فصل ۱

روش پیشنهادی

۱.۱ مقدمه

پس از آشنایی با روش‌های پیشین که برای حل مسئله مشابه مورد استفاده قرار گرفته‌اند، حال می‌توانیم به معرفی و تشریح روش‌های پیشنهادی خود برای حل مسئله پیش رو بپردازیم. در این فصل ابتدا داده‌های ورودی مسئله را همراه با فرضیات در نظر گرفته شده بیان می‌کنیم و پس از آن دو روش پیشنهادی متفاوت را بیان خواهیم نمود. در روش اول که به رویکردهای پیشین نزدیک‌تر است با تغییری از جنس روش‌های نوین در مراحل میانی به یک روش جدید می‌رسیم که به علت افزایش سرعت همگرایی می‌توان فرض و داده‌های جدیدی را از طریق حذف و تغییر تعداد کپی[1] به آن افزود و پاسخ گرفت. اما روش دوم کاملا متفاوت بوده و با رویکردی جدید در حوزه یادگیری ماشین همراه است که به کمک یادگیری تقویتی به حل مسئله مورد نظر می‌پردازد.

۲.۱ معرفی دادگان ورودی

قبل از وارد شدن به بخش روش‌های پیشنهادی نیاز است تا دادگان ورودی را مشخص و معرفی نماییم. دادگان ورودی در این پایان‌نامه همگی به صورت فایل‌های خام اسکی[2] هستند که حاوی اطلاعات جهش‌های ماتریس

[1] Copy number variation (CNV)
[2] Ascii

ژن-سلول (SNV) و اطلاعات مربوط به حذف و تغییر تعداد کپی هستند.

در ادامه جدول ۱.۱ را برای معرفی اندیس‌های بکار گرفته شده در روابط مربوط به روش پیشنهادی اول معرفی می‌نماییم.

جدول ۱.۱: اندیس‌های به کار رفته در روابط روش پیشنهادی اول

D	ماتریس داده نویزی در دسترس که مقادیر ۰ و ۱ در آن قرار دارد
E	ماتریس داده حقیقی بدون نویز که به دنبال آن هستیم
T	درخت فیلوژنی جهش‌ها
σ	بردار انتصابات
$\wp$	بردار پذیرش فقدان
X_T	ماتریس متناظر درخت T
N	تعداد سلول‌های نمونه
M	تعداد جهش‌ها
$\mathcal{N}$	مجموعه سلول‌های متمایز از هم
$\mathcal{M}$	مجموعه جهش‌های متمایز از هم
α	نرخ خطای مثبت کاذب
β	نرخ خطای منفی کاذب

۳.۱ روش پیشنهادی برای مدیریت داده‌های از دست رفته

در ادامه این بخش به معرفی روش‌های پیشنهادی پرداخته خواهد شد اما در ابتدا به دلیل وجود داده‌های از دست رفته در پایگاه‌داده‌های مورد استفاده لازم است تا به بررسی و ارائه رویکردی برای حل این مشکل پرداخته شود و در ادامه پس از معرفی روش پیشنهادی برای مددریت این داده‌های از دست رفته، هر کدام از روش‌های پیشنهادی به تفضیل شرح داده شود.

همان‌گونه که در داده‌های حقیقی مشاهده شد در پایگاه داده‌های حقیقی ما با اطلاعات از دست رفته مواجه هستیم و به همین دلیل نیز سعی کردیم تا در پایگاه داده مجازی تولید شده نیز به مشابه داده‌های حقیقی، شامل اطلاعات از دست رفته باشد. در این بخش به رویکرد روش محاسبه استاتیک برای مدیریت این داده‌های از دست رفته می‌پردازیم و در بخش بعد به معرفی روشی برای بدست آوردن درخت فیلوژنی پرداخته خواهد شد. همان‌گونه

که در ادامه بررسی خواهد شد، این اطلاعات از دست رفته در پایگاه داده‌های مختلف نرخ‌های متفاوتی دارد که تاثیر این تغییرات نیز در روشی پیشنهادی بررسی خواهد شد.

