



## یک روش مبتنی بر اتوماتاهای یادگیر برای آموزش جداول احتمال شرطی در شبکه های بیزی

محمد رضا میبیدی

آزمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

تهران ایران

mmeybodi@aut.ac.ir

نبی الله رضوانی

آزمایشگاه محاسبات نرم

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

تهران ایران

nabi.rezvani@gmail.com

میشود و مقادیر نامعلوم در مثال ها بر مبنای وضعیت شبکه و میزان نزدیکی آن به توزیع احتمالی توام حاکم بر مثال های آموزشی، استنتاج می گردد [10]. با وجود اینکه الگوریتم MLE عموماً در برآورد توزیع احتمالی موجود در مثال های آموزشی خوب عمل می کند، دارای اشکالاتی نیز می باشد. برای مثال هنگامی که در مثال های آموزشی نویز وجود داشته باشد، این الگوریتم برای همگرایی به توزیع احتمالی مطلوب نیازمند در اختیار داشتن مثال های آموزشی نسبتاً زیادی است. همچنین شبکه بیزی آموزش داده شده، پس از مرحله آموزش قابل به روز رسانی با مثال های جدید نیز نمی باشد.

اتوماتاهای یادگیر یک مدل یادگیری تقویتی برای بدست آوردن سیاست بهینه می باشد و بر اساس بازخورد محیط کار می کنند. از اتوماتا های یادگیر به عنوان یک مدل غیر قطعی برای جستجوی پارامتر های مدل های محاسباتی استفاده شده است. استفاده در بهینه ساز توابع [11]، به عنوان دسته بندی کننده [7]، و تنظیم کننده پارامتر های ماژول بینایی [3] مثال هایی از این کاربرد ها هستند. در این مقاله یک روش مرحله به مرحله مبتنی بر اتوماتاهای یادگیر برای آموزش جداول احتمال شرطی شبکه های بیزی پیشنهاد می گردد. نتایج آزمایشها نشان می دهند که الگوریتم پیشنهادی با تعداد مثال های آموزشی کمتری دارای دقت دسته بندی مشابه با الگوریتم های کلاسیک مانند برآورد امکان پیشنهادی عمل آموزش و تست به صورت همزمان پیش می رود و پس از آموزش با هر مثال شبکه آماده تست می باشد در حالیکه در الگوریتم MLE باید فاز آموزش پایان یابد تا بتوان از شبکه به عنوان یک دسته بندی کننده استفاده کرد. البته بایستی ذکر شود در الگوریتم پیشنهادی برای بدست آوردن نتیجه مناسب باید تعداد قابل قبولی از مثال ها توسط شبکه مشاهده شده باشند. علاوه بر این در الگوریتم پیشنهادی به ازای هر مثال آموزشی عمل به روز رسانی احتمالات در جداول انجام میشوند در حالیکه در الگوریتم MLE به ازای هر سطر از هر جدول باید کلیه مثال های آموزشی پیمایش شوند و نتیجتاً الگوریتم

**چکیده:** شبکه های بیزی به عنوان یک دسته بندی کننده در کاربرد های مختلف یادگیری ماشین مورد استفاده قرار گرفته اند. یکی از مسایل مهم در شبکه های بیزی، آموزش شبکه های بیزی بهینه بر اساس مثال های آموزشی دردسترس می باشد. آموزش شبکه های بیزی شامل آموزش مقادیر احتمالی موجود در جداول احتمال شرطی و نیز آموزش ساختار شبکه می باشد. در این مقاله یک روش مرحله به مرحله مبتنی بر اتوماتاهای یادگیر برای آموزش جداول احتمال شرطی شبکه های بیزی پیشنهاد می گردد. نتایج آزمایشها نشان می دهند که الگوریتم پیشنهادی با تعداد مثال های آموزشی کمتری دارای دقت دسته بندی مشابه با الگوریتم های کلاسیک مانند برآورد امکان پیشنهادی می باشد.

**واژه های کلیدی:** شبکه های بیزی، آموزش، اتوماتاهای یادگیر

### ۱- مقدمه

شبکه های بیزی مدل های گرافیکی برای بیان یک توزیع احتمالی توام بر روی مجموعه ای از متغیر های تصادفی هستند [4]. از شبکه های بیزی بار ها به عنوان دسته بندی کننده استفاده شده است. یکی از مهم ترین چالش ها در ارتباط با دسته بندی کننده های بیزی، آموزش شبکه بر مبنای مثال های آموزشی موجود می باشد. در حالت کاملاً مشاهده پذیر (وقتی مقدار کلیه متغیر ها در تمام مثال های آموزشی معلوم است) از الگوریتم برآورد امکان پیشنهادی<sup>۱</sup> برای آموزش پارامتر های شبکه استفاده می گردد [9][4]. این الگوریتم که مبنای آن شمردن فرکانس رخداد وقایع مختلف در مثال های آموزشی است، نتایج نسبتاً خوبی را نیز ارائه می کند. مبنای این الگوریتم فرض انجام شده در دسته بندی کننده بیزی ساده<sup>۲</sup> می باشد که فرض می کند کلیه متغیر های تصادفی از یکدیگر مستقل هستند [8]. در حالت تاحدودی مشاهده پذیر، از تکنیک پیشینه سازی انتظار<sup>۳</sup> برای برآورد جداول احتمال شرطی استفاده می شود که یک روش مبتنی بر باز خورد می باشد [4]. در این روش از الگوریتمی چون MLE برای برآورد استفاده

