

الگوریتمهای ترکیبی برای دسته‌بندی الگوها (آناتامونهای یادگیر + روش‌های کلاسیک)

غلامرضا رضایی محمدرضا میبدی

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

تهران - ایران

چکیده: نکنکهای شناسایی الگو به دسته‌بندی یک کلاس الگو از سایر کلاس‌های الگو موجود در حوزه کاربرد مورد نظر من پردازد. استفاده از تابع تابیز، یکی از روش‌های دسته‌بندی الگوها می‌باشد. برای تعیین تابع تابیز، روش‌های مختلفی پیشنهاد شده که گروهی از آنها براساس آناتامونهای یادگیر می‌باشد.^[۲۱] [۲۲] [۱۹] [۲۱]. در [۲۱] کارایی آناتامونهای یادگیر تک‌خطی در تعیین پارامترهای تابع تابیز از طریق ارزی آناتامونهای یادگیر با پاسخ یکسان بررسی شد و آناتامونهای یادگیر با ساختار ثابت جدیدی که کارایی بالاتری در تعیین تابع تابیز (ازدیگر به) بهینه از خود شان می‌دهند، پیشنهاد گردید. در [۲۲] با توجه آناتامونهای یادگیری ارائه شده در [۲۱] آناتامونهای یادگیر سلسله‌مراتبی با ساختار ثابت جدید ارائه گردیده و کارایی آنها در تعیین پارامترهای تابع تابیز شان داده شده است. در این مقاله، از ترکیب آناتامونهای یادگیر با روش‌های کلاسیک، الگوریتمهای یادگیر با پیش‌نمایش تابع تابیز ارائه می‌گردد. الگوریتمهای پیشنهادی با استفاده از آناتامونهای یادگیر تک‌خطی با سلسله‌مراتبی و روش‌های کلاسیک، سعی در تعیین خودکار حدود هر کلاس الگو می‌نمایند. از هر یک از آناتامونهای یادگیر ارائه شده در [۲۱] و [۲۲] می‌توان به عنوان پایزگران بازی (اجزای سیستم یادگیری) استفاده نمود. همچنین، در این مقاله، کاربرد روش‌های پیشنهادی پیش‌نمایش تشخیص آواهای گفتاری زبان فارسی مورد بررسی قرار گرفته است.

کلمات کلیدی: شناسایی الگو، آناتامونهای یادگیر، تابع تابیز، الگوریتمهای ترکیبی، تشخیص آوا

۱ - مقدمه

نکنکهای شناسایی الگو، با ارائه قواعد تصمیم‌گیری، سعی در جذب‌سازی یک کلاس الگو از سایر کلاس‌های الگو می‌نماید. در صورتیکه اطلاعات متنه بطور کامل موجود باشد، قانون بیز^[۱] [۲] [۱۳] می‌تواند قاعده تصمیم‌گیری که احتمال دسته‌بندی غلط را مینیمیز کند را تولید نماید. در شرایطی که اطلاعات کافی درباره متنه در دست نیست، قواعد تصمیم‌گیری با استفاده از یک سری نمونه‌های آموزشی موجود در حوزه کاربرد موردنظر، یادگرفته می‌شود. یک روش دسته‌بندی الگوها، استفاده از تابع تابیز می‌باشد. در یک متنه در کلاس، اگر $(-)g(x) \geq 0$, قاعده تصمیم‌گیری به صورت if $g(x) > 0$, then $x \in W_1$, else $x \in W_2$ می‌باشد.

رووش‌های متعددی برای تعیین تابع تابیز ارائه شده‌اند که گروهی از آنها براساس آناتامونهای یادگیر می‌باشد. Andrew Barto و Roshni Banam A_{R-P}

که هدف آن شناسایی تابع تابیز خطی $X^T \theta = g(X)$ با استفاده از آناتامونهای یادگیر می‌باشد.^[۳] در این روش یک رابطه تجمنی (Associative Relation) بین هر عضو از نمونه روروی و هر عمل تصمیم‌گیری ایجاد می‌شود و آناتامونهای یادگیر سعی می‌کنند که با مشاهده هر نمونه و روروی، تصمیم صحیح را اتخاذ کنند. روش Sastry را جهت یادگیری تابع تابیز بهینه از فضای تابع تابیز با استفاده از بازی آناتامونهای یادگیر با پاسخ یکسان ارائه کرده‌اند.^[۴] الگوریتم Thatachar ارائه شده در این روش براساس آناتامونهای یادگیر با ساختار غیرمی‌باشد که در آن، احتمال هر عمل با استفاده از تخمین از ماتریس پاداش (Pay-off Matrix) ارزی شده در طول یادگیری ساخته می‌شود، پهنه‌گام می‌باشد. هر آناتامون شرکت کننده در بازی، یکی از پارامترهای تابع تابیز را تعیین می‌کند. مجموعه اعمال هر آناتامون، مجموعه مقادیر ممکن برای آن پارامتر می‌باشد. Phananskarj Thatachar روشی را براساس شبکه‌ای پیشخور (Feed Forward Network) از 3-Layer Linear Piecewise Classification^[۵] ارائه کرده‌اند. در این روش با استفاده از تابع تابیز خطی قطعه به قطعه به قطعه آناتامونهای یادگیر، جهت پیامهازی یک سیستم دسته‌بندی سیستمی براساس دسته‌بندی خطی قطعه به قطعه (Parameterized LA) را ارائه کرده‌اند.^[۶] در این روش با استفاده از تابع تابیز خطی قطعه به قطعه محدوده‌های هر کلاس تعیین زده می‌شود. در این روش از آناتامونهای یادگیر با ساختار متغیر بوده و قابلیت تغییر با شیوه بهنگام سازی I_{R-I} استفاده و نشان داده شده که سیستم می‌تواند به یک نقطه بهینه محلی (Local Optimum) (و نه الزاماً سراسری) اگر شرایط اولیه مناسب باشد، همگراشود. روشی را جهت شناسایی نقطه بهینه سراسری (Global Optimum) با استفاده از شبکه‌های پیشخور پیشنهاد کرده‌اند.^[۷] در این روش، بهجای آناتامونهای استاندارد از آناتامون جدیدی به نام PLA که دارای ساختار متغیر بوده و قابلیت دریافت بردار محتوی (Context Vector) از محیط را دارد، استفاده و نشان داده شده است که تحت شرایط ویژه‌ای، مجموعه آناتامونها می‌تواند به نقطه بهینه سراسری همگراشود.

رضایی و میبدی در [۲۱] و [۲۲] کارایی آناتامونهای یادگیر با ساختار ثابت در تعیین پارامترهای (بهینه) تابع تابیز با استفاده از بازی آناتامونهای یادگیر با پاسخ

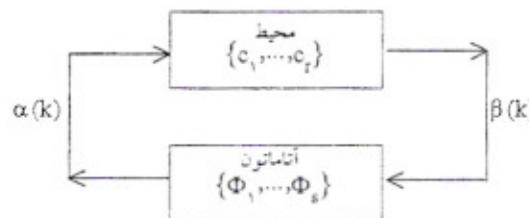
پکسان مورد بررسی قرار داده‌اند. در [۲۱] آناماتونهای یادگیر نک سطحی با ساختار ثابت جدیدی ارائه شده که قادرند پارامترهای تابع تمايز را با کاریز بالای تعیین نمایند. در [۲۲] با توسعه آناماتونهای ارائه شده در [۲۱]، آناماتونهای یادگیر سلسه‌مراتبی جدید که قادرند در محدوده وسیعتری پارامترهای تابع تمايز را تعیین کنند ارائه گردیده است.

روش‌های ارائه شده در [۲۱] و [۲۲] برای تعیین پارامترهای تابع تمايزی که فرم کلی آن از قبل معلوم باشد، قابل استفاده می‌باشد. همچنین محدوده پارامترهای تابع تمايز که باید تخمین زده شود، باید از قبیل شخص باشد. به این ترتیب، باید اطلاعات زیادی (نوع تابع تمايز و حدود پارامترهای آن) درباره مسئله موردنظر در دست باشد. در این مقاله از ترکیب روش آناماتونهای یادگیر ارائه شده در [۲۱] و [۲۲] با روشهای کلاسیک، محدوده هر کلاس را بطور خودکار تعیین می‌نماییم. روشهای کلاسیک متعددی برای تعیین توابع تمايز خطی ارائه شده‌اند که غالباً برای نوع خاصی از توزیع گوکها (مثلًا توزیع نرمال) بهینه می‌باشند [۱] و به طور عام نیز توان از این روشهایی حل مسائل استفاده نمود. با درنظر گرفتن محدوده‌ای برای هر پارامتر تابع تمايز اولیه بدست آمده از روشهای کلاسیک را اصلاح نمود. آناماتون یادگیر به عنوان یک مدل یادگیری از صفر، می‌تواند عمل اصلاح هر پارامتر را انجام دهد. همچنین با ترکیب دست‌خطهای تمايز و اصلاح موقعیت آنها با استفاده از باری آناماتونهای یادگیر من نوان بطور خودکار منحنی پوشش هر کلاس گک را تقریب زد و به این ترتیب، هر کلاس را از سایر کلاس‌های گک جدا نمود. همچنین با استفاده از ساختارهای درختی می‌توان حدود تصمیم‌گیری برای جذب‌سازی کلیه کلاس‌های گک موجود در مسئله موردنظر را تعیین نمود. در این مقاله روش آناماتونهای یادگیر با الگوریتم‌های کلاسیک شده و در قالب الگوریتم‌های نکمطحی و سلسه‌مراتبی یک کلاس گک را از سایر کلاس‌های گک جدا نماید. سایر بخشهای بعدی مقاله به صورت زیر تنظیم شده است.

