فهرستِ مطالب

۱ مقدمه

امروزه دسته‌بندی داده‌ها[1] به صورت وسیعی در علوم کامپیوتر[2] و هوش مصنوعی[3] کاربرد دارند. مساله دسته‌بندی داده‌ها معمولا به صورت تشخیص یک دسته مناسب از بین چند دسته موجود برای یک داده ورودی تعریف می‌شود که در آن دسته‌های موجود بر اساس تعدادی از اعضای موجود در آن دسته (مجموعه آموزش) توصیف شده‌اند. در حالت کلی نمی‌توان یک روش برای دسته‌بندی ارائه داد که بتواند داده‌ها را همیشه به صورت درست دسته‌بندی نماید، اما می‌توان روش‌هایی بهینه و یا نیمه‌بهینه برای این کار ارائه داد. روش‌های مختلفی برای دسته‌بندی داده‌ها مطرح شده است که بسیاری از این روش‌ها، ابتکاری[4] هستند. روش‌های دسته‌بندی داده‌ها را به دو دسته کلی روش‌های هوش‌مصنوعی و روش‌های آماری تقسیم می‌کنند که روش‌های هوش‌مصنوعی بیشتر به دنبال استفاده از تکنیک‌های هوش‌مصنوعی برای حل این مساله هستند و روش‌های آماری بیشتر تلاش در جهت مدل‌سازی مساله توسط یک فرآیند تصادفی دارند و سپس از ابزارها و روش‌های موجود در آمار و احتمال برای حل مساله استفاده می‌کنند. روش Naive Bayes یکی از ساده‌ترین روش‌ها از نظر پیاده‌سازی است که در موارد خاص دارای بهترین کارایی است و می‌توان از آن برای مسائل پیچیده‌تر (مسائلی که در شرط‌های لازم برای کارآیی Naive Bayes نمی‌گنجند) نیز استفاده کرد. Naive Bayes را به طور کلی به عنوان یک روش آماری می‌شناسند.

تئوری اطلاعات، شاخه‌ای از ریاضیات و علوم ارتباطی است که به بررسی مساله‌هایی از قبیل مقدار آشفتگی در پیام‌های ارسالی از یک منبع، خطاهای ممکن حین انتقال پیام از مبدا به مقصد و همچنین فشرده‌سازی داده‌ها می‌پردازد. این شاخه از علم با انتشار مقاله‌ای از Shannon در سال ۱۹۴۸ رسما متولد شد.

در مساله‌های دسته‌بندی می‌توان مساله را به نوعی منبع ارسال‌کننده پیام تشبیه کرد که تعدادی از پیام‌های آن منبع را در اختیار داریم و می‌خواهیم در مورد امکان ارسال شدن یک پیام توسط آن منبع نظر بدهیم، یعنی پیش‌بینی کنیم که آیا منبع اطلاعاتی می‌تواند یک پیام مشخص را ارسال کند یا خیر. روش‌های مختلفی برای ایجاد یک منبع اطلاعاتی با استفاده از تعدادی از پیام‌های آن منبع وجود دارد.

در این پایان‌نامه تلاش شده است تا یک روش برای دسته‌بندی داده‌ها بر اساس تئوری

[1] Data Classification

[2] Computer Science

[3] Artificial Intelligence

[4] Heuristic

اطلاعات (ITDC) ارائه شود و به صورت ریاضی ثابت شود که این روش (ITDC) در صورت برقرار بودن شرایطی مانند Naive Bayes عمل می‌کند و در حقیقت یک گسترش برای Naive Bayes محسوب می‌شود و سپس با دو مثال عملی به بررسی کارآیی ITDC پرداخته شده است. شرط‌های مورد نیاز برای عملکرد مشابه Naive Bayes و ITDC بیشتر معطوف به مجموعه آموزش است که شامل به اندازه کافی بزرگ بودن و همچنین گستردگی کافی آن مجموعه است. شرایط مورد نظر در حقیقت شرایطی هستند که در آن‌ها Naive Bayes و سایر الگوریتم‌های مبتنی بر Bayes به درستی (با دقت خوبی) کار می‌کنند.

ساختار پایان‌نامه به این صورت می‌باشد که در فصل دوم به معرفی تئوری اطلاعات و مرور کلی بر مفاهیم اولیه آن پرداخته شده است. فصل سوم، شامل معرفی دقیق مساله دسته‌بندی و همچنین بررسی اجمالی چند روش موجود با تاکید بر Naive Bayes خواهد بود و در ادامه فصل چهارم به بررسی دقیق‌تر Naive Bayes می‌پردازد. در فصل پنجم، به معرفی ITDC پرداخته شده است. فصل‌های ششم و هفتم نیز شامل چند مساله معروف در زمینه دسته‌بندی داده‌ها می‌باشد.

۲ معرفی تئوری اطلاعات

در این بخش به معرفی مفاهیم اساسی تئوری اطلاعات[۵] می پردازیم. تئوری اطلاعات در مقاله‌ای توسط Claude E. Shannon به نام A mathematical theory of computation[۲۶]، که شامل نتایج پایه‌ای در رابطه با منابع بدون حافظه و کانال‌ها و معرفی مدل‌های جامع‌تر ارتباطی، از جمله منابع و کانال‌ها با تعداد حالات متناهی بود بوجود آمد[۱۲].

۲-۱ تئوری اطلاعات

Shannon به مطالعه ارتباطات در مجردترین حالت پرداخت. برای این منظور او نیاز داشت تا پیام‌ها(که انتقال آن‌ها بین فرستنده و گیرنده یک ارتباط را تشکیل می‌دهد) را بررسی کند. برای این منظور Shannon به جای در نظر گرفتن پیام‌ها، یک سامانه[۶] را به عنوان فرستنده پیام در نظر گرفت. هر یک از این سامانه‌ها می‌توانند تعداد متناهی و یا نامتناهی پیام را با احتمالات متفاوت ایجاد نمایند.

Entropy

Shannon نیاز به یک معیار برای سنجش مقدار اطلاعاتی که توسط سامانه مبدا هنگامی که یک پیام توسط آن برای ارسال انتخاب می‌شد (به عبارت دیگر مقدار عدم‌اطمینانی که در پیام خروجی مبدا به وجود می‌آید) داشت. Hartly پیش از Shannon، با مثال‌های زیادی نشان داده بود که تعداد خروجی‌های ممکن یک منبع (پیام‌ها) در مدت زمان T به صورت نمایی با T رشد می‌کند [۱] و بنابراین مشاهده، پیشنهاد ارائه تعریف میزان اطلاعات یک منبع بر اساس لگاریتم این رشد را داده بود[۲]. Shannon این ایده را با وارد کردن خصوصیت‌های آماری منبع تولید کننده پیام گسترش داد.
تعریف پیشنهادی Shanoon در ساده‌ترین حالت ممکن که در آن خروجی یک منبع به وسیله یک متغیر تصادفی[۷] X که مقادیر خود را از یک مجموعه مانند $\Sigma = \{۱, ۲, \cdots, N\}$ با احتمال $p_i, i \in \Sigma$ انتخاب می‌کند (با فرض مخالف صفر بودن همه p_iها) به صورت زیر

[5] Information Theory

[6] System

[7] Random Variable

است:

$$H(X) = \sum_i p_i \log \frac{۱}{p_i} \qquad (۲-۱)$$

$H(X)$ را آشفتگی منبع[8] می‌نامند. اگر لگاریتم در عبارت بالا در پایه دو محاسبه گردد، آشفتگی منبع بر اساس $\frac{bits}{symbols}$ (تعدادبیت‌ها به ازای هر سمبل) خواهد بود. در مبحث فشرده‌سازی[9]، از مقدار Entropy یک مجموعه داده به عنوان بهترین نرخ فشرده‌سازی استفاده می‌شود[۱۹].
Shannon نشان‌داد که $H(X)$ در خاصیت‌هایی که از یک معیار اندازه‌گیری مقدار اطلاعات به طور بدیهی انتظار می‌رود صدق می‌کند. این خاصیت‌ها به صورت زیر هستند:[۱۹]

۱. اگر منبع تولید کننده پیام دارای احتمالات مساوی باشد (یعنی $\forall\ i,j \in\ \Sigma,\ p_i = p_j$) میزان آشفتگی منبع برابر خواهد بود با:

$$H(X) = \log |\Sigma|$$

و با تعریف Hartley[۸] هم‌خوان است.

۲. اگر منبعی یک پیام را با احتمال ۱ تولید کند و در نتیجه سایر پیام‌ها دارای احتمالی برابر صفر باشند، میزان آشفتگی منبع برابر خواهد بود با:

$$H(X) = ۰$$

که با شهود ما از یک چنین منبعی مطابقت دارد. در حقیقت در مورد خروجی این منبع هیچ عدم قطعیتی وجود ندارد و به صورت قطعی خروجی آن را شناخته شده است.

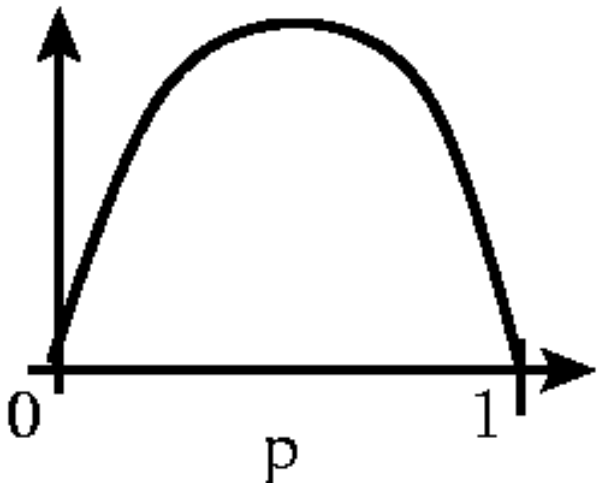
۳. اگر یک مجموعه از پیام‌ها مانند Σ موجود باشد به طوری که $\Sigma' \subset \Sigma$ و $\forall i,j \in \Sigma' : p_i = p_j$ و $\forall i \in \Sigma - \Sigma' : p_i = ۰$ آن‌گاه میزان آشفتگی منبع تولید کننده Σ برابر خواهد بود با:

$$H(X) = \log |\Sigma'|$$

این موضوع با شهود ما از این‌که مقدار آشفتگی در پیام‌های تولید شده توسط یک منبع با خصوصیات توصیف شده فوق معادل است با مقدار آشفتگی منبعی که پیام‌های Σ' را تولید می‌کند صدق می‌کند.

[8] Entropy of the Source

[9] Compression



شکل ۲ـ۱: میزان آشفتگی به ازای مقادیر مختلف احتمالات در حالت دو متغیره

۴. مقدار آشفتگی (عدم قطعیت) هر منبع بزرگ‌تر یا مساوی با صفر است یعنی $H(X) \geq ۰$

$$\forall i : p_i \log \frac{۱}{p_i} = -p_i \log p_i$$

$$\forall i : ۰ \leq p_i \leq ۱$$

اگر $۰ < p_i \leq ۱$ باشد، داریم $\log p_i \leq ۰$ و در نتیجه $-p_i \log p_i \geq ۰$ خواهد بود. در حالت خاص $p_i = ۰$ از حالت حدی برای تعریف مقدار $p_i \log p_i$ استفاده می‌کنیم:

$$\lim_{t \to ۰} t \log t = ۰$$

در نتیجه مجموع چند مولفه نامنفی (مثبت و یا برابر با صفر) همیشه مثبت خواهد بود.

شکل ۲ـ۱، نشان‌دهنده مقدار آشفتگی منبع تولیدکننده دو پیام با احتمال‌های p و $p - ۱$ به‌ازای مقادیر مختلف $p \in [۰, ۱]$ است. تصویر نشان‌گر این موضوع است که بیشینه مقدار آشفتگی (عدم قطعیت) منبع تولیدکننده دو پیام هنگامی است که $p = ۰.۵$ باشد، یعنی هنگامی که منبع تولیدکننده پیام‌ها یک تولیدکننده تصادفی پیام‌ها با احتمال برابر باشد مقدار آشفتگی آن بیشینه است.

آشفتگی توامان X و Y

آشفتگی بین X و Y [10] به صورت زیر تعریف می‌شود:

$$H(X,Y) = \sum_{(xy)\in X\times Y} -P(x,y)\log P(x,y) \qquad (۲-۲)$$

اگر دو متغیر تصادفی X و Y از هم مستقل باشند یعنی $P(x,y) = P(x)P(y)$ خواهیم داشت:

$$H(X,Y) = H(X) + H(Y)$$

که به آسانی با استفاده از رابطه ۲-۲ با جایگزین کردن $P(x,y)$ با $P(x)P(y)$ بدست می‌آید.

آشفتگی شرطی X با دانستن مقدار Y

آشفتگی شرطی X با دانستن مقدار Y [11] برابر است بامیانگین، روی Y، از مقدار آشفتگی X با دانستن مقدار Y و به صورت زیر تعریف می‌شود:

$$H(X|Y) = -\sum_{y\in Y} P(y)(\sum_{x\in X} P(x|y)\log P(x|y))$$

$$= -\sum_{(x,y)\in X\times Y} P(x,y)\log(P(x|y))$$

این کمیت، میزان تردید (عدم قطعیت) باقی‌مانده در مورد x هنگامی که مقدار y مشخص شود را بیان می‌کند [۷].

قاعده زنجیر در مورد میزان اطلاعات

قانون Bayes در احتمالات بیان می‌کند:

$$P(A|B) = \frac{P(A\cap B)}{P(B)}$$

[10] Joint Entropy

[11] Conditional Entropy of X given Y

و با توجه به رابطه Bayes داریم:

$$\log(P(x,y)) = \log(P(x)P(y|x)) = \log P(x) + \log P(y|x)$$

و با توجه به تعریف آشفتگی توامان X و Y داریم:

$$H(X,Y) = \sum_{(x,y)\in X\times Y} -P(x,y)\log P(x,y) = -\sum_{(x,y)\in X\times Y} P(x,y)(\log P(x)+\log P(y|x))$$

$$= -\sum_{(x,y)\in X\times Y} P(x,y)\log P(x) - \sum_{(x,y)\in X\times Y} P(x,y)\log P(y|x))$$

$$= H(X) + H(Y|X)$$

رابطه بالا بیان می‌کند که میزان تردید در اطلاعات (توامان) X و Y برابر با جمع تردید در مقدار X و تردید در مقدار Y با دانستن مقدار متغیر X می‌باشد. به راحتی می‌توان دید که $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$ [۳۰].

اطلاعات مشترک

مقدار اطلاعات مشترک[12] بین X و Y با رابطه زیر تعریف می‌شود:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

این کمیت بیان می‌کند که چه میزان از عدم قطعیت درباره X با دانستن مقدار Y کاسته می‌شود و یا به صورت معادل، چه میزان اطلاعات در مورد Y را می‌توان از X بدست آورد[۱۴]. این معیار یک معیار متقارن است و می‌توان این موضوع را با استفاده از قاعده زنجیر اثبات کرد[۷].

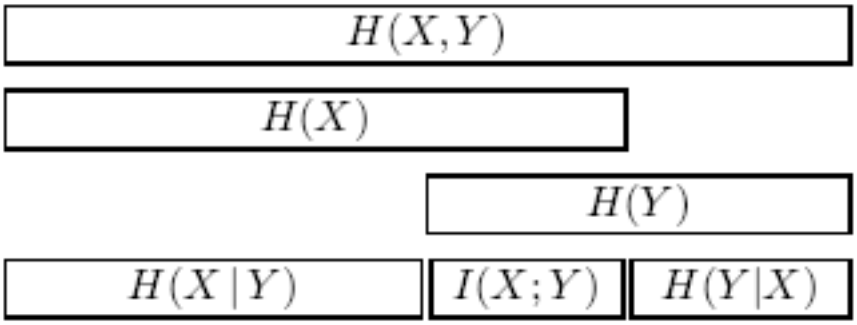
آشفتگی نسبی

مقدار آشفتگی نسبی[13] یا Kullback-Leibler divergence بین دو توزیع احتمال گسسته $p(x)$ و $q(x)$ به صورت زیر تعریف می‌شود:

$$D_{KL}(p||q) = \sum_x p(x)\log\frac{p(x)}{q(x)}$$

[12] Mutual Information

[13] Relative Entropy



شکل ۲ـ۲: رابطه بین آشفتگی توامان، آشفتگی شرطی و آشفتگی هر یک از متغیر

آشفتگی نسبی مقداری بزرگ‌تر یا مساوی با صفر دارد و تساوی فقط هنگامی اتفاق می‌افتد که $p = q$ باشد. این نکته که آشفتگی نسبی که گاهی از آن با نام "فاصله KL" یاد می‌شود کمیت متقارنی نیست قابل توجه است، یعنی در حالت کلی $D_{KL}(p||q) \neq D_{KL}(q||p)$. آشفتگی نسبی در دسته‌بندی داده‌ها و شبکه‌های عصبی نیز علاوه بر تئوری اطلاعات کاربرد دارد[۲۷]
شکل ۲ـ۲، نشان‌دهنده رابطه بین مفاهیم مطرح شده در بالا است.

۲-۲ فرآیندهای تصادفی

خاصیت مارکف

یک فرآیند تصادفی دارای **خاصیت مارکف** است اگر احتمال هر یک از حالت‌های بعدی فقط بر اساس حالت کنونی و نه بر اساس حالات گذشته قابل بیان باشد.[۱۱]
اگر بخواهیم تعریف فوق را به صورت ریاضی بیان کنیم خواهیم داشت:

$$\forall h > ۰ : p[X(t+h) = y|\ X(s) = x(s), s \leq t] = p[X(t+h) = y|\ X(t) = x(t)]$$

یک فرآیند تصادفی را مستقل از زمان می‌نامیم اگر

$$p[X(t+۱) = y|X(t) = x(t)] = p[X(۱) = y|X(۰) = x(۰)]$$

در تئوری اطلاعات، به فرایندهای تصادفی[۱۴] که دارای خاصیت مارکف[۱۵] باشند فرآیند مارکف گویند[۱۶][۳]. زنجیرهای مارکف[۱۷] نوع خاصی از فرآیندهای مارکف هستند که با زمان گسسته کار می‌کنند.

زنجیرهای مارکف

همان‌طور که اشاره شد زنجیرهای مارکف یک فرآیند تصادفی زمان-گسسته[۱۸] هستند که دارای خاصیت مارکف می‌باشند.
یک زنجیر مارکف دنباله‌ای از متغیرهای تصادفی $X_۱, X_۲, \cdots$ با خاصیت مارکف است و می‌توان آن را به صورت زیر نمایش داد:

$$p(X_{n+۱} = x|X_n = x_n, \cdots, X_۱ = x_۱, X_۰ = x_۰) = p(X_{n+۱} = x|X_n = x_n)$$

مقادیر ممکن برای متغیر X_i باید یک مجموعه شمارا باشد که بدان فضای حالت [۱۹] گفته می‌شود. برای نمایش یک زنجیرمارکف معمولا از یک گراف جهت‌دار استفاده می‌شود به طوری که یال‌ها با احتمال تغییر حالت برچسب گذاری شده است.
یکی از مثال‌های زنجیرهای مارکف، ماشین‌های دارای حالات متناهی[۲۰] است. اگر ماشین M در لحظه n در حالت x باشد آن‌گاه ماشین با احتمالی که فقط وابسته به حالت‌های x و y است و زمان در آن تاثیری ندارد به حالت y منتقل می‌شود.
یک زنجیر مارکف ایستا[۲۱] نامیده می‌شود اگر در رابطه زیر صدق کند:

$$\forall n \in N; p(X_{n+۱} = x|X_n = y) = p(X_n = x|X_{n-۱} = y)$$

[14] Stochastic Process
[15] Markov Property
[16] Markov Process
[17] Markov Chain
[18] Discrete-Time
[19] State Space
[20] Finite State Machine
[21] Stationary Markov Chain

خصوصیت‌های زنجیرهای مارکف

اگر p_{ij}^n را برابر با احتمال انتقال از حالت i به حالت j در زمان n در نظر بگیریم داریم:

$$p_{ij}^n = p(X_n = j | X_{n-1} = i)$$

و اگر زنجیر مارکف مستقل از زمان[22] باشد می‌توانیم p_{ij} را برابر با احتمال انتقال از حالت i به حالت j تعریف کنیم:

$$p_{ij} = p(X_1 = j | X_0 = i)$$

در ادامه تمرکز ما بیشتر روی **زنجیرهای مارکف مستقل از زمان** خواهد بود.
مجموع احتمال‌های خروجی از هر حالت برابر با یک است:

$$\forall\ i : \sum_k p_{ik} = 1$$

حالت j از حالت i **قابل دسترسی** است اگر حالت i با احتمال بزرگ‌تر از صفر به حالت j می‌رود یعنی:

$$p_{ij} \neq 0$$

گوییم حالات i و j با هم در ارتباط هستند اگر هم i از j قابل دسترسی باشد و هم j از i. یک مجموعه از حالات که اعضای آن دو به دو با هم در ارتباط هستند را یک **کلاس ارتباطی**[23] گوییم. کلاس ارتباطی C را بسته نامند اگر و فقط اگر احتمال ترک کردن کلاس (انتقال به حالتی که عضو C نیست) صفر باشد.
یک زنجیر مارکف را **ساده‌نشدنی**[24] می‌نامند اگر کل حالات آن در یک کلاس ارتباطی قرار داشته باشند. این بدان معناست که هر حالتی در یک چنین زنجیری از هر حالت دیگری قابل دست‌یابی است و در نتیجه در صورت حذف هر یک از حالات آن، مجموعه پیام‌های تولیدی توسط آن‌زنجیر به‌کلی تغییر می‌کند.
می‌گوییم حالت i دارای دوره تناوب k است اگر هر بازگشت دوباره به حالت i مستلزم طی مضربی از k گام باشد. برای مثال، هنگامی که برای بازگشت به حالت i مجبور باشیم تا تعداد زوجی گام برداریم، آن‌گاه i متناوب با دوره تناوب ۲ است.

$$k = gcd\{n : p(X_n = i | X_0 = i) > 0\}$$

[22] Time-Stationary Markov Chain

[23] Communicating Class

[24] Irreducible

در رابطه بالا gcd تابع محاسبه کننده بزرگترین مقسوم‌علیه مشترک است و k دوره تناوب i خواهد بود. اگر $k = ۱$ باشد حالت را **غیرنوسانی**[۲۵] می‌نامند و در غیر این‌صورت آن را تناوبی با دوره تناوب k می‌نامند.