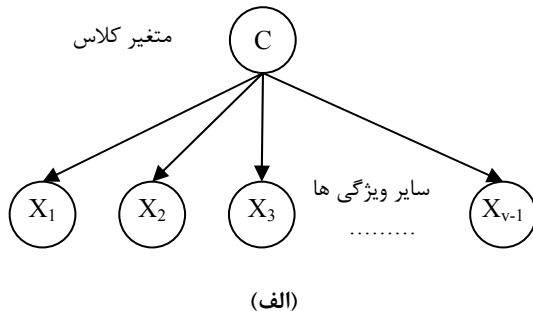
$$L = \frac{1}{M} \log \prod_{m=1}^M \Pr(D_m | G) = \frac{1}{M} \sum_{i=1}^n \sum_{m=1}^M \log P(X_i | P_a(X_i), D_m) \quad (2)$$

که  $Pa(X_i)$  مجموعه والدین نود  $X_i$  هستند. مشاهده می شود که تابع امتیاز دهی امکان لگاریتمی<sup>۸</sup> بر اساس ساختار گراف تجزیه شده و بنابراین می توان میزان تاثیر امکان لگاریتمی برای هر نود را به صورت جداگانه بیشینه کرد (با این فرض که نود ها، متغیر های مستقلی در نظر گرفته شوند).

همانطور که مشخص است، این رابطه با فرض استقلال شرطی همه متغیر ها، یک مقدار امکان محاسبه می کند که دقیقاً مشابه چیزی است که در مورد دسته بندی کننده های بیزی ساده صورت می گیرد [8]. این روش، برآورد امکان بیشینه نام دارد. برآورد امکان بیشینه روشی مبتنی بر شمردن فرکانس وقوع رخداد های مختلف در مثال های آموزشی است. علاوه بر این از فرض احتمال شرطی کلیه متغیر ها نیز استفاده می کند. بنا براین اولاً فرآیند آموزش شبکه و پرسش<sup>۹</sup> از آن دو فرآیند مستقل خواهند بود. از طرفی برای بدست آوردن نتایج قابل قبول باید تعداد مثال های آموزشی زیادی وجود داشته باشند. روش آموزش مبتنی بر اتوماتاهای یادگیر که در این مقاله شرح داده می شود، سعی می کند این نقایص را مرتفع نماید.

## ۲-۲ دسته بندی کننده های بیزی

اگر فرض کنیم هر مثال آموزشی یک بردار از خصیصه های  $(X_1, X_2, \dots, X_{v-1}, C)$  باشد، هدف دسته بندی پیش بینی صحیح یک مقدار از متغیر کلاس  $C = x_v$  با دانستن  $(x_1, x_2, \dots, x_{v-1})$  است. اگر معیار ارزیابی دقت دسته بندی (درصد پیش بینی های صحیح بر روی مثال های آزمایشی) باشد، پیش بینی بهینه برای  $(x_1, x_2, \dots, x_{v-1})$  کلاسی است که  $P(C | x_1, x_2, \dots, x_{v-1})$  را بیشینه می کند. اگر یک شبکه بیزی برای  $(x_1, x_2, \dots, x_{v-1}, C)$  داشته باشیم این احتمالات را می توان با استنتاج بر روی آن محاسبه کرد. به طور مثال دسته بندی کننده بیزی ساده یک شبکه بیزی است که در آن متغیر کلاس والدی ندارد و هر متغیر دیگر متغیر کلاس را به عنوان تنها والدش دارد. الگوریتم TAN فریدمن [13] یک اشتقاق از الگوریتم چو و لیو [14] است که در آن هر متغیر یک والد دیگر علاوه بر متغیر کلاس دارد. شکل ۱ بهتر این دسته بندی کننده ها را نشان می دهد.



پیشنهادی در مقایسه با الگوریتم MLE دارای پیچیدگی زمانی کمتری میباشد.

ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ شبکه های بیزی به اختصار شرح داده میشود. در بخش ۳ اتوماتاهای یادگیر معرفی می شوند. سپس در بخش ۴ روش آموزش پیشنهادی معرفی می گردد. در بخش ۵ نتایج پیاده سازی آورده شده و بخش ۶ نتیجه گیری میباشد.

## ۲- شبکه های بیزی

یک شبکه باور بیزی<sup>۴</sup> توزیع احتمالی حاکم بر مجموعه ای از متغیرها را با تعریف یک سری استقلال احتمالی و نیز یک سری احتمالات شرطی توصیف می کند [4]. به بیان دیگر یک شبکه باور بیزی یک توزیع احتمال توأم بر روی مجموعه ای از متغیر ها را تعریف می کند [2]. برای توصیف یک شبکه بیزی باید دو مورد را فراهم آوریم: توپولوژی (ساختار) گراف، و پارامتر ها یا همان جداول احتمال شرطی مربوط به تمام متغیر ها. با داشتن این دو مورد می توان توزیع احتمالی توأم حاکم بر متغیر های شبکه را نمایش داد. به بیان ساده می توان گفت که این توزیع احتمال توأم، برابر با حاصلضرب احتمال شرطی هر متغیر به شرط متغیر های والد بلافاصله اش می باشد. این رابطه به صورت زیر تعریف می شود:

$$P(y_1, \dots, y_n) = \prod_{i=1}^n P(y_i | Parents(Y_i)) \quad (1)$$

مساله ای که معمولاً برای ساخت شبکه های بیزی مطرح است، یادگیری هر دوی این موارد بر اساس مجموعه مثال های آموزشی است. مساله یادگیری بسته به اینکه ساختار شبکه معلوم بوده و فقط نیاز به یادگیری جداول احتمالی باشد و یا اینکه ساختار شبکه نیز نا معلوم باشد، ابعاد مختلفی پیدا می کند. همچنین داده های آموزشی ممکن است کاملاً قابل مشاهده<sup>۵</sup> و یا تا حدودی قابل مشاهده<sup>۶</sup> باشند. قابل مشاهده بودن بدان معناست که برای هر مثال آموزشی مقادیر تمام ویژگی ها معلوم باشند.

## ۱-۲ برآورد امکان بیشینه

ساده ترین حالت آموزش پارامتر های شبکه وقتی است که ساختار شبکه مشخص بوده و کلیه متغیر ها کاملاً مشاهده پذیر باشند. در این حالت فرض می شود که هدف یادگیری پیدا کردن برآورد امکان بیشینه<sup>۷</sup> (MLE) پارامتر های جداول احتمال شرطی هر یک از متغیرها می باشد. این بدان معناست که مقادیری را برای این پارامتر ها انتخاب کنیم که امکان داده های آموزشی که شامل  $M$  نمونه هستند را بیشینه کند. مقدار نرمالیزه شده امکان لگاریتمی، برای مجموعه آموزشی  $D = \{D_1, D_2, \dots, D_M\}$  یک جمع بر روی تمام نود ها به صورت زیر می باشد:

الگوریتم یادگیری می باشد که  $p(n+1) = T[\alpha(n), \beta(n), p(n)]$  این اتوماتا به صورت زیر عمل می کند. بر مبنای مجموعه احتمالات عمل ها  $p$  اتوماتا یک عمل  $\alpha_i$  را به طور اتفاقی انتخاب کرده و آن را به محیط اعمال می کند. پس از دریافت سیگنال تقویتی از محیط، اتوماتا مجموعه احتمالات عمل هایش را برای پاسخ های مطلوب با رابطه (۳) و برای پاسخ های نامطلوب با رابطه (۴) به روز رسانی می کند.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a \cdot (1 - p_i(n)) \\ p_j(n+1) &= (1-a) \cdot p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (3)$$

$$\begin{aligned} p_i(n+1) &= (1-b) \cdot p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b) \cdot p_j(n) \quad \forall j, j \neq i \end{aligned} \quad (4)$$

در این دو رابطه  $a$  و  $b$  به ترتیب پارامترهای پاداش و جریمه هستند. هرگاه  $a = b$  باشد الگوریتم یادگیری پاداش-جریمه خطی<sup>۱۰</sup> یا  $L_{R-P}$  نامیده می شود، برای  $a \ll b$  پاداش-جریمه خفیف خطی<sup>۱۱</sup> یا  $L_{R\epsilon P}$ ، و برای  $b = 0$  پاداش-هیچ کار خطی<sup>۱۲</sup> یا  $L_{R-I}$  گفته می شود.

این نوع  $LA$  با نام  $LA$  با مجموعه عمل محدود<sup>۱۳</sup> (FALA) نیز نامیده شده است. ویژگی این  $LA$  این است که بردار احتمالات بیانگر حالت داخلی  $LA$  بوده و برای محیطی که  $LA$  در آن کار می کند مجموعه حالات در نظر گرفته نمی شود.

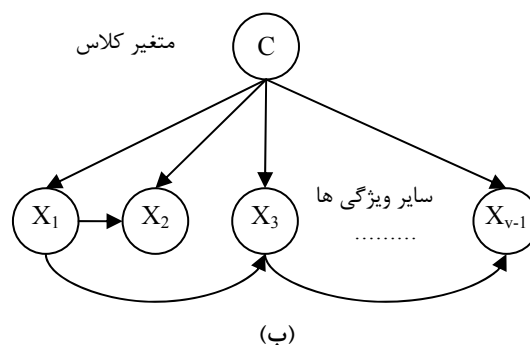
### ۳-۲ اتوماتاهای یادگیر تعمیم یافته<sup>۱۴</sup> (GLA)

برای حل مساله یادگیری تقویتی انجمنی<sup>۱۵</sup> به یک  $LA$  با تعریف متفاوت نیاز است. در مسائل یادگیری تقویتی انجمنی هدف حل یک مساله تشخیص الگو می باشد. در این مسائل در هر مرحله کاری  $LA$ ، یک بردار از مقادیر ویژگی های یک نمونه از توزیع مورد نظر نیز به عنوان ورودی  $LA$  مطرح است. این بردار از ویژگی ها بردار زمینه<sup>۱۶</sup> نیز نامیده می شود.

یک راه حل این مسائل استفاده از  $GLA$  است. در  $GLA$  ساختار  $LA$  به نحوی تغییر یافته که اجازه می دهد بردار زمینه به عنوان ورودی به آن اعمال شود. یک  $GLA$  به صورت چند تایی زیر تعریف می شود.

$$GLA \equiv \langle X, \alpha, \beta, U, g, T \rangle \quad (5)$$

که  $X$  مجموعه کلیه بردار های زمینه است که می توانند ورودی  $GLA$  باشند.  $\alpha$  مجموعه (متناهی) از خروجی ها یا اعمال  $LA$  است.  $\beta$  مجموعه مقادیری است که سیگنال تقویتی می تواند اختیار کند (که معمولاً در بازه  $[0,1]$  در نظر گرفته می شود).  $g$  تابع تولید کننده احتمالات است. و  $U$  حالت داخلی است که یک بردار از اعداد حقیقی می باشد.  $T$  الگوریتم یادگیری است که  $U$  را به روز رسانی می کند. اگر



شکل ۱ الف) دسته بندی کننده ساده بیزی یا NB، ب) TAN

پس از اینکه ساختار شبکه بیزی دسته بندی کننده مشخص شد، مهم است که پارامترها به نحوی برآورد شوند که شبکه بتواند بهترین برآورد را برای مقدار متغیر کلاس در مثال های آزمایشی ارائه دهد.