در بخش ۲ به معرفی اجمالی آناماتونهای یادگیر می‌پردازم و بازی آناماتونهای یادگیر با پاسخ پکسان را معرفی می‌نمایم. در بخش ۳ الگوریتم «بهبود تابع خطی» ارائه می‌گردد. در این الگوریتم بیندا یک تابع خطی (غیر دقیق) با استفاده از روشهای کلاسیک، برای جذب‌کردن یک کلاس از سایر کلاس‌های، تخمین زده می‌شود. سپس پارامترهای تابع خطی بدست آمده بوسیله بازی آناماتونهای یادگیر با پاسخ پکسان تصحیح می‌شود. همچنین روش برای غاییش درجه تابع خطی تخمین زده شده (به درجه ۲ و ۳ و...) ارائه می‌گردد. در بخش ۴ الگوریتم «دست‌بندی قطعه به قطعه تطبیقی» بیشترین مورد توزیع هر کلاس گک بوسیله مجموعه‌ای از خطوط، تقریب زده می‌شود. این تقریب توسط روش‌های کلاسیک الجام من گردید. سپس خطوط غیرضروری حذف شده و مجموعه خطوط باقیمانده، بوسیله آناماتونهای یادگیر اصلاح می‌شوند. در بخش ۵ الگوریتم «دست‌بندی درختی» بیشترین مورد توزیع هر کلاس گک بوسیله جذب‌کردن یک ساختار درختی از کلاس‌ها ایجاد می‌گردد. به این ترتیب کلاس‌های قرار گرفته در سطوح بالاتر این ساختار، با دقت بیشتر و کلاس‌های قرار گرفته در سطوح پایین‌تر، با دقت کمتر قابل جذب‌سازی می‌باشد. سپس تابع تمايز جذب‌کننده هر گره توسط الگوریتم‌های ارائه شده در بخشهای ۳ و ۴ تخمین زده شده و تصحیح می‌گردد. در بخش ۶ کارایی الگوریتم‌های بیشترهای در مسئله جذب‌سازی آواهای زبان فارسی مورد بررسی قرار می‌گیرند. در این قسمت از آنالیز MFCC جهت استخراج بردارهای ویژگی استفاده گردیده است. بخش ۷ خلاصه مقاله می‌باشد.

۲ - آناماتون یادگیر

یک آناماتون یادگیر، یک آناماتون احتمالی (Stochastic Automaton) است که با محیط خود به صورت پسخور (Feedback) تراکنش دارد. خروجی آناماتون عملی است که به عنوان ورودی، به محیط داده می‌شود و خروجی محیط پاسخ محیط به عمل تجام شده است که به عنوان ورودی به آناماتون ارسال می‌گردد. آناماتون یادگیر براساس پاسخ محیط استراتژی خود را در تصمیم‌گیری‌های آینده به‌هنگام آینده ایجاد می‌نماید. هدف از فعالیت آناماتون، انتخاب اعمالی از مجموعه اعمال آناماتون است که بهترین پاسخ را از محیط دریافت کند. رابطه محیط و آناماتون که بصورت پسخور می‌باشد در شکل (۱) نشان داده شده است.



شکل ۱ : رفتار متقابل محیط و آناماتون

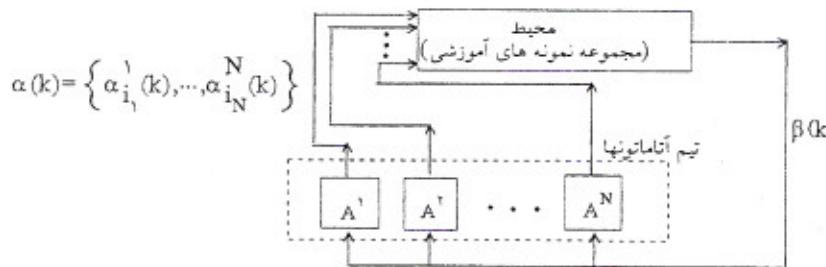
از نظر ساختاری، آناماتونهای یادگیر به دو دسته آناماتونهای یادگیر با ساختار ثابت (Fixed Structure Learning Automata) و آناماتونهای یادگیر با ساختار متغیر (Variable Structure Learning Automata) تقسیم می‌شوند. در آناماتونهای با ساختار ثابت، نحوه تغییر حالت آناماتون از قبیل شخص می‌شود. براساس حالت آناماتون در مرحله k ام، عمل $\alpha(k)$ با استفاده از تابع انتخاب عمل آناماتون، به محیط اعمال می‌شود و با دریافت پاسخ $\beta(k)$ از جانب محیط، حالت بعدی آناماتون تعیین می‌گردد. در آناماتونهای یادگیر با ساختار متغیر، نحوه تغییر حالت آناماتون از قبل مشخص نیست. در این مدل، از بردار احتمال عمل (Action Probability Vector) یا بردار احتمال حالت (State Probability Vector) آناماتون استفاده می‌گردد. بردار احتمال عمل یا حالت، احتمال انتخاب یک عمل با قرار گرفتن آناماتون در یک حالت را در مرحله k ام تعیین می‌کند. در آناماتونهای یادگیر با ساختار متغیر، یک توزیع اولیه (معمولًا بتکتواخت) به عنوان مقادیر اولیه بردار احتمال عمل در نظر گرفته می‌شود. در هر مرحله، آناماتون یک عمل را با توجه به احتمالهای عملها، انتخاب و به محیط اعلام می‌نماید و پاسخ آن را از محیط دریافت می‌کند. براساس پاسخ دریافت شده، آناماتون، این احتمالها را به هنگام من نماید و در مرحله بعدی بر لامس احتمالهای به‌هنگام شده، عمل بعدی را انتخاب می‌کند. هدف از این به‌هنگام سازی، یافتن عمل مناسب با هدف دریافت بیشترین احتمال پاسخ مطلوب از محیط است. به نحوه تغییر احتمالها، الگوریتم یادگیری (Learning Algorithm) یا روش تقویتی (Reinforcement Scheme) گفته می‌شود که من تواند تابع خطی یا غیرخطی از احتمالها پاسخ بدهد.

محیط براساس مجموعه β (مجموعه پاسخ محیط) به سه دسته تقسیم می‌شود: مدل P که در آن، مجموعه β در عضوی است که یک عضو بیانگر پاسخ مطلوب و عضو دیگر بیانگر پاسخ نامطلوب می‌باشد. مدل Q که در آن، مجموعه خروجی به صورت مجموعه محدود $\{\beta_1, \dots, \beta_m\}$ تعریف می‌شود و مدل S که در آن، مجموعه خروجی، یک مجموعه نامحدود به صورت $\{(a, b)\} = \beta$ می‌باشد. در این مدل، پاسخ محیط می‌تواند یکی از مقادیر بوسیله در ناحیه (a, b) را اختیار کند. در مدل‌های Q و S معمولاً مقدار بیشتر برای پاسخ محیط به عنوان پاسخ بهتر در نظر گرفته می‌شود.

برلسان مجموعه \mathcal{C} (مجموعه مشخصات داخلی محیط)، محیط به دو دسته است (Stationary) و پریا (Non Stationary) می‌شود. در مدل است مقادیر α_i ها (احتمال دادن باسخ نامطلوب به عمل i) در نتیجه مشخصات محیط در طول زمان ثابت است، در صورتیکه در مدل پریا، مشخصات داخلی محیط در طول پادگیری در حال تغییر می‌باشد.

از نظر تئوری، یک مسئله را می‌توان با یک آنماتون منفرد که با محیط در ارتباط است حل کرد که در این صورت، ممکن است مجبور به استفاده از آنماتونی با تعداد عملهای بسیار زیاد شویم که این، منجر به سرعت همگرایی نسیار پایین می‌گردد [۹]. برای حل مشکل همگرایی، می‌توان از مجموعه‌ای از آنماتونها که به صورت یک تیم در حل مسئله شرکت می‌کنند، استفاده کرد. مدل‌های مختلف از بازی آنماتونها را به شده است. در مدلی که در این مقاله در نظر گرفته شده است، همه بازیگران در انتهای هر بازی پاسخ یکسانی را از محیط دریافت می‌کنند [۹].