۱.۳.۱ روش محاسبه استاتیک

در این روش قصد داریم تا به یک‌باره بتوانیم مقادیر مناسب برای داده‌هایی که از دست رفته‌اند را تخمین بزنیم. در این روش باید توجه شود که ما لزوما به دنبال جایگذاری مقدار از دست رفته با مقدار درست واقعی نیستیم. اگرچه چنین بیانی در نگاه اول ممکن است تعجب‌آور باشد اما با دقت بیشتر متوجه خواهیم شد که ما در آینده برای خطاهای موجود در پایگاه داده مدل‌سازی‌های محدودی داریم. مدل‌هایی که بهترین آن‌ها نیز ممکن است با واقعیت نویز افزوده شده به دادگان متفاوت باشد. در نتیجه اگر مطمئن بودیم که تمام داده‌هایی که موجود می‌باشند بدون خطا هستند در آن صورت ما نیز به دنبال یافتن جایگذاری با مقدار واقعی بودیم اما در حال حاضر که درصدی از داده‌های در دسترس خود همراه با خطا می‌باشند، ما به دنبال جایگذاری‌ای هستیم که بتواند در مجموع با مدل‌سازی خطایی که در نظر می‌گیریم بیشترین سازگاری را داشته باشد کما اینکه ممکن است در حقیقت جایگزاری اشتباهی انجام داده باشیم. حال با توجه به توضیحی که بیان شد به تشریح این روش می‌پردازیم.

با توجه به فرض مدل مکان‌های بی‌نهایت می‌دانیم که جهش‌های اتفاق افتاده در والد در تمامی نسل‌های آینده باقی خواهد ماند. بنابرین اگر تمامی جهش‌های نمونه (سلول) a در نمونه‌ای دیگر مانند b قرار داشته باشد، بنابرین می‌توان نتیجه گرفت که a یکی از اجداد b خواهد بود. همین فرضیه هسته اصلی روش پیشنهادی درنظر گرفته شده را تشکیل می‌دهد. بنابرین اگر جهش i در سلول a از دست رفته است، با توجه به اینکه آن جهش در سلول b چه وضعیتی دارد می‌توان تصمیم‌گیری کرد. اگر $b(i) = ۰$ باشد، در این صورت $a(i)$ حتما باید ۰ باشد وگرنه فرض اولیه مدل مکان‌های بی‌نهایت نقض خواهد شد. اما اگر $b(i) = ۱$ باشد، آنگاه نتیجه خاصی نمی‌توان گرفت و باید به دنبال نمونه والد a یعنی نمونه d باشیم. حال اگر $d(i) = ۱$ باشد، آنگاه $a(i)$ حتما باید ۱ باشد. اما اگر $d(i) = ۰$ بود آنگاه انتخاب هر مقداری برای $a(i)$ تقریبا آزاد خواهد بود زیرا با فرض اولیه تناقضی ندارد و اینکه ساختار فیلوژنی را تغییر نمی‌دهد. اما از آنجایی که خود داده‌های در دسترس شامل خطا می‌باشند و هر نمونه‌ای که حاوی اطلاعات از دست رفته است لزوما یک نواده یا یک والد ندارد، مجموعه‌ای از سلول‌های فرزند یا والد خواهند بود که متناسب با پارمترهای خطایی که در نظر می‌گیریم و فاصله ژنی‌ای که دارند می‌توانند در

تصمیم‌گیری تاثیرگزار باشند. صورت دقیق‌تر توضیحات داده شده را می‌توان به صورت فرمولی که در ادامه آمده است به نمایش درآورد.

در ابتدا تابعی به نام $F_s(D_{ij})$ تعریف می‌کنیم که به نوعی با توجه به ارزشی که به سلول‌های نواده شده از سلول j می‌دهد سعی دارد تا اطمینان ۰ بودن داده از دست رفته D_{ij} را بیان کند.