### ۳-۳ اتوماتاهای یادگیر

اتوماتای یادگیر، ماشینی است که می تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می گیرد. هدف نهایی این است که اتوماتا یاد بگیرد تا از بین اعمال خود، بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند [6].

محیط را می توان توسط سه تایی  $E \equiv \{\alpha, \beta, c\}$  نشان داد که در آن  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه ورودی ها،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$  مجموعه خروجی ها و  $c \equiv \{c_1, c_2, \dots, c_r\}$  مجموعه احتمال های جریمه می باشد. هرگاه  $\beta$  مجموعه ای دو عضوی باشد، محیط از نوع  $P$  است. در چنین محیطی  $\beta_1 = 1$  به عنوان جریمه و  $\beta_2 = 0$  به عنوان پاداش در نظر گرفته می شود. در محیط از نوع  $Q$ ،  $\beta(n)$  می تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله  $[0,1]$  را اختیار کند و در محیط از نوع  $S$ ،  $\beta(n)$  متغیر تصادفی در فاصله  $[0,1]$  است.  $c_i$  احتمال اینکه عمل  $\alpha_i$  نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا، مقادیر  $c_i$  بدون تغییر می مانند، حال آن که در محیط غیرایستا این مقادیر در طی زمان تغییر می کنند.

### ۳-۱ اتوماتاهای یادگیر با ساختار متغیر (VSLA)

اتوماتاهای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است را می توان توسط چهار تایی  $\{\alpha, \beta, p, T\}$  نشان داد که  $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  مجموعه عمل های اتوماتا،  $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_r\}$  مجموعه ورودی ها،  $p \equiv \{p_1, \dots, p_r\}$  بردار احتمال انتخاب هریک از عمل ها و  $T$

▪ **تابع تولیدکننده احتمالات (g):** همان طور که در بخش ۳ گفته شد، در GLA مقادیر احتمالی مربوط به هر عمل به صورت ایستا در بردار موجود نمی باشند، بلکه برای هر مثال آموزشی و حالت داخلی شبکه بیزی، تابع تولیدکننده احتمال مقدار احتمال هر عمل LA را محاسبه خواهد کرد. برای مثال آموزشی  $S_i$  احتمال تولید شده برای هر عمل  $c_j$  برابر با  $P(c_j | a_{1i}, a_{2i}, \dots, a_{ni})$  می باشد که از روی شبکه قابل محاسبه است.  $a_{ni}$  برابر با مقدار  $n$  امین ویژگی در مثال  $i$  ام است.

▪ **انتخاب عمل:** بر خلاف اتوماتهای یادگیر VSLA های معمول که انتخاب عمل بعدی تنها بر اساس مقادیر موجود در بردار احتمالات عمل ها انجام می گیرد، در این GLA انتخاب عمل بعدی بسته به مثال آموزشی مشاهده شده نیز می باشد. همان طور که گفته شد، این وابستگی در محاسبه مقادیر بردار احتمالات به صورت پویا نهفته است.

▪ **پاداش و جریمه:** همانطور که گفته شد، بردار حالات داخلی این GLA ( $U$ ) در واقع همان مقادیر موجود در جداول احتمال شرطی نود ها هستند. بردار زمینه  $X$  مجموعه مثال های آموزشی است که به شبکه ارائه می شود. برای اعمال پاداش و جریمه به شبکه، به یک تابع به روز رسانی  $T$  لازم است که  $U$  را به روز سانی می کند.

برای این منظور از یک روش مشابه پاداش و جریمه خطی ( $L_R-P$ ) استفاده شده است. همان طور که گفته شد، دسته انتخاب شده برای یک مثال توسط شبکه، در صورت انطباق آن با مقدار واقعی در مثال آموزشی شامل پاداش و در غیر این صورت شامل جریمه خواهد شد. از آنجا که احتمال محاسبه شده برای هر کلاس (هر عمل) حاصلضرب مقادیر مربوطه در هر جدول احتمال شرطی هر متغیر است، لذا کلیه این مقادیر در اعمال پاداش و جریمه سهیم هستند. هر سطر از جدول احتمال شرطی هر نود معادل یک بردار احتمالی است و می توان مقادیر احتمالات موجود در آن را با رابطه مشابه با اتوماتا های خطی به روز رسانی کرد. روابط پاداش و جریمه به صورت زیر است.

$$p_{ik}(n+1) = p_{ik}(n) + a \cdot (1 - p_{ik}(n))$$

$$p_{jk}(n+1) = (1-a) \cdot p_{jk}(n) \quad \forall j, j \neq i \quad (\text{پاداش})$$

$$p_{ik}(n+1) = (1-b) \cdot p_{ik}(n)$$

$$p_{jk}(n+1) = \frac{b}{r_k - 1} + (1-b) \cdot p_{jk}(n) \quad \forall j, j \neq i \quad (\text{جریمه})$$

$$(8)$$

سطر  $k$  ام سطر مربوطه در جدول احتمال شرطی هر نود است. ستون  $i$  ام ستونی است که احتمال از آن استخراج شده است. سایر ستون ها با اندیس  $r$  نشان داده شده اند. تعداد ستون های موجود در هر جدول با  $r_k$  نشان داده شده است.  $a$  و  $b$  پارامترهای یادگیری هستند

مجموعه اعمال GLA،  $\alpha = \{\alpha_1, \dots, \alpha_r\}$  باشد، احتمال اعمال GLA به صورت زیر محاسبه خواهند شد.

$$\text{Prob}[a(k) = \alpha_i | \mathbf{u}, \mathbf{x}] = g(\mathbf{x}, \alpha_i, \mathbf{u}) \quad (6)$$

مشاهده می شود که به جای وجود بردار احتمالات، از تابع  $g$  برای محاسبه احتمال اعمال استفاده می شود. تابع  $g$  دارای شرایط زیر می باشد.

$$g(\mathbf{x}, \alpha_i, \mathbf{u}) \geq 0, \quad \forall \alpha_i, \mathbf{u}, \mathbf{x}$$

$$\sum_{i=1}^r g(\mathbf{x}, \alpha_i, \mathbf{u}) = 1, \quad \forall \mathbf{u}, \mathbf{x} \quad (7)$$

در هر مرحله  $k$ ، الگوریتم یادگیری  $T$  مقدار  $\mathbf{u}(k)$  را بر مبنای مقادیر فعلی  $\mathbf{x}(k)$  و  $\mathbf{u}(k)$ ، سیگنال تقویتی  $\beta(k)$  و عمل انتخاب شده توسط GLA  $\alpha(k)$ ، به روز رسانی می کند. وابستگی این به روز رسانی به بردار زمینه  $\mathbf{x}(k)$  ویژگی اصلی GLA است.

انگیزه تعریف GLA این است که بتوان مسائل یادگیری تقویتی انجمنی را مستقیماً حل کرد. بنا براین با داشتن یک بردار حالت ثابت  $\mathbf{u}$ ، احتمالاتی که GLA با آنها اعمال متفاوتی را انتخاب می کند می توانند (و معمولاً اینگونه است) که به بردار زمینه وابسته باشند. به همین خاطر است که تابع تولید احتمال GLA هم به بردار حالت  $\mathbf{u}$  و هم به بردار زمینه  $\mathbf{x}$  وابسته است. این مهم ترین ویژگی GLA است که انتخاب عمل بعدی در آن علاوه بر حالت داخلی LA به بردار زمینه ورودی نیز بستگی دارد.

#### ۴- روش آموزش پیشنهادی

در این بخش یک روش مرحله به مرحله مبتنی بر اتوماتاهای یادگیر برای آموزش شبکه های بیزی که شامل آموزش مقادیر احتمالی موجود در جداول احتمال شرطی می باشد پیشنهاد میگردد. این مدل سعی خواهد کرد مقدار پارامترها را از یک توزیع یکنواخت شروع کرده و با دیدن هر یک از مثال های آموزشی توزیع پارامترها را به توزیع حاکم بر مجموعه مثال های پردازش شده نزدیک تر کند.

برای این منظور یک اتوماتای یادگیر تعمیم یافته (GLA) با ویژگی های زیر طراحی شده است:

- اعمال  $LA(\alpha)$ : در LA مذکور به ازای هر یک از مقادیر متغیر کلاس، یک عمل در نظر گرفته شده است.
- **محاسبه بازخورد ( $\beta$ ):** دریافت بازخورد از محیط، بر اساس بررسی هر یک از مثال های آموزشی صورت می گیرد. به این صورت که ابتدا یک عمل (یک مقدار برای متغیر کلاس) انتخاب شده و سپس مقدار واقعی متغیر کلاس از روی مثال آموزشی دریافت می گردد. در صورت یکسان بودن این دو مقدار به عمل انتخاب شده پاداش تعلق گرفته و در غیر این صورت این عمل جریمه می شود.



که مقدار آنها قابل تنظیم بین صفر و یک است. هرچه مقدار این پارامترها بیشتر باشد همگرایی سریع تر بوده اما دقت آن کمتر است. در آزمایشات این مقادیر نزدیک به ۱ در نظر گرفته شده اند. شکل ۲ الگوریتم اجرای روش پیشنهادی را نشان می دهد.

جدول ۱ مجموعه داده های استفاده شده و خصوصیات هر یک

مجموعه داده	کاربرد	تعداد ویژگی ها	تعداد مثال ها	تعداد کلاس ها
adult	پیش بینی میزان دستمزد افراد بر اساس سایر اطلاعات آنها	۱۵	۳۲۵۶۱	۲
iris	تشخیص سه تیره از یک گیاه خاص بر اساس ویژگی های آنها	۴	۱۵۰	۳
mushroom	تشخیص سمی یا غیر سمی بودن قارچ ها بر اساس ویژگی های آنها	۲۲	۸۱۲۴	۲
wine	تشخیص میزان ترکیبات مختلف در نوشیدنی های الکلی دار	۱۳	۱۷۸	۳
cancer	تشخیص ابتلا به سرطان سینه بر اساس ویژگی های نمونه ها	۳۲	۵۶۹	۲

۱- به هر مقدار احتمالی موجود در کلیه جداول مقدار  $l/r_{n,k}$  را نسبت بده که  $n$  اندیس نود و  $k$  اندیس سطر مربوطه در جدول نود  $n$  است (شروع از توزیع یکنواخت).  
۲- برای هر مثال آموزشی  $S_d$  کلیه مراحل زیر را انجام بده  
۱-۲- برای هر مقدار  $c$  از متغیر کلاس  
۱-۲- برای هر نود  $X_n$  در شبکه  
۱-۱-۲- سطر  $k$  و ستون  $i$  از جدول هر نود را بیاب که احتمال  $v(n)=x_n$  باید از آن استخراج شود.  $P(v(n)|v(Parents(n)))$  مقدار متغیر  $X_n$  در مثال آموزشی  $S_d$  است.  
۲-۱-۲- احتمالات بدست آمده از کلیه نود ها را در هم ضرب کرده و به عنوان احتمال کلاس  $P(c|x_1, \dots, x_{v-1})$  در نظر بگیر.  
۲-۲- کلاسی که بیشترین احتمال برای آن محاسبه شده را به عنوان حدس دسته بندی ارائه کن.  
۳-۲- حدس دسته بندی را با مقدار واقعی کلاس در مثال آموزشی مقایسه کن.  
۳-۲- اگر دو مقدار برابر است بر اساس رابطه پاداش، کلیه احتمالات در سطر های مربوطه را با رابطه پاداش به روز رسانی کن. در غیر این صورت احتمالات را با رابطه جریمه به روز رسانی کن.