فرض کنیم تابع تغییر به فرم $g(\theta_1, \dots, \theta_N, x) = h(\theta_1, \dots, \theta_N, x)$ باشد بطوریکه θ_i پارامترهای تابع g هستند که باید پاد گرفته شوند و x بردار نمونه ریزگری‌ها است که باید دستیابی شود. با استفاده از N آنماتون پادگیر در قالب یک بازی با پاسخ یکسان مطابق شکل (۲) می‌توان پارامترهای فوق را تخمین زد. مطابق شکل (۲)، آنماتونهای A^1, A^2, \dots, A^N در بازی شرکت کرده‌اند. هر آنماتون A^i عمل $\alpha_{i,1}, \dots, \alpha_{i,N}$ می‌باشد. هر عمل آنماتون A^i یک مقدار معکن برای پارامتر θ_i از تابع تابع نسبت در نظر گرفته می‌شود.



شکل ۲: بازی آنماتونهای پادگیر با پاسخ یکسان

مقادیر معکن برای پارامتر θ_i (و در نتیجه اعمال آنماتون A^i) از قبل تعیین شده‌اند. هدف از این بازی، پیدا کردن مقادیری برای پارامترهای θ_i می‌باشد بطوریکه تعداد دستیابی‌های خلط مینیمم گردد. در هر مرحله، آنماتون A^i با توجه به شیوه تصمیم‌گیری خود، یک عمل (مقدار) از مجموعه اعمال خود (مجموعه مقادیر معکن برای θ_i) را انتخاب و به محیط اعلام می‌نماید. یک (یا چند) نمونه از بردارهای ریزگری موجود در محیط توسط تابع تغییر بدست آمده (با استفاده از پارامترهای انتخاب شده) دستیابی می‌شود و نتیجه بصورت نسبت تعداد دستیابی‌های صحیح به کل نمونه‌های آزمایش شده، به تیم ارسال می‌شود. برلسان پاسخ پدست آمده، آنماتونهای شرکت کننده در بازی، شیوه تصمیم‌گیری خود را به نگام می‌نمایند. هدف از بازی، بدست اوردن تابع تابعی است که بر اساس آن احتمال دریافت پاسخ نامطلوب و با عبارت دیگر تعداد دستیابی‌های خلط نمونه‌ها مینیمم گردد. برای اطلاعات بیشتر درباره آنماتونهای پادگیر، می‌توانید به [۹] [۱۰] [۱۱] [۱۲] [۱۳] [۱۴] [۱۵] مراجعه نمایید. نتیجه در ادامه این بخش می‌آید، شرح مختصری است از یکی از آنماتونهای پادگیر که در [۱۶] پیشنهاد شده است. این آنماتون که SL_{R-I} نامیده می‌شود، در این مقاله مورد استفاده غرر گرفته است.

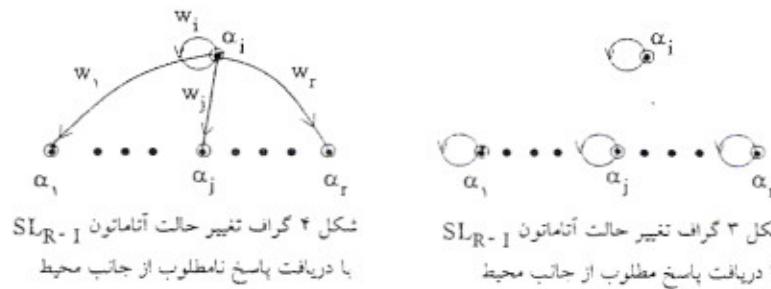
۱-۲ - آنماتون SL_{R-I}

آنماتون با حالت $\Phi_i, \Phi_j, \dots, \Phi_N$ و عمل $\alpha_1, \dots, \alpha_N$ بطوریکه در هر حالت Φ_i عمل α_i توسط آنماتون انتخاب شود را در نظر گیرید. برای هر عمل α_i وزن w_i با توزیع اولیه پکتو راحت (مقدار $\frac{1}{N}$ برای هر Φ_i) در نظر گرفته می‌شود. در صورتیکه در مرحله k ام، عمل $\alpha_i(k)$ توسط آنماتون انتخاب شود (آنماتون در حالت $\Phi_i(k)$ باز) با دریافت پاسخ محیط، آنماتون تشویق یا تنبیه می‌شود. در صورت تشویق شدن، وزن عملهای آنماتون مطابق رابطه (۱) به نگام می‌شود بدین معنی که با وزن عمل انتخاب شده مقداری اضافه شده و از وزن سایر عملهای کم می‌شود. در صورت تنبیه شدن، بدرو اینکه وزن عملهای آنماتون تغییر کند، با احتمال $w_i(k)$ در حالت فعلی باقی مانده و با احتمال (k) به حالت Φ_i تغییر حالت می‌دهد. عملی که وزن آن از حد مشخصی (مثلاً $0.99/0.01$) بیشتر شود، به عنوان عمل بهینه انتخاب می‌شود. W عددی است صحیح که پارامتر پادگیری نامیده می‌شود.

$$\begin{aligned} w_i(k+1) &= w_i(k) + \frac{1}{W}(1-w_i(k)) \\ w_j(k+1) &= \left(1 - \frac{1}{W}\right)w_j(k) \quad j \neq i \end{aligned} \quad (1)$$

شکل‌های (۳) و (۴) نحوه تغییر حالت آنماتون با دریافت پاسخ مطلوب و نامطلوب از جانب محیط را نشان می‌دهد. فرض شده است که آنماتون در حالت Φ قرار دارد، به این ترتیب عمل α_i را انتخاب می‌نماید. با دریافت پاسخ مطلوب از جانب محیط، آنماتون تشویق می‌گردد و وزن عملهای آن مطابق رابطه (۱) به نگام می‌گردد. با دریافت پاسخ نامطلوب، آنماتون با احتمال w_i در حالت Φ باقی می‌ماند و با احتمال w_j ($j \neq i$) به حالت Φ تغییر حالت می‌دهد. باید

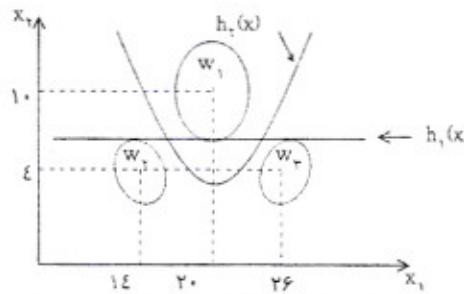
توجه داشت که همواره $1 = \sum_{j=1}^N w_j$ می‌باشد. برای کسب اطلاعات بیشتر درباره این آنماتون می‌توانید به [۱۶] مراجعه نمایید.



۳- بهبود توابع خطی

روش‌های کلاسیک متعددی برای تعیین توابع تمايز خطی برای جداسازی یک کلاس از کلاس‌های دیگر موجود در حوزه مسئله مورد حل، ارائه شده است. گروهی از این روشها بصورت تکراری (Iterative) عمل می‌کنند و گروهی دیگر براساس مطالعات آماری بین کلاسها می‌پاشند [۱] [۲]. غالباً الگوریتم‌های ارائه شده برای جداسازی در کلاس با توابع توزیع نرمال و اغلب با شرط یکسان بودن ماتریس‌های کوواریانس آنها، بهینه می‌پاشند.

شکل (۵) شکل توزیع کلاس‌های یک مسئله سه کلاسی فرضی را نشان می‌دهد. مانظور که در شکل (۵) دیده می‌شود، تابع تمايز (x) $h_i(x)$ تابع تمايز خطی است که می‌تواند کلاس W_i را از دو کلاس دیگر جداسازد، اما تابع $h_i(x)$ $- h_j(x)$ که پک تابع درجه ۲ می‌باشد $-$ با دقت بسیار بیشتری کلاس W_i را دسته‌بندی می‌نماید. با درنظر گرفتن تابع (x) $h_i(x)$ به عنوان یک حدس اولیه برای جداسازی کلاس W_i و تصحیح آن، می‌توان تابع تمايز (x) $h_i(x)$ را بدست آورد.



شکل ۵: توزیع کلاس‌های مسئله نمونه شماره ۱

اگر (x) $h_i(x)$ تابع تمايز اولیه خطی برای جداسازی کلاس W_i از سایر کلاسها باشد، فرم عمومی (x) بصورت (۲) می‌باشد:

$$h_i(x) = \begin{cases} v^T x + v_0 < 0 & \Rightarrow x \in W_i, \quad v = (v_1, \dots, v_n)^T \in \mathbb{R}^n \\ v^T x + v_0 > 0 & \Rightarrow x \notin W_i, \quad x = (x_1, \dots, x_n)^T \in \mathbb{R}^n \end{cases} \quad (2)$$

و یا :

$$h_i(x) = \sum_{i=1}^n v_i x_i + v_0 \quad (3)$$

فضای مسئله \mathbb{R}^n فرض می‌شود و با درنظر گرفتن v_0 به عنوان حد آستانه، تابع (x) $h_i(x)$ دارای $n+1$ پارامتر می‌باشد. در صورتیکه هدف تعیین تابع درجه ۲ باشد، فرم عمومی تابع بصورت (۴) خواهد بود.

$$h_i(x) = \sum_{i=1}^n \sum_{j=1}^n \theta_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i + \theta_0 \quad (4)$$

بنابراین تابع (x) $h_i(x)$ دارای $\frac{n(n+1)}{2} + n+1$ پارامتر می‌باشد. برای تعیین (x) $h_i(x)$ با استفاده از حدس اولیه (x) $h_i(x)$ ، می‌توان مقدار پارامترهای $\theta_0, \theta_1, \dots, \theta_n$ در v_0, v_1, \dots, v_n و مقدارهای θ_{ij} را حول مقدار در نظر گرفت. الگوریتم (۱)، نحوه تبدیل تابع (x) $h_i(x)$ را با استفاده از بازی آتاماتونهای یادگیری با پاسخ یکسان، بیان می‌کند.

در الگوریتم (۱) رویه Learn-h2 پارامترهای تابع تمايز (x) $h_i(x)$ را تعیین می‌نماید. تابع Discriminate در هر مرحله از یادگیری، بردار نمونه ویژگی و روش را دسته‌بندی نموده و پاسخ صحیح یا غلط را باز می‌گرداند. رویه UpdateGame با نتیجه به نتیجه دسته‌بندی (DiscRes) وضعیت آتاماتونها را به نکام می‌نماید. در رویه Learn-h2، بینداز تابع (x) $h_i(x)$ با یک روش آماری (غیر دقیق) تعیین می‌گردد. آتاماتونهای i, j $< i, j < n$ که در الگوریتم ایجاد می‌شوند، وظیفه یادگیری پارامترهای θ_{ij} (صرایب $x_i x_j$) از تابع تمايز (x) $h_i(x)$ را بر عهده دارند. از آنجا که (x) $h_i(x)$ با استفاده از (x) $h_i(x)$ ساخته می‌شود و این ضرایب در وجود ندارند (صفر می‌باشند)، مقدارهای ممکن برای این پارامترها حول مقدار صفر در نظر گرفته می‌شود. تعداد مقدارها (تعداد عملهای آتاماتون j) برابر $1 + 2r + r^2 = K$ با قدمهای $\frac{1}{K}$ در نظر گرفته می‌شود و مقدار صفر به عنوان یکی از عملهای آتاماتون در نظر گرفته می‌شود. r و K پارامترهای الگوریتم می‌باشند. آتاماتونهای

نیز مأمور پادگیری فراهم θ_1 در تابع $(x_i h_i)$ می‌باشد. مجموعاً $\sum_{i=1}^n \theta_i h_i$ حول مقدار پارامتر $\sum_{i=1}^n \theta_i$ از تابع $(x_i h_i)$ برای پارامتر θ_1 درنظر گرفته می‌شود. به این ترتیب یک فضای برای پارامترهای تابع تمايز ایجاد می‌شود و مجموعه آناماتونهای پادگیر سعی در پیدا کردن پارامترهای بهینه برای تابع تمايز مورد نظر در فضای تعریف شده می‌نمایند.

Procedure Learn-h2

```

    Guess  $h_i(x)$  using training samples;
    for i := 1 to n do
        for j := i to n do
             $LA_{ij}$  := create a learning automata;
            for k := -r' to r' do  $LA_{ij}.Actions.Add(k / K)$ ;
            end for {j}
        end for {i}
        for i := 0 to n do
             $LA_i$  := create a learning automata;
            for j := -r' to r' do  $LA_i.Actions.Add(v_i^{*(1+j/K)})$ ;
        end for {i}
    Repeat
         $x$  := Next training sample;
        DiscResult := Discriminate ( $x$ );
        UpdateGame (DiscResult);
    Until EndOfGame;
    Optimal Actions are parameters of function  $h_i(x)$ ;
End Learn-h2;
```

Function Discriminate(x) : Boolean;
$$hx := \sum_{i=1}^n \sum_{j=1}^n (LA_{ij}.Action) \cdot x_i x_j + \sum_{i=1}^n (LA_i.Action) \cdot x_i + LA_0.Action$$

Result := (($hx \leq 0$) and ($x.ClassID = 1$) or ($hx > 0$) and ($x.ClassID \neq 1$));

End Discriminate;

الگوریتم (۱) تابع تمايز خطی ($x_i h_i$) را به تابع درجه ۲ ($x_i h_i$) تبدیل می‌نماید. بطور کلی برای تبدیل تابع خطی ($x_i h_i$) به یک تابع چند جمله‌ای با درجه $n > 1$ می‌توان از روش مشابه الگوریتم (۱) استفاده نمود. به عنوان مثال برای تابع تمايز درجه ۲، فرم کلی بصورت (۵) می‌باشد:

$$(5) \quad h_2(x) = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \theta_{ijk} x_i x_j x_k + \sum_{i=1}^n \sum_{j=1}^n \theta_{ij} x_i x_j + \sum_{i=1}^n \theta_i x_i + \theta_0$$

برای تعیین پارامترهای ممکن برای θ_{ijk} ، مقدارهای صفر و مقدارهای ممکن برای θ_{ij} حول $\sum_{i=1}^n \theta_i$ درنظر گرفته می‌شود. در قسمت پادگیری تابع تمايز در الگوریتم (۱)، روش UpdateGame روشی برای آناماتونها را به عنوان آناماتونها می‌نماید. آناماتونها می‌توانند هر یک از اثواب آناماتونهای ساختار ثابت و پا منغیر که قادر به همگرا شدن در بازی آناماتونها با پاسخ یکسان باشند، انتخاب شوند. در صورتیکه تعداد عملها زیاد باشد، می‌توان از ساختارهای سلسه‌مراتبی نیز استفاده نمود.

مسئله نمونه ۱: محاسبه تابع تمايز برای جداولی کلاس W از دو کلاس دیگر با توزیعهای نرمال با مشخصات زیر:

$$(6) \quad W_1 : \Sigma = \begin{pmatrix} 4 & 6 \\ 6 & 6 \end{pmatrix}, \quad m = [14, 10]^T \quad W_2 : \Sigma = \begin{pmatrix} 2 & 10/5 \\ 10/5 & 2 \end{pmatrix}, \quad m = [12, 1]^T$$

$$W_3 : \Sigma = \begin{pmatrix} 2 & 10/5 \\ 10/5 & 2 \end{pmatrix}, \quad m = [13, 10]^T$$

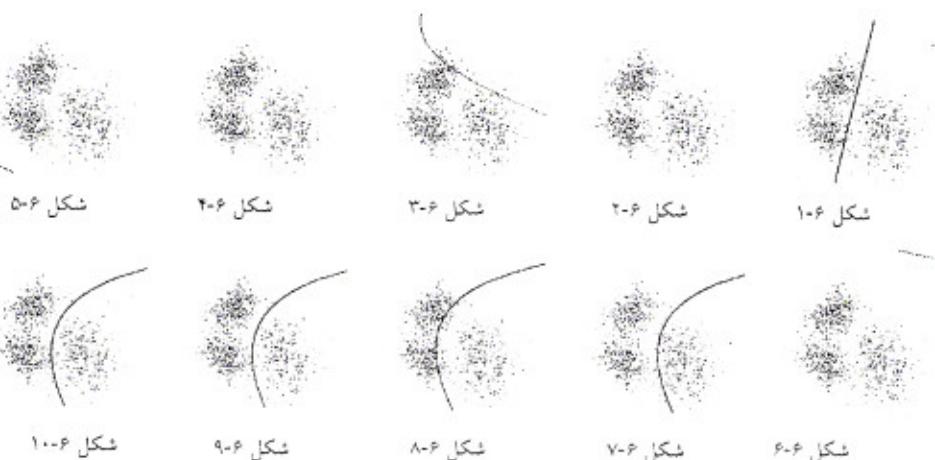
ج ماتریس کوواریانس توزیعها و m بردار میانگین توزیعها می‌باشد. جدول (۱) نتیجه ۱۰ بار اجرای الگوریتم (۱) را برای حل مسئله نمونه ۱ نشان می‌دهد.

جدول ۱

	۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	شماره آزمایش
دققت اولیه	۹۶/۵	۹۷/۱	۹۶/۵	۹۶/۵	۹۵/۷	۹۷/۱	۹۵/۱	۹۶/۱	۹۵/۸	۹۶/۳	
دققت نهایی	۹۸/۵	۹۸/۱	۹۸/۷	۹۸/۱	۹۷/۲	۹۸/۱	۹۸/۱	۹۷/۸	۹۸/۳	۹۸/۵	
تعداد مراحل	۲۴۶۴۸	۱۲۰۹۹	۱۲۲۱۷	۲۰۲۰۷	۱۷۴۷۲	۲۲۲۲۸	۱۵۳۰۲	۱۲۱۸۴	۲۲۲۲۶	۱۵۹۱۰	

در پیاده‌سازی نجام شده، مقدار $K = 5$ و $r = 3$ درنظر گرفته شده و برای پادگیری هر آناماتون از الگوریتم I $SL_R = 1000$ با پارامتر SL_R است. در قسمت پادگیری تابع تمايز درجه ۲ در میانگین توزیعهای اولیه ساخته می‌گردد. در جدول (۱) دققت اولیه، دققت تابع تمايز خط اولیه تخمین زده شده بوسیله تابع فیشر [۱] برای یک سری نمونه آموزشی با توزیع

مسئله نموده ۱ را نشان می‌دهد. شکل (۲-۶) وضیعت اولیه منحنی درجه ۲ در لبتدای پادگیری ر شکلهای دیگر، وضعیت خط پادگرفته شده، را بعد از هر ۲۰۰۰ مرحله نشان می‌دهد.



۴ - الگوریتم دسته‌بندی قطعه به قطعه تطبیقی (Adaptive Piecewise Classification)

الگوریتم ارائه شده در بخش ۳ قادر به بهبود تابع تمایز موردنظر می‌باشد. در این الگوریتم فرم عمومی تابع تمایز نهایی باید معلوم باشد. با استفاده از این الگوریتم دسته‌بندی قطعه به قطعه و تغییر زدن منحنی بوش هر دسته الگو، می‌توان این عمل را نیز بطور خودکار انجام داد. الگوریتم (۲) اسکلت اصلی این روش را نشان می‌دهد.

```

Procedure Adaptive-Piecewise
    Estimate-Basic-Lines;
    Discard-Extra-Lines;
    Learn-Modified-Lines;
End Learn-Piecewise;

```

الگوریتم ۲

در الگوریتم (۲)، رویه Estimate-Basic-Lines، مجموعه ثوابت خطوط اولیه برای جداسازی تغییری کلاس W_i از سایر کلاسهای موجود در حوزه مسئله را تعیین می‌کند. این نوع مسئله تعداد زیاد و با دقت پایین باشند. رویه Discard-Extra-Lines خطوط غیرضروری را حذف می‌کند به این ترتیب که مجموعه خطوط که بیشترین دقت جداسازی را ازخود نشان می‌دهند، حفظ شده و سایر خطوط کنار گذاشته می‌شوند. در قدم سوم، رویه Learn-Modified-Lines پارامترهای خطوط تمایز بدست آمده را اصلاح می‌کند. در این قسمت حتی می‌توان درجه خطوط تمایز قطعه به قطعه بدست آمده از قدم قبلی را نیز با استفاده از الگوریتم (۱) افزایش داد. در الگوریتم‌هایی که در ادامه بخش خواهد آمد، فرض شده است که کلاس الگو W_i توسط یک منحنی محدب از سایر کلاسهای قابل جداسازی باشد. در ادامه بحث، مراحل مختلف الگوریتم فوق به تفصیل آمده است.

۴-۱- تعیین خطوط اولیه

در این قسمت الگوریتمی برای تغییرپذیردن خطوط اولیه تشکیل دهنده بوش کلاس الگو W_i از سایر کلاسهای ارائه می‌گردد. برای انجام این عمل، از نقاط کناره‌ای توزیع کلاس الگو W_i استفاده می‌گردد. نقاط کناره‌ای، نقاطی درنظر گرفته می‌شوند که فاصله آنها از نقاط متعلق به سایر کلاسهای الگو از حد مشخصی کمتر باشد. مفهوم از نقاط همان بردار و بیزگی متعلق به مجموعه آموزشی حوزه مسئله می‌باشد. الگوریتم (۳) (مجموعه خطوط اولیه برای جداسازی کلاس W_i از سایر کلاسهای W_j) MWD (Max Within Distance) و MBD (Max Between Distance) در پارامتر الگوریتم می‌باشد. تعیین مسئله در الگوریتم (۳) (مقدارهای MBD و MWD) در مجموعه $Samples$ نمونه‌های آموزشی موجود در مجموعه $Samples$ کلاس W_i را نشان می‌دهد. آرایه G مجموعه قطعات نقاط تغییری برای شناسایی لبه کلاس W_i را ذبحه می‌کند و هر عضو آرایه V قطعه خط تغییر زده شده برای جداسازی کلاس W_i با استفاده از اطلاعات موجود در عضو مشاهده در آرایه G را ذبحه می‌کند. بطور خلاصه الگوریتم فوق بصورت زیر عمل می‌کند: ابتدا به ازای هر نمونه آموزشی موجود در کلاس W_i ، کلیه نمونه‌های با فاصله کمتر از مقدار MBD که متعلق به سایر کلاسهای باشند را جدا می‌کند و درصورتیکه چنین نمونه‌هایی وجود داشته باشد، گروهی در آرایه G ساخته می‌شود که اعضای آن نمونه اولیه کلاس W_i و سایر نمونه‌های بدست آمده می‌باشد. در قدم بعد گروههای تزدیک بهم با یکدیگر ترکیب می‌شوند، به این ترتیب که اگر فاصله نمونه‌های متعلق به کلاس W_i از هر گروه از مقدار MWD کمتر باشد، اعضای دو گروه در قالب یک گروه ترکیب می‌گردند. به این ترتیب مجموعه نمونه‌های آموزشی موجود در حوزه مسئله به گروههای کوچکتر تقسیم می‌گردد. در قدم سوم خط جدا کننده کلاس W_i از سایر کلاسهای با استفاده از اطلاعات محلی موجود در گروهها تعیین و در آرایه V ذبحه می‌گردد. خروجی الگوریتم، آرایه V می‌باشد که اعضای آن ضرایب خطوط تمایز اولیه می‌باشد.

Procedure Learn-Basic-Lines

```

for i := 1 to Wi.Count do
    SampleAdded := False;
    for j := 1 to Samples.Count do
        if Samples[j].ClassID < Samples[i].ClassID and
           Distance(Samples[j],Samples[i]) < MBD then
            if not SampleAdded then
                SampleAdded := True;
                Gi := Make-A-Group-For-Samples[i];
            end if;
            Gi.AddSample (Samples[j]);
        end if;
    end for {j};
end for {i};

for i := 1 to G.Count do
    for j := i+1 to G.Count do
        if Distance (Gj.Sample-Wi, Gj.Sample-Wi) < MWD then
            Gi := CombineGroups (Gi, Gj)
        end if;
    end for {j}
    Vi := Estimate-Line (Gi)
end for {i}
end Learn-Basic-Lines;

```

الگوریتم ۲

در الگوریتم فوق با درنظر گرفتن پارامتری برای تعیین لبه، می‌توان تعداد خطوط اضافی را کم کرد به این ترتیب که تنها خطوطی در مجموعه V ذخیره گردند که دقت جداسازی آنها برای جداسازی نمونه‌های کلاس W از حد مشخص BP (Border Parameter) پیشتر باشد.

۲-۴ - کاهش تعداد خطوط

اگرتابع تمايز خطوط با بردار ضرايب γ به صورت

$$\text{if } v^T \underline{x} < \cdot \text{ then } \underline{x} \in W_i \text{ else } \underline{x} \notin W_i \quad (7)$$

درنظر گرفته شود، در جداسازی قطعه به قطعه توسط آرایه خطوط V با m عضو (V_i)، خواهیم داشت:

(8)

تعدادی از خطوط در مجموعه خطوط بدست آمده توسط الگوریتم (۳) غیرضروری می‌باشد و همچنین بدليل دقیق بودن الگوریتم تقریب لبه، این مجموعه حاوی خطوط با دقت پایین نیز می‌باشد. در قدم بعدی، خطوط غیرضروری و خطوط که دارای دقت پایین می‌باشند، حذف می‌شوند و با به عبارت دیگر بهترین زیر مجموعه n عضوی از مجموعه m عضوی که بهترین دقت جداسازی را بدست می‌دهد باید تعیین گردد. بهترین جواب با آزمایش کلیه زیرمجموعه‌های مجموعه V روی مجموعه نمونه‌های آموزشی حاصل می‌گردد که دارای پیجیدگی $O(2^n)$ می‌باشد. الگوریتم (۴) زیرمجموعه‌ای خوب (نه الزاماً بهینه) را اینین کلیه زیرمجموعه‌ها، با پیجیدگی $O(m^n)$ جدا می‌کند.

در الگوریتم (۴)، آرایه‌های Max-Masks و Masks در دسته‌بندی الگوها شرکت می‌نمایند که عضو از این در آرایه، متاخر با عضوی از آرایه V می‌باشد. تنها اعضایی از آرایه خطوط V در دسته‌بندی الگوها شرکت می‌نمایند که عضو متاخر آن در آرایه Masks برابر با «یک» باشد. مقادیر اولیه این در آرایه صفر فرض می‌شود. متغیر Precision دقت جداسازی در هر مرحله و متغیر Max-Precision حداکثر دقت جداسازی بدست آمده تا آن مرحله را در خود ذخیره می‌کنند. اعضای آرایه Max-Masks مجموعه خطوطی که بیشترین دقت جداسازی را نولید کرده‌اند را نشان می‌دهند. همانطور که در الگوریتم مشاهده می‌شود، الگوریتم دارای m^n تعداد خطوط آرایه V قدم (حلقه‌ای که با متغیر α مشخص شده است) می‌باشد. در هر قدم فرض می‌شود که عضو متاخر آن قدم (عضو α)، در جداسازی شرکت می‌نماید و درنهایت با توجه به الگوریتم اوله شده، زیرمجموعه‌ای از سایر خطوط که بیشترین دقت جداسازی را از خود نشان می‌دهند، تعیین می‌گردد. دقت جداسازی با توجه به آرایه Masks در هر مرحله توسط تابع DiscriminateByMasks تعیین می‌گردد. بهترین دقت بدست آمده در مراحل مختلف به عنوان نتیجه الگوریتم درنظر گرفته می‌شود. الگوریتم (۵) تابع DiscriminateByMasks می‌باشد.

پیجیدگی الگوریتم (۴) از مرتبه $O(m^n)$ می‌باشد و در صورتیکه مجموعه M بردار نمونه در مجموعه نمونه‌ای آموزشی وجود داشته باشد، تعداد مقایسه‌ها در مرتبه $O(m^n)$ خواهد بود. خروجی الگوریتم (۴) بردار Max-Masks می‌باشد که نشان می‌دهد چه خطوطی در تعیین لبهای کلاس الگوی W شرکت خواهند داشت.

```

Procedure Discard-Extra-Lines (v : List of Lines)
  Max-Precision := 0;
  for i := 1 to v.Count do
    for j := 1 to Masks.Count do
      Masks[j] := 0;
    end for {j};
    Masks[i] := 1;
    Precision := DiscriminateByMasks;
    for (j := 1 to Masks.Count) and (j <> i) do
      Masks[j] := 1;
    if DiscriminateByMasks > Precision then
      Precision := DiscriminateByMasks
    else
      Masks[j] := 0;
    end if;
  end for {j}
  if Precision > Max-Precision then
    Max-Precision := Precision;
    Max-Masks := Masks;
  end if;
end for {i}
End Discard-Extra-Lines

```

الگوریتم ۴

```

Function DiscriminateByMasks : Real;
  N-Correct := 0;
  for i := 1 to Samples.Count do
    V-Side := True;
    for (j := 1 to v.Count) and (Masks[j] = 1) do
      v-Side := v-Side and
        (DiscriminateByV(v[i], Sample) <= 0)
    end for {j}
    if (v-Side and (Samples[i] in Wv)) or
      (not v-Side and not (Samples[i] in Wv)) then
        inc (N-Correct);
    end for {i}
  Result :=  $\frac{N-Correct}{Samples.Count}$ 
End DiscriminateByMasks;

```

الگوریتم ۵

۲-۴ - تصحیح خطوط

مجموعه خطوط بدست آمده از الگوریتم‌های (۳) و (۴)، بدلیل دقیق نبودن الگوریتم‌های ایجاد کننده آنها، خطوط غیر دقیق هستند. در این بخش، الگوریتمی براساس آناتامونهای پادگیر برای تصحیح خطوط بدست آمده از آنها معرفی می‌گردد. روش کلی به اینصورت می‌باشد که برای هر خط موجود در آرایه V آرایه Max-Masks برابر یک باشد. یک تیم از آناتامونها مطابق الگوریتم (۱) تشکیل می‌شود. در هر مرحله، شیوه‌ای بطور تصادفی انتخاب و توسط تمثیل آناتامونهای شرکت کننده در بازی دسته‌بندی می‌گردد. سپس آناتامونهای شرکت کننده، براساس نتیجه دسته‌بندی، استراتژی تقصیم‌گیری خود را برای مرحله بعدی بهنگام می‌نمایند. الگوریتم‌های (۶) تا (۸) نحوه تصحیح خطوط توابع ثابت بدست آمده از الگوریتم‌های (۳) تا (۵) را اشان می‌دهند.

```

Procedure Learn-Modified-Lines
  for (i := 1 to v.Count) and if (Max-Masks[i] = 1) then
    LA-List := Create LAis for modifying
    line v[i] like Algorith (1);
    GCi.LA-List := LA-List;
  end for {i}
  Repeat
    Sample := Select-A-Random-Sample;
    DiscRes := DiscriminateByGC (Sample);
    for i := 1 to GC.Count do GCi.UpdateGame (DiscRes);
  until EndofLearn;
End Learn-Modified-Lines;

```

الگوریتم ۶

```

Function DiscriminateByGC (Sample) : Boolean
  LA-Side := True;
  for i := 1 to GC.Count-1 do
    LA-Side := LA-Side and
      (Discriminate(GC.LA-List, Sample) <= 0);
  end for {i}
  Result := (LA-Side and (Sample in Wi)) or
    (not LA-Side and not (Sample in Wi));
End DiscriminateByGC;

```

الگوریتم ۷

```

Function EndOfLearn : Boolean;
  Result := True if for All GC : GCi.EndOfGame = True;
End EndOfLearn;

```

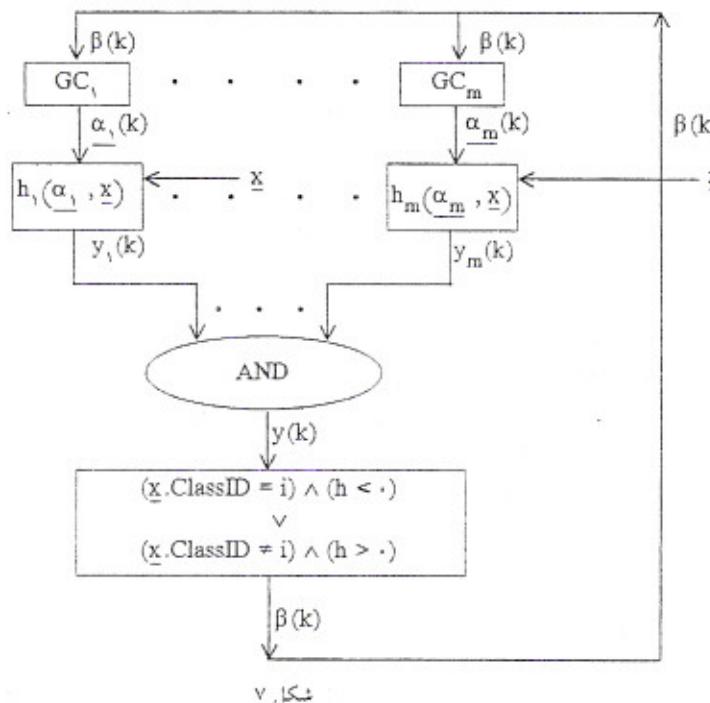
الگوریتم ۸

هر عضو GC_i از آرایه GC (Game Controller) تیم آناتلونهای اصلاح کننده خط V_i را نگهداری می‌نماید. به ازای هر خط V_i تولید شده، توسط الگوریتمها (۳) تا (۵) و با توجه به روش تصویح خط تمایز (اصلاح ضرایب تابع خطي، غواصی درجه ثالث و ...)، یک تیم تصویح کننده مشابه الگوریتم (۴) ایجاد و در GC_i ذخیره می‌گردد. ممکن در قسمت پادگیری پارامترهای تابع تمایز، در هر مرحله، یک تیم نمونه تصادفی از مجموعه نمونه‌های آموزشی انتخاب و بوسیله کلیه تیمهای شرکت کننده در تعیین پوش کلاس W_i دسته‌بندی می‌گردد و با توجه به نتیجه دسته‌بندی، آناتلونهای شرکت کننده در تیمهای مختلف استراتژی تصویح گیری خود را به عنوان نسبت پوش کلاس W_i تعیین می‌نمایند. هر مرحله پادگیری پارامترهای تابع تمایز در الگوریتم (۶) توسط خلقه Repeat...Until شان داده شده است.

الگوریتم (۷) نحوه دسته‌بندی یک بردار نمونه را نشان می‌دهد. در این الگوریتم تابع Discriminate در هر مرحله، با استفاده از پارامترهای پادگیری شده برای تابع تمایز تا آن مرحله، مقدار تابع را برای بردار ویژگی نمونه ورودی تعیین می‌کند. خروجی الگوریتم (۷) به صورت دسته‌بندی صحیح با خطوط می‌باشد.

الگوریتم (۸) ، شرط خاتمه پادگیری را معین می‌کند. پادگیری زمانی خاتمه پیدا می‌کند که کلیه تیمهای آناتلونهای شرکت کننده در تعیین خطوط تمایز، همگرا شده باشند یعنی آناتلونهای شرکت کننده در هر تیم، عمل بهینه خود را پیدا کرده باشند.

الگوریتم (۹) تا (۱۰) شبهه‌ای از تیمهای آناتلونهای پادگیری را بوسود می‌آورند. در هر مرحله، پاسخ محیط که همان نتیجه دسته‌بندی می‌باشد، برای کلیه آناتلونهای شرکت کننده، در هر تیم و برای کلیه تیمهای شرکت کننده در شکل پادگیری، ارسال می‌شود. شکل (۷) نتایی از این شبکه در لحظه k می‌باشد. تیمهای GC_1, \dots, GC_m پارامترهای هر یک از m خط تابع تمایز اصلاح شده را تعیین می‌نمایند. هر تیم، بردار عمل $\alpha_i(k)$ را که بینگر مقدارهای تغییر زده شده برای مجموعه پارامترهای خط V_i در مرحله k می‌باشد را انتخاب می‌نماید (در شکل، خط V_i بصورت تابع $h_i(\underline{\alpha}_i, \underline{x})$ نشان داده شده است). نتیجه دسته‌بندی هر تیم بصورت $y_i(k)$ برای بردار نمونه تصادفی \underline{x} در هر مرحله محاسبه شده و نتیجه کلی پازی با اجرای عملگر AND روی اجرای عملگر AND می‌باشد. این نتیجه به عنوان پاسخ برای شبکه پادگیری درنظر گرفته شده و به همه تیمها ارسال می‌گردد.



شکل ۷

مسئله نمونه ۲ : پیدا کردن توابع تمایز برای جداسازی کلاس P از هیج چهار کلاس W_1, W_2, W_3, W_4 با توزیعهای نرمال. مشخصات کلاسهای W_1, W_2, W_3, W_4 مطابق مسئله نمونه ۱ و ماتریس کوواریانس و بردار میانگین توزیع P مطابق رابطه (۸) می‌باشد.

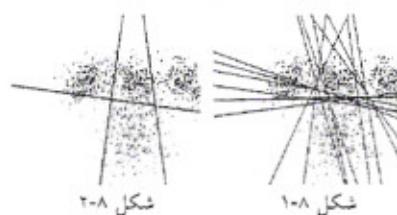
$$W_P : \Sigma = \begin{pmatrix} 1 & -0.5 \\ 0.5 & 2 \end{pmatrix}, \quad m = [20, 1]^T \quad (8)$$

الگوریتم (۲) به ازای $K = 100, I = 3, \sigma^2 = 0.5, MWD = 0.8, BP = 0.8, SL_{R-I} = 2000$ با استفاده از آناتلونهای SL_{R-I} پادگیری می‌باشد. برای حل مسئله نمونه ۲ پیدا کردن تابع تمایز برای ۱۰ بار اجرای الگوریتم را با پارامترهای ذکر شده نشان می‌دهد. در جدول (۲) مطری منوطه تعداد خطوط اولیه تعداد خطوط اولیه حدس زده شده در مرحله اول الگوریتم را نشان می‌دهد که با مقدار مشخص شده، توسط دقت اولیه کلاس الگوی P را از سایر کلاسهای جدا می‌کند. تعداد خطوط ثانویه، تعداد خطوط باقی مانده پس از انجام مرحله حذف خطوط غیرضروری را نشان می‌دهد که با دقت مشخص شده، توسط دقت ثانویه کلاس P را جدا می‌کند. دقت نهایی، بینگر دقت خطوط اصلاح شده، توسط آناتلونهای پادگیر می‌باشد.

شکل (۱۰-۸) تا (۱۰-۱۰) مراحل پادگیری توابع تمایز برای جداسازی کلاس P در مسئله نمونه ۲ را نشان می‌دهد. شکل (۱۰-۸) خطوط اولیه، تمحین زده شده را نشان میدهد. شکل (۱۰-۹)، ۳ خط باقی مانده پس از حذف خطوط غیرضروری را نشان می‌دهد. شکل‌های (۱۰-۱۰) تا (۱۰-۱۰) مراحل مختلف تصویح توابع تمایز که توسط آناتلونهای SL_{R-I} انجام می‌گیرد را نشان می‌دهد. شکل (۱۰-۱۰) وضعیت اولیه خطوط در شروع پادگیری و شکل‌های (۱۰-۱۰) تا (۱۰-۱۰) توابع تمایز اصلاح شده پس از

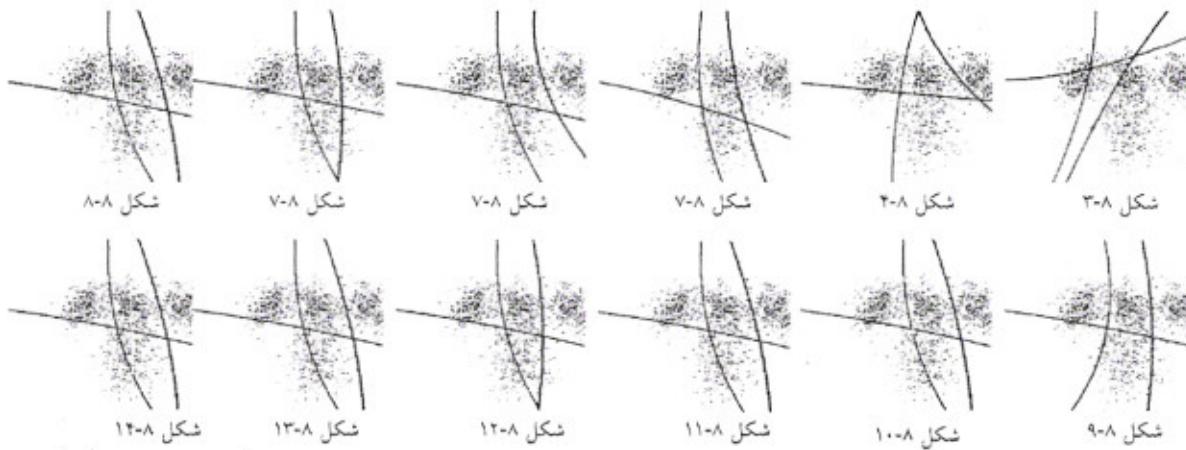
جدول ۲

۱۰	۹	۸	۷	۶	۵	۴	۳	۲	۱	شماره آزمایش
۱۷	۱۷	۱۰	۱۰	۲۰	۲۰	۱۷	۱۷	۱۶	۱۷	تعداد خطوط اولیه
۹۰/۳	۸۵/۱	۸۷/۹	۹۱/۴	۸۹/۴	۹۰/۲	۹۰/۷	۸۷/۳	۹۰	۸۸/۷	دقت اولیه
۳	۴	۳	۳	۵	۴	۴	۳	۴		تعداد خطوط ثانویه
۹۰/۲	۹۰/۰	۹۰/۲	۹۴/۷	۹۶/۳	۹۰/۶	۹۰/۹	۹۰/۰	۹۴/۶	۹۰/۸	دقت ثانویه
۹۰/۳	۹۰/۲	۹۶/۲	۹۰/۰	۹۶/۰	۹۰/۸	۹۶/۱	۹۰/۹	۹۶/۱	۹۰/۹	دقت نهایی
۳۹۱۳۱۷	۴۴۳۹۲۵	۲۲۱۰۷	۲۸۵۵۹۱	۲۱۴۴۴۹	۲۷۴۰۷۱	۲۲۴۰۷۶	۲۹۳۹۵۷	۱۴۷۷۹۱	۲۹۰۳۷۸	تعداد مراحل



شکل ۲-۸

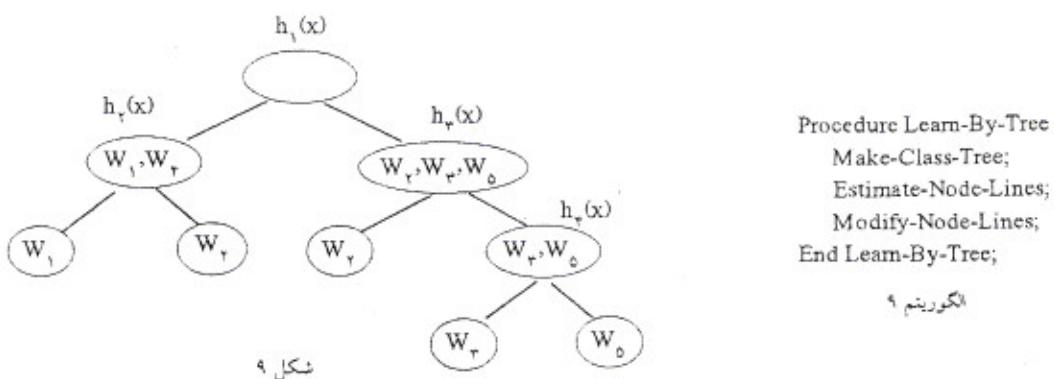
شکل ۱-۸



۵ - دسته‌بندی درختی

الگوریتم ارائه شده در بخش (۲) یک کلاس الگو را از سایر کلاس‌های الگو جدا می‌نماید. در صورتی که n کلاس W_1, W_2, \dots, W_n در حوزه مسئله مورد نظر وجود داشته باشد، با اجرای الگوریتم (۲) برای کلاس‌های مختلف مسئله، می‌توان هر کلاس را از سایر کلاسها جدا نمود. با استفاده از دسته‌بندی درختی، به شکل مولتیتری می‌توان عمل دسته‌بندی را انجام داد. روش کار پذیرن صورت است که ابتدا توسط یک الگوریتم قطعه‌بندی (Clustering Algorithm)، از طریق ترکیب برشی (Cutting and Joining) کلاس‌های مسئله با یکدیگر و ساختن کلاس‌های جدید، قطعات (Cluster) بزرگتر که قابلیت جداسازی بیشتری دارند، ساخته می‌شود و در مراحل بعدی، کلاس‌های موجود در هر قطعه از یکدیگر جدا می‌گردند. به این ترتیب یک درخت قطعه‌بندی برای جداسازی کلاس‌های مسئله مورد نظر تشکیل می‌شود بطوریکه در سطح بالایی، قطعات بزرگتر و در سطح پایین، قطعات کوچکتر و در برگها، کلاس‌های مسئله جدا می‌شوند. شکل (۹) یک مسئله با ۵ کلاس می‌باشد که بصورت درختی قطعه‌بندی شده است را نشان می‌دهد. فاصله تصمیم‌گیری هر گره، در قطعه سمت چپ و راست هر گره را از یکدیگر جدا می‌نماید.

تقسیم‌بندی کلاسها در هر گره به گونه‌ای تجامیم می‌گیرد که دقت جداسازی در آن گره بیشترین مقدار ممکن باشد، به این ترتیب، کلاس‌های مشابه در یک قطعه قرار می‌گیرند. با استفاده از الگوریتم دسته‌بندی درختی، دسته‌بندی‌های واضح‌تر (کلاس‌هایی که به راحتی از یکدیگر جدا می‌گردند) در سطح بالاتر و دسته‌بندی‌های غامض‌تر (دسته‌بندی کلاس‌هایی که بطور ساختاری به یکدیگر نزدیکتر می‌باشند) در سطح پایین‌تر (Make-Class-Tree) انجام می‌گیرد. الگوریتم (۹)، الگوریتم دسته‌بندی درختی را نشان می‌دهد. در الگوریتم (۹)، روش Estimate-Node-Lines درخت قطعات کلاس‌های مسئله را بوجود می‌آورد. روش Make-Class-Tree درخت قطعات کلاس‌های مسئله را بوجود می‌آورد. روش Estimate-Node-Lines خطوط تابع اولیه مربوط به جداسازی قطعه کلاسها در هر گره از درخت را ایجاد می‌کند. قطعات قرار گرفته در دو طرف هر گره می‌توانند توسط یک خط تابعیز و با دسته‌ای از خطوط تابعیز (مانند الگوریتم (۲) از یکدیگر جدا شوند. روش Modify-Node-Lines خطوط اولیه تقریب‌زده شده در هر گره را تصحیح می‌نماید.



۱-۵ - ایجاد درخت دسته‌بندی

فرض کنید n کلاس W_1, W_2, \dots, W_n در مسئله وجود داشته باشد. در صورتیکه بخواهیم این کلاسها را به دو گروه با $p \leq n$ کلاس تقسیم کنیم، مجموعاً $\binom{n}{p}$ حالت را می‌توان در نظر گرفت. در صورتیکه بخواهیم به هر شکل ممکن مجموعه کلاسها را به دو گروه تقسیم نماییم بطوریکه در هر گروه حداقل یک کلاس وجود داشته باشد ($n < p < 1$) تعداد حالتهای ممکن دسته‌بندی برابر با $\sum_{p=1}^n \binom{n}{p} = 2^n$ می‌باشد. بنابراین پیچیدگی الگوریتم محاسبه کننده بهترین قطعه‌بندی (تقسیم کلاسها به درقطعه بطوریکه بهترین دقت جداسازی بوجود آید) در هر گروه از مرتبه $O(2^n)$ می‌باشد. الگوریتم (۱۰)، الگوریتمی است که در هر گروه با مرتبه $O(n^3)$ اجرا می‌شود و کلاس‌های موجود در هر گروه را به دو دسته که بطور خطی از یکدیگر جدا نمی‌باشند، تقسیم بندی می‌نماید.

```

Procedure Make-Class-Tree (ClassTree : TClassTree; ClassSet : TClassLabelSet);
  if ClassSet.Count = 1 then Exit;
  ClassTree.LeftSet = [];
  ClassTree.Precision := 0;
  Repeat
    NextIndex := -1;
    for i := 1 to ClassSet.Count do
      LeftSet := ClassTree.LeftSet + ClassSet.Item[i];
      V := Estimate-Line (LeftSet);
      Precision := DiscriminateSamples (V);
      if Precision > ClassTree.Precision then
        NextIndex := i;
        ClassTree.Precision := Precision;
      end if;
    end for;
    if NextIndex > 0 then
      ClassTree.LeftSet := ClassTree.LeftSet + ClassSet.Item[NextIndex];
      ClassSet := ClassSet - ClassSet.Item[NextIndex];
    end if;
  Until NextIndex < 0;
  Make-Class-Tree (ClassTree.Left, ClassTree.LeftSide);
  Make-Class-Tree (ClassTree.Right, ClassSet);
End Make-Class-Tree;

```

الگوریتم ۱۰

الگوریتم (۱۰) یک الگوریتم بازگشتی است و دارای دو ورودی می‌باشد. در پaramتر $ClassTree$ ، درخت تجزیه کلاسها ساخته و ذخیره می‌گردد. مجموعه $ClassSet$ ، مجموعه کلاس‌هایی که باید تقسیم‌بندی شوند، می‌باشد. اگر تعداد کلاسها n باشد، حلقة داخلی الگوریتم حداقل $\frac{n(n+1)}{2}$ بار اجرا می‌گردد. توسط این الگوریتم مجموعه کلاسها به دو گروه سمت چپ و سمت راست تقسیم می‌گردد. ابتدا کلاسی که به بهترین شکل بوسیله یک تابع خطی قابل جداسازی باشد، از بقیه کلاسها جدا شده و به عنوان عضوی از گروه کلاس‌های سمت چپ درنظر گرفته می‌شود. در هر بار اجرای بعدی تا زمانی که با برداشتن کلاسی از کلاس‌های باقیمانده و

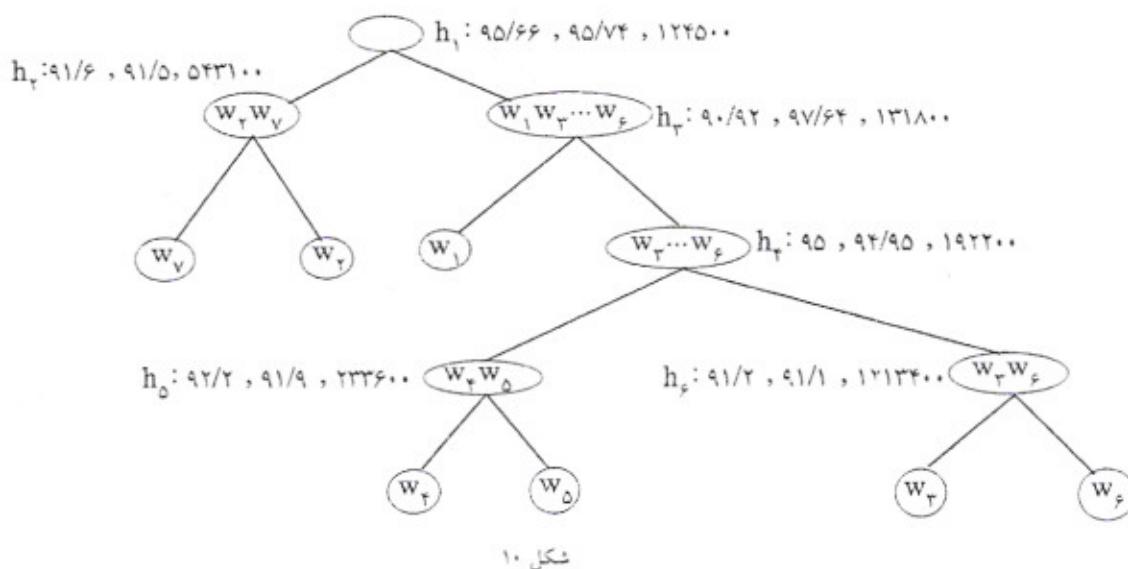
اضافه نمودن آن به گروه کلاس‌های سمت چپ دقت بالاتری بدست بیاید، الگوریتم ادامه پیدا می‌کند. در غیر اینصورت کلاس‌های جدیده به عنوان کلاس‌های گروه سمت چپ و کلاس‌های باقی‌مانده به عنوان کلاس‌های گروه سمت راست درنظر گرفته می‌شوند. الگوریتم بطور بازگشی برای گروه سمت چپ و راست اجرا می‌گردد. شرط خاتمه الگوریتم زمانی است که در هر گره انتها فقط یک کلاس وجود داشته باشد.

مسئله نمونه ۳: بدست آوردن تابع تعابیر برای جداسازی ۷ کلاس $W_1 \dots W_7$ با توزیع کلاس‌های $W_1 \dots W_7$ مطابق مسئله نمونه ۲ می‌باشد. ماتریس کوواریانس و بردار میانگین کلاس‌های $W_1 \dots W_7$ در زیر آمده است.

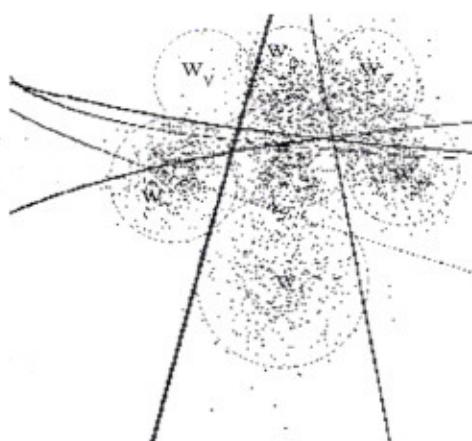
$$W_5 : \Sigma = \begin{pmatrix} 2 & 0/5 \\ 0/5 & 2 \end{pmatrix}, \quad m = [20, 0]^T \quad W_6 : \Sigma = \begin{pmatrix} 3 & 0/5 \\ 0/5 & 3 \end{pmatrix}, \quad m = [24, 0]^T$$

$$W_7 : \Sigma = \begin{pmatrix} