–219. Morgan Kaufmann Publishers Inc., 2000.

[21] Gibbs, M. and MacKay, D.J.C. Efficient implementation of gaussian processes. 1997.

[22] Gibbs, M.N. *Bayesian Gaussian processes for regression and classification*. Ph.D. thesis, 1997.

[23] Gilks, W.R., Best, NG, and Tan, KKC. Adaptive rejection metropolis sampling within gibbs sampling. *Applied Statistics*, pp. 455–472, 1995.

[24] Gilks, W.R. and Wild, P. Adaptive rejection sampling for gibbs sampling. *Applied Statistics*, pp. 337–348, 1992.

[25] Green, P.J. Penalized likelihood for general semi-parametric regression models. *International Statistical Review/Revue Internationale de Statistique*, pp. 245–259, 1987.

[26] Hastings, W.K. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[27] Heskes, T. Solving a huge number of similar tasks: a combination of multi-task learning and a hierarchical bayesian approach. in *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 233–241, 1998.

[28] Heskes, T. Empirical bayes for learning to learn. in *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pp. 367–374, 2000.

[29] Higdon, D. A process-convolution approach to modelling temperatures in the north atlantic ocean. *Environmental and Ecological Statistics*, 5(2):173–190, 1998.

[30] Jawanpuria, P. and Nath, J.S. Multi-task multiple kernel learning. *interpretation*, 1:2sl.

[31] Jeruchim, M. Techniques for estimating the bit error rate in the simulation of digital communication systems. *Selected Areas in Communications, IEEE Journal on*, 2(1):153–170, 1984.

[32] Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., and Saul, L.K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[33] Kass, R.E. and Raftery, A.E. Bayes factors. *Journal of the american statistical association*, pp. 773–795, 1995.

[34] Keerthi, S. and Chu, W. A matching pursuit approach to sparse gaussian process regression. *Advances in neural information processing systems*, 18:643, 2006.

[35] Koiran, P. Efficient learning of continuous neural networks. in *Proceedings of the seventh annual conference on Computational learning theory*, pp. 348–355. ACM, 1994.

[36] Koller, D. and Friedman, N. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[37] Koller, D., Lerner, U., and Angelov, D. A general algorithm for approximate inference and its application to hybrid bayes nets. in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 324–333. Morgan Kaufmann Publishers Inc., 1999.

[38] Kullback, S. and Leibler, R.A. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[39] Lauritzen, S.L. Propagation of probabilities, means, and variances in mixed graphical association models. *Journal of the American Statistical Association*, pp. 1098–1108, 1992.

[40] Lawrence, N., Platt, J., and Jordan, M. Extensions of the informative vector machine. *Deterministic and Statistical Methods in Machine Learning*, pp. 56–87, 2005.

[41] Lawrence, N.D. and Jordan, M.I. Semi-supervised learning via gaussian processes. *Advances in neural information processing systems*, 17:753–760, 2005.

[42] Lawrence, N.D. and Platt, J.C. Learning to learn with the informative vector machine. in *Proceedings of the twenty-first international conference on Machine learning*, p. 65. ACM, 2004.

[43] Lawrence, N.D., Seeger, M., and Herbrich, R. Fast sparse gaussian process methods: The informative vector machine. *Advances in neural information processing systems*, 15:609–616, 2002.

[44] Lawrence, N.D., Seeger, M., and Herbrich, R. The informative vector machine: A practical probabilistic alternative to the support vector machine. *Dept. Computer Science, University of Sheffield, Tech. Rep*, 2005.

[45] Lawrence, N.D. and Urtasun, R. Non-linear matrix factorization with gaussian processes. in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 601–608. ACM, 2009.

[46] Lee, H.K.H., Holloman, C.H., Calder, C.A., and Higdon, D.M. Flexible gaussian processes via convolution. *Duke University*, 2002.

[47] MacKay, D.J.C. Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6(3):469–505, 1995.

[48] MacKay, D.J.C. et al. Comparison of approximate methods for handling hyperparameters. *Neural computation*, 11(5):1035–1068, 1999.

[49] Maybeck, P.S. *Stochastic models, estimation and control*, vol. 1. Academic Pr, 1979.

[50] Melkumyan, A. and Ramos, F. A sparse covariance function for exact gaussian process inference in large datasets. *The 21st IJCAI*, 2009.

[51] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21:1087, 1953.

[52] Micchelli, C.A. and Pontil, M. Kernels for multi-task learning. *Advances in Neural Information Processing Systems*, 17:921–928, 2004.

[53] Micchelli, C.A. and Pontil, M. Regularized multi-task learning. 2004.

[54] Minka, T.P. Expectation propagation for approximate bayesian inference. in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 362–369. Morgan Kaufmann Publishers Inc., 2001.

[55] Minka, T.P. *A family of algorithms for approximate Bayesian inference*. Ph.D. thesis, Massachusetts Institute of Technology, 2001.

[56] Neal, R.M. *Bayesian learning for neural networks*, vol. 118. Springer Verlag, 1996.

[57] Neal, R.M. Monte carlo implementation of gaussian process models for bayesian regression and classification. *Arxiv preprint physics/9701026*, 1997.

[58] Nilsback, M.E. and Zisserman, A. Automated flower classification over a large number of classes. in *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pp. 722–729. IEEE, 2008.

[59] Opper, M. and Winther, O. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

[60] Osborne, M. A., Roberts, S. J., Rogers, A., Ramchurn, S. D., and Jennings, N. R. Towards real-time information processing of sensor network data using computationally efficient multi-output gaussian processes. in *Proceedings of the 7th international conference on Information processing in sensor networks*, IPSN '08, pp. 109–120, Washington, DC, USA, 2008. IEEE Computer Society.

[61] Psorakis, I., Damoulas, T., and Girolami, M.A. Multiclass relevance vector machines: Sparsity and accuracy. *IEEE Transactions on Neural Networks*, 21(10):1588–1598, 2010.

[62] Quinonero-Candela, J. and Rasmussen, C.E. A unifying view of sparse approximate gaussian process regression. *The Journal of Machine Learning Research*, 6:1939–1959, 2005.

[63] Raftery, A.E. and Lewis, S. How many iterations in the gibbs sampler. *Bayesian statistics*, 4(2):763–773, 1992.

[64] Rasmussen, C.E. and Ghahramani, Z. Occam's razor. *Advances in neural information processing systems*, pp. 294–300, 2001.

[65] Rasmussen, C.E. and Nickisch, H. Gaussian processes for machine learning (gpml) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010.

[66] Rasmussen, C.E. and Williams, CKI. Gaussian processes for machine learning. *The MIT Press, Cambridge, MA, USA*, 38:715–719, 2006.

[67] Seeger, M., Williams, C.K.I., and Lawrence, N.D. Fast forward selection to speed up sparse gaussian process regression. in *Workshop on AI and Statistics*, vol. 9, p. 2003, 2003.

[68] Shahrbaf, M. and Khashabi, D. Facial feature tracking, extraction and reduction. tech. rep., Amirkabir University of Technology(Tehran Polytechnic), 2011. Under the supervision of Mahdi M. Kalayeh and Hamid Sheikhzadeh Nadjar.

[69] Silverman, B.W. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–52, 1985.

[70] Smith, A.F.M. and Roberts, G.O. Bayesian computation via the gibbs sampler and related markov chain monte carlo methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 3–23, 1993.

[71] Smola, A.J. and Bartlett, P. Sparse greedy gaussian process regression. in *Advances in Neural Information Processing Systems 13*, 2001.

[72] Snelson, E. and Ghahramani, Z. Sparse gaussian processes using pseudo-inputs. 2006.

[73] Teh, Y.W., Seeger, M., and Jordan, M.I. Semiparametric latent factor models. in *Workshop on Artificial Intelligence and Statistics*, vol. 10, pp. 333–340. Citeseer, 2005.

[74] Thayananthan, A. Template-based pose estimation and tracking of 3d hand motion. *Cambridge, UK: Department of Engineering, University of Cambridge*, 2005.

[75] Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P., and Cipolla, R. Multivariate relevance vector machines for tracking. *Computer Vision–ECCV 2006*, pp. 124–138, 2006.

[76] Thayananthan, A., Navaratnam, R., Stenger, B., Torr, P.H.S., and Cipolla, R. Pose estimation and tracking using multivariate regression. *Pattern Recognition Letters*, 29(9):1302–1310, 2008.

[77] Tipping, A.C.F.M.E. and Faul, AC. Analysis of sparse bayesian learning. *Advances in Neural Information Processing Systems*, 14:383–389, 2002.

[78] Tipping, M.E. Sparse bayesian learning and the relevance vector machine. *The Journal of Machine Learning Research*, 1:211–244, 2001.

[79] Tipping, M.E., Faul, A., et al. Fast marginal likelihood maximisation for sparse bayesian models. in *Proceedings of the ninth international workshop on artificial intelligence and statistics*, vol. 1. Jan, 2003.

[80] Titsias, M.K. and L zaro-Gredilla, M. Spike and slab variational inference for multi-task and multiple kernel learning. *Advances in Neural Information Processing Systems(NIPS 2011)*, 2011.

[81] Tresp, V. A bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

[82] Vijayakumar, S., D'souza, A., and Schaal, S. Incremental online learning in high dimensions. *Neural Computation*, 17(12):2602–2634, 2005.

[83] Vijayakumar, S., D'souza, A., Shibata, T., Conradt, J., and Schaal, S. Statistical learning for humanoid robots. *Autonomous Robots*, 12(1):55–69, 2002.

[84] Vijayakumar, S. and Schaal, S. Locally weighted projection regression: An o (n) algorithm for incremental real time learning in high dimensional space. in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, vol. 1, pp. 288–293, 2000.

[85] Wahba, G., Lin, X., Gao, F., Xiang, D., Klein, R., and Klein, B. The bias-variance tradeoff and the randomized gacv. *Advances in Neural Information Processing Systems*, 11(5), 1999.

[86] Williams, C. and Seeger, M. Using the nyström method to speed up kernel machines. in *Advances in Neural Information Processing Systems 13*, 2001.

[87] Williams, C.K.I. and Barber, D. Bayesian classification with gaussian processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(12):1342–1351, 1998.

[88] Williams, C.K.I. and Rasmussen, C.E. Gaussian processes for regression. 1996.

[89] Yu, K., Tresp, V., and Schwaighofer, A. Learning gaussian processes from multiple tasks. in *Proceedings of the 22nd international conference on Machine learning*, pp. 1012–1019. ACM, 2005.

[90] Zhou, M., Chen, H., Paisley, J., Ren, L., Sapiro, G., and Carin, L. Non-parametric bayesian dictionary learning for sparse image representations 1. 2009.

Surname: *Khashabi* Name: *Daniel*

Title: Analysis and Implementation of of Bayesian Methods for Modelling Correlated Multioutput Information(Multitask Learning)

Supervisor: Hamid Sheikhzadeh Nadjar

Degree: Bachelor of Science Subject: Electrical Engineering
Analysis and Implementation of of Bayesian Methods for Modelling Correlated Multioutput Information(Multitask Learning)
Field: Systems and Communication

Amirkabir University of Technology(Tehran Polytechnic) Electrical Engineering Department
Date: 2012 Number of pages: 85

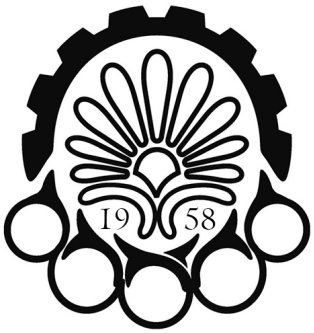
Keywords: Machine Learning, Bayesian Inference, Sparse Representation, Multioutput inference, Multitask Learning

Abstract

In this dessertation Bayesian Multitask Learning is considered. A great deal of this text is about Bayesian modeling and learning and recent breakthroughs in efficient Bayesian inference. After considering conventional low-dimensional Bayesian learning trends, we have introduced two recent methods for Bayesian Multitask learning and their results on toy data. It is worthy of mentioning that there have been a recent explosion of works in Multitask learning, though we have considered only two of them in detail. For the rest of methods, we have briefly introduced their general ideas.

In the first chapter, we will give a brief introduction to Bayesian learning and its concepts, plus notion of Multitask learning and its applications. The next two chapters are dedicated to conventional Bayesian learning methods, for scalar output and more than one output, respectively, without considering correlation between different output variables. After that we will consider recent developments in Multitask learning. The chapter also includes results of several experiments on toy data. In appedices, we have included, basic mathamat-

ics we need for probablistic and linear algebraic derivations, Bayesian inference methods and some tidious proofs for several algorithms, respectively.



Amirkabir University of Technology(Tehran Polytechnic)
Electrical Engineering Department

Dissertation Submitted in Partial

Fulfillment of The Requirements For The

Degree of Bachelor of Science in

Electrical Engineering

Analysis and Implementation of of Bayesian Methods for Modelling Correlated Multioutput Information(Multitask Learning)

Supervisor

Hamid Sheikhzadeh Nadjar

by

Daniel Khashabi

8723001

2012