برای محاسبه این تابع می‌دانیم که ابتدا سلول‌های مختلف با توجه به احتمال نواده بودنشان باید رتبه‌بندی شوند و وزن بگیرند. پس از آن هر سلول متناسب با ارزش تاثیرگزاری خود می‌تواند در مورد جایگاه جهش i برای سلول j نظر دهد.

$$F_s(D_{ij}) = \sum_{n \in \mathcal{N}} (۱ - D_{mj}) \prod_{m=۱}^{M} W(D_{mn}, D_{mj}) \tag{۱.۱}$$

در فرمول ۱.۱ مجموعه $\mathcal{N}$ برابر با مجموعه سلول‌های متمایز از هم است. زیرا که در بسیاری از پایگاه‌داده‌ها از یک نمونه سلول ممکن است چندین نمونه وجود داشته باشد که وجود آن‌ها باعث بایس در محاسبات ما خواهد شد. همچنین تابع $W_s(c,p)$ به ارزش‌دهی جهش c در برابر p به عنوان نواده بودن می‌پردازد که در فرمول ۲.۱ تعریف شده است.

$$W(c,p) = \begin{cases} ۱ & \text{if} \quad c = ۱, p = ۱ \\ ۱ - \xi & \text{if} \quad c = ۱, p = ۰ \\ ۰ & \text{if} \quad c = ۰, p = ۱ \\ ۱ & \text{if} \quad c = ۰, p = ۰ \end{cases} \tag{۲.۱}$$

مقدار ξ عددی بین $(۰, ۱)$ است که پارامتری در جهت میزان ارزش‌دهی به نوادگان با فواصل مختلف می‌باشد. هرچه این عدد بزرگتر باشد به معنی کم‌ارزش‌تر شدن نوادگان با فواصل بیشتر است و بلاعکس.

به همین صورت برای اولاد سلول j نیز می‌توان مشابه حالت قبل عمل کرد که روابط آن به صورت فرمول ۳.۱ خواهد شد.

$$F_a(D_{ij}) = \sum_{n \in \mathcal{N}} D_{mj} \prod_{m=۱}^{M} W(D_{mj}, D_{mn}) \tag{۳.۱}$$

حال دو نکته در استفاده از روابط بالا باقی خواهد ماند.

نکته اول وجود داده‌های دیگر از دست رفته در محاسبه توابع است که به دو صورت می‌توان با آن‌ها برخورد نمود. رویکرد اول این است که در آن‌جایگاه ژنی از محاسبه آن خود داری شود و رویکرد دوم استفاده از از مقدار ۰٫۵ یا فراوانی نسبی آن جهش در محسبات است که ما رویکرد اول را در این گزارش استفاده خواهیم کرد.

نکته دوم وجود خطا در داده‌هاست. برای مدیریت این مشکل می‌توان با مدل‌سازی خطا که به صورت فرمول ۴.۱ بیان می‌شود، برخورد کرد.

$$\begin{aligned} P(D_{ij}=1|E_{ij}=0)=\alpha, \qquad & P(D_{ij}=0|E_{ij}=0)=1-\alpha \\ P(D_{ij}=0|E_{ij}=1)=\beta, \qquad & P(D_{ij}=1|E_{ij}=1)=1-\beta \end{aligned} \tag{۴.۱}$$

پس از تعریف مدل‌سازی خطا می‌توان روابط قبلی را مجددا به صورتی که در ادامه آمده است بازنویسی کرد.

$$W_e(c,p)=\sum_{i,j\in\{0,1\}} P(c|E_c=i)P(p|E_p=j)W(i,j) \tag{۵.۱}$$

که در این صورت توابع F_p و F_a نیز به صورت زیر همراه با مدل‌سازی خطا بازتعریف خواهند شد.

$$\begin{aligned} \hat{F}_s(D_{ij}) &= \sum_{n\in\mathcal{N}}[1-D_{mj}(1-\alpha)]\prod_{m=1}^{M} W_e(D_{mn},D_{mj}) \\ \hat{F}_a(D_{ij}) &= \sum_{n\in\mathcal{N}} D_{mj}(1-\beta)\prod_{m=1}^{M} W_e(D_{mj},D_{mn}) \end{aligned} \tag{۶.۱}$$

حال پس از محاسبه مقادیر $\hat{F}_s$ و $\hat{F}_a$ می‌توان در مورد داده نامعلوم D_{ij} به صورت فرمول ۷.۱ تصمیم گرفت.

$$D_{ij}=\begin{cases} 0 & \text{if} \quad \hat{F}_s \geq \hat{F}_a \\ 1 & \text{if} \quad \hat{F}_s < \hat{F}_a \end{cases} \tag{۷.۱}$$

همچنین با کمی دقت در فرمول‌بندی انجام شده اگر برای تمام i,jهای ماتریس D این مقادیر توابع $\hat{F}$ محاسبه شوند، خود می‌توانند معیاری برای ارزیابی پایگاه‌داده در دسترس و احتمال درستی فرض مدل مکان‌های بی‌نهایت

باشند.

۱.۱.۳.۱ تصادفی

پر کردن کاملا تصادفی میس‌ها. در این روش به صورت تصادفی مقادیر از دست رفته را مقدار دهی می‌کنیم. تنها نکته‌ای که در این روش وجود دارد این است که نباید این پرکردن تصادفی داده‌های از دست رفته باعث شود تا پارامترهای مدل‌سازی‌ای که از قبل در نظر گرفته بودیم با این روش نادقیق شوند.

۴.۱ روش پیشنهادی اول (درخت‌بازی)

در این روش ما بر حسب بهتر کردن یک پاسخی که از پیش داشتیم به دنبال رسیدن به بهترین پاسخ ممکن در طی تکرار[3] پشت سر هم هستیم. برای مشخص شدن نحوه کارکرد روش پیشنهادی در مراحلی که در ادامه بیان خواهد شد به عنوان مثال یک ماتریس

$$D = \begin{bmatrix} ۰ & ۱ & ۱ & ۰ \\ ۰ & ۰ & ۱ & ۱ \\ ۱ & ۰ & ۱ & ۰ \\ ۱ & ۱ & ۰ & ۱ \\ ۱ & ۰ & ۱ & ۱ \\ ۰ & ۱ & ۱ & ۱ \end{bmatrix} \tag{۸.۱}$$

را به عنوان ورودی مساله به همراه پارامترهای α و β در نظر بگیرید. (برای راحتی کار فرض کرده‌ایم که داده از دست رفته در D نداریم.)

[3] Iteration

۱.۴.۱ پیش‌پردازش

قبل از شروع باید بر روی داده‌ها یک پیش‌پردازش اعمال کنیم که وابسته به سیاست درنظر گفته شده می‌تواند باعث تغییر در پاسخ نهایی نیز شود. به این منظور داده‌هایی که miss شده‌اند با روش‌های زیر می‌تواند برای ورود به مرحله بعد تخمین زده شود.

به صورت تصادفی

۲.۴.۱ اولین پاسخ (درخت تصادفی)

همان‌گونه که از قبل می‌دانستیم خروجی نهایی ما برابر با درختی خواهد بود که نودهای آن برابر با جهش‌های ماتریس ورودی ما و برگ‌های آن برابر با نمونه‌های مشاهده شده خواهند بود. در روش پیشنهادی اول ما به دنبال بهتر کردن این درخت به عنوان پاسخ هستیم. از این رو پایه این روش پیشنهادی اول بر مبنای بهتر کردن پاسخ فعلی بنا نهاده شده است. در نتیجه ما همواره پاسخی به عنوان جواب نهایی داریم که تلاش خواهیم نمود تا با استفاده از ابزرهایی بتوانیم ابا ایجاد تغییری در این پاسخ به پاسخی جدید برسیم که قابل مقایسه با پاسخ فعلی برای انجام مراحل بعدی باشد.

با توجه به توضیحاتی که داده شد ما برای شروع الگوریتم پیشنهادی اول خود نیاز به یک پاسخ داریم. این پاسخ که درخت فیلوژنی هست با توجه پارامترهای ورودی و انتخاب یک نود (ژن) $root$ به عنوان ریشه این درخت به صورت زیر حاصل می‌شود.

$$\begin{aligned} &\mathcal{M} = \{۱ \dots M\} \\ &\hat{B}_{T_۱} = [R_۱(\mathcal{M} - |۱|), R_۲(\mathcal{M} - |۲|), \dots, R_{root}(\{\}), \dots, R_M(\mathcal{M} - |M|)] \end{aligned} \tag{۹.۱}$$

که در این رابطه $\mathcal{M}$ برابر با مجموعه تمامی جهش‌های متمایز از شماره ۱ تا M است و $\hat{B}$ مشخص‌کننده نود پدر در درخت برای جهش iام در این لیست خود است که توسط تابع $R_i(X)$ به صورت کاملا یکنواخت[۴] از اعضای مجموعه X انتخاب می‌شود.

با توجه به مثالی که در رابطه ۸.۱ زده شد فرض کنید مقدار بردار $\hat{B}$ با ریشه $root = ۲$ به صورت رابطه ۱۰.۱

[4] Uniform

شود.

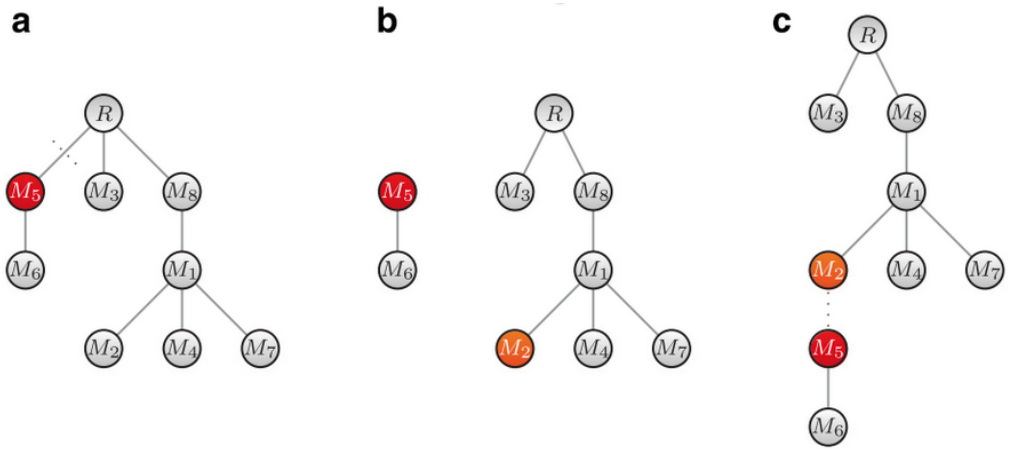
$$\hat{B} = [۲, ۱, -, ۰] \tag{۱۰.۱}$$

که درخت شکل ۱.۱ را نتیجه می‌دهد.

شکل ۱.۱: درخت تصادفی اول

۳.۴.۱ پاسخی جدید

تا به اینجا ما یک درخت فیلوژنی به عنوان پاسخ داریم که در این بخش می‌خواهیم با انجام تغییراتی بر روی آن به یک پاسخ جدید برسیم تا در گام‌های بعدی بتوانیم با مقایسه آن‌ها تصمیمات لازم را برای ادامه الگوریتم بگیریم. به همین منظور تقریبا مشابه با روش [؟] به صورت هرس و اتصال دوباره[۵] قصد داریم تا درخت پاسخ فعلی را برای رسیدن به یک پاسخ دیگر تغییر دهیم. در شکل ۲.۱ مثالی از این روش آورده شده است. در این



شکل ۲.۱: نحوه انجام کار روش هرس و اتصال دوباره [؟].

شکل مطابق با قسمت a یکی از نودهای درخت (به جز ریشه) انتخاب می‌شود (در اینجا نود M_5) و اتصال از پدرش قطع می‌شود. در این حالت به دو درخت مشابه با شکل b می‌رسیم. حال در درخت باقی مانده یک نود

[5] Prune and reattach

دیگر (M_2) به عنوان پدری جدید انتخاب می‌شود تا با این تغییر به درخت جدید شکل c برسیم.
در نتیجه با این تکنیک می‌توان به پاسخ‌های جدید رسید اما سوالی که باقی ماند این است که این دو نود چگونه باید انتخاب شوند؟ این دو انتخاب به صورت هوشمند توسط دو شبکه عمیق گرفته می‌شود. این شبکه‌ها که در اصل شبکه‌های یادگیری تقویتی[۶] عمیق هستند، از پیش برای این منظور آموزش داده شده‌اند. شبکه اول برای انتخاب محل برش (هرس) مورد استفاده قرار می‌گیرد و شبکه دوم با دریافت خروجی ××××××× دو درخت راهبر-پیرو[۷] مکان‌های مناسب برای بازاتصال را ارزش‌گزاری می‌کند. این دو شبکه در قسمت‌های ۶.۴.۱ و ۷.۴.۱ به تفضیل شرح داده شده‌اند.

۴.۴.۱ مقایسه و ارزیابی پاسخ‌ها

پس از اینکه از پاسخ فعلی به یک پاسخ جدید رسیدیم حال می‌توان کیفیت این دو پاسخ را باهم مقایسه کرد و پس از آن با توجه به امتیاز دو پاسخ در مورد پذیرش یا عدم پذیرش پاسخ جدید در برابر پاسخ فعلی تصمیم گرفت. این فرآیند شامل دو بخش اصلی است که در دو زیربخشی که در ادامه آمده است بیان شده‌اند.

۱.۴.۴.۱ تبدیل درخت پاسخ به ماتریس

برای ادامه روش پیشنهادی و مقایسات لازم است تا درخت پاسخ را به ماتریس X تبدیل کنیم که قابل بررسی با داده‌های مشاهده شده D باشد. ماتریس X مشابه با ماتریس D متشکل از مقادیر ۰ و ۱ خواهد بود که به عنوان مثال $X_{i,j} = ۱$ به این معنی است که طبق درخت T در سلول i جهش j مشاهده نشده است. ما هر درخت T را می‌توانیم با مقادیر مختلفی از σ و $\wp$ مزین کنیم و به ماتریس‌های مختلفی برسیم. اما در نهایت مهمترین پارامترها که بیشترین امتیاز را برای درخت ما بوجود می‌آورند مطلوب ما خواهند بود و ماتریس متناظر با آن حالت را X می‌نامیم و به مراحل بعدی برای محاسبات انتقال می‌دهیم. در نتیجه کار ما در این بخش این خواهد بود که به ازای درخت دلخواه T بتوانیم بهترین σ و $\wp$ را بدست آوریم. به رابطه

$$A = \qquad (۱۱.۱)$$

[6] Reinforcement learning
[7] Master-slave

در این مرحله لازم است تا درخت را به صورت یک ماتریس در فرم D داشته باشیم که پس از آن بتوانیم به مقایسه این ماتریس بدست آمده که آن را از این پس X می‌گوییم،

همان‌گونه که از ابتدا می‌دانیم ما به دنبال درخت فیلوژنی حقیقی داده‌های نویزی مشاهده شده D هستیم. این درخت در این روش برابر با درختی است که،

- نحوه قرارگیری ژن‌ها در ساختار درخت (T)
- محل‌هایی در درخت که جهش‌های قبلی در آن‌ها حذف می‌شوند ($\wp$)
- نحوه انتصاب نمونه‌های مشاهده شده به درخت (σ)
- و در نهایت پارامترهایی که برای مدل‌سازی خطای بوجود آمده در داده‌های در دسترس‌مان تعیین شده است (θ)

به گونه‌ای انتخاب شوند که محتمل‌ترین حالت را برای مشاهده داده‌های D بوجود آورند که در این حالت ما رابطه علت توضیحات مجدد این موارد به این دلیل است که این بخش مهمترین بخش در ساختار روش پیشنهادی اول است.

۲.۴.۴.۱ مقایسه پاسخ فعلی با پاسخ آرمانی

پس از استخراج ماتریس مناسب از درخت می‌توان به ارزش‌گزاری و محاسبه درست‌نمایی[8] پرداخت. این عمل به صورت رابطه ۱۱.۱ محاسبه می‌شود.

$$L : P(D|T, \sigma, \wp, \theta) = \prod_{n=1}^{n=N} \prod_{m=1}^{m=M} P(D_{nm}|X_{nm}) \tag{۱۲.۱}$$

که X برابر ماتریس بدست آمده از درخت T با توجه به بردارهای σ و $\wp$ است. این رابطه بیانگر احتمال مشاهده ماتریس داده ورودی D در صورتی است که درخت فیلوژنی صحیح T و پارامترهای حقیقی θ باشد که توسط بردارهای σ و $\wp$ تثبیت شده است. هرچه این احتمال بالاتر باشد نمایانگر این است که درخت، پارامترها و

[8] Likelihood

بردارهای کنترلی ما بگونه‌ای انتخاب شده‌اند که محتمل‌ترین حالت برای مشاهده داده‌های ورودی ما هست و در این صورت بهترین پاسخ برای ما همان پاسخی خواهد بود که محتمل‌ترین باشد. از این رو با دانستن θ ما به دنبال Tای به همراه بردارهای مربوطه آن هستیم که پاسخ رابطه باشد.

$$(T, \sigma, \wp)_{\mathrm{ML}} = \arg \max_{(T,\sigma,\wp)} P(D|T, \sigma, \wp, \theta) \tag{۱۳.۱}$$

اما همانگونه که می‌دانیم ما به دنبال بهترین درخت T هستیم که σ و $\wp$ در آن درخت برای ما اهمیت دارند. در واقع هر درخت T دارای امتیاز $S(T)$ است که به صورت رابطه تعریف می‌شود.

$$S(T) = P(D|T, \sigma^*, \wp^*), \qquad \sigma^* = \arg \max_{\sigma} P(D|T, \sigma, \wp) \tag{۱۴.۱}$$

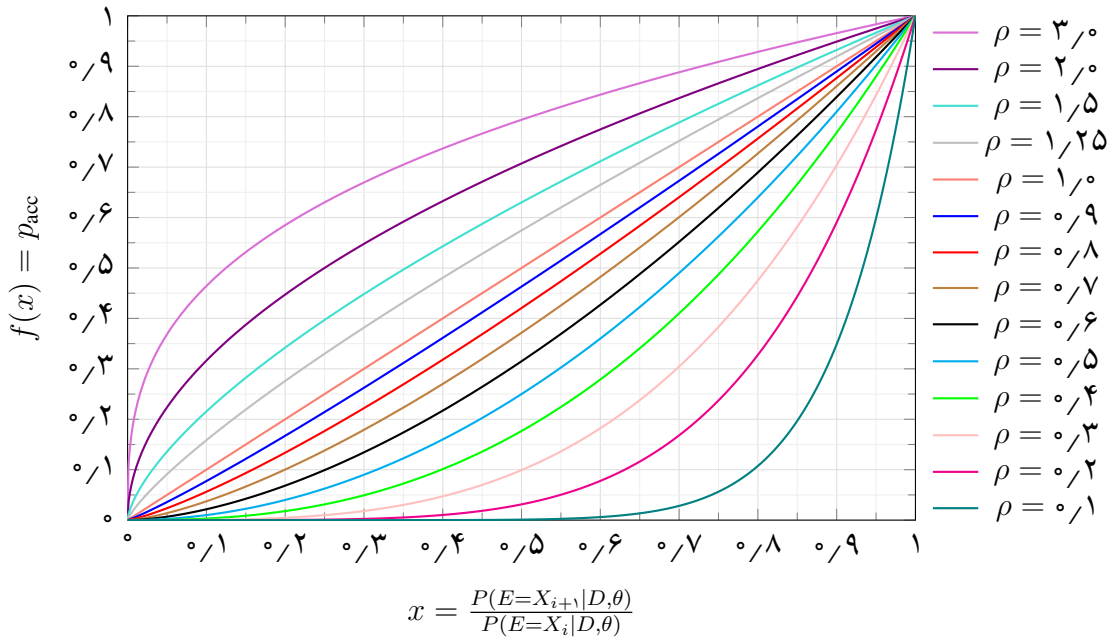
۵.۴.۱ پذیرش پاسخ‌های جدید و یافتن بهترین پاسخ

در این مرحله ما دو پاسخ با امتیازهایشان در اختیار داریم که می‌توانیم برحسب آن‌ها برای ورود به تکرار بعد تصمیم‌گیری نماییم. این فرآیند توسط رابطه‌ای که در ادامه آمده است انجام می‌شود.

$$p_{\mathrm{acc}} = \min \left[۱, \left(\frac{P(E = X_{i+۱}|D, \theta)}{P(E = X_i|D, \theta)} \right)^{\rho^{-۱}} \right] \tag{۱۵.۱}$$

در رابطه ۱۴.۱، اگر پاسخ جدید بهتر از پاسخ فعلی باشد بیان می‌کند که افزایش بهینگی در پاسخ جدید باعث می‌شود تا صورت کسر مقداری بیش از مخرج بگیرید که در این صورت p_{acc} که برابر با احتمال پذیرش پاسخ جدید است، برابر ۱ خواهد شد که یعنی حتما پاسخ جدید به عنوان پاسخ پابرجا برای ورود به تکرار بعد در نظر گرفته می‌شود. اما اگر پاسخ جدید (درخت جدید) بهتر از پاسخ فعلی ارزیابی نشود ما آن را مستقیما رد نمی‌کنیم و به احتمالی کمتر از ۱ ممکن است آن را بپذیریم. دلیل این پذیرش جلوگیری از به دام افتادن الگوریتم در پاسخ‌مان در بیشینه‌های محلی[9] است. در شکل تاثیر تغییر پارامتر ρ در احتمال پذیرش پاسخ‌های جدیدی که مطلوب‌تر از پاسخ فعلی نیستند نمایش داده شده است.

[9] Local maxima



شکل ۳.۱: نمودار تغییر احتمال پذیرش پاسخ جدید نامطلوب‌تر با توجه به مقدار پارامتر ρ.

۶.۴.۱ شبکه هرس‌کننده

۷.۴.۱ شبکه بازاتصال‌کننده

۸.۴.۱ جمع‌بندی

روش پیشنهادی ما الگو گرفته شده از روش ارائه شده در مقاله سایت بود اما با این تفاوت که در آنجا فرض مکان‌های بی نهایت بود ولی ما فرض اسکارلت را جایگزین کردیم که در این بین چون فضای جست و جو بزرگتر شد در نتیجه مجبور شدیم نحوه برداشتن گام های خود را تغییر بدیم و هوشمندانه تر جلو برویم که در این فضای بزرگتر بتوانیم به جواب مناسب برسیم. در سایت برای رسیدن به درخت بهتر به دنبال تنظیم کردن پارامترهای خطا بود در حالی که ممکن بود با جواب واقعی فاصله داشته باشد اما چون بدنبال جواب با امتیاز بالا بود در نتیجه این پارامتری بودن و جست و جو برای مقادیر بهینه خطا در روش آن وجود داشت اما ما برای بهتر کردن امتیاز به جای تغییر پارامترهای خطا به ازای یک درخت جست و جوی ضمیمه کردن های مختلف سلول‌ها و لاس شدن جهش ها را جست و جو میکنیم.

۵.۱ روش پیشنهادی دوم

۱.۵.۱ مقدمه

۲.۵.۱ دادگان ورودی

۳.۵.۱ تبدیل داده‌ها به بردار ویژگی

۴.۵.۱ مدل‌سازی مساله

۱.۴.۵.۱ معماری شبکه

۵.۵.۱ تابع هزینه

۶.۵.۱ اصلاح خطا و یافتن درخت جواب

۶.۱ نتیجه‌گیری

جدول ۲.۱: پارامترهای مدل ریاضی

t_{ik}	زمان خدمت‌دهی به بیمار در مرحله kام
$\tilde{t}_{ik}$	زمان فاری خدمت‌دهی به بیمار در محله kام
t^p_{ik}	مقدار بدبینانه (حداکثر) برای زمان خدمت‌دهی به بیمار در مرحله kام
t^m_{ik}	محتمل‌ترین مقدار برای زمان خدمت‌دهی به بیمار در مرحله kام
t^o_{ik}	مقدار خوشبینانه (حداقل) برای زمان خدمت‌دهی به بیمار در مرحله kام

جدول ۳.۱: متغیرهای مدل ریاضی

X_{ild_k}	متغیر صفر-یک تخصیص بیمار به تخت/اتاق عمل
S_{ild_k}	زمان شروع خدمت‌دهی به بیمار
Y_{ijkl_k}	متغیر صفر-یک توالی بیماران
V_{ni}	متغیر صفر-یک تخصیص جراح به بیمار