بازگشت

به حالت i **گذرا** گویند اگر با شروع از حالت i، با احتمالی مخالف صفر هیچ‌گاه به i باز نگردیم. اگر T_i اولین باری باشد که بعد از خروج از حالت i به i بازمی‌گردیم:

$$T_i = min\{n : X_n = i | X_۰ = i\}$$

حالت i یک حالت گذرا است اگر T_i با احتمالی مخالف یک متناهی باشد یعنی:

$$p(T_i < \infty) < ۱$$

اگر حالتی گذرا نباشد می‌گوییم آن **حالت بازگشتی**[۲۶] یا **پایدار**[۲۷] است.
میانگین زمان بازگشت برای یک حالت به صورت $M_i = E[T_i]$ تعریف می‌شود. اگر M_i متناهی باشد حالت i را بازگشتی مثبت[۲۸] و در غیر این‌صورت آن‌را ناپایدار[۲۹] گویند. حالات پایدار حالاتی هستند که در تولید رشته‌های نامتناهی می‌توان بینهایت بار از آن‌ها عبور کرد. یعنی می‌توان نشان داد که یک حالت پایدار است اگر و فقط اگر

$$\sum_{n=۰}^{\infty} p_{ii}^n = \infty$$

فرآیندهای Ergodic

حالت i را Ergodic گویند اگر غیرنوسانی باشد و امیدریاضی طول بازگشت آن متناهی باشد[۲]. اگر همه حالات یک زنجیر مارکف Ergodic باشند آن زنجیر را Ergodic گویند.

[25] Aperiodic

[26] Recurrence

[27] Persistent

[28] Positive Recurrent

[29] Null Persistent

بررسی حالات دائمی و محدودکردن توزیع‌ها

اگر زنجیر مارکف مستقل از زمان باشد می‌توان آن‌را با استفاده از یک ماتریس p_{ij} که درایه i و j آن احتمال انتقال از حالت i به حالت j را نشان می‌دهد بیان کرد. بردار Π، بردار توزیع ایستایی[۳۰] برای یک زنجیر است اگر مجموع عناصر Π برابر با یک باشد و در رابطه صدق کند:

$$\Pi_j = \sum_i \Pi_i p_{i,j}$$

یک زنجیر مارکف ساده نشدنی دارای بردار توزیع ایستایی است اگر و فقط اگر همه حالات آن بازگشتی مثبت باشند:

$$\Pi = \Pi P$$

در حقیقت، بردار توزیع ایستایی Π برابر با بردار ویژه[۳۱] چپ ماتریس P با مقدار ویژه ۱ است. اگر ماتریس P دارای یک بردار ویژهΠ با مقدارویژه ۱ باشد می‌توان آن‌را با رابطه زیر بدست آورد:

$$\lim_{k\to\infty} P^k = 1\Pi$$

که در آن 1 یک بردار ستونی است که مقدار همه درایه‌های آن برابر با ۱ است.

۲-۳ فرآیندهای مارکف

نوع دیگری از فرآیندهای تصادفی، فرآیندهای مارکف گسسته است که به صورت زیر تعریف می‌شوند[۲۶]: مجموعه متناهی از حالات مانند $S_1, S_2, \cdots, S_n$ در کنارمجموعه‌ای از احتمالات جابه‌جایی مانند آن‌چه در زنجیرهای مارکف وجود دارد تشکیل می‌شود. برای تبدیل این فرآیند مارکف به یک منبع اطلاعات، فرض می‌شود که هر جابه‌جایی از یک حالت به حالت دیگر (یال در گراف متناظر) یک سمبل تولید کند. در بخش ۵ به صورت دقیق‌تر به بررسی این گونه فرآیندهای مارکف می‌پردازیم.

[30] Stationary Distribution

[31] Eigenvector

۴-۲ رابطه بین دسته‌بندی و Entropy

هدف نهایی در دسته‌بندی داده‌ها این است که روشی برای تشخیص تعلق و یا عدم تعلق یک داده به هر یک از دسته‌ها ارائه شود. راه‌های مختلفی برای این منظور وجود دارد که به بخشی از آن‌ها در فصل دوم اشاره شد. در دسته‌بندی معمولا تعدادی از اعضای یک دسته در اختیار است که در این صورت یکی از روش‌های تعیین عضویت می‌تواند استفاده از میزان شباهت بین داده با اعضای هر دسته باشد. به این ترتیب که با استفاده از یک معیار به سنجش میزان شباهت داده با اعضای هر یک از دسته‌ها بپردازیم و مشابه‌ترین دسته را به عنوان دسته برنده اعلام کنیم.
همان‌طور که دیدیم، یکی از راه‌های محاسبه میزان شباهت استفاده از مفهوم مشابه میزان تردید (عدم قطعیت) است. در بخش بعدی به توضیح چگونگی استفاده از این مفهوم برای دسته‌بندی خواهیم پرداخت.

۳ دسته‌بندی داده‌ها

مساله دسته‌بندی داده[۳۲] یکی از مساله‌های اساسی در هوش‌مصنوعی[۳۳] است که سال‌ها مورد مطالعه قرار داشته است و راه حل‌های متعددی برای حل این مساله ارائه شده است. در این بخش، ابتدا تلاش خواهیم کرد تا مساله را به صورت ریاضی بیان کنیم و سپس به توضیح چند روش معمول در این زمینه بپردازیم.

۳-۱ تعریف ریاضی مساله دسته‌بندی داده‌ها

مساله دسته‌بندی داده‌ها درباره حدس زدن و یا پیش‌بینی یک دسته برای یک داده ورودی است. داده ورودی معمولا یک مجموعه از اعداد و یا داده‌های نمادین[۳۴] است که به صورت یک بردار dـتایی مانند x نمایش داده می‌شود ولی ممکن است در برخی کاربردها داده ورودی به صورت یک تصویر و یا یک نمودار نیز باشد. دسته‌ای که x بدان تعلق دارد را با y نشان می‌دهیم. در ساده‌ترین حالت، y یکی از دو مقدار $\{۱-, ۱\}$ را اختیار می‌کند[۴].
در این پایان‌نامه توجه اصلی به دسته‌بندی داده‌ها بین دو کلاس(دسته‌بندی دوگانه [۳۵]) خواهد بود و از کلمه دسته‌بندی به معنای دسته‌بندی دوگانه استفاده شده است. به راحتی می‌توان با استفاده از یک ساختار سلسله مراتبی، از دسته‌بندی دوگانه برای حل هر مساله دسته‌بندی دیگری استفاده کرد. دلیل این انتخاب ساده‌تر بودن دسته‌بندی دوگانه ولی در عین حال حفظ قابلیت کلی مساله‌های دسته‌بندی در حالت عمومی است.
در دسته‌بندی به دنبال ایجاد تابعی به صورت $g : X \to \{-۱, ۱\}$ هستیم که دسته پیشنهادی برای داده(بردار) ورودی x، به صورت $y = g(x)$ محاسبه شود. به چنین تابعی، **تابع دسته‌بندی‌کننده**[۳۶] می‌گوییم. در برخی از تابع‌های دسته‌بندی کننده برد تابع را به $[-۱, ۱]$ (به جای $\{-۱, ۱\}$) تغییر می‌دهند. خطای یک تابع دسته‌بندی کننده روی یک مجموعه از داده‌ها ($\{(x_i, y_i)\}$) به صورت زیر تعریف می‌شود:

$$Err(g, S) = \sum_{(x_i, y_i) \in S} |\frac{g(x_i) - y_i}{۲}|$$

[32] Data Classificatoin

[33] Artificial Inteligene

[34] Symbolic Data

[35] Binary Classifcation

[36] Classifier function

برای بیان مساله یادگیری، از یک مدل احتمالاتی استفاده می‌کنیم که در آن (X,Y) یک زوج تصادفی از $X \times \{-۱, ۱\}$ شامل داده و دسته مربوط به آن است. علت استفاده از مدل تصادفی این است که ممکن است یک $X = x$ به هر دو دسته $Y = \{-۱, +۱\}$ متناظر شود. یکی از دلیل‌های این حالت می‌تواند کافی نبودن اطلاعات ویژگی‌های انتخابی باشد. به عنوان مثال، حالتی را در نظر بگیرید که دسته یک مجموعه از داده‌ها توسط سه ویژگی به طور کامل مشخص شود ولی ما فقط از دو تا از این ویژگی‌ها در X استفاده کرده‌ایم. دلیل دیگر برای این حالت، می‌تواند طبیعت احتمالاتی مساله باشد. توزیع احتمال زوج (X,Y) را می‌توان براساس توزیع احتمال X (توسط $P(X)$) و $\eta(x) = P\{Y = ۱|X = x\}$. می‌توان خطای کارآیی تابع g را بر اساس تابع توزیع احتمال خطا اندازه‌گیری کرد:

$$L(g) = P\{g(x) \neq Y\}$$

در صورتی که تابع η یا تخمین مناسبی از آن در دسترس باشد می‌توان با استفاده از تصمیم‌گیری حریصانه[۳۷] یک تابع دسته‌بندی کننده با کمترین احتمال خطا ارائه داد:

$$g^*(x) = \begin{cases} 1 & \text{if } \eta(x) > \frac{1}{2} \\ -1 & \text{otherwise} \end{cases}$$

به راحتی می‌توان اثبات کرد:

$$\forall g : X \mapsto \{-۱, ۱\} : L(g^*) \leq L(g)$$

کمترین مقدار خطا $L^* = L(g^*)$ را خطای Bayes[۳۸] می‌نامیم[۹]. به صورت دقیق‌تر، می‌توان ثابت کرد[۶] که

$$L(g) - L^* = E[۱_{g(X) \neq g^*(X)} | ۲\eta(X) - ۱ \geq ۰]$$

معمولا دسته‌بندی بر اساس تابع g^* را، دسته‌بندی براساس روش $Bayes$[۳۹] می‌نامند. در روش‌های آماری[۴۰] عموما یک مجموعه از داده‌ها $D_n = \{(X_i, Y_i) : ۱ \leq i \leq n\}$ را در اختیار داریم. فرض می‌کنیم که داده‌های مجموعه D_n توسط یک توزیع تصادفی مستقل[۴۱] از بین زوج‌های ممکن از $X \times \{-۱, ۱\}$ استخراج می‌شود و در نتیجه $(X_۱, Y_۱), \cdots, (X_n, Y_n)$ با دقت خوبی دارای توزیعی مانند (X,Y) می‌باشد.

[37] Greedy

[38] Bayes Error

[39] Bayes Classifier

[40] Statistical Model

[41] Indepdent Identically Distribution

تابع دسته‌بندی کننده‌ای که بر پایه $D_n = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ بدست می‌آید را g_n می‌نامیم. کارایی تابع g_n را می‌توان بر پایه احتمال خطای آن محاسبه کرد:

$$L(g_n) = P\{g_n(X) \neq Y | D_n\}$$

توجه اصلی در تئوری و کاربردهای عملی دسته‌بندی، ایجاد تابع g_n به طوری است که احتمال خطای آن نزدیک‌ترین مقدار ممکن L^* است.
یک راه ساده برای مساله دسته‌بندی داده‌ها این است که یک کلاس مانند C از تابع‌های دسته‌بندی‌کننده مانند $g : X \mapsto \{-1, 1\}$ تشکیل دهیم و سپس با استفاده از D_n یک تخمین برای احتمال خطای هر $g \in C$ یعنی $L(g)$ حساب کرده و بر اساس آن یک تابع را انتخاب کنیم. شاید بدیهی‌ترین راه برای تخمین احتمال خطای $L(g) = P\{g(X) \neq Y\}$ استفاده از

$$L_n(g) = \frac{1}{n} \sum \{1_{g(X_i) \neq Y_i}\}$$

$L_n(g)$ را خطای تجربی[۴۲] می‌نامیم.
هدف در الگوریتم‌های دسته‌بندی مبتنی بر یادگیری ماشین[۴۳] تولید توابعی است که می‌توانند هر نمونه از داده‌های ورودی را به یک دسته مناسب نظیر کنند. روش‌های گوناگون یادگیری ماشین برای کاربردهای گوناگون براساس مدل‌های احتمالاتی و یا آماری و الگوریتم‌های گوناگون ارائه شده است. در بین این روش‌ها، شبکه‌های عصبی[۴۴] در کاربردهایی که در آن عناصر بردارهای اعداد حقیقی هستند، و درخت‌های تصمیم[۴۵] ، Naive Bayes در کاربردهایی که عناصر بردار ورودی آن نمادین هستند از معروف‌ترین روش‌ها هستند[۱۰]. در ادامه به توضیح مختصر دو روش اول می‌پردازیم و روش آخر (Naive Bayes) را به علت ساختار ریاضی آن و رابطه نزدیکی که با روش مورد بحث در این پایان‌نامه دارد را به طور دقیق‌تر بررسی خواهیم کرد.
تقریبا همه الگوریتم‌های دسته‌بندی دارای دو مرحله هستند:

۱. مرحله آموزش: در این مرحله یک مدل (بسته به الگوریتم مورد استفاده) انتخاب می‌شود و براساس مجموعه‌های آموزش پارامترهای مدل مربوطه تعیین می‌شود. این مرحله، معمولا فقط یک‌بار اجرا می‌شود و بارها از نتیجه آن استفاده می‌شود بنابراین پیچیدگی محاسبه در این مرحله، زیاد مورد توجه قرار نمی‌گیرد.

[42] Empirical Error

[43] Machine Learning

[44] Neural Network

[45] Decision Tree

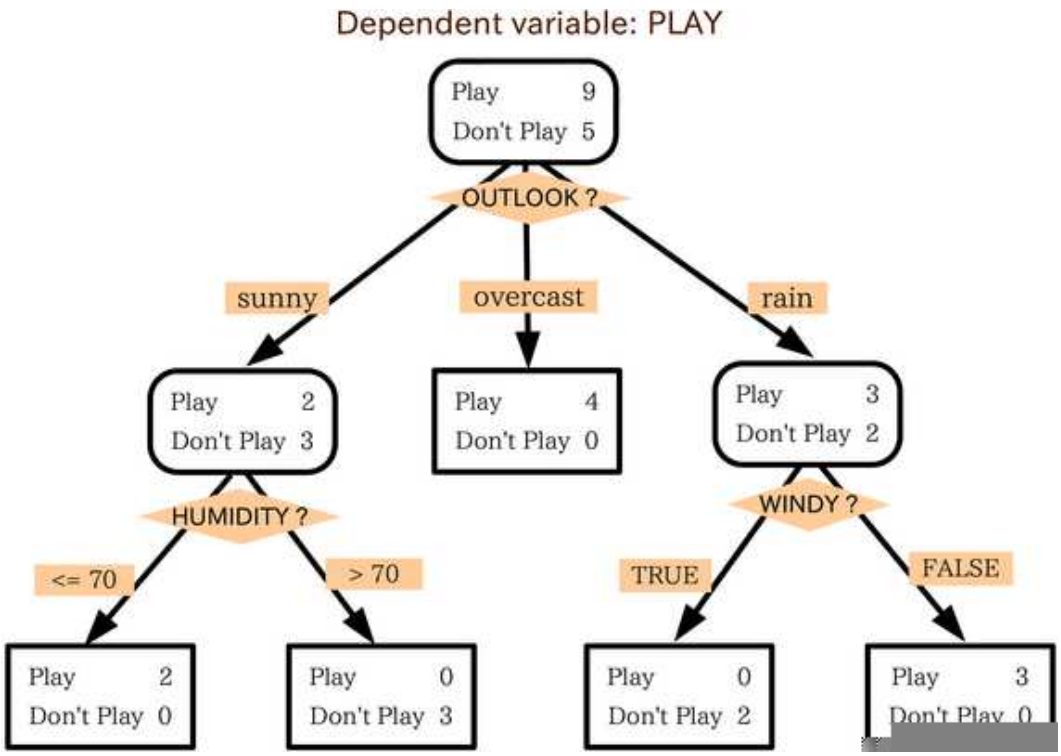
۲. مرحله استفاده: در این مرحله، از مدل و پارامترهای بدست آمده در مرحله آموزش استفاده شده و با به‌کارگیری آن‌ها، در مورد بردار ورودی اظهار نظر می‌شود. برخلاف مرحله قبل، این مرحله بارها مورد استفاده قرار می‌گیرد و بنابراین پیچیدگی محاسبه آن بسیار مهم است.

۳-۲ درخت‌های تصمیم

درخت‌های تصمیم دارای یک ساختار درختی هستند که راس‌های هر سطح آن متناظر با یک ویژگی بوده و دارای دو و یا بیشتر فرزند است که بسته به مقدار آن ویژگی در بردار ورودی یک و یا چند تا از فرزندان انتخاب می‌شوند و در نهایت برگ‌ها به یکی از دسته‌ها مربوط می‌شوند.
در مرحله آموزش این روش، ابتدا ساختار درخت (ترتیب قرارگرفتن ویژگی‌ها در درخت) تعیین می‌شود و سپس با استفاده از مجموعه‌آموزشی دسته‌هایی که برگ‌ها بدان‌ها باید اشاره کنند تعیین می‌شود. پیچیدگی درخت تصمیم تولید شده (تعداد راس های درخت) بسیار وابسته به ساختار درخت است. روش‌های ابدایی[۴۶] برای تعیین ترتیب قرارگرفتن ویژگی‌ها در درخت تصمیم وجود دارد ولی هیچ روش اثبات شده بهینه‌ای در این رابطه وجود ندارد.
در هنگام استفاده از یک درخت تصمیم، با شروع از ریشه در هر راس، براساس مقدار ویژگی اختصاص داده شده به راس، به یکی از فرزندان آن راس می‌رویم. و این روند را تا رسیدن به یک برگ ادامه می‌دهیم و دسته متناظر با برگ و یا برگ‌های متناظر به عنوان دسته(دسته‌های) مربوطه گزارش می‌شود[۲۴].
مزیت اساسی درخت‌های تصمیم بر سایر الگوریتم‌های دسته‌بندی، در کم بودن هزینه محاسبه برای انتخاب دسته پیشنهادی و قابلیت کار با داده‌های عددی و نمادین است و همچنین در حالاتی که داده دارای ویژگی‌های گم‌شده[۴۷] باشد درخت‌های تصمیم کارآیی خوبی دارند.
یکی دیگر از امتیازات درخت‌های تصمیم قابلیت نمایش ساختارهای تصمیم پیچیده در آن‌ها است. شکل ۳ـ۱، یک نمونه از درخت‌های تصمیم را نشان می‌دهد.
یکی از بهترین الگوریتم‌هایی که برای تولید درخت‌های تصمیم به‌کار گرفته می‌شود الگوریتم C4.5[۲۰] است.

[46] Hueristic

[47] Missing Variables



شکل ۳ـ۱: یک نمونه از درخت‌های تصمیم برای بازی گلف

۳-۳ شبکه‌های عصبی

شبکه‌های عصبی مصنوعی[۴۸] که معمولا با نام شبکه عصبی[۴۹] شناخته می‌شوند از یک گروه به هم متصل از سلول‌های عصبی مصنوعی[۵۰](نرون) تشکیل شده و از یک مدل ریاضی یا محاسباتی برای پردازش اطلاعات بر حسب اتصالات بین نرون‌ها استفاده می‌کند. در بیشتر مواقع شبکه عصبی مصنوعی یک سامانه سازگارشونده[۵۱] است که ساختار آن در طول فرآیند بر اساس اطلاعات داخلی و یا خارجی تغییر می‌کند.

در کاربرد، شبکه عصبی یک ابزار برای مدل‌سازی داده‌های آماری غیر خطی است. از آن می‌توان برای مدل کردن رابطه‌های پیچیده بین ورودی‌ها و خروجی‌ها و یا پیدا کردن الگو استفاده کرد.

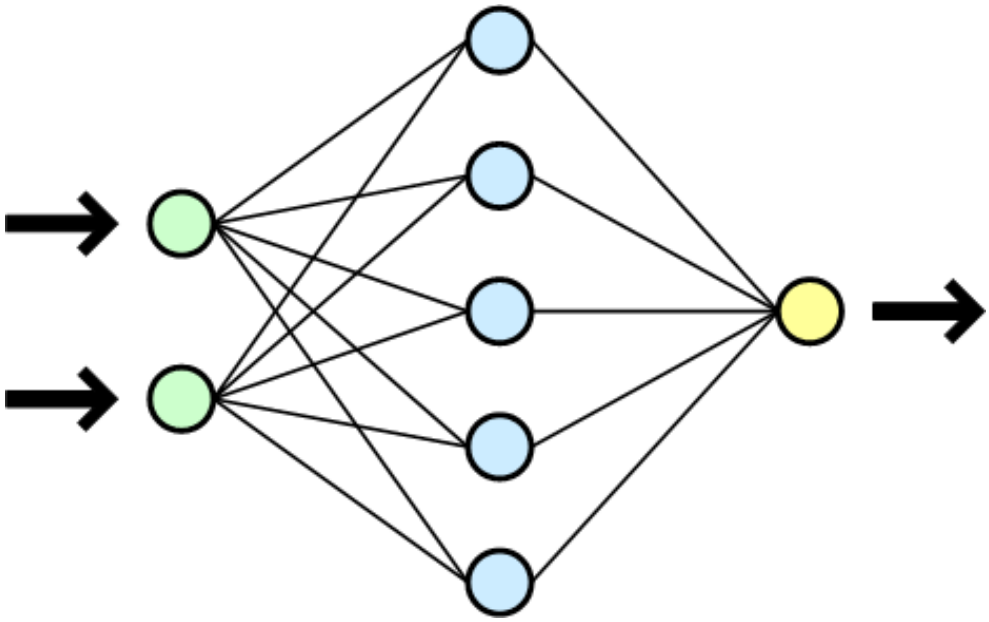
تعریف دقیق و مورد قبول عمومی برای شبکه عصبی وجود ندارد. یکی از تعاریف به صورت زیر است: شبکه عصبی از یک شبکه از عناصر پردازش‌گر ساده(Neurons) تشکیل شده است

[48] Artificial Neural Network

[49] Neural Network

[50] Artificial Neurons

[51] Adaptive System



شکل ۳ـ۲: یک نمونه شبکه عصبی با سه لایه

و می‌تواند ساختارها و رفتارهای پیچیده را با چگونگی ارتباط بین عناصر پردازش‌گر شبکه و داده‌های ورودی و خروجی بیان کرد. ایده اصلی در شبکه‌های عصبی مصنوعی از سیستم عصبی انسان گرفته شده است که در آن فعالیت‌ها به صورت جمعی و به صورت موازی بوسیله هر واحد ساده انجام می‌شود[۱۳].

شکل ۳ـ۲ یک سامانه شبکه عصبی را نشان می‌دهد. هر نرون دارای یک و یا چند یال ورودی و دقیقا یک یال خروجی است. مقدار خروجی هر نرون به صورت زیر محاسبه می‌شود:

$$\sum_{i=۱}^{n} g(w_i I_i)$$

که در آن، هر w_i یک عدد حقیقی است که به هریال ورودی نسبت داده می‌شود و I_i مقدار ورودی iـام است. در حقیقت w_i ها مقدار اثر هر ورودی را تعیین می‌کنند و از این رو معمولا w_i را «وزن» نیز می‌نامند. و g تابع فعالیت [۵۲] نرون است. تابع فعالیت نرون‌ها می‌تواند هر تابع تام از اعداد حقیقی به به بازه $[-۱,+۱]$ باشد ولی معمولا از تابع Sigmoid به شکل $\frac{۱}{۱+e^{-at}}$ استفاده می‌شود[۱۳].

معمولا در شبکه‌های عصبی، هر نرون در لایه خروجی نشان‌گر یک دسته است و دسته پیشنهادی توسط یک شبکه عصبی دسته‌ای است که خروجی نرون مربوطه (در حالاتی که

[52] Acivation Function

چند نرون به یک کلاس مربوط شده اند بیشینه مقدار خروجی نرون‌ها) بیشترین باشد. شبکه‌های عصبی بر حسب نوع ارتباطاتی که نرون‌ها می‌توانند با هم داشته باشند و همچنین بر حسب تابع فعالیت نرون‌ها به دسته‌های مختلفی تبدیل می‌شوند. شبکه‌های عصبی بازخورد خطا[۵۳][۳۴] از معروف‌ترین شبکه‌های عصبی می‌باشند.

۴-۳ Naive Bayes

روش استدلال Bayes[۵۴] از روش‌های آماری برای استنتاج استفاده می‌کند. در این روش فرض این است که ویژگی‌های[۵۵] مورد علاقه بر اساس توزیع احتمالی که وجود دارد ولی احتمالا ما از آن بی‌اطلاع هستیم قابل توصیف هستند و می‌توان بر اساس یک تخمین از توزیع احتمال (که بر اساس مجموعه آموزشی به‌دست می‌آید) تصمیم بهینه گرفت. روش‌هایی که بر این اساس کار می‌کنند را الگوریتم‌های یادگیری Bayes[۵۶] می‌نامند. در بین الگوریتم‌های یادگیری Bayes، می‌توان به روش Naive Bayes اشاره کرد که به علت سادگی پیاده‌سازی بسیار مورد توجه است. این الگوریتم در کنار سادگی، از کارآیی بسیار خوبی مخصوصا در زمینهٔ دسته‌بندی متون[۵۷] برخوردار است[۲۱]. در [۱۷]، به مقایسه کارآیی دسته‌بندی با استفاده از Naive Bayes با سایر روش‌ها از جمله درخت‌های تصمیم و شبکه‌های عصبی پرداخته شده است. آمار ارائه شده در این مقاله نشان می‌دهد که Naive Bayes یک رقیب خوب برای سایر الگوریتم‌های دسته‌بندی داده‌ها است و حتی در بعضی از مساله‌ها، بهتر از سایر روش‌ها عمل می‌نماید.
تئوری Bayes در نظریه احتمال یک روش برای محاسبه کردن احتمال رخ‌دادن h به شرط اتفاق افتادن D یعنی $P(h|D)$ با استفاده از احتمال احتمال رخ‌دادن D به شرط اتفاق افتادن h یعنی $P(D|h)$ و احتمال رخ‌دادن D و h به صورت زیر ارائه می‌دهد[۳۱]:

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

برای بیان الگوریتم Naive Bayes و سایر الگوریتم‌های وابسته به روش Bayes، در جزئیات مساله دسته‌بندی باید تغییراتی ایجاد کنیم . فرض کنید به ازای هر دسته مانند

[53] Back Propagation

[54] Bayesian Resoning

[55] Features

[56] Bayesian Learning Methods

[57] Document Classification

$c \in \{c_1, \cdots, c_m\}$، یک بردار (نامعلوم) θ^c وجود دارد که داده‌های متعلق به آن دسته را به صورت مستقل تولید می‌کند. داده‌هایی که دسته مربوط به آن‌ها شناخته شده است مجموعه آموزشی (به ازای هر دسته مانند c مجموعه مانند D^c) نامیده می‌شوند و بقیه داده‌های ممکن (داده‌های مجاز) مجموعه آزمون را تشکیل می‌دهند[۲۱].
پیدا کردن (تخمین زدن) مقادیر پارامتر مجهول یکی از مهم‌ترین بخش‌های روش Naive Bayes است. روش‌های متفاوتی برای برآورد کردن این بردار (θ^c) وجود دارد که در ادامه به بیان جزئیات دو روش پرکاربردتر خواهیم پرداخت:

ML Naive Bayes

منطقی‌ترین راه برای تخمین‌زدن بردار θ^c استفاده از تنها داده شناخته‌شده یعنی D^c است. در روش بیشترین احتمال[۵۸]، فرض بر این است که هیچ دانشی در مورد چگونگی توزیع احتمال دسته‌ها در دست نیست و بنابراین توزیع یکنواخت برای آن در نظر گرفته می‌شود. بردار θ^c توسط بردار $\hat{\theta^c}$ چنان تخمین زده می‌شود که با بیشترین احتمال ممکن مجموعه آموزشی را تولید کند یعنی:

$$\hat{\theta_c} = \arg\max_{\theta} p(D^c|\theta)$$

و سپس از بردار $\hat{\theta^c}$ به جای θ^c استفاده می‌شود. این روش دارای نقاط ضعفی است که شاید از مهمترین آن‌ها، وابستگی بسیار زیاد به چگونگی انتخاب مجموعه آموزش است. برای مثال، اگر یک ویژگی در مجموعه آموزش پدیدار نشده باشد در این روش فرض می‌شود که ویژگی مذکور در هیچ یک از داده‌های عضو آن دسته، ظاهر نخواهد شد. برای بدست آوردن احتمال تولید مجموعه آموزشی با فرض در اختیار بودن بردار θ می‌توان از روابط زیر استفاده کرد:

$$P(D^c|\theta) = \frac{N^c!}{\prod_k N_k^c!} \prod_k \theta_k^{N_k^c}$$

که در آن N_k^c نشان‌دهنده تعداد دفعاتی است که ویژگی k (f_k) در مجموعه آموزشی مربوط به دسته c ظاهر شده و $N^c = \sum_k N_k^c$ برابر با کل ویژگی‌هایی است که (با در نظر گرفتن تکرار) در D^c وجود دارد. فرض کنید θ_k، kـامین مولفه بردار θ باشد و نشان‌دهنده احتمال پدیدار شدن ویژگی kـام در هر مولفه از داده‌های عضو D^c باشد. در تخمین ML بر اساس D^c (مقداری برای $\hat{\theta^c}$ که احتمال $p(D^c|\hat{\theta^c})$ را بیشینه می‌کند) به ازای تمام kها، $\hat{\theta_k^c} = \frac{N_k^c}{N^c}$ خواهد بود.

[58] Maximum Likelihood

همان‌طور که اشاره شد Naive Bayes، از $\hat{\theta^c}$ برای ارزیابی این که آیا داده آزمون، d، به دسته c تعلق دارد (توسط تولید کننده دسته c تولید شده است) استفاده می‌کند. از آن‌جایی که تخمین بردار پارامتر و در نتیجه قضاوت در مورد داده‌های آزمون بسیار وابسته به چگونگی انتخاب داده‌های مجموعه آموزشی، D^c، است این مجموعه(D^c) می‌تواند تاثیر زیادی روی کارآیی داشته باشد.
انتخاب جواب بهینه طبق رابطه Bayes چنین خواهد بود:

$$H_{ML}(d) = \arg\max_c \ p(c|D,d) = \arg\max_c \ p(d|\hat{\theta^c})p(c)$$

اگر همه دسته‌ها، دارای شانس تقریبا برابر باشند (یعنی $p(c)$ دارای توزیع یکسان[۵۹] باشد)، فرآیند انتخاب دسته جواب را می‌توان به صورت زیر ساده کرد:

$$H_{ML}(d) = \arg\max_c \ p(d|\hat{\theta^c}) = argmax_c \ \prod_k (\frac{N_k^c}{N^c})^{N_k^d}$$

N_k^d برابر است با تعداد دفعاتی که ویژگی k‌ـام در داده آزمون (d) ظاهر شده است. همان‌طور که اشاره شد در بسیاری از کاربردها، ممکن است N_k^c برای یک یا چند ویژگی صفر باشد و در این صورت مقدار $P(d|c)$ برابر با صفر خواهد بود و ممکن است به گمراهی H_{ML} منتهی شود. برای جلوگیری از این حالت، به یک سامانه Naive Bayes علاوه بر داده‌های آموزشی، یک دنباله از اعداد a_k به عنوان ورودی داده می‌شود و از این دنباله که a_k با ویژگی k‌ـام در ارتباط است و ($a = \Sigma a_k$) به صورت زیر استفاده می‌شود:

$$H_{ML}(d) = \arg\max_c \prod (\frac{N_k^c + a_k}{N^c + a})$$

به این روش (اضافه کردن یک دنباله از اعداد به Naive Bayes) M-Estimation گفته می‌شود[۲۹]. در کاربرد، تعیین این دنباله به صورت بهینه کار غیرممکنی است و بنابراین تنها از این خاصیت رابطه جدید، که در صورت برابر با صفر بودن N_k^c، $P(d|c)$ صفر نخواهد بود استفاده می‌شود (با انتخاب a_k ها به صورت پیش‌فرض). در بیشتر به‌کارگیری‌های M-Estimation، از دنباله یکنواخت ($\forall\ i,j : a_i = a_j$) استفاده می‌شود که در آن $a_i = ۱$ است[۲۱].

MAP Naive Bayes

ML Naive Bayes با فرض این‌که هیچ دانشی از چگونگی توزیع احتمال دسته‌ها در دسترس نیست کار می‌کند ولی اگر چنین دانشی موجود باشد آن‌گاه می‌توان از روش MAP Naive

[59] Uniform Distribution

Bayes[۶۰] استفاده کرد.MAP estimation یک گسترش از ML است و مانند ML Bayes کار می‌کند یعنی بازهم بهترین $\hat{\theta^c}$ بر حسب داده‌های مجموعه آموزشی انتخاب کرده و سپس از آن و توزیع احتمال دسته‌ها برای دسته‌بندی استفاده می‌کنیم.
در MAP estimation، θ^c توسط $\hat{\theta^c}$ (مشابه ML) تخمین زده می‌شود:

$$\hat{\theta^c} = \ argmax_\theta \ p(D^c|\theta)$$

فرآیند تصمیم در MAP به صورت زیر است:

$$H_{MAP}(d) = argmax_c \ p(c|d) = argmax_c \ P(d|\theta^c)p(c)$$

که در آن $p(c)$ احتمال دسته c است.
برای روشن‌تر شدن چگونگی کاربرد تئوری Bayes در دسته‌بندی و همچنین روش‌های ML و MAP در تخمین پارامترها و استدلال، می‌توان به مساله زیر اشاره کرد[۱۸]
فرض کنید در مساله تشخیص یک نوع سرطان یکی از دو فرض "(۱) مراجعه‌کننده به نوع خاصی از سرطان دچار است" و یا "(۲) مراجعه‌کننده دچار بیماری سرطان نیست" برقرار باشد. داده‌های موجود در این مساله از یک آزمایشگاه به صورت نتیجه یک آزمون دارای دو حالت مثبت ($\oplus$) و یا منفی ($\ominus$) است، و همچنین آزمون انجام شده در آزمایشگاه، یک نشان‌گر غیردقیق از وجود بیماری است. از میان ۱۰۰۰ آزمایش‌شونده در حالتی که نتیجه آزمون مثبت است ۹۸۰ نفر واقعا بیمار بوده اند و هنگامی که آزمون جواب منفی می‌دهد ۹۷۰ نفر سالم بوده‌اند. و در سایر موارد پاسخ آزمون نادرست است.

$$\begin{aligned} &\mathrm{P}(\oplus|\ \text{cancer}) = 0.98 \qquad \mathrm{P}(\ominus|\ \text{cancer}) = 0.02 \\ &\mathrm{P}(\oplus|\ \neg\ \text{cancer}) = 0.03 \quad \mathrm{P}(\ominus|\ \neg\ \text{cancer}) = 0.97 \end{aligned}$$

۱ـ فرض کنید بیماری وجود دارد که نتیجه آزمون او مثبت بوده است. آیا این بیمار سرطان دارد؟
۲ـ اگر بدانیم تنها ٪۰/۸ از افراد جامعه ممکن است مبتلا به این‌گونه از سرطان باشند، آیا بیماری با آزمون مثبت، سرطان دارد؟
در این مساله فقط یک ویژگی (نتیجه آزمون که می‌تواند $\oplus$ و یا $\ominus$ باشد) و دو دسته ($cancer$ و $\neg cancer$) وجود دارد و داده آزمون به صورت $d =< \oplus >$ است.
۱ـ در حالت اول، دانشی در مورد توزیع احتمال دسته‌ها وجود ندارد و برای حل مساله از ML استفاده کنیم:

[60] Maximum A Posteriori

$p(d|caner) = p(\oplus|cancer) = 0.98,\ p(d|\neg caner) = p(\oplus|\neg cancer) = 0.3$

$\Rightarrow H_{ML}(d) = cancer$

۲ـ در این حالت، دانشی از چگونگی توزیع احتمال دسته‌ها وجود دارد و بنابراین باید (بهتر است) از ML استفاده شود.

P(cancer)= 0.008	P($\neg$ cancer)= 0.992
P($\oplus$| cancer)= 0.98	P($\ominus$| cancer)= 0.02
P($\oplus$| $\neg$ cancer)= 0.03	P($\ominus$| $\neg$ cancer)= 0.97

احتمال مبتلا بودن مراجعه کننده با نتیجه آزمون مثبت ($d =< \oplus >$) به سرطان طبق MAP از رابطه زیر بدست می‌آید:

$$H_{cancer}(d) = p(d|\theta^{cancer})p(cancer) = ۰/۹۸ * ۰/۰۰۸ = ۰/۰۰۷۸$$

و احتمال مبتلا نبودن وی به سرطان نیز از رابطه مشابه زیر بدست می‌آید:

$$H_{\neg cancer}(d) = p(d|\theta^{\neg cancer})p(\neg cancer) = ۰/۰۳ * ۰/۹۹۲ = ۰/۰۲۹۸$$

حال اگر بخواهیم با استفاده از Naive Bayes، داده d را به یکی از دو دسته اختصاص دهیم باید اعلام کنیم که مراجعه کننده سالم است.

۴ بررسی تخمین‌های پارامترهای Naive Bayes

Naive Bayes یک الگوریتم حریصانه است که براساس احتمال عضویت داده در کلاس تصمیم می‌گیرد. برای انجام این‌کار، Naive Bayes نیاز به تخمین زدن پارامترهای هر کلاس (که در بخش قبل بدان‌ها اشاره شد) دارد و کیفیت تخمین پارامترها به طور مستقیم بر کارآیی الگوریتم تاثیر دارد. در این بخش نشان خواهیم داد که تخمین‌های به‌کار رفته در Naive Bayes پایدار[۶۱] هستند و سپس کارکرد روش‌های تخمین را روی داده‌های آموزشی متناهی با تحلیل جابه‌جایی[۶۲] و انحراف از معیار[۶۳] بررسی خواهیم کرد. نشان خواهیم داد که جابه‌جایی در تخمین با افزایش داده‌های آموزشی، به سمت صفر میل خواهد کرد. تحلیل نشان خواهد داد که در صورت عدم وجود مجموعه آموزشی به اندازه کافی گسترده در یک دسته، می‌تواند روی کارآیی کل سامانه تاثیر داشته باشد. انحراف از معیار سامانه برابر با حاصل جمع انحراف هر یک از مولفه‌های آن سامانه است. در نتیجه اگر یک دسته دارای انحراف بالایی باشد، مقدار انحراف کل سامانه نیز بزرگ خواهد بود.[۲۱].

۴-۱ پایداری

پایداری بدین معناست که با افزایش اندازه مجموعه آموزش، مقدار تخمین‌زده شده به مقدار واقعی آن میل می‌کند. ML، m-estimation و MAP سعی می‌کنند که با استفاده از $\{a_k\}$ (در ML همه آن‌ها برابر با صفر هستند) تخمین $\hat{\theta}$ از θ را به صورت زیر تولید کنند:

$$\hat{\theta_k^c} = \frac{a_k + N_k^c}{a + N^c}$$

Cover و Thomas در کتاب "Elements of Information Theory"[۵] روشی را برای توصیف خصوصیات توزیع‌های تجربی شرح می‌دهند. فرض کنید X یک متغیر تصادفی با پارامترهای θ_k^c باشد و p_x نشان‌گر توزیع پارامترهای X و p_y نشان‌گر توزیع احتمال هنگامی که N^c نمونه برداشته شده از X دارای تعداد تکرار $\{N_k^c\}$ باشند. در این حالت $\hat{\theta_k^c} = \frac{a_k+N_k^c}{a+N^c}$ خواهد بود.
احتمال مشاهده یک‌چنین $\{N_k^c\}$ و در نتیجه ایجاد تخمین $\hat{\theta}$ برابر است با:

$$p(\hat{\theta^c}|\theta^c) = \frac{N^c!}{\Pi_k N_k^c!}\Pi_k(\theta_k^c)^{N_k} = \frac{N^c!}{\Pi_k N_k^c!}۲^{-N(H(p_Y)+D(p_Y||p_X))}$$

[61] Consistent

[62] Bias

[63] Variance

میانگین تخمین به روش بالا برابر با $\frac{a_k+N^c\theta_k^c}{a+N^c}$ خواهد بود و با بزرگ شدن N^c $(N^c \to \infty)$ به θ_k^c میل می‌کند. همچنین انحراف از معیارتخمین نیز از رابطه زیر بدست می‌آید:

$$\delta_{c,k}^{۲} = \frac{N^c\theta_k(۱-\theta_k^c)}{(a+N^c)^{۲}}$$

$\delta_{c,k}^{۲}$ با بزرگ شدن N^c به سمت صفر میل می‌کند. پس تخمین حاصل از روش‌های فوق پایدار هستند زیرا در حالت حدی، به سمت مقدار دقیق میل می‌کند و دارای انحراف از معیار صفر است. در نتیجه با افزایش تعداد داده‌های عضو مجموعه‌آموزشی، تخمین ما به مقدار واقعی نزدیک‌تر می‌شود.

۴-۲ جابه‌جایی

از آن‌جایی که در کاربردهای عادی، همیشه اندازه مجموعه‌آموزشی متناهی است، درک چگونگی رفتار روش‌های تخمین پارامترهای Naive Bayes مهم است.
جابه‌جایی به صورت اختلاف بین مقدار تخمین زده شده و مقدار واقعی تعریف می‌شود یعنی

$$bias(\hat{\theta_k^c}) = \hat{\theta_k^c} - \theta_k^c = \frac{a_k+N^c\theta_k^c}{a+N^c} - \theta_k^c = \frac{a_k - a\theta_k^c}{a+N^c}$$

در رابطه بالا، اگر $\theta_k^c > \frac{a_k}{a}$، امید ریاضی مقدار تخمین زده شده از مقدار واقعی کوچک‌تر خواهد بود و اگر $\theta_k^c < \frac{a_k}{a}$، امید ریاضی مقدار تخمین زده شده از مقدار واقعی بزرگ‌تر خواهد بود. ولی به هرحال با رشد N^c، مقدار bias به سمت صفر میل می‌کند.

۴-۳ انحراف از معیار

فرض کنید که N_k^d تعداد تکرار ویژگی kـام در داده‌آزمون (d) باشد و $N^d = \Sigma_k N_k^d$ آن‌گاه:

$$z_c = -N^d \log(a+N^c) + \Sigma N_k^d \log(a_k + N_k^c)$$

z_c برابر است با لگاریتم امتیازی که داده‌آزمون d از دسته c کسب می‌نماید. d به دسته‌ای که در آن بیشترین امتیاز را آورده باشد منصوب می‌شود. جمع‌های بکار رفته در عبارت بالا از هم مستقل هستند. با توجه به این که $\{N_k^d\}$ برای یک داده‌آزمون ثابت بوده ولی $\{N_k^c\}$ ممکن است در دسته‌های مختلف متفاوت باشد خواهیم داشت:

$$var(z_i) = \Sigma(N_k^d)_k^{۲}\ var(\log(a_k + N_k^i))$$

از طرفی می‌دانیم:

$$var(\log(a_k + N_k^i)) = E[(\log(a_k + N_k^i))^۲] - E[\log(a_k + N_k^i)]^۲$$

با در نظر گرفتن هر یک از N_k ها به صورت توزیع دوجمله‌ای با پارامتر θ_k^c خواهیم داشت:

$$E(\log(a_k + N_k^i)] = \Sigma_n \log(a_k + n)(_n^N)(\theta_k^i)^n(۱ - \theta_k^i)^(N^i - n)$$

رابطه بالا با استفاده از تخمین N_k^i بوسیله توزیع پواسن[۶۴] با $\Lambda = \theta^c$ و استفاده از فرمول استرلینگ[۶۵] برای $n!$ بدست آمده است.
بر اساس تحلیل Rennie در [۲۱]، مقدار $var(\log(۱ + N_k))$ هنگامی که امید ریاضی تعداد رخ‌دادهای f_k (kامین ویژگی) در داده آزمون بین ۱ تا ۲ برابر مجموعه‌آموزشی باشد، به مقدار بیشینه خود می‌رسد.