شکل ۲ الگوریتم روش پیشنهادی

## ۵- نتایج آزمایشها

برای ارزیابی الگوریتم پیشنهادی از ۵ مجموعه داده کلاسیک از پایگاه یادگیری ماشین دانشگاه UCI استفاده شده است [12]. در جدول ۱ شرح مختصری از این مجموعه داده ها آورده شده است.

ابتدا بر روی مجموعه داده هایی که دارای متغیر های عددی پیوسته بوده اند، عمل گسسته سازی<sup>۱۷</sup> بر روی مثال ها انجام شده است. ساختار آموزش داده شده طبق الگوریتم TAN<sup>۱۸</sup> آماده شده و با ثابت در نظر گرفتن این ساختار، نتایج الگوریتم آموزش پارامتر MLE با الگوریتم پیشنهادی مقایسه شده است. برای بروز رسانی بردار احتمالات اتوماتاهای یادگیر از الگوریتمهای یادگیر LR-P و LR-I استفاده شده است. نمودار یادگیری MLE و الگوریتم پیشنهادی (LA Based) برای مجموعه داده های مختلف زمانیکه از الگوریتم یادگیر LR-P استفاده میکند در شکل ۳ و زمانیکه از الگوریتم یادگیر LR-I استفاده میکند در شکل ۴ آمده است.

همان طور که مشاهده می شود، الگوریتم پیشنهاد شده برای برخی مجموعه داده ها نتیجه چندان خوبی ارائه نداده است، اما برای برخی دیگر نتایج قابل قبول است. در مقایسه نتایج بدست آمده از دو آزمایش با به روز رسانی احتمالات با الگوریتم های LR-P و LR-I می توان گفت که به روز رسانی با روابط LR-P تنها باعث می شود همگرایی به توزیع با سرعت بیشتری صورت گیرد.

در مورد مجموعه داده هایی که دارای نویز بالایی هستند مانند مجموعه داده Adult، گرچه تعداد مثال های آموزشی زیادی در دسترس است، اما مشاهده می شود که چه الگوریتم MLE و چه الگوریتم پیشنهاد شده نتایج دسته بندی چندان خوبی ندارند. همچنین در مورد این مجموعه داده الگوریتم پیشنهاد شده با تعداد مثال های آموزشی کم نیز نتیجه بهتری ارائه نکرده است.

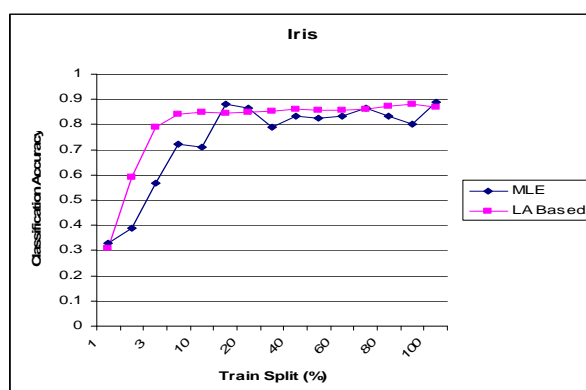
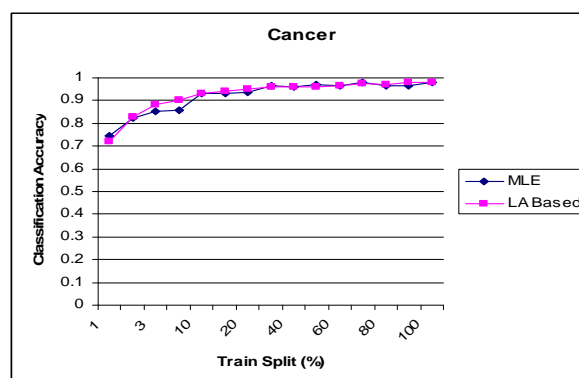
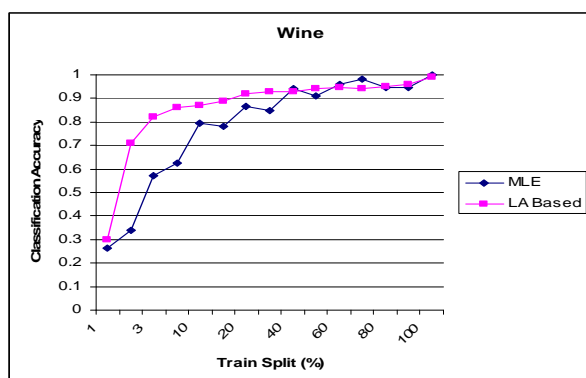
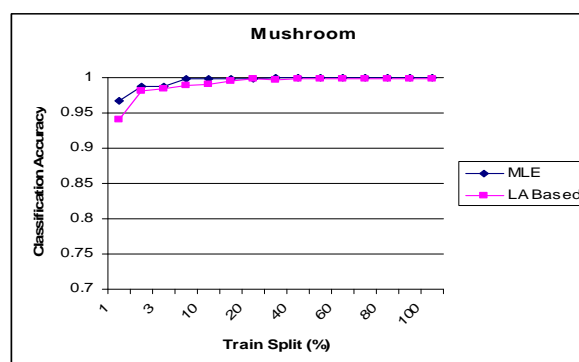
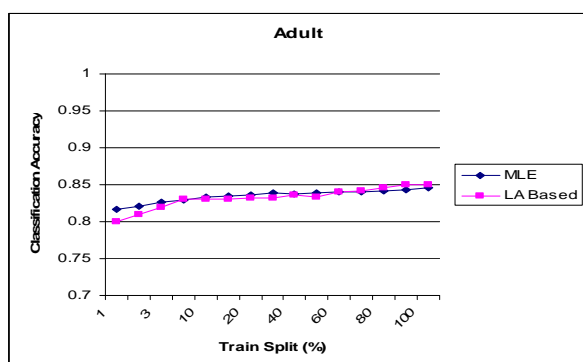
بنا به تعریف، وجود نویز در یک مثال آموزشی به این معناست که مقادیر موجود در متغیر های این مثال از تابع حاکم بر کلیه مثال های



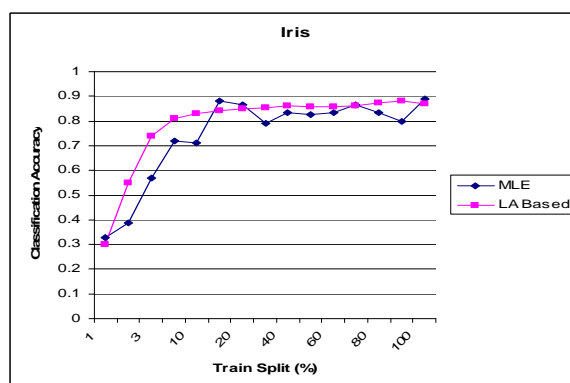
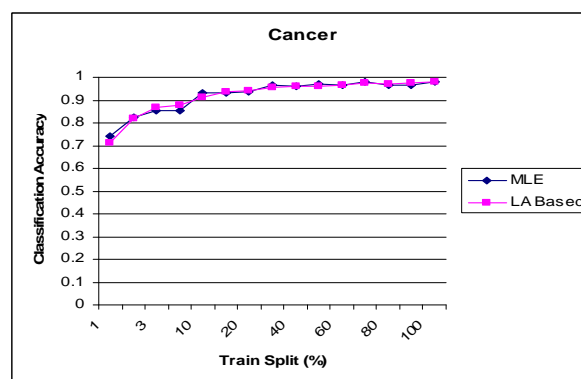
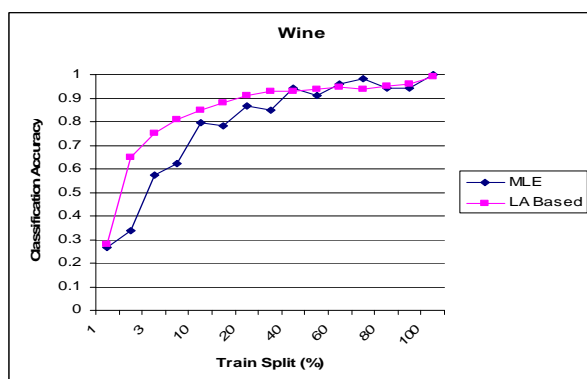
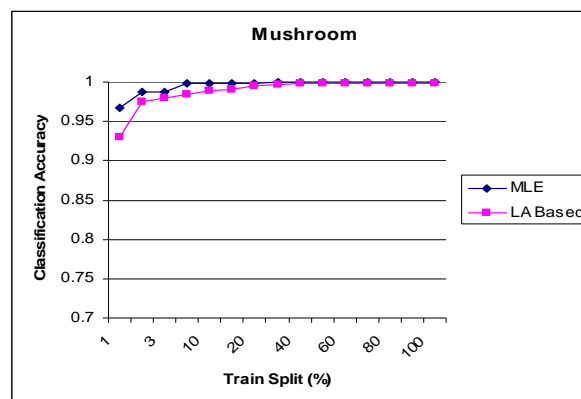
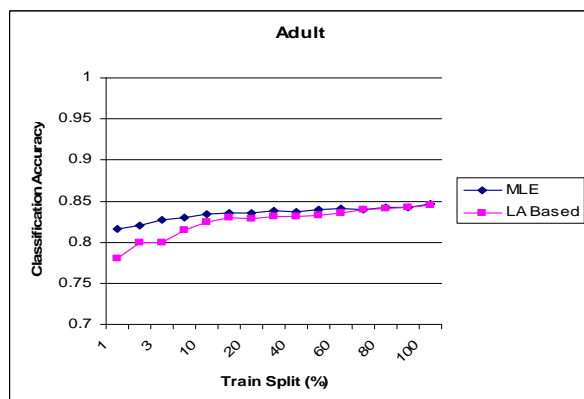
های آموزشی نمی کنند [2]. در مسائل تشخیص الگو، اصولاً مقدار متغیر کلاس تابعی از مقادیر سایر متغیرها در کلیه مثال های آموزشی و آزمایشی می باشد. یک مثال دارای نویز از این خصیصه پیروی نمی کند.

در مورد مجموعه داده Mushroom به علت وجود نویز بسیار کم، با تعداد مثال های کم نیز الگوریتم MLE دقت خوبی نشان می دهد و الگوریتم پیشنهاد شده برتری خود را در زمینه تعداد مثال های آموزشی نشان نمی دهد. اما در مورد سه مجموعه داده دیگر می توان همگرایی سریع الگوریتم پیشنهاد شده به دقت دسته بندی نهایی را با تعداد مثال

نتیجه اینکه وقتی تعداد مثال های آموزشی کم و مثال ها دارای نویز متوسط باشند، الگوریتم پیشنهاد شده می تواند در قیاس با الگوریتم کلاسیک MLE، با داشتن مثال های کمتر به دقت دسته بندی مشابه دست یابد.



شکل ۳ به روز رسانی احتمالات با استفاده از الگوریتم  $L_R-P$



شکل ۴ به روز رسانی احتمالات با استفاده از الگوریتم  $L_R-1$

داشته و در این الگوریتم مراحل آموزش و تست شبکه می توانند به صورت همزمان انجام شوند.

**سپاسگزاری:** این کار تحقیقاتی توسط مرکز تحقیقات مخابرات ایران حمایت مالی شده است که از این طریق سپاسگزاری می گردد.

## مراجع

- [1] Cheng, J., Greiner, R., "Comparing Bayesian Network Classifiers", Morgan Kaufmann, In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, Pages 101—107, 1998.
- [2] Mitchell, T. M., "Machine Learning", McGraw-Hill, 1997.
- [3] Sarkar, S., Chavali, S., "Modeling Parameter Space Behavior of Vision Systems Using Bayesian Networks", Elsevier Science Inc., Computer Vision and Image Understanding, Volume 79, Pages 183 – 223, 2000.

## ۶- نتیجه گیری

در این مقاله یک روش مبتنی بر اتوماتای یادگیر برای آموزش جداول احتمال شرطی در شبکه های بیزی پیشنهاد گردید. برای تست الگوریتم پیشنهاد شده از ۵ مجموعه داده مختلف استفاده گردید. دقت دسته بندی الگوریتم پیشنهاد شده با دقت دسته بندی الگوریتم MLE، برای تعداد متفاوت مثال های آموزشی بررسی و منحنی یادگیری مربوطه ترسیم شد. در نهایت نتایج نشان دادند که هرگاه مقدار نویز در مثال های آموزشی مجموعه داده مربوطه مقدار متوسطی باشد، الگوریتم پیشنهاد شده می تواند در قیاس با الگوریتم کلاسیک MLE، با داشتن مثال های کمتر به دقت دسته بندی مشابه دست یابد. به علاوه الگوریتم پیشنهادی پیچیدگی زمانی کمتری نسبت به الگوریتم MLE



- [10] Haug, C., Darviche A., "Inference in Belief Networks: A Procedural Guide", Elsevier Science, International Journal of Approximate Reasoning, Volume 15, Pages 225-263, 1994.
- [11] Beygi, H., Meybodi M. R., "A New Action-Set Learning Automaton for Function Optimization", Int. J. Franklin Inst. Volume 343, Pages 27-47, 2006.
- [12] Murphy, P. M., Aha D. W., UCI Repository of Machine Learning Databases, 1995, Available: <http://www.ics.uci.edu/~mllearn/MLRepository.html>
- [13] Friedman, J. H., "On Bias, Variance, 0/1 - Loss, and the Curse-of-Dimensionality". Data Mining and Knowledge Discovery, Volume 1, Pages 55-77, 1997.
- [14] Chow, C. K., Liu, C. N., "Approximating Discrete Probability Distributions with Dependence Trees". IEEE Transactions on Information Theory, Volume 14, Pages 462-467, 1968.
- [4] Murphy, K. P., "An Introduction to Graphical Models", Technical Report, Intel Research Technical Report, 2001.
- [5] Cheng, J., Greiner, R., "Learning Bayesian Belief Network Classifiers: Algorithms and System", Proceedings of 14 Biennial Conference of the Canadian Society for Computational Studies of Intelligence, Volume 2056, Pages 141-151, 1998.
- [6] Thathachar, M. A. L., Sastry, P. S., "Varieties of Learning Automata: An Overview", IEEE, Transaction on Systems, Man, and Cybernetics, Volume 32, Pages 711- 722, 2002.
- [7] Zahiri, S. H., "Learning Automata Based Classifier", Elsevier Science, Pattern Recognition Letters, Volume 29, Pages 40-48, 2008.
- [8] McCallum, A, Nigam K., "A Comparison of Event Models for Naive Bayes Text Classification", In Proceedings of AAAI/ICML-98, Workshop on Learning for Text Categorization, 1998.
- [9] Cooper, G. F., Herskovits H., "A Bayesian Method for the Induction of Probabilistic Networks from data", Machine Learning, Volume 9, Pages 309-347, 1992.

<sup>1</sup> Maximum likelihood estimation (MLE)

<sup>2</sup> Naïve bayes classifier

<sup>3</sup> Expectation maximization

<sup>4</sup> Bayesian belief network

<sup>5</sup> Fully observable

<sup>6</sup> Partially observable

<sup>7</sup> Maximum likelihood estimate

<sup>8</sup> log-likelihood

<sup>9</sup> Querying

<sup>10</sup> Linear Reward-Penalty

<sup>11</sup> Linear Reward epsilon Penalty

<sup>12</sup> Linear Reward Inaction

<sup>13</sup> Finite action-set learning automata

<sup>14</sup> Generalized learning automata

<sup>15</sup> Associative reinforcement learning

<sup>16</sup> Context vector

<sup>17</sup> discretization

<sup>18</sup> Tree Augmented Naïve Bayes