۴-۴ گسترش Naive Bayes

در روش Naive Bayes فرض بر این است که ویژگی‌های بکار رفته در مساله از هم مستقل هستند و در نتیجه احتمال عضویت داده در دسته برابر با حاصل‌ضرب احتمال هر یک از اعضای آن داده در دسته مورد نظر است. ولی در بسیاری از موارد مساله پیچیده‌تر از این حالت ساده است. برای حل چنین مسائلی از فرآیندهای مارکف گسسته به عنوان فرآیند تصادفی تولید کننده داده‌ها استفاده شده و از روش‌های ML و MAP برای تخمین پارامترهای مجهول مدل (با روشی همانند آن‌چه در بخش قبل توضیح داده شده بود) استفاده می‌شود.
برای دسته‌بندی براین اساس، احتمال تولید شدن داده مورد پرسش را توسط مدل مفروض و پارامترهای تخمین‌زده‌شده محاسبه می‌شوند و داده در دسته‌ای که با بیشترین احتمال آن داده را تولید می‌کند دسته‌بندی می‌شود.
با روشی مشابه Naive Bayes می‌توان ثابت کرد که در صورتی که مدل درست انتخاب شود (مدل مفروض بتواند تخمین خوبی برای مدل غیرقابل دسترس ما باشد) این گسترش از Naive Bayes نیز به صورت بهینه کار می‌کند.

[64] Poisson

[65] Stirling's Formula

۵ روش دسته‌بندی داده‌ها بر اساس تئوری اطلاعات

برای دسته‌بندی داده‌ها در هوش مصنوعی روش‌های متنوعی وجود دارد که در فصل سوم به چند نمونه از آن‌ها اشاره‌شد. در این فصل تلاش داریم تا این کار را بااستفاده از تئوری اطلاعات انجام دهیم.

۵-۱ علت نیاز به روش دیگر هنگامی که روش Bayes وجود دارد

همان‌طور که در ۴ـ۳ توضیح داده شد روش Bayes در حالت کلی دقیق‌ترین نوع دسته‌بندی کننده‌ها است. حال سوالی که در ابتدا باید بدان پاسخ داد این است که در صورت وجود یک روش دقیق مانند روش Bayes آیا نیازی به روش دیگری مبتنی بر احتمالات است؟ در چه مواقعی نمی‌توان از روش Bayes استفاده کرد؟ در چه مواقعی روش Bayes از کارآیی خوبی برخوردار نیست؟
در چهار حالت زیر، روش Bayes جواب مناسب تولید نمی‌کند:

۱. اگر مدل درست انتخاب نشده باشد: در بسیاری از موارد اطلاع دقیقی از جزئیات مدل در دست نیست و مدل باید با توجه به داده‌های تولید شده توسط مدل تخمین زده‌شود. در این حالت انتخاب پارامتر تعداد حالات و چگونگی ارتباط حالات با یکدیگر پارامتر تعیین کننده‌ای است. در مواردی وضعیت از این مورد هم بدتر است، بدین معنا که در حقیقت منابع تولیدکننده داده‌ها واقعا یک فرآیند تصادفی (با تعداد حالات متناهی) نیستند، اما به دلیل استفاده از روش Bayes ناگزیر به مدل کردن آن با یک فرآیند تصادفی هستیم.

۲. اگر مجموعه آموزش جامعیت کافی نداشته باشد: در این حالت حتی اگر مدل درست انتخاب شده باشد به دلیل جامع نبودن مجموعه آموزش به‌کار رفته احتمالات جابه‌جایی بین دو حالت درست محاسبه نمی‌شود. حالت وخیم‌تر هنگامی است که یک یا چند ویژگی از مجموعه ویژگی‌ها، در عبور از یک حالت به حالت دیگر در مجموعه آموزش رخ نداده باشند.

۳. احتمال وجود اختلال در داده‌های مجموعه آموزشی و داده‌های آزمون: اختلال[۶۶] معمولا به علت دقیق نبودن ابزارهای نمونه‌گیری رخ‌می‌دهد و در برخی از کاربردها

[66] Noise

غیرقابل اجتناب هستند. به عنوان مثال مساله تشخیص چهره افراد را در نظر بگیرید. در این مساله، مجموعه آموزشی از تعدادی عکس از افراد مختلف که با کد صاحب عکس مشخص شده تشکیل می‌شود. در کاربرد، عکسی که از چهره یک فرد گرفته شده به عنوان ورودی به سامانه داده می‌شود و سامانه باید شبیه‌ترین فرد را پیدا کند. از معمول‌ترین مشکلات در این چنین مساله‌هایی پراکندگی رنگ‌ها، تغییر رنگ‌ها، تداخل رنگ‌ها و مانند این‌ها است. شاید قسمتی از این اختلالات را می‌توان با پیش‌پردازش[۶۷] حذف کرد ولی باز هم قسمتی از اختلال باقی می‌ماند. در این صورت، اگر اندازه مجموعه آموزش نسبتا بزرگ باشد (معمولا چنین است) احتمالا پارامترهای مدل مورد نظر با دقت خوبی قابل تخمین خواهند بود. ولی در صورتی که در داده‌آزمون هم اختلال پیش آید آن‌گاه به علت کوچک بودن احتمال خطا (این یک فرض معمول است) احتمال عضویت داده در هر یک از دسته‌ها بسیار کوچک خواهد بود و چه بسا برابر با صفر باشد.

۴. ۱اگر مجموعه در حال تغییر باشد: گاهی از اوقات دسته‌ها در حال تغییر هستند. بدین معنا که به مرور زمان برخی از احتمالات در فرآیند تصادفی متناظر با دسته‌ها، تغییر می‌کنند. به طور خاص می‌توان این مساله را در مورد خبرها دید.

به عنوان مثال، تا چندی پیش «انرژی هسته‌ای» احتمالا در حوزه اخبار سیاسی ایران اصلا ظاهر نشده بود ولی به تازگی این واژه به صورت بسیار زیادی در این حوزه خبری (دسته) ظاهر می‌شود. به عنوان مثالی دیگر، تا چندی پیش، واژه «قانا» با احتمال بیشتری در حوزه اخبار گردشگری یافت می‌شد اما برای مدتی این واژه به صورت فراوان در اخبار سیاسی ظاهر شد و اکنون وجود آن در حوزه‌های مختلف به صورت قبل بازگشته است.

در صورتی که تغییر احتمالات دسته‌ها، بسیار زیاد باشد احتمالا نمی‌توان روش موثری برای بهبود روش‌های موجود ارائه داد ولی در مثال‌های بالا، واژه‌های کلیدی دسته‌ها (خبرهای سیاسی) مانند «سازمان ملل»، «سیاست» و «ایران» تغییر نکرده است ولی احتمال یک یاچند واژه در این حوزه تغییر کرده است.

مورد ۱، به علت عدم دانش کافی کاربر از مساله رخ می‌دهد و نمی‌توان راه حل موثری برای این مشکل ارائه داد. مورد ۲، در حالت کلی یک مشکل اساسی در مساله دسته‌بندی است زیرا تنها دانش‌ما از داده‌های عضو یک دسته، مجموعه‌آموزشی است اما اگر مجموعه آموزشی شامل اطلاعات نسبتا جامعی از فرآیند باشد ولی بخشی از جزئیات مربوط به دسته

[67] Preprocessing

را در خود نداشته باشد می‌توان راه‌حلی برای رفع این مشکل ارائه داد. یکی از مشکلات اساسی روش‌های Bayes در رویارویی با اختلال‌هااست[۲۲]. به دلیل این‌که روش‌های Bayes بسیار وابسته به احتمالات تخمین‌زده شده هستند معمولا نمی‌توانند در مورد ۴، پاسخ مناسبی ارائه دهند.
یکی دیگر از مواردی که روش‌های Bayes در حل آن به مشکل برمی‌خورند حالت زیر است: حالتی را در نظر بگیرید که در آن یک دنباله $T = f_1, f_2, \ldots, f_n$ داده شده باشد. ماشین M به عنوان مدل دسته C فراخوانی می‌شود و در گام i تولید T در مجموعه حالات $\{S_{i_1}, S_{i_2}, \ldots, S_{i_k}\}$ قرار داریم و هیچ یک از S_iها با ویژگی (کارکتر) f_i به حالت دیگری خروجی نداشته باشند. در این حالت باید چه تصمیمی در مورد مقدار احتمال تولید T توسط ماشین گرفته‌شود؟ آیا این احتمال واقعا صفر است؟ آیا عدم وجود یال به علت ناقص بودن و یا ناکافی بودن مجموعه آموزشی است؟ به این سوالات به سه طریق می‌توان پاسخ داد:

۱. ساده‌ترین روش برخورد با این مساله صرف‌نظر کردن از f_i است. به این ترتیب که f_i را از T حذف کنیم و به جای محاسبه احتمال T، احتمال T' (دنباله جدید) را محاسبه کنیم. همان‌طور که گفته شد این راه‌حل ساده‌ترین روش برخورد با این مشکل است و احتمالا جواب مناسبی برای این مساله نیست (هیچ توجیه علمی نیز برای آن وجود ندارد).

۲. روش دیگر برخورد با این مساله، صفر در نظر گرفتن احتمال f_i در آن حالت است. این کار را می‌توان این‌گونه توجیه کرد که فرض کردیم مجموعه آموزشی ما یک مجموعه کامل و پوشا بوده و عدم وجود یک ویژگی در مجموعه آموزشی به معنای صفر بودن امکان آن ویژگی است. این روش در صورت کامل بودن مجموعه آموزشی درست است ولی چگونه می‌توان از جامع بودن مجموعه آموزشی اطمینان حاصل کرد؟ اگر مجموعه کلاس در حال تغییر باشد چطور؟ در این صورت ممکن است T شامل یک ویژگی جدید که تا به حال در مجموعه‌آموزشی وجود نداشته است باشد. در فصل بعدی این مورد را به همراه کاربرد عملی آن به طور کامل‌تر توضیح می‌دهیم.

۳. روش دیگر استفاده از یک راه‌حل بینابینی است. به این ترتیب که در صورت عدم وجود یال خروجی از یک حالت مشخص، احتمال آن را برابر با عددی از پیش تعیین‌شده مانند v قرار دهیم. مقدار v باید متناسب با مقدار جامعیت مجموعه آموزشی باشد. با قرار دادن $v = ۰$ راه اول و $v = ۱$ راه دوم بدست می‌آید. نکته قابل ذکر این است که در این روش، روش Bayes به صورت درست و کامل پیاده شده است زیرا در روش Bayes، در حقیقت باید $P(T|M, H)$ محاسبه شود که در آن H مجموعه‌ای از فرض‌هاست. H می‌تواند شامل این فرض باشد که "در صورت عدم امکان رخ دادن

یک ویژگی در یک حالت خاص با فلان احتمال به فلان حالت برو" . مقدار v را می‌توان با تنظیم پارامترهای a_i و a معرفی شده در بخش سوم بدست آورد.

در حقیقت مشکل اساسی این روش چگونگی تعیین این پارامتر است.

در ادامه یک روش جایگزین ارائه می‌شود و ثابت می‌کنیم که در صورت وجود شرایط خاصی (که این شرایط، شرایط محدود کننده‌ای نیستند) مانند روش Bayes کار می‌کند و علاوه بر این می‌تواند مشکلات اشاره شده در ۲ و ۳ و ۴ را حل کند و علاوه بر آن در مورد بالا نیز، نیاز به تنظیم پارامتر خاصی ندارد.

۵-۲ مدل احتمالات

همان‌طور که در معرفی روش Bayes گفته شد، روش Bayes و سایر روش‌های احتمالاتی نیاز به یک فرآیند تصادفی برای مدل کردن تولید رشته‌ها دارند. معمولا از مدل‌های مارکف (زنجیره‌های مارکف، پردازه‌های مارکف گسسته) برای این منظور استفاده می‌شود. در روش‌های احتمالاتی دسته‌بندی داده‌ها، هر داده توسط یک دنباله از ویژگی‌ها[۶۸] بیان می‌شود. این دنباله‌ها الزاما برگشت پذیر و یک به یک نیستند. به ازای هر دسته، یک سامانه در نظر گرفته می‌شود که در سامانه همه داده‌های عضو آن دسته با بیشترین احتمال تولید می‌شوند. در این روش‌ها مفهوم عضویت با مفهوم احتمال عضویت جایگزین شده است و پس از محاسبه احتمال‌های عضویت، در مورد هردسته فرآیند تصمیم‌گیری انجام می‌شود. از روش‌های معمول برای این‌کار می‌توان به آستانه‌گیری[۶۹][۲۵] ویا ماکزیمم‌گیری اشاره کرد. ماشین توصیف کننده یک دسته، دنباله‌های مربوط به داده‌های آن دسته را با احتمال بیشتری تولید می‌کند.
سامانه موردنظر برای هر دسته معمولا یک پردازه مارکف گسسته است که آن‌را به صورت زیر بیان کنیم:

۱. الفبا: ماشین مورد استفاده در این روش‌ها، باید دنباله‌ای از ویژگی‌ها را تولید کند. در نتیجه، باید به ازای هر ویژگی از مجموعه ویژگی‌ها یک حرف الفبا اختصاص یابد. مجموعه الفبا را با $F = \{f_۱, f_۲, \ldots, f_L\}$ نشان می دهیم.

۲. حالت‌ها: مجموعه حالات را با $S = \{S_۱, S_۲, \ldots, S_n\}$ نشان می‌دهیم.

[68] feature

[69] Thresholding

۳. انتقال‌ها: هر حالت مانند S_i با احتمالی بزرگ‌تر یا مساوی صفر به تمام حالت‌های دیگر S_j متصل می‌باشد و در نتیجه استفاده از این یال، کارکتر اختصاص داده‌شده به آن یال در خروجی پدیدار می‌شود. احتمال انتقال از حالت i به حالت j و تولید کارکتر f را با $p^f_{i,j}$ نشان می‌دهیم. با تثبیت تعداد حالات و نحوه اتصال آن‌ها(وجود یا عدم وجود یال حامل کارکتر f بین S_i و S_j)، می‌توان احتمال انتخاب شدن یال مورد نظر ($p^f_{i,j}$) را برحسب مجموعه‌آموزشی بدست آورد. $n^f_{i,j}$ را تعداد کارکترهای f موجود در مجموعه آموزش که برای تولید آن‌ها ماشین باید از حالت S_i و S_j با کارکتر f برود می‌نامیم. $n^f_i = \sum_j n^f_{i,j}$ را برابر با تعداد دفعاتی که ماشین برای تولید مجموعه آموزشی، از حالت S_i با کارکتر f خارج می‌شود قرار می‌دهیم. به همین ترتیب $n_i = \sum_f n^f_i$ را برابر با تعداد دفعاتی که ماشین برای تولید مجموعه‌آموزش از S_i خارج می‌شود قرار می‌دهیم. کل تعداد کارکترهای مجموعه‌آموزش را با $N = \sum_i n_i$ نشان می‌دهیم. تعداد رویداد کارکتر f در مجموعه آموزش را با n^f_M و تعداد رخدادهای f را در رشته d با n^f_d نشان می‌دهیم.

در ادامه در برخی از موارد از نمادهای معرفی در این بخش به عنوان یک تابع استفاده شده است که آرگومان ورودی آن می‌تواند یک مجموعه از بردارهاو یا یک بردار باشد.

محاسبه Entropy یک پردازه مارکف گسسته

Entropy یک سامانه تولید کننده مجموعه‌ای از رشته‌ها با احتمالات مختلف برابر با Entropy تمام رشته‌های قابل تولید آن سامانه با در نظر گرفتن احتمال هر رشته است. در حالت کلی هر پردازه می‌تواند تعداد نامتناهی رشته تولید کند و در نتیجه نمی‌توان به تولید تمام رشته‌ها پرداخت و سپس Entropy را محاسبه کرد. روش مستقیمی برای محاسبه Entropy مجموعه تمامی رشته‌های قابل تولید توسط یک پردازه مارکف گسسته به صورت زیر وجود دارد:

$$E = -\sum_i \pi_i \sum_j \sum_c p^c_{i,j} \log p^c_{i,j}$$

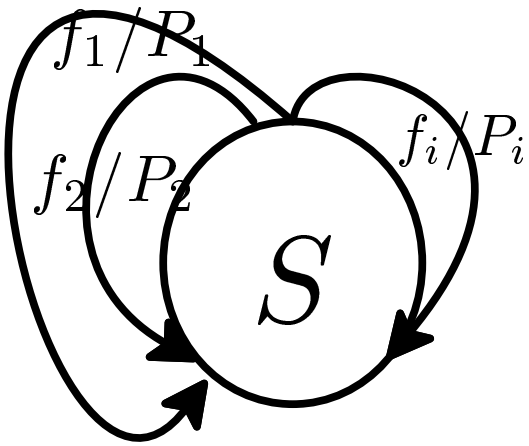
که در آن $p^c_{i,j}$ احتمال انتقال از حالت S_i به حالت S_j و تولید کارکتر c است و π_i احتمال قرار داشتن در حالت S_i است. این پارامترها در بخش ۲ توضیح داده شده بود.
برای راحت‌تر شدن اثبات‌ها، فرض کرده‌ایم هرحالت M با تولید کارکتر c فقط به یک حالت دیگر می‌تواند منتقل می‌شود، به راحتی می‌توان تحقیق کرد که این محدودیت هیچ کاهشی در قدرت این ماشین‌ها ایجاد نمی‌کند.

چند نکته در مورد چگونگی استفاده از پردازه‌های مارکف گسسته به عنوان منبع

همان‌گونه که اشاره شد هدف اصلی در مساله دسته‌بندی، بدست آوردن یک تعمیم بر اساس داده‌های موجود در مجموعه‌آموزشی برای کل دامنه مساله می‌باشد، به همین دلیل، هنگامی که می‌خواهیم یک مساله را به صورت احتمالاتی و با استفاده از منابع اطلاعاتی حل کنیم از یک منبع اطلاعاتی که توانایی تولید تعداد نامتناهی (شمارا) رشته را دارد استفاده می‌کنیم. از این پس منابع مورد استفاده ما هیچ رشته متناهی تولید نمی‌کنند و فقط رشته‌های نامتناهی تولید می‌کنند. این بدان معناست که هر راس موجود در گراف نمایش منبع در حداقل یک دور قرار دارد.
معمولا برای منابع اطلاعاتی، حالت شروعی در نظر گرفته نمی‌شود و حالت i با احتمال Π_i (احتمال ایستایی در حالت iکه در بخش ۲ تعریف شد) می‌تواند حالت شروع منبع مورد نظر باشد. منبع S برای تولید رشته نامتناهی T، از یک حالت به صورت تصادفی و با توزیع احتمال Π شروع می‌کند و با توجه به توزیع احتمال انتقال در حالت کنونی فرآیند، یکی از انتقال‌های ممکن (انتقالی که مبدا آن حالت کنونی باشد) را انتخاب می‌کند و پس از اضافه کردن کارکتر مرتبط با آن انتقال به رشته T، حالت کنونی فرآیند را به‌روز می‌کند.
گوییم رشته (متناهی و یا نامتناهی) T توسط منبع S تولید می‌شود اگر در هر رشته (نامتناهی) تولیدی توسط S مانند T'، رشته T به عنوان زیررشته بی‌نهایت بار تکرار شده باشد. به صورت مشابه تولید شدن مجموعه‌ای از رشته‌ها توسط یک منبع به این معناست که هر عضو مجموعه موردنظر در T' بی‌نهایت بار تکرار شود.
احتمال تولید شدن داده (بردار) d توسط ماشین M را برابر با احتمال تولید شدن $|d|$ کارکتر اول رشته d توسط M ($P(d|M)$) تعریف می‌کنیم. برای محاسبه آن، نیاز به تعاریف زیر داریم:

- $n^c_{i,j,d}$: تعداد دفعاتی که در یک مسیر از پیش تعیین شده در یک فرآیند برای تولید d از حالت i به حالت j رفته‌ایم.
- T^M_d: تمامی دنباله‌های $|d|+1$ عنصری که هر عنصر آن برابر با نام یک حالت است و با دنبال کردن این دنباله بتوان $|d|$ کارکتر اول d را تولید کرد. به علت فرض ما مبنی بر این که در هر حالت فقط یک انتقال با کارکتر c از آن حالت خارج می‌شود می‌توان هر عضو $t \in T_d$ را به صورت یکتا توسط حالت اولیه t تعریف کرد.
- $P(d|t)$: که در آن $t \in T^M_d$ است برابر با احتمال تولید شدن $|d|$ کارکتر اول d در مسیر t است و برابر است با:

$$P(d|t) = \prod_i p^{d_i}_{t_i t_{i+1}}$$



شکل ۵ـ۱: فرآیند مربوط به حالتی که در آن ترتیب و چگونگی قرارگرفتن ویژگی‌ها مهم نباشد و فقط وجود یا عدم وجود ویژگی‌ها مهم باشد

$P(d|M)$ از رابطه زیر بدست می آید:

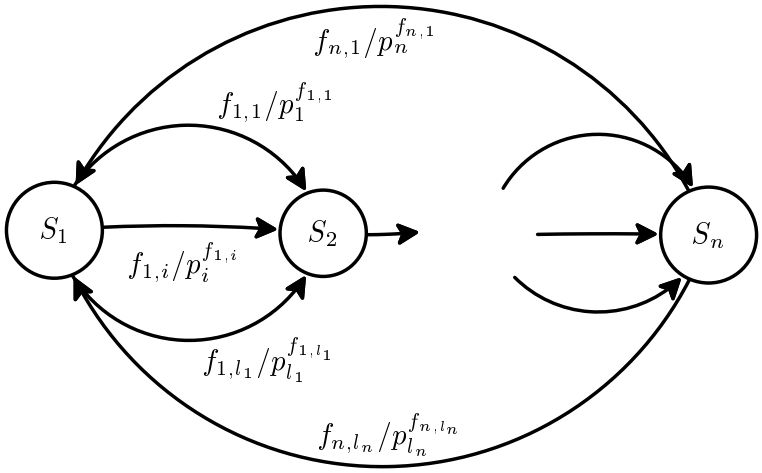
$$p(d|M) = \sum_{t \in T_d^M} \Pi_{t_0} p(d|t)$$

در ادامه به توضیح چند حالت خاص از پردازه‌های مارکف گسسته که برای کار ما مناسب است می‌پردازیم:

۱. حالتی که در آن ترتیب و چگونگی قرارگرفتن ویژگی‌ها مهم نباشد و فقط وجود یا عدم وجود ویژگی‌ها مهم باشد: در این حالت به جای دنباله‌ای از ویژگی‌ها، با یک مجموعه از ویژگی‌ها سروکار داریم. پردازه‌مارکف زیر برای بیان این حالت مناسب است.

در این حالت مدل دارای یک راس S و L یال است و باید مقدار L پارامتر یعنی احتمال‌های مجهول، $p_{S,S}^{f_l}$ تعیین شود. باید نشان دهیم که این مدل‌توانایی نمایش مساله‌هایی با خصوصیت مورد بحث را دارد. برای این‌کار کافی‌است نشان دهیم هر مجموعه از ویژگی‌ها (که به صورت یک دنباله نشان داده شده است) و در نتیجه هر مجموعه‌آموزشی در این فرآیند قابل نمایش است. به علت وجود تنها یک حالت، بدیهی است که به ازای هر دنباله متناهی از داده‌ها می‌توان آن‌را در صورت وجود عناصر آن دنباله در فرآیند تولید کرد. از کاربردهای این مدل می‌توان به دسته‌بندی متون[۷۰] اشاره کرد. این گونه فرآیندها برای مدل‌سازی مساله‌هایی که در آن ویژگی‌ها به صورت مستقل از هم اتفاق می‌افتند دقیقا هم‌خوانی دارد همچنین می‌توان در

[70] Text Classifcation



شکل ۵ـ۲: حالتی که در آن ترتیب عناصر مهم است و هر ویژگی بسته به مکان آن ویژگی احتمال به‌خصوصی دارد

مواردی که تعداد ویژگی‌ها زیاد است و یا اندازه مجموعه‌آموزشی به اندازه کافی بزرگ نیست که بتوان بر اساس آن همه احتمالات لازم را به صورت دقیق پیدا کرد از آن‌ها استفاده نمود.

برای تخمین زدن پارامترهای این گونه فرآیندها، می‌توان به صورت زیر عمل کرد:

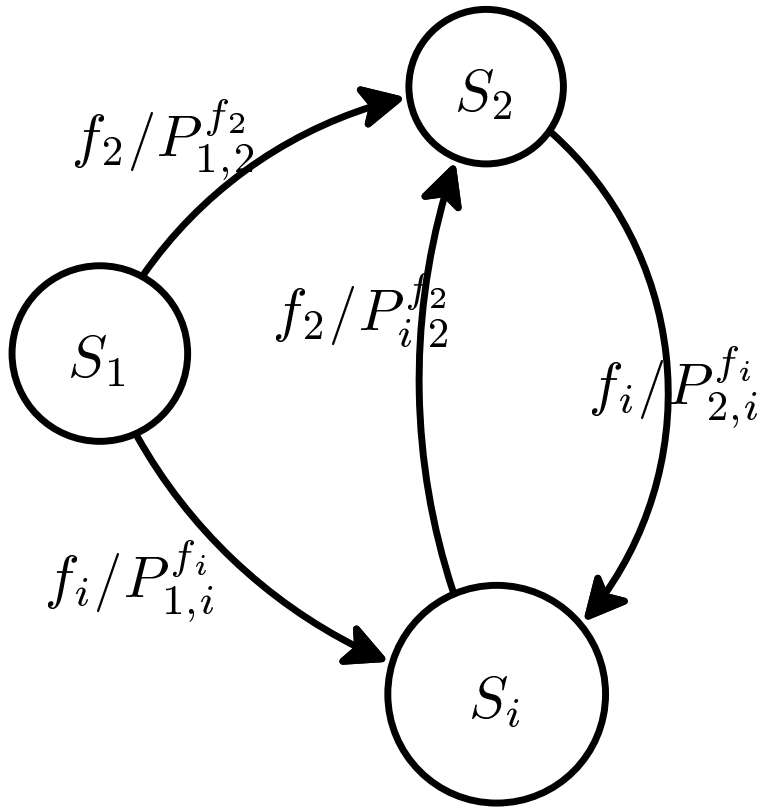
اگر n_f تعداد تکرار ویژگی f در مجموعه‌آموزشی و $n = \sum_f n_f$ باشد داریم $p_f = \frac{n_f}{n}$

۲. حالتی که در آن ترتیب عناصر مهم است و هر ویژگی بسته به مکان آن ویژگی احتمال به‌خصوصی دارد: در این حالت واقعا با مفهوم دنباله‌ای از ویژگی‌ها (با طول ثابت l) سروکار داریم که برد مولفه iـام دنباله (F_i)، همیشه در مجموعه $F_i = \{f_{i,j}\}$ است. در حقیقت F_i مجموعه تمام ویژگی‌های ممکن در مکان iـام است. برای سادگی فرض می‌کنیم $\forall i \neq j : F_i \cap F_j = \emptyset$.

پردازه شکل ۵ـ۲ برای بیان این حالت پیشنهاد می‌شود:

در این حالت، مدل دارای l راس و $\sum |F_i|$ یال است و باید مقدار $\sum |F_i|$ پارامتر احتمالات مجهول p_i^f تعیین شود. برای نشان دادن توانایی این گونه از پردازه‌ها برای توصیف مساله‌هایی به صورت بالا نشان می‌دهیم که هر دنباله $T =< f_{۱,t_۱}, f_{۲,t_۲}, \cdots, f_{l,t_l} >$ توسط یک پردازه از نوع بالا قابل تولید است.

به دلیل وجود تنها یک دور در گراف فرآیند بالا، پس از حداکثر تولید l کارکتر به حالت $S_۱$ می‌رسیم. از $S_۱$ می‌توان با تولید $f_{۱,t_۱}$ به $S_۲$ رفت و به همین ترتیب از S_i با تولید f_{i,t_i} به حالت $S_{i+۱}$ می‌رویم.



شکل ۵ـ۳: حالتی که در آن ترتیب عناصر مهم است و هر ویژگی بسته به عنصر قبلی دارای احتمال متفاوتی است

برای محاسبه احتمال‌های مجهول پردازه می‌توان از روش زیر استفاده کرد:

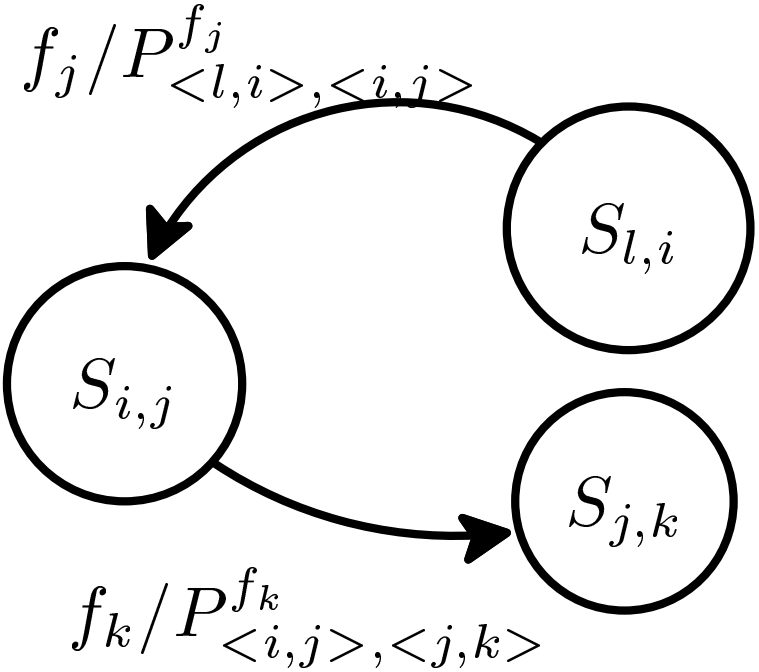
اگر n_i^f تعداد تکرار ویژگی f در مجموعه‌آموزشی در مکان iـام باشد و $n_i = \sum_f n_i^f$ باشد داریم $p_i^f = \frac{n_i^f}{n_i}$

فرض اشتراک تهی برای بردهای مولفه‌های مختلف یک شرط ضروری نیست ولی به خوانایی و درک بهتر از مدل منتهی می‌شود.

۳. حالتی که در آن ترتیب عناصر مهم است و هر ویژگی بسته به عنصر قبلی دارای احتمال متفاوتی است: در این حالت نیز واقعا با مفهوم دنباله‌ای از ویژگی‌ها سروکار داریم، اما این بار هر عنصر به‌جای وابستگی به مکان خود به عنصر قبلی خود وابسته است. زنجیره‌مارکف زیر برای بیان این حالت مناسب است.

در این حالت مدل دارای L راس (هر راس نشان‌گر آخرین کارکتر تولید شده است) و L^2 یال است و باید مقدار L^2 پارامتر احتمالات مجهول $p_{i,j}^{f_i}$ تعیین شود.

در گراف شکل ۵ـ۳ همه راس ها در حداقل یک دور قرار دارند و هر راسی از هر راس



شکل ۵ـ۴: حالتی که در آن ترتیب عناصر مهم است و هر ویژگی بسته به دو عنصر قبلی دارای احتمال متفاوتی است

دیگر قابل دسترسی است. احتمال انتقال از حالت i به حالت j (و در نتیجه تولید کارکتر f_j) را با $p^{f_j}_{i,j}$ نشان می‌دهیم. بدیهی است که در حالت کلی $\forall i,j,k : k \neq j \; p^{f_j}_{i,j} \neq p^{f_k}_{i,k}$

به راحتی می‌توان مشاهده کرد که هر دنباله‌ای از ویژگی‌ها(کارکترها) توسط فرآیند فوق قابل تولید است

روش مشابهی مانند آن‌چه در بالا بدان اشاره شد را می‌توان برای این نوع ماشین‌ها به‌کار برد و به این ترتیب عمل کرد:

برای تولید هر دنباله توسط فرآیند فوق فقط یک مسیر وجود دارد و در نتیجه می‌توان به راحتی $n^{f_j}_{i,j}$ را که تعداد دفعاتی است که در دنباله پس از ویژگی f_i، ویژگی f_j آمده است روی تمام دنباله‌های موجود در مجموعه آموزشی محاسبه کرد و سپس $n_i = \sum_j n^{f_j}_{i,j}$ که برابر با تعداد دفعاتی است که فرآیند برای تولید مجموعه آموزش وارد حالت i می‌شود. حال می‌توان $p^{f_j}_{i,j} = \frac{n^{f_j}_{i,j}}{n_i}$ را تعریف کرد.

۴. حالتی که در آن ترتیب عناصر مهم است و هر ویژگی بسته به دو عنصر قبلی دارای احتمال متفاوتی است: زنجیره‌مارکف زیر برای بیان این حالت مناسب است.

در این حالت، مدل دارای L^2 راس و L^3 یال است و باید مقدار L^3 پارامتر احتمالات مجهول $p^{f_j}_{<l,i>,<i,j>}$ تعیین شود.

روش مشابهی مانند آن‌چه در بالا بدان اشاره شد را می‌توان برای این نوع ماشین‌ها به کار برد.

یک نکته قابل توجه این است که می‌توان از هر یک از فرآیندهای معرفی شده دربالا برای توصیف یک مساله استفاده کرد اما در صورت استفاده از یک مدل نامناسب احتمالات بدست آمده دیگر کاربردی نیستند. به عنوان مثال، یک مجموعه آموزشی شامل ۱۰۰ نمونه از دنباله‌ای به صورت $< f_1, f_2, \cdots, f_L >$ را درنظر بگیرید. اگر بخواهیم این دنباله را براساس فرآیندی مانند فرآیند ۱ توصیف کنیم خواهیم داشت $p_f = \frac{1}{L}$. و اگر آن‌را براساس فرآیندی مانند فرآیند ۲ ($l = L$) توصیف کنیم بردار توزیع احتمال p به صورت زیر خواهد بود:

$$p_j^{f_i} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

و اگر آن را با فرآیندی مانند فرآیند ۳ تخمین بزنیم خواهیم داشت

$$p_{i,j}^{f_j} = \begin{cases} 1 & \text{if } j = i+1 \text{ or } (j = 0 \ and \ i = L) \\ 0 & \text{otherwise} \end{cases}$$

۳-۵ چگونگی محاسبه Entropy برای چند فرآیند خاص

طبق تعریف ارائه شده در بخش ۱ برای فرآیندهای مارکف و تعمیم آن فرآیندهای مارکف گسسته، می‌توان Entropy یک فرآیند را در حالت کلی می‌توان به صورت زیر محاسبه کرد:

$E_M = -\sum_{s \in States} \pi_s \sum_{s' \in States} \sum_{f \in F} p_{s,s'}^f \log p_{s,s'}^f$

در برخی از فرآیندهای خاص می‌توان، یک رابطه ساده‌تر برای محاسبه E_M بدست آورد که در ادامه به بررسی آن‌ها می‌پردازیم:

حالت اول

در این گونه فرآیندها، فقط یک حالت موجود است و بنابراین بردار π برابر با $< ۱ >$ خواهد بود و با توجه به رابطه‌های مربوط به محاسبه p_f داریم:

$$E_M = -\sum_{s \in States} \pi_s \sum_{s' \in States} \sum_{f \in F} p_{s,s'}^f \log p_{s,s'}^f$$

$$= -\sum_{f \in F} p_f \log p_f = -\sum \frac{n_f}{n} \log \frac{n_f}{n} = \log n \sum \frac{n_f}{n} - \sum \frac{n_f}{n} \log n_f$$

$$= \log n - \frac{1}{n} \sum n_f \log n_f$$

$$\Rightarrow E_M = \log n - \frac{۱}{n} \sum n_f \log n_f \qquad (۵-۳)$$

حالت دوم

در این گونه فرآیندها، l حالت موجود است و از هر حالت فقط به حالت بعدی می‌رویم و در نتیجه احتمال حضور در هر حالت برابر با احتمال حضور در سایر حالات است یعنی $\pi_i = \frac{۱}{l}$ و با توجه به رابطه‌های مربوط به محاسبه p_i^f داریم:

$$E_M = - \sum_{s \in States} \pi_s \sum_{s' \in States} \sum_{f \in F} p^f_{s,s'} \log p^f_{s,s'}$$

$$= - \sum_{s \in States} \pi_s \sum_{f \in F} p_i^f \log p_i^f = -\frac{1}{l} \sum \sum \frac{n_i^f}{n_i} \log \frac{n_i^f}{n_i}$$

$$= \frac{1}{l} \sum_i \left[\sum_f \frac{n_i^f}{n_i} \log n_i - \sum_f \frac{n_i^f}{n_i} \log n_i^f \right]$$

$$= \frac{1}{l} \sum_i \left[\log n_i - \sum_f \frac{n_i^f}{n_i} \log n_i^f \right]$$

همان‌طور که در توصیف این‌گونه فرآیندها گفته شد $\forall i, j : n_i = n_j = n$ که در آن n برابر با تعداد اعضای مجموعه‌ای است که فرآیند بر اساس آن ساخته شده است، و در نتیجه رابطه بالا به صورت زیر ساده می‌شود:

$$E_M = \log n - \frac{۱}{l} \sum_i \sum_f \frac{n_i^f}{n_i} \log n_i^f \qquad (۵-۴)$$

حالت سوم

در این گونه فرآیندها، L حالت موجود است و به علت کلی بودن ماتریس احتمالات P تعیین مقدار ویژه این ماتریس به صورت یک فرمول بسته امکان‌پذیر نیست:

$$E_M = -\sum_{s\in States} \pi_s \sum_{s'\in States} \sum_{f\in F} p^f_{s,s'} \log p^f_{s,s'}$$

$$= -\sum_i \pi_i \sum_j \sum_f \frac{n^f_{i,j}}{n_i} \log \frac{n^f_{i,j}}{n_i} = \sum_i \pi_i \sum_j \sum_f \left[\frac{n^f_{i,j}}{n_i} \log n_i - \frac{n^f_{i,j}}{n_i} \log n^f_{i,j} \right]$$

$$= \sum_i \pi_i \sum_j \sum_f \frac{n^f_{i,j}}{n_i} \log n_i - \sum_i \pi_i \sum_j \sum_f \frac{n^f_{i,j}}{n_i} \log n^f_{i,j}$$

$$= \sum_i \frac{\pi_i \log n_i}{n_i} \sum_j \sum_f n^f_{i,j} - \sum_i \frac{\pi_i}{n_i} \sum_j \sum_f n^f_{i,j} \log n^f_{i,j}$$

$$= \sum_i \pi_i \log n_i - \sum_i \frac{\pi_i}{n_i} \sum_j \sum_f n^f_{i,j} \log n^f_{i,j}$$

$$E_M = \sum_i \pi_i \log n_i - \sum_i \frac{\pi_i}{n_i} \sum_j \sum_f n^f_{i,j} \log n^f_{i,j} \qquad (۵-۵)$$

۵-۴ رابطه بین دسته‌بندی و Entropy

هدف نهایی در دسته‌بندی داده‌ها این است که روشی برای تشخیص تعلق و یا عدم تعلق یک داده به هر یک از دسته‌ها ارائه شود. راه‌های مختلفی برای این منظور وجود دارد که به بخشی از آن‌ها در فصل دوم اشاره شد. در دسته‌بندی معمولا تعدادی از اعضای یک دسته در اختیار است. یکی از روش‌های تعیین عضویت می‌تواند استفاده از میزان شباهت بین داده با اعضای هر دسته باشد. به این ترتیب که، با استفاده از یک معیار به سنجش میزان شباهت داده با اعضای هر یک از دسته‌ها بپردازیم و مشابه‌ترین دسته را به عنوان دسته برنده اعلام کنیم.
همان‌طور که دیدیم، یکی از راه‌های محاسبه میزان شباهت استفاده از مفهوم مشابه میزان تردید (عدم قطعیت) است. در بخش بعدی به توضیح چگونگی استفاده از این مفهوم برای دسته‌بندی خواهیم پرداخت.

۵-۵ ایده کلی روش

همان‌طور که در بخش ۳ـ۱ بحث شد یکی از راه‌های اندازه‌گیری ریاضی میزان آشفتگی داده‌ها، استفاده از Entropy است. در صورت داشتن یک معیار برای اندازه‌گیری میزان آشفتگی داده‌ها، می‌توان از این معیار در دسته‌بندی داده‌ها به صورت زیر استفاده کرد:

در مساله دسته‌بندی داده‌ها، دسته c براساس یک مجموعه آموزشی جامع(S_c) تعریف شده است و می‌خواهیم در مورد تعلق داده d به دسته c نظر بدهیم. اگر $S'_c = S_c \cup \{d\}$ باشد با توجه به فرض ما مبنی بر جامع بودن مجموعه آموزشی، منطقی است که انتظار داشته باشیم اگر d واقعا به c متعلق باشد پارامترهای تعیین شده (در مرحله آموزش الگوریتم‌ها) توسط مجموعه های S_c و S'_c احتمالا بسیار مشابه خواهد بود. برای مقایسه دو توزیع احتمال راه‌های فراوانی موجود است که برای اطلاع از آن‌ها می‌توانید به [۱۶] مراجعه کنید. در کاربرد مورد بحث در این پایان‌نامه، به دلیل رابطه به‌خصوص دو توزیع با یکدیگر، به نظر می‌رسد که استفاده از Entropy یک معیار خوب باشد. به این ترتیب که Entropy فرآیند تولیدکننده S_c و Entropy فرآیند تولیدکننده S'_c را با هم مقایسه کنیم و در حقیقت به مطالعه میزان تغییرات در Entropy در اثر افزوده شدن داده d بپردازیم. این تغییر(Δ_E) می‌تواند مثبت، منفی و یا صفر بسته به شرایط باشد که در ادامه به بررسی هر یک خواهیم پرداخت:

۱. $\Delta_E > ۰$: بدین معناست که Entropy در حالتی که پارامترهای فرآیند بر اساس مجموعه آموزش تعیین شده بود از Entropy همان فرآیند هنگامی که پارامترهای آن براساس مجموعه‌آموزشی و داده d تعیین شده است کمتر است یعنی با اضافه‌شدن d، مقدار آشفتگی در بین رشته‌های تولید شده توسط منبع بیشتر گردیده‌است. این می‌تواند بدان معنا باشد که داده d موجب تغییرات اساسی در پارامترهای فرآیند شده است و این تغییرات بیشتر در جهت آشفته‌تر کردن دنباله‌های تولیدی بوده است. به عنوان مثال، فرض کنید برای یک مساله فرآیند ۱ را برای مدل‌سازی انتخاب کرده‌ایم و براساس یک مجموعه آموزشی مقادیر n_fها به صورت زیر باشد ($L = ۴$):

$$n_f = \begin{cases} 100 & \text{if } f \leq 3 \\ 10 & \text{otherwise} \end{cases}$$

Entropy این فرآیند برابر با ۱/۷۴۰ خواهد بود. اگر توزیع ویژگی‌ها در داده d به صورت زیر باشد:

$$n_f^d = \begin{cases} 0 & \text{if } f \leq 3 \\ 31 & \text{otherwise} \end{cases}$$

در مورد پارامترهای فرآیند حاصل از مجموعه‌آموزش و d خواهیم داشت که $p =< ۰/۲۹۳, ۰/۲۹۳, ۰/۲۹۳, ۰/۱۲۱ >$ و در نتیجه Entropy فرآیند جدید برابر با ۱/۹۲۵ خواهد بود.

در این حالت احتمالا داده به دسته متعلق نیست و یا احتمال عضویت داده در آن دسته به‌خصوص، کم است.

۲. $\Delta_E = 0$: بدین معناست که Entropy در حالتی که پارامترهای فرآیند بر اساس مجموعه آموزش تعیین شده بود و Entropy فرآیند هنگامی که پارامترهای آن براساس مجموعه آموزشی و d تعیین می‌شود با هم برابر بوده است. اگر خوش شانس باشیم این بدان معناست که توزیع احتمال فرآیند پیش و پس از افزوده شدن d اصلا تغییری نکرده است، یعنی d توزیعی کاملا مشابه با مجموعه آموزش داشته است. در مثال بالا، اگر توزیع ویژگی‌ها در d به صورت زیر باشد:

$$n_f^d = \begin{cases} 10 & \text{if } f \leq 3 \\ 1 & \text{otherwise} \end{cases}$$

خواهیم داشت $\Delta_E = 0$.

۳. $\Delta_E < 0$: بدین معناست که Entropy در حالتی که پارامترهای فرآیند بر اساس مجموعه آموزش تعیین شده بود از Entropy فرآیند هنگامی که پارامترهای آن براساس مجموعه آموزشی و d تعیین می‌شود بیشتر است. یعنی با اضافه‌شدن d، مقدار آشفتگی در بین رشته‌های تولید شده توسط منبع کمتر گردیده‌است و در حقیقت یک نظم جدید با اضافه شدن d به S کشف شده است. این می‌تواند بدان معنا باشد که داده d موجب تغییراتی (اساسی) در پارامترهای فرآیند شده است و این تغییرات بیشتر در جهت منظم‌تر کردن دنباله‌های تولیدی بوده است. اگر در مثال بالا توزیع ویژگی‌ها در d به صورت زیر باشد:

$$n_f^d = \begin{cases} 10 & \text{if } f \leq 3 \\ 0 & \text{otherwise} \end{cases}$$

در مورد پارامترهای فرآیند حاصل از مجموعه‌آموزش و d خواهیم داشت که $p =< ۰/۳۲۲, ۰/۳۲۲, ۰/۳۲۲, ۰/۰۳۲ >$ و در نتیجه Entropy فرآیند جدید برابر با ۱/۷۳۸ خواهد بود.

خواهیم داشت $\Delta_E < 0$.

چگونگی دسته‌بندی

برای تعیین دسته یک داده مجهول در بین چند دسته داده‌شده می‌توان از روش زیر استفاده کرد:

۱. به ازای هر دسته مانند c یک پردازه‌مارکف گسسته با مدل مناسب در نظر بگیر و بر اساس مجموعه‌آموزشی متناظر آن دسته (S_c)، احتمالات انتقال بین حالت‌ها را

شکل ۵ـ۵: $\{< ۰۱۱ >, < ۱۰۰ >\}$یک فرآیند با ۹ حالت برای مجموعه داده‌های

محاسبه کرده و مقدار Entropy پردازه (E^c_{Old}) را ذخیره کن.

الف‌ـ مجموعه $S'_c = S_c \cup \{d\}$ را تشکیل بده و به محاسبه مجدد احتمالات انتقال بین حالات بدون ایجاد تغییر در مدل بپرداز و سپس Entropy مدل جدید (E_{New}) را محاسبه کن.

ب‌ـ مقدار Δ^c_E را برابر با $E^c_{New} - E^c_{Old}$ قرار بده.

۲. داده مجهول متعلق به کلاسی است که مقدار Δ^c_E آن کمترین باشد.

چگونگی محاسبه

در الگوریتم بالا، قسمت ۱ فقط یک بار اجرا می‌شود و به ازای هر دسته E^c_{Old} محاسبه می‌شود. در حقیقت این مرحله، همان مرحله آموزش الگوریتم است. قسمت‌های ۱ـالف و ۱ـب به ازای هر داده مورد آزمون فراخوانی می‌شود. از آن‌جا که معمولا یک سیستم برای دسته‌بندی یک بار آموزش داده شده و پس از آن برای تشخیص استفاده می‌شود توجه اصلی در پیچیدگی این نوع سیستم‌ها، پیچیدگی مرحله آزمون آن‌هاست[۲۳].
در کاربرد مورد بحث ما، میزان آشفتگی یک مجموعه از داده‌ها به فرآیند تولید کننده داده‌ها وابسته است. برای مثال اگر بخواهیم Entropy مجموعه داده‌های $\{< ۰۱۱ >, < ۱۰۰ >\}$ را برای یک فرآیند با یک حالت محاسبه کنیم در حقیقت فرآیند مورد نظر یک فرآیند کاملا تصادفی خواهد بود(با احتمال $\frac{1}{2}$ یک و با $\frac{1}{2}$ صفر تولید می‌کند)، ولی اگر تلاش کنیم که این مجموعه را توسط یک مدل با ۹ حالت به مانند ۵ـ۵ تولید کنیم Entropy مجموعه بسیار کمتر خواهد بود. باید توجه داشت که علت این اختلاف در این است که ما مجموعه مورد نظر را به عنوان نمونه داده‌های تولید شده توسط یک فرآیند در نظر گرفته‌ایم و در نتیجه تخمین ما از Entropy مجموعه‌آموزش بسته به چگونگی محدودیت مجموعه و فرآیند تغییر می‌کند.

۵-۶ اثبات هم‌ارزی روش مبتنی بر تئوری اطلاعات با Naive Bayes

همان‌طور که گفته شد دسته‌بندی براساس Naive Bayes، در حالتی که اطلاعات کافی در اختیار داشته باشیم (مساله توسط Naive Bayes قابل حل باشد)، بهترین راه است. در این بخش تلاش خواهیم کرد تا روش ارائه شده در بالا را با روش Naive Bayes مقایسه کنیم و

ثابت کنیم در حالات اول و دوم که مساله می‌تواند توسط Naive Bayes حل شود روش ارائه شده هم مانند Naive Bayes عمل می‌کند. این بدان معناست که این روش در حالات اول و دوم به نوعی یک گسترش برای Naive Bayes است، و در (حداقل برخی از) حالاتی که Bayes، خیلی خوب کار نمی‌کند نتایج خوبی می‌دهد که با مثال‌هایی در دو بخش بعدی به بررسی این موضوع خواهیم پرداخت.
برای این منظور فرض کنید برای هر دسته c یک پردازه مناسب مانند M_c با استفاده از S_c تشکیل شده باشد. در نتیجه $P(d|S_c) = P(d|M_c)$ و $P(S_c|d) = P(M_c|d)$ همچنین $\forall M_c; n_d << n_{S_c}$ یعنی اندازه مجموعه‌های آموزشی بسیار بزرگ‌تر از اندازه یک داده‌است و در نتیجه می‌توانیم فرض‌کنیم $\forall i, j, c, M : n^c_{i,j,d} \ll n^c_{i,j,M}$. فرض فوق از آن‌جا ناشی می‌شود که ما در حال مقایسه شرایطی هستیم که در آن‌ها Naive Bayes خوب کار می‌کند.
همان‌گونه که اشاره شد در بسیاری از کاربردها، احتمال وقوع یک دسته مشخص نیست و بنابراین از توزیع یکنواخت برای آن استفاده می‌کنند. طبق رابطه Bayes داریم

$$P(M_c|d) = \frac{P(d|M_c)P(M_c)}{P(d)}$$

و به دلیل این‌که عبارت $P(d)$ در همه $P(M_c|d)$ ظاهر می‌شود و همچنین با برابر در نظر گرفتن همه $P(M_c)$ ها داریم:

$$P(M_c|d) \approx P(d|M_c)$$

فرض می‌کنیم داده d و تعدادی دسته (که توسط پردازه‌های مارکف توصیف شده‌اند) در اختیار داریم. می‌خواهیم از بین دسته‌های موجود دسته‌ای را به عنوان دسته‌ای که d بدان تعلق دارد اعلام کنیم.
طبق روش Bayes باید احتمال $P(M_c|d)$ را به ازای همه دسته‌ها محاسبه کنیم و سپس دسته‌ای که احتمال آن بیشتر بود را به عنوان جواب اعلام کنیم، اما با توجه به داده‌های مساله داریم $P(M_c|d) \approx P(d|M_c)$ و در نتیجه کافی است به محاسبه $P(d|M_c)$ بپردازیم. روش مبتنی بر Entropy، به محاسبه Δ^c_E می‌پردازد. برای اثبات هم‌ارزی این دو روش کافی است ثابت کنیم

$$\Delta^c_E = f(P(d|M_C))$$

که در آن f تابعی اکیدا نزولی است. در این صورت به ازای هر دو دسته c و c' داریم:

$$\Delta^c_E < \Delta^{c'}_E \iff P(d|M_C) > P(d|M_{C'})$$

و در نتیجه $t = \arg\min(\Delta^c_E) \iff t = \arg\max(P(d|M_C))$

حالت اول

پردازه مارکف گسسته این حالت در شکل ۵ـ۱، آمده است. در این حالت خاص، پردازه در حقیقت احتمال وقوع هر ویژگی را مستقل از ویژگی‌های قبلی در نظر می‌گیرد و در حقیقت می‌توان از Naive Bayes نیز در این گونه مساله‌ها استفاده کرد.
برای محاسبه Δ_E^c نیاز به محاسبه Entropy برای M_c داریم و همان گونه که اشاره شد (در رابطه۵-۳) از رابطه زیر بدست می‌آید:

$$E_{Old} = E_M(S) = \log n(S) - \frac{1}{n(S)} \sum n_f(S) \log n_f(S)$$

$$E_{New} = E_M(S \cup \{d\}) = \log n(S \cup \{d\}) - \frac{1}{n(S) \cup \{d\}} \sum n_f(S \cup \{d\}) \log n_f(S \cup \{d\})$$

$$= \log (n(S) + n(d)) - \frac{1}{n(S) + n(d)} \sum (n_f(S) + n_f(d)) \log (n_f(S) + n_f(d))$$

و در نتیجه $\Delta_E(S, d)$ به صورت زیر محاسبه می‌شود:

$$\Delta_E^c(S,d) = \log(n(S)+n(d)) - \sum_f \frac{n_f(S)+n_f(d)}{n(S)+n(d)} \log(n_f(S)+n_f(d)) - \log n(S) + \sum_f \frac{n_f(S)}{n(S)} \log n_f(S)$$

$$= \log \frac{n(S)+n(d)}{n(S)} - \sum_f \left[\frac{n_f(S)+n_f(d)}{n(S)+n(S)} \log (n_f(S)+n_f(d)) - \frac{n_f(S)}{n(S)} \log n_f(S) \right]$$

با فرض این‌که $n(d) \ll n(S)$ داریم:

$$\frac{n(S)+n(d)}{n(S)} = ۱ + \frac{n(d)}{n(S)} \approx ۱$$

$$\Rightarrow \log \frac{n(S)+n(d)}{n(S)} \approx ۰$$

$$\Rightarrow \Delta_E^c(S,d) \approx - \sum_f \left[\frac{n_f(S)+n^f(d)}{n(S)+n(d)} \log (n_f(S)+n_f(d)) - \frac{n_f(S)}{n(S)} \log n_f(S) \right]$$

با توجه به فرض ما مبنی بر $N = n(S) \gg n(d)$ اگر داده و مجموعه‌آموزشی توسط یک مدل یکسان تولید شده باشند باید $n_f(S) \gg n_f(d)$ (این فرض در حالاتی است که Naive Bayes خوب کار می‌کند برقرار است) و در نتیجه می‌توان $\log (n_f(S) + n_f(d))$ را با $\log n_f(S)$ تخمین زد.

$$\Rightarrow \Delta_E^c(S,d) \approx -\sum_f \left[\left(\frac{n_f(S)+n_f(d)}{n(S)+n(d)} - \frac{n_f(S)}{n(S)}\right)\log n_f(S)\right]$$

$$= -\sum_f \left[\frac{n_f(d)n(S) - n_f(S)n(d)}{n_S\left(n(S)+n(d)\right)}\log n_f(S)\right]$$

$$= -\sum_f \left[\frac{n_f(d)n(S)}{n_S\left(n(S)+n(d)\right)}\log n_f(S) - \frac{n_f(S)n(d)}{n_S\left(n(S)+n(d)\right)}\log n_f(S)\right]$$

$$= -\sum_f \left[\frac{n_f(d)}{n(S)+n(d)}\log n_f(S) - \frac{n_f(S)n(d)}{n_S\left(n(S)+n(d)\right)}\log n_f(S)\right]$$

در عبارت دوم در رابطه بالا یعنی $\sum_f \frac{n_f(S)n(d)}{n_S(n(S)+n(d))}\log n_f(S)$ مقادیر $n(S)$ و $n(d)$ مستقل از مقدار f هستند و در نتیجه

$$\sum_f \frac{n_f(S)n(d)}{n_S\left(n(S)+n(d)\right)}\log n_f(S)$$

$$= \frac{n(d)}{n_S\left(n(S)+n(d)\right)}\sum_f n_f(S)\log n_f(S)$$

مقدار عبارت $\sum_f n_f(S)\log n_f(S)$ با شرط $\sum_f n_f(S) = n(S)$ در رابطه زیر صدق می‌کند:

$$n(S)\log n(S) - n(S)\log L \leq \sum_f n_f(S)\log n_f(S) \leq n(S)\log n(S)$$

$$\Rightarrow \frac{\log n(S) - \log L}{n(S)+n(d)} \leq \frac{\sum_f n_f(S)\log n_f(S)}{n_S\left(n(S)+n(d)\right)} \leq \frac{\log n(S)}{n(S)+n(d)}$$

فرض کردیم که $n(S)$، عدد بزرگی است. اگر $n(S)$ طوری بزرگ باشد که $\log n(S) \ll n(S)$ آن‌گاه خواهیم داشت:

$$\frac{\log n(S) - \log L}{n(S)+n(d)} \leq \frac{\sum_f n_f(S)\log n_f(S)}{n_S\left(n(S)+n(d)\right)} \leq \frac{\log n(S)}{n(S)+n(d)}$$

$$\Rightarrow 0 \leq \frac{\sum_f n_f(S)\log n_f(S)}{n_S\left(n(S)+n(d)\right)} \leq 0$$

$$\frac{\sum_f n_f(S)\log n_f(S)}{n_S\left(n(S)+n(d)\right)} \approx 0$$

در کاربردهای عملی (به عنوان مثال مباحث الکترونیک) گوییم $a \ll b$ اگر $a < \frac{b}{۱۰}$ و در نتیجه اگر $n > ۶۴$ باشد شرط بالا برقرار است و در نتیجه می‌توان Δ_E^c را به صورت زیر تخمین زد:

$$\Delta_E^c(S,d) \approx C - \frac{\sum_f n_f(d)\log n_f(S)}{n(S)+n(d)} + \frac{\sum_f n_f(S)\log n_f(S)n(d)}{n_S\,(n(S)+n(d))}$$
$$\approx C - \frac{\sum_f n_f(d)\log n_f(S)}{n(S)+n(d)}$$

و همچنین برای $P(d|M)$ داریم:

$$p(d|M) = \sum_{t\in T_d^M} \pi_{t_1} p(d|t)$$

از آن‌جا که در این نوع فرآیندها، فقط یک مسیر برای تولید بردار $d =< f_1, f_2, \cdots, f_l >$ وجود دارد خواهیم داشت:

$$p(d|t) = \prod_i p_{f_i}(S) = \prod_f p_f(S)^{n_f(d)}$$
$$\Rightarrow \log p(d|t) = \sum_f n_f(d)\log p_f(S)$$
$$= \sum_f n_f(d)\log\frac{n_f(S)}{n(S)} = \sum_f n_f(d)\log n_f(S) - \log n(S)\sum_f n_f(d)$$
$$= \sum_f n_f(d)\log n_f(S) - \log n(S)n(d) = \sum_f n_f(d)\log n_f(S) - C'$$
$$\Rightarrow \Delta_E^c(S,d) \approx C_1 - C_2\log\left(p(d|t)\right)$$

پس تابع f که به دنبال آن بودیم بدست آمده است و در نتیجه روش ارائه شده در پایان‌نامه در شرایط اشاره شده می‌تواند مانند Naive Bayes رفتار کند.

حالت دوم

فرآیند مارکف گسسته این مدل در شکل ۵ـ۲، آمده است. در این حالت اگر احتمال وقوع هر ویژگی مستقل از از ویژگی قبلی باشد و فقط به مکان آن ویژگی بستگی داشته باشد، می‌توان از Naive Bayes در این گونه مساله‌ها استفاده کرد. به این صورت که احتمال رخ‌دادن هر یک از دنباله ویژگی‌های با طول l را در هر فرآیند محاسبه کرد و سپس از Naive Bayes روی فرآیندهای حاصل (که در آن‌ها احتمال وقوع دنباله‌های lـتایی از هم مستقل هستند) استفاده کرد.
برای محاسبه Δ_E^c نیاز به محاسبه Entropy برای M_c داریم و همان گونه که اشاره شد از رابطه زیر بدست می آید:

$$E_{Old} = E_M(S) = \log n(S) - \frac{1}{l}\sum_i \sum_f \frac{n_i^f(S)}{n_i(S)} \log n_i^f(S)$$

$$E_{New} = E_M(S \cup \{d\}) = \log n(S \cup \{d\}) - \frac{1}{l}\sum_i \sum_f \frac{n_i^f(S \cup \{d\})}{n_i(S \cup \{d\})} \log n_i^f(S \cup \{d\})$$

$$= \log\left(n(S) + n(d)\right) - \frac{1}{l}\sum_i \sum_f \left(\frac{n_i^f(S) + n_i^f(d)}{n_i(S) + n_i(d)}\right) \log\left(n_i^f(S) + n_i^f(d)\right)$$

با توجه به این‌که d باید یک بردار lـتایی باشد ($n(d) = n_i(d) = ۱$)داریم:

$$E_{New} = \log\left(n(S) + 1\right) - \frac{1}{l}\sum_i \sum_f \left(\frac{n_i^f(S) + n_i^f(d)}{n_i(S) + 1} \log\left(n_i^f(S) + n_i^f(d)\right)\right)$$

و در نتیجه $\Delta_E(S, d)$ به صورت زیر محاسبه می‌شود:

$$\Delta_E^c(S, d) =$$

$$\log\left(n(S) + 1\right) - \frac{1}{l}\sum_i \sum_f \left(\frac{n_i^f(S) + n_i^f(d)}{n(S) + 1} \log\left(n_i^f(S) + n_i^f(d)\right)\right) - \log n(S) +$$

$$\frac{1}{l}\sum_i \sum_f \frac{n_i^f(S)}{n(S)} \log n_i^f(S)$$

$$= \log\frac{n(S)+1}{n(S)} - \frac{1}{l}\sum_i \sum_f \left[\frac{n_i^f(S) + n_i^f(d)}{n(S)+1} \log\left(n_i^f(S) + n_i^f(d)\right) - \frac{n_i^f(S)}{n(S)} \log n_i^f(S)\right]$$

$$= \log\frac{n(S)+1}{n(S)} - \frac{1}{l}\sum_i \sum_f \left[\frac{n_i^{f_i}(S) + 1}{n(S)+1} \log\left(n_i^{f_i}(S) + 1\right) - \frac{n_i^{f_i}(S)}{n(S)} \log n_i^{f_i}(S)\right]$$

اگر دنباله $d =< f_{d_۱}, f_{d_۲}, \cdots, f_{d_l} >$ باشد داریم:

$$\Delta_E^c(S, d) =$$

$$= \log\frac{n(S)+1}{n(S)} - \frac{1}{l}\sum_i \left[\frac{n_i^{f_{d_i}}(S) + 1}{n(S)+1} \log\left(n_i^{f_{d_i}}(S) + 1\right) - \frac{n_i^{f_{d_i}}(S)}{n(S)} \log n_i^{f_{d_i}}(S)\right]$$

با توجه به فرض ما مبنی بر بزرگ بودن $n(S)$ خواهیم داشت:

$$\log\frac{n(S)+1}{n(S)} \approx \log 1 = 0$$

و مانند بخش قبل اگر بتوان $\log\left(n_i^f(S)+1\right)$ را با $\log n_i^f(S)$ تخمین بزنیم می‌توان Δ_E^c را به صورت زیر تخمین زد:

$$\Delta_E^c = \log\frac{n(S)+1}{n(S)} + \frac{1}{l}\sum_i\left[\frac{n_i^{f_{d_i}}(S)}{n(S)}\log n_i^{f_{d_i}}(S) - \frac{n_i^{f_{d_i}}(S)+1}{n(S)+1}\log\left(n_i^{f_{d_i}}(S)+1\right)\right]$$

$$\approx 0 + \frac{1}{l}\sum_i\left[\frac{n_i^{f_{d_i}}(S)}{n(S)} - \frac{n_i^{f_{d_i}}(S)+1}{n(S)+1}\right]\log n_i^{f_{d_i}}(S)$$

$$= \frac{1}{l}\sum_i\frac{n_i^{f_{d_i}}(S)-n(S)}{n(S)\left(n(S)+1\right)}\log n_i^{f_{d_i}}(S)$$

$$= \frac{\sum_i n_i^{f_{d_i}}(S)\log n_i^{f_{d_i}}(S)}{l*n(S)\left(n(S)+1\right)} - \frac{\sum_i\log n_i^{f_{d_i}}(S)}{l\left(n(S)+1\right)}$$

می‌دانیم $0 \le n_i^{f_{d_i}}(S) \le n_i(S) = n(S)$ و درنتیجه $\sum_i^l n_i^{f_{d_i}}(S)\log n_i^{f_{d_i}}(S)$ مقداری در بازه $[n(S)\left(\log n(S)-\log l\right), n(S)\log n(S)]$ خواهد بود که در مقایسه با $n(S)^2$ قابل صرف نظر است. پس

$$\Delta_E^c \approx 0 - \frac{\sum_i\log n_i^{f_{d_i}}(S)}{l\left(n(S)+1\right)}$$

از طرفی $P(d|M)$ به علت فرض ما مبنی بر متفاوت بودن مجموعه ویژگی‌های ممکن در هر حالت ($F_i\cap F_j=\emptyset$) فقط یک مسیر $t =< t_1, t_2, \cdots, t_l >$ برای تولید d وجود دارد صورت زیر محاسبه می‌شود:

$$P(d|M) = P(d|t) = \prod_i p_i^{f_{d_i}}(S)$$

$$\Rightarrow \log P(d|M) = \sum\log\frac{n_i^{f_{d_i}}(S)}{n_i(S)} = \sum\log n_i^{f_{d_i}}(S) - \sum\log n_i(S)$$

$$= \sum\log n_i^{f_{d_i}}(S) - l\log n(S) = \sum\log n_i^{f_{d_i}}(S) - C$$

و در نتیجه:

$$\Delta_E^c \approx -C_1\log P(d|M)$$

تابع f اکیدا نزولی به صورت $f(x) = -C_1 x$ می‌باشد.

تعمیم حالت دوم

گاهی اوقات به علت ساختار دسته‌ها، نمی‌توان از ساختار مشابه حالت دوم استفاده کرد در این‌گونه موارد احتمال وقوع برخی ویژگی‌ها تنها وابسته به مکان آن ویژگی نیست بلکه می‌تواند بسته به ویژگی‌های دیگر تغییر کند. به عنوان مثال اگر بخواهیم در مورد عضویت بردارهای $l = ۵$ عنصری در یک دسته که به صورت زیر تعریف شده است فرآیندی مانند حالت دوم ارائه دهیم احتمالا به مشکل برخواهیم خورد:

$$l \in C \leftrightarrow (l_1 = f_1) \vee (l_2 = f_2)$$

احتمال رخ‌دادن ویژگی $f_۲$ در مکان دوم یک بردار در C وابسته به مقدار $l_۱$ است بدین معنا که اگر $l_۱ \neq f_۱$ باشد آن‌گاه باید داشته‌باشیم $l_۲ = f_۲$ (ویژگی $f_۲$ با احتمال ۱ رخ خواهد داد) ولی در صورت $l_۱ = f_۱$ نمی‌توان در مورد احتمال وقوع ویژگی $f_۲$ در مکان دوم ادعایی کرد. برای رفع این مشکل می‌توان از ساختاری مشابه حالت دوم ولی با کمی تغییرات استفاده کنیم، به این صورت که فرآیند دارای دورهایی با طول l باشد و فقط یک راس مانند $(S_۰)$ موجود باشد که در همه دورها مشترک است. در صورت انتقال از هر یال (عبور از هر یال) کاراکتر (ویژگی منسوب به آن یال) در رشته ظاهر می‌شود. در حقیقت راس $S_۰$ به عنوان نقطه شروع هر بردار در نظر گرفته شود و هر مسیر شامل l یال با شروع از هر حالتی باید از $S_۰$ عبور می‌نماید. برای تعیین احتمال انتقال بین حالات بر اساس مجموعه‌آموزشی نیز به صورت زیر عمل می‌کنیم:

- از حالت $S_۰$ با توجه به کارکتر مربوط به هر یال به حالت مناسب بعدی برو.
- به علت شروع از حالت $S_۰$ و وجود دورهایی به طول l در انتها به راس $S_۰$ بازگردیم

اگر $n_{i,j}$ را برابر با تعداد دفعاتی که از حالت S_i به حالت S_j برای تولید کل مجموعه‌آموزشی منتقل شویم و در صورتی که فرآیند مارکف گسسته‌ای با خصوصیات بالا در نظر بگیریم خواهیم داشت:

$$\forall i : \sum_j n_{i,j} = \sum_j n_{j,i}$$

در این حالت خاص خواهیم داشت:

$$\pi_i = \frac{n_i}{\sum_j n_j}$$

همچنین گوییم دنباله $V =< v_1, \cdots, v_l >$ در مکان iـام رشته نامتناهی T تولید شده توسط یک فرآیند ظاهر شده است اگر بردار $< T_i, T_{i+1}, \cdots, T_{i+l-1} >$ برابر با یکی از l بردار $< v_1, v_2, \cdots, v_l >, < v_2, v_3, \cdots, v_l, v_1 >, \cdots, < v_l, v_1, \cdots, v_{l-1} >$ باشد. بنابر تعریف فوق احتمال تولید شدن دنباله V توسط یک فرآیند مارکف را می‌توان به صورت زیر محاسبه کرد (با فرض این‌که ساختار فرآیند به گونه‌ای باشد که هر بردار با شروع از هر حالت فقط به یک طریق قابل تولید باشد):

$$P(V|M) = \sum_{t \in T} \pi_{t_0} \prod_i p_{t_i, t_{i+1}}^{v_i}$$

به علت فرض بالا مبنی بر یکتا بودن مسیر تولید یک رشته در صورت شروع از یک حالت و همچنین فرض قبلی ما در حالت دوم مبنی بر متفاوت بودن مجموعه ویژگی‌های قابل قبول در هر مکان خواهیم داشت:

۱. اگر V توسط M قابل تولید باشد، الزاما یک مسیر با شروع از S_0 نیز وجود دارد که V را تولید کند (با توجه به تعریف). این مسیر را با $T(V, M)$ نشان می‌دهیم.

۲. سایر اعضای T را می‌توان با جابه‌جاکردن بردار T_M^V به صورت دایره‌ای بدست آورد.

$$\begin{aligned} P(V|M) &= \sum_i \pi_{T(V,M)_i} \prod_j p_{T(V,M)_{(i+j) mod l}, T(V,M)_{(i+j+1) mod l}}^{V_j} \\ &= \prod_j p_{T(V,M)_j, T(V,M)_{j+1}}^{V_j} \sum_i \pi_i \\ &= \prod_j p_{T(V,M)_j, T(V,M)_{j+1}}^{V_j} \end{aligned}$$

با در دست بودن Π می‌توان به صورتی مشابه حالت دوم ثابت کرد که $P(V|M)$ با $\Delta_E(S, V)$ نسبت معکوس دارد.
در حقیقت این نوع فرآیندها برای هماهنگ سازی کار با روش‌های تعمیم Bayes که در بخش ۴ـ۴ معرفی شده بود می‌باشد. در فصل ۶ به یک نمونه از این فرآیندها در کاربرد عملی اشاره می‌شود.

حالت‌های سوم و چهارم

در این حالت، به علت تغییر احتمال رخ‌دادن ویژگی‌ها بر حسب ویژگی قبلی نمی‌توان از روش Naive Bayes انتظار عمل کرد خوبی داشت (به علت فرض Naive Bayes مبنی بر مستقل

بودن ویژگی‌ها) اما می‌توان حدس زد که الگوریتم ارائه شده در این پایان‌نامه، بتواند در این موارد نیز از کارآیی خوبی برخوردار باشد. علت این حدس این است که این روش بر خلاف Naive Bayes می‌تواند با مدل‌ها و فرآیندهای پیچیده‌تری منطبق شود. در حقیقت می‌توان گفت $\Delta_E(S,d)$ یک تخمین برای $P(d|M)$ است.

۵-۷ کارآیی

No Free Lunch Theorem یک قضیه در زمینه بهینه‌سازی ترکیبیاتی[۷۱] است که توسط دو فیزیک‌دان با نام‌های David H. Wolpert و Willian G. Macready ثابت شده است[۱۵]. در این قضیه ادعا شده است:
کارآیی همه الگوریتم‌هایی که یک نقطه کمینه و یا بیشینه[۷۲] از یک تابع را جستجو می‌کنند هنگامی که روی همه توابع در نظر گرفته شوند با هم برابر است[۳۲]. در مساله دسته‌بندی داده‌ها، در حقیقت به دنبال یافتن یک نقطه کمینه برای تابع Err تعریف شده در بخش ۱ـ۳ هستیم و در نتیجه NFLT برای مساله دسته‌بندی قابل اعمال است، و در نتیجه هر الگوریتم دلخواهی، از جمله الگوریتمی که همیشه دسته ۱ را به عنوان برنده اعلام می‌کند و یا الگوریتمی که به ازای هر ورودی به صورت تصادفی یک دسته را اعلام می‌کند درروی کلیه توابع ممکن از کارآیی یکسانی برخوردار هستند. از NFLT می‌توان نتیجه گرفت که هیچ دو الگوریتم دسته‌بندی قابل مقایسه با هم نیستند یعنی نمی‌توان گفت که الگوریتم A در حالت کلی بهتر از الگوریتم B است. این بدان معناست که در صورت وجود یک مساله به طوری که A بهتر از B کار کند می‌توان مساله‌ای (شاید به صورت مصنوعی) ایجاد کرد که در آن B بهتر از A کار کند. البته توجه به این نکته ضروری است که مسائل عملی و کاربردی قسمت بسیار کوچکی از کل مسائل ممکن را تشکیل می‌دهند و بنابراین معمولا برای مقایسه دو الگوریتم از چند کاربرد عملی استفاده می‌شود.
برای نشان‌دادن کارآیی روش مبتنی بر تئوری اطلاعات (در حالت‌های اول و دوم)، با استفاده از دو بانک اطلاعاتی استاندارد موجود در وب به مقایسه نتایج Naive Bayes و الگوریتم فوق پرداختیم. بانک اول شامل مجموعه‌ای از نامه‌های الکترونیکی بیست گروه خبری[۷۳] که در آدرس http://people.csail.mit.edu/jrennie/20Newsgroups/ موجود است[۲۱] و به عنوان یک مجموعه آزمون استاندارد در زمینه دسته‌بندی متون[۷۴] شناخته شده است بود.

[71] Combinatorics Optimization

[72] Extremum

[73] 20 news groups

[74] Text Classification

بانک دوم مورد آزمون، مجموعه‌ای از داده‌های مصنوعی بود که با نام MONK معروف هستند و به عنوان یک مجموعه آزمون استاندارد در زمینه یادگیری ماشین شناخته شده است[۲۸]. در فصل‌های ۶ و ۷ به توضیح صورت مساله‌ها و نتایج بدست آمده روی هر یک از بانک‌ها پرداخته‌ایم.

۶ یک مثال

۶-۱ معرفی مساله

مساله MONK یک مساله دسته‌بندی داده‌هاست که برای دسته‌بندی ربات‌های مصنوعی بکار می‌رود. در این مساله ربات‌ها بر اساس شش ویژگی متفاوت خود توصیف می‌شوند[۲۸]:

$x_1 : \quad Head\_Shape \in \{Round, Square, Octagon\}$

$x_2 : \quad Body\_Shape \in \{Round, Square, Octagon\}$

$x_3 : \quad Is\_Smiling \in \{Yes, No\}$

$x_4 : \quad Holding \in \{Sword, Balloon, Flag\}$

$x_5 : \quad Jacket\_Color \in \{Red, Yellow, Green, Blue\}$

$x_6 : \quad Has\_Tie \in \{Yes, No\}$

این مساله شامل سه زیر مساله است که به صورت مستقل از هم و با یک عبارت منطقی مطرح می‌شوند. هر ربات به دسته توصیف شده توسط عبارت منطقی هر زیر مساله، متعلق است و یا نیست. در مساله به جای بیان عبارت منطقی که بر اساس آن دسته‌بندی صورت می‌گیرد شماری از ۴۳۲ ربات ممکن و دسته مربوط به هر یک را به ما داده‌اند و از ما خواسته شده تا تمام ۴۳۲ ربات را در یکی از دو دسته مناسب قرار دهیم. سه زیر مساله به صورت زیر هستند.

مساله ۱

در این مساله اعضای دسته + ربات‌هایی هستند که در عبارت زیر صدق می‌کنند:

$$(x_۱ = x_۲) \vee (x_۵ = Red)$$

از ۴۳۲ ربات ممکن، ۱۲۴تای آن‌ها به صورت تصادفی انتخاب شده‌اند و در مجموعه‌آموزش قرارگرفته‌اند. در این مساله هیچ‌گونه آشوب و یا دسته‌بندی اشتباه در مجموعه آموزش وجود ندارد.

مساله ۲

در این مساله اعضای دسته + ربات‌هایی هستند که دقیقا دو تا از شش ویژگی $(x_۱, \cdots, x_۶)$ آن مقدار نخست خود را اختیار کنند. از ۴۳۲ ربات ممکن، این بار ۱۶۹تای آن‌ها به صورت

تصادفی انتخاب شده‌اند و در مجموعه‌آموزش قرارگرفته‌اند. باز هم در این مساله هیچ‌گونه آشوب و یا دسته‌بندی اشتباه در مجموعه آموزش وجود ندارد.

مساله ۳

در این مساله اعضای دسته +، ربات‌هایی هستند که در عبارت زیر صدق کنند:

$$(x_۵ = Green \wedge x_۴ = Sword) \vee (x_۵ \neq Blue \wedge x_۲ \neq Octagon)$$

از ۴۳۲ ربات ممکن، ۱۲۲تای آن‌ها به صورت تصادفی انتخاب شده‌اند و در مجموعه‌آموزش قرارگرفته‌اند. این‌بار ٪۵ از اعضای مجموعه‌آموزش به اشتباه دسته‌بندی شده‌اند.
مساله ۱، یک مساله استاندارد به صورت DNF[۷۵] است و احتمالا همه الگوریتم‌های یادگیری نمادین مانند درخت‌های تصمیم باید به خوبی از پس آن برآیند. بر خلاف مساله ۱، مساله ۲ یک مساله کنترل‌کردن زوجیت است اما توصیف آن به صورت یک فرمول بسته DNF و یا CNF[۷۶] کمی دشوار است. مساله ۳، را به راحتی می‌توان به صورت رابطه DNF نوشت ولی این‌بار الگوریتم باید قابلیت دسته‌بندی براساس آشوب را داشته باشد.[۲۸]

۶-۲ تبدیل مساله به مدل مارکف مربوطه

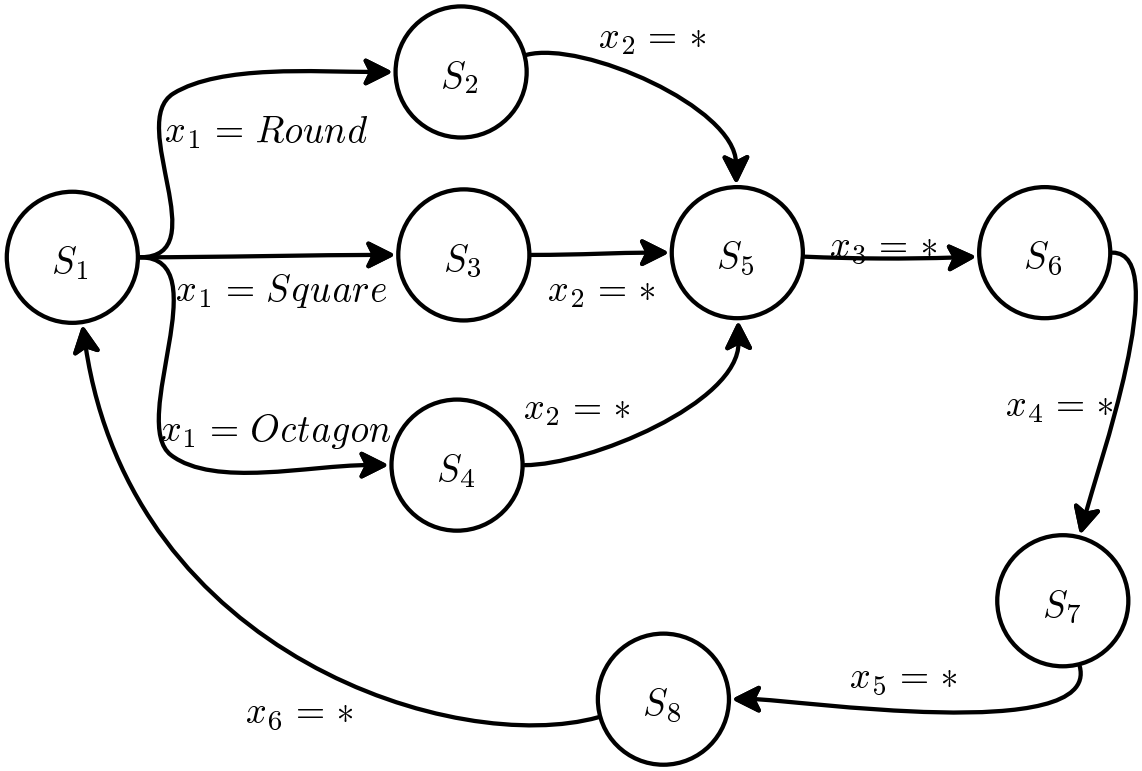
برای حل هر یک از زیر مساله‌های مطرح شده در بالا از یک فرآیند نوع ۲ با جزئیات متفاوت (بسته به رابطه دسته‌بندی) استفاده شد.

مساله ۱

برای حل این مساله از فرآیند نشان داده شده در شکل ۶ـ۱ استفاده شد:
علت پیشنهاد داده شدن فرآیند ۶ـ۱ این بوده است که اعضای دسته +، دارای یک رابطه بین ویژگی های ۱ و ۲ خود ($x_۱$ و $x_۲$) و ویژگی ۵ ($x_۵$) هستند و در فرآیند پیشنهادی اگر داده‌ای در این دسته نباشد پس قطعا دارای $x_۱$ و $x_۲$ متفاوتی خواهد بود و این موضوع باعث می‌شود از یکی از راس‌های $S_۲$، $S_۳$ و یا $S_۴$ یال جدیدی خارج شود که می‌توان انتظار افزایش

[75] Disjunctive Normal Form

[76] Conjunctive Normal Form

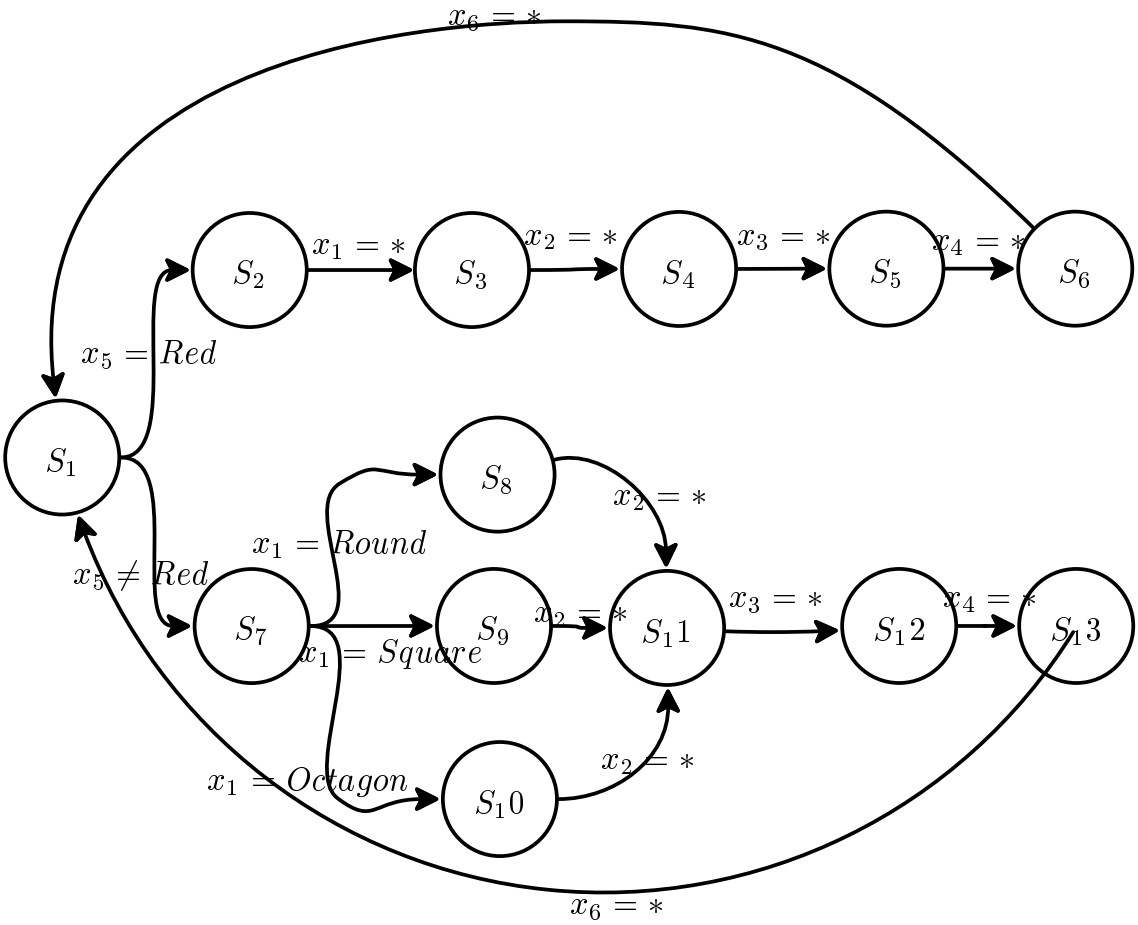


شکل ۶ـ۱: یک فرآیند پیشنهادی برای مساله ۱

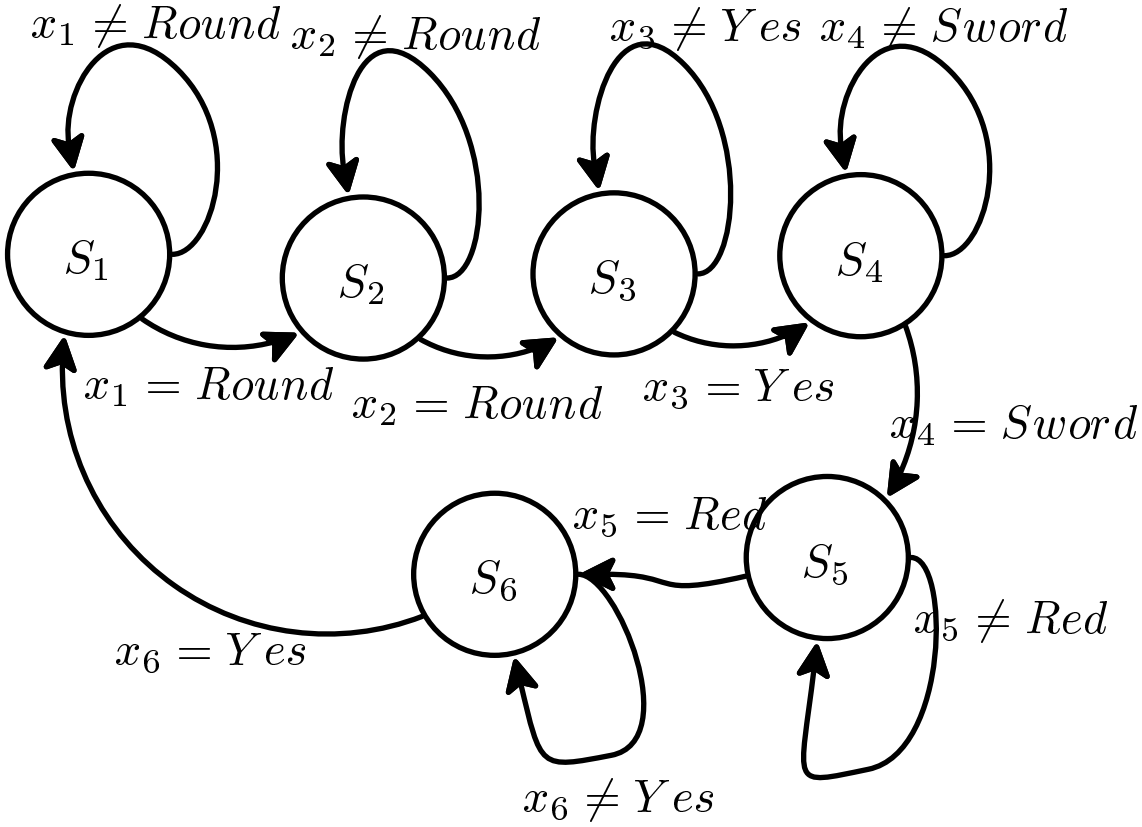
آشفتگی در بین احتمالات باشد اما به دلیل وجود داده‌هایی در این دسته که دارای x_1 و x_2 متفاوت هستند این فرآیند ممکن است کارآیی خوبی نداشته باشد. فرآیند ۶ـ۲ فرآیند دقیق‌تری برای توصیف این دسته‌ها می‌باشد.

مساله ۲

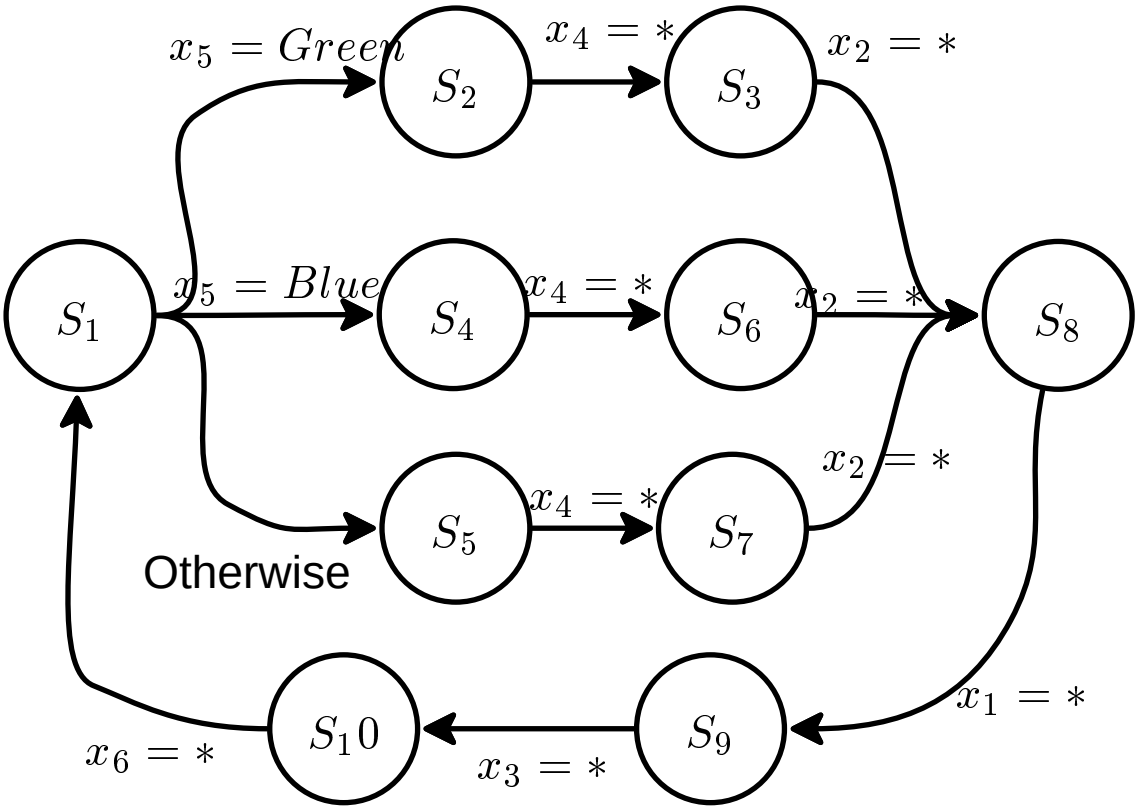
برای حل این مساله از فرآیند نشان داده شده در شکل ۶ـ۳ استفاده شد:
علت پیشنهاد داده شدن این فرآیند این بوده است که اعضای دسته +، دارای دقیقا دو ویژگی با مقدار نخست خود هستند و در نتیجه تعداد دفعاتی که فرآیند برای تولید اعضای مجموعه‌آموزش وارد حالات S_4 S_5 و S_6 شده است برابر با صفر است و در نتیجه با ورود داده‌ای که سه و یا بیشتر ویژگی آن دارای ویژگی نخست باشد Entropy فرآیند را بالا می‌برد همچنین اگر داده‌ای صفر و یا یک ویژگی با مقدار نخست داشته باشد تعداد ورود در حالت S_1 و یا S_2 (n_1 و n_2) تغییر نامتناسبی نسبت به داده‌های مجموعه آموزش خواهد کرد و در نتیجه Entropy تغییرات مثبتی خواهد داشت.
برای این مساله فرآیندهای پیچیده‌تری را نیز می‌توان ارائه داد ولی هیچ تضمینی برای بهتر



شکل ۶ـ۲: یک فرآیند دیگر برای مساله ۱



شکل ۶ـ۳: یک فرآیند پیشنهادی برای مساله ۲



شکل ۶ـ۴: یک فرآیند پیشنهادی برای مساله ۳

کارکردن این فرآیندها وجود ندارد.

مساله ۳

برای حل این مساله از فرآیند نشان داده شده در شکل ۶ـ۴ استفاده شد:
علت پیشنهاد داده شدن این فرآیند این بوده است که اعضای دسته +، دارای نوعی محدودیت روی ویژگی های ۲، ۴ و ۵ خود (x_2، x_4 و x_5) خود است و در فرآیند پیشنهادی اگر داده‌ای در این دسته نباشد احتمالا آشفتگی زیادی در بین احتمالات ایجاد می‌کند و در نتیجه Entropy را افزایش می‌دهد.

۶-۳ بحث در مورد تخمین‌ها

همان‌گونه که در فصل ۵ شرح داده شد در صورت وجود شرایط خاص، می‌توان از ITDC انتظار کارآیی خوبی داشت (یعنی به خوبی Naive Bayes عمل کند). به همین دلیل در این

بخش به بررسی برقرار بودن آن شرایط در هر یک از سه مساله بالا می‌پردازیم.
فرض اولیه در فصل ۵، بسیار بزرگ‌تر بودن اندازه مجموعه آموزش نسبت به داده‌آزمون بوده است ($n(s) \gg n(d)$). همان‌طور که شرح داده شد $n(s) = ۱۲۴ * ۶$ بوده در حالی که $n(d) = ۱ * ۶$ و در نتیجه می‌توان با دقت خوبی فرض کرد که شرط فوق برقرار است.

مساله ۱

در این گونه از فرآیندهای مارکف، شرط اشاره شده در بخش ۶ـ۵، باید شرط زیر برقرار باشد:
$\log n_c(S) = \log n_c(d)$: این شرط نیز با توجه به داده‌های موجود در مجموعه آموزشی و با توجه به این واقعیت که $n_c(d) \leq ۱$ قابل ارضا می‌باشد.

مساله ۲

برای فرآیند مارکف از نوع دوم که در فصل ۵ بدان‌ها اشاره شد، باید دو شرط زیر برقرار باشد:

۱. $\log n_c(S) = \log n_c(d)$: این شرط نیز با توجه به داده‌های موجود در مجموعه آموزشی و با توجه به این واقعیت که $n_c(d) \leq ۱$ قابل ارضا می‌باشد.

۲. $\log \frac{(n(S)+۱)}{n(S)} \approx ۰$: با توجه به $n(S) = ۱۶۹$ خواهیم داشت:

$\log \frac{170}{169} = 0.0085 \approx 0$

مساله ۳

برای فرآیند مارکف مورد استفاده در این مساله نیز مانند مساله دوم، باید دو شرط زیر برقرار باشد:

۱. $\log n_c(S) = \log n_c(d)$: این شرط نیز با توجه به داده‌های موجود در مجموعه آموزشی و با توجه به این واقعیت که $n_c(d) \leq ۱$ قابل ارضا می‌باشد.

۲. $\log \frac{(n(S)+۱)}{n(S)} \approx ۰$: با توجه به $n(S) = ۱۲۲$ خواهیم داشت:

$\log \frac{123}{122} = 0.0118$

که اگر ۵٪ خطای موجود در مجموعه‌آموزشی را در نظر نگیریم این عدد بزرگ‌تر نیز می‌شود. اما با این حال ITDC در این مساله نیز جواب قابل قبولی داده است.

۶-۴ نتایج حاصل

ستون اول جدول زیر نشان دهنده میزان دقت الگوریتم دسته‌بندی بر اساس تئوری اطلاعات (ITDC) است که در این پایان‌نامه معرفی شد و در ستون‌های بعدی دقت چند الگوریتم دیگر آمده است. این اطلاعات از [۲۸] استخراج شده است. باید توجه داشت که احتمالا فرآیندهای پیشنهادی بهترین فرآیندهای ممکن نبودند. یکی از کارهای پیشنهادی در ادامه این پایان‌نامه ایجاد خودکار فرآیند با استفاده از مجموعه‌آموزش است که در فصل ۸ بدان اشاره شده است.

Problem	ITDC	Decision Trees	Naive Bayes
M1	100%	100%	75%
M2	94%	81%	61%
M3	100%	100%	97%

جدول بالا نشان می‌دهد که ITDC از Naive Bayes حداقل در برخی موارد از کارآیی بهتری برخوردار است. دلیل این برتری در این مساله، ذاتا تصادفی نبودن و همچنین صدق نکردن مساله در شرط استقلال ویژگی‌ها است که از کارآیی Naive Bayes کاسته است.

۷ یک مثال

۷-۱ معرفی مساله

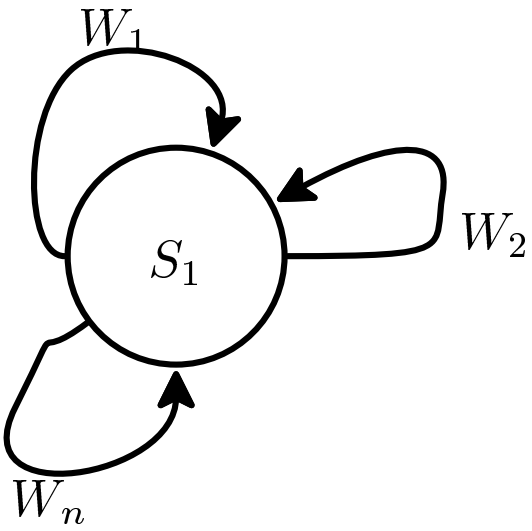
بانک 20 NewsGroup از حدود ۲۰۰۰۰ متن e-mail که از بین ۲۰ گروه خبری متفاوت جمع‌آوری شده‌اند تشکیل شده‌است. این بانک توسط Ken Lang به عنوان یک مجموعه‌آزمون برای مقاله خود با نام Newsweeder: Learning to filter netnews تهیه شده است. این مجموعه امروزه به عنوان یک مجموعه استاندارد در آزمایشات و اندازه‌گیری کیفیت الگوریتم‌ها استفاده می‌شود. کاربرد اصلی این مجموعه در دسته‌بندی متن‌ها[۷۷] و کاربردهای آن در هوش‌مصنوعی است.
۲۰ گروه خبری مورد استفاده در این مجموعه، موضوعات متفاوت و متنوعی را در برمی‌گیرد. بعضی از موضوعات به صورت زیادی مشابه هستند مانند comp.sys.ibm.pc.hardware و comp.sys.mac.hardware، و برخی دیگر کاملا متفاوت به نظر می‌رسند مانند misc.forsale و soc.religion.christian. در جدول زیر این ۲۰ گروه بر حسب ارتباط در موضوعات مورد بحث در آن‌ها آورده شده‌اند[۲۲].

comp.graphics comp.os.ms-windows.misc comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	rec.autos rec.motorcycles rec.sport.baseball rec.sport.hockey	sci.cryp sci.electronics sci.med sci.space
misc.forsale	talk.politics.guns talk.politics.mideast talk.politics.misc	talk.religion.misc alt.atheism soc.religion.christian

۷-۲ تبدیل مساله به مدل مارکف مربوطه

به علت گسترده بودن دامنه واژگان موجود در مجموعه (حدود واژه) از یک فرآیند بدون حافظه مانند شکل ۷ـ۱ برای مدل کردن فرآیند تولید این داده‌ها استفاده شد. در پیاده‌سازی هر واژه به عنوان یک ویژگی در نظر گرفته شد و برای راحت‌تر کردن پیاده‌سازی به هر واژه یک عدد طبیعی به عنوان کد آن واژه اختصاص داده شد. برای محدود کردن تعداد

[77] Text Classification



شکل ۷ـ۱: فرآیند پیشنهادی برای مساله دسته‌بندی خبرها

ویژگی‌ها، واژه‌هایی که دارای طولی بیشتر از یک مقدار از پیش تعیین شده بود (در پیاده‌سازی ما ۱۵) با صفر (یک ویژگی که برای این‌منظور در نظر گرفته شد) نمایش داده شد.

۷-۳ نتایج حاصل

برای بدست آوردن نتیجه بهتر، از روش چند سطحی استفاده شد. به این ترتیب که عمل تشخیص در دو مرحله انجام می‌گیرد. در مرحله اول، داده به یک گروه از دسته‌ها بین چندین گروه تخصیص داده می‌شود و سپس در مرحله بعد در مورد دسته نهایی بین اعضای آن گروه تصمیم‌گیری می‌شود. رده بندی گروه‌ها به صورت نمایش داده شده در جدول زیر بوده است (مانند [۳۳]):

Level 1	Level 2	Level3
alt.atheism misc.forsale soc.religion.christian		
comp.	graphics os.ms-windows.misc windows.x	
	sys.	ibm.pc.hardware mac.hardware
rec.	autos motor-cycle	
	sport.	baseball hockey
sci.	crypt electronics med space	
talk	religion.misc	
	politics.	guns mideast misc

متوسط دقت دسته‌بندی در جدول پایین با چند روش دیگر از [۲۱] مقایسه شده است:

Method	Accuracy
Information Theoretic	88.0%
H-SVN	89.1%
MNB	84.8%
TWCNB	86.1%
SVM	86.2%

در این جدول میزان دقت در دسته‌بندی روی هر یک از دسته‌ها را نشان می‌دهد:

Group Name	Accuracy
alt.theism	74%
comp.graphics	79%
comp.os.ms-windows.misc	70%
comp.sys.ibm.pc.hardware	67%
comp.sys.mac.hardware	80%
comp.windows.x	80%
misc.forsale	85%
rec.autos	63%
rec.motorcycles	89%
rec.sport.baseball	92%
rec.sport.hockey	95%
sci.crypt	83%
sci.electronics	80%
sci.med	91%
sci.space	96%
soc.religion.christian	70%
talk.politics.guns	93%
talk.politics.mideast	90%
talk.politics.misc	84%
talk.religion.misc	87%

۸ کارهای آتی

به علت محدود بودن تعریف این پایان‌نامه به دسته‌بندی داده‌های متنی، تمرکز اصلی روی منابع بدون حافظه و Naive Bayes بود و الگوریتم بدست آمده در طول تحقیق فقط در همین زمینه‌ها آزمایش شد. تعمیم ریاضی این روش در حالاتی که منبع دارای حافظه است و مقایسه آن با روش‌های Bayes و سایر روش‌ها و همچنین انجام یک‌سری مقایسه عملی برای مشاهده کارآیی الگوریتم ارائه شده در حالات حافظه دار را می‌توان به عنوان اصلی‌ترین کار آینده معرفی کرد.
یکی از مشکلاتی که روش پیشنهادی (ITDC) با آن روبرو است نیاز به طراحی یک فرآیند دقیق و کارا برای هر مساله (هر دسته) است. همان‌طور که در فصل ۶ دیدیم کارایی ITDC وابسته به فرآیند تغییر می‌کند. یکی دیگر از کارهای مرتبط با این پایان‌نامه را می‌توان ایجاد یک ساختار در مرحله آموزش برای پیشنهاد دادن فرآیندهایی بر اساس مجموعه‌آموزشی تعریف کرد. بدین صورت که در ابتدا فرآیند یک فرآیند ساده باشد و با افزوده شدن داده‌های جدید آموزشی به یک دسته فرآیند پیچیده‌تر و در عین حال اختصاصی‌تر برای هر دسته گردد ولی باید توجه داشت که فرآیند بسیار وابسته به مجموعه‌آموزشی نگردد.

References

[1] J. Aczel, B. Forte, and C. Ng, *Why the Shannon and Hartley Entropies Are'Natural'*, Advances in Applied Probability, 6 (1974), pp. 131–146.

[2] P. Ahammad, K. Daskalakis, O. Etesami, and A. Frome, *Claude Shannon and A Mathematical Theory of Communication.*

[3] P. Bergner, *Stochastic processes and applications in biology and medicine. Part 1Theory. Part 2Models*, Bulletin of Mathematical Biology, 36 (1974), pp. 607–610.

[4] S. Boucheron, O. Bousquet, and G. Lugosi, *Theory of classification: a survey of some recent advances*, ESAIM P AND S, 9 (2005), p. 323.

[5] T. Cover and J. Thomas, *Elements of information theory*, Wiley New York, 1991.

[6] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*, Springer, 1996.

[7] M. D.J.C, *Information Theory, Inference and Learning Algorithm*, Cambridge University Press, 2005.

[8] B. Forte and C. Sastri, *Is something missing in the Boltzmann entropy?*, Journal of Mathematical Physics, 16 (2006), pp. 1453–1456.

[9] K. Fukunaga and D. Hummels, *Bayes error estimation using Parzen and k-NN procedures*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 9 (1987), pp. 634–643.

[10] D. Gao, M. Madden, D. Chambers, and G. Lyons, *Bayesian ANN classifier for ECG arrhythmia diagnostic system: a comparison study*, Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on, 4 (2005).

[11] S. Goldstein, *Remarks on the Global Markov Property*, Commun. math. Phys, 74 (1980), pp. 223–234.

[12] R. Gray, *Entropy and information theory*, Springer-Verlag New York, Inc. New York, NY, USA, 1990.

[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall PTR Upper Saddle River, NJ, USA, 1998.

[14] C. Hillman, *An Entropy Primer*, tech. rep., Mimeo retrieved from URL:¡ http://www. math. washington. edu/hillman, 1996.

[15] Y. Ho and D. Pepyne, *Simple Explanation of the No-Free-Lunch Theorem and Its Implications*, Journal of Optimization Theory and Applications, 115 (2002), pp. 549–570.

[16] R. Mason, D. Lind, and W. Marchal, *Statistics: an introduction*, Harcourt Brace Jovanovich, 1983.

[17] D. Michie, D. Spiegelhalter, and C. Taylor, *Machine learning of rules and trees*, Machine Learning, Neural and Statistical Classification, (1994).

[18] T. Mitchell, *Machine Learning*, McGraw-Hill Higher Education, 1997.

[19] M. Nelson and J. Gailly, *The Data Compression Book*, MIS: Press New York, NY, USA, 1995.

[20] J. Quinlan, *Bagging, Boosting, and C4. 5.*

[21] J. Rennie, *Improving Multi-class Text Classification with Naive Bayes*, (2001).

[22] J. Rennie, L. Shih, J. Teevan, and D. Karger, *Tackling the Poor Assumptions of Naive Bayes Text Classifiers*, Proceedings of the Twentieth International Conference on Machine Learning, 41 (2003), p. 18.

[23] O. B. S. Boucheron and G. Lugosi, *Theory of classification: A survey of some recent advances*, (2001).

[24] S. Safavian and D. Landgrebe, *A survey of decision tree classifier methodology*, IEEE Transactions on Systems, Man, and Cybernetics, 21 (1991), pp. 660–673.

[25] P. Sahoo, S. Soltani, A. Wong, and Y. Chen, *A survey of thresholding techniques*, Computer Vision, Graphics, and Image Processing, 41 (1988), pp. 233–260.

[26] C. Shannon, *A mathematical theory of communication*, The Bell Syst Tech J, 27 (1948), pp. 379–423.

[27] F. Thollard, P. Dupont, and C. de la Higuera, *Probabilistic DFA inference using Kullback-Leibler divergence and minimality*, Proc. Int. Conf. on Machine Learning, (2000), pp. 975–982.

[28] S. Thrun et al., *The MONK's Problems: A Performance Comparison of Different Learning Algorithms*, School of Computer Science, Carnegie Mellon University, 1991.

[29] S. van de Geer, *Empirical Processes in M-Estimation*, Cambridge University Press, 2000.

[30] A. Wehrl, *General properties of entropy*, Reviews of Modern Physics, 50 (1978), pp. 221–260.

[31] D. Williams, *Weighing the Odds: A Course in Probability and Statistics*, Cambridge University Press, 2001.

[32] D. Wolpert, W. Macready, I. Center, and C. San Jose, *No free lunch theorems for optimization*, Evolutionary Computation, IEEE Transactions on, 1 (1997), pp. 67–82.

[33] Y. Yoon, C. Lee, and G. Lee, *Systematic Construction of Hierarchical Classifier in SVM-based Text Categorization.*

[34] J. Zurada, *Introduction to artificial neural systems*, West Publishing Co. St. Paul, MN, USA, 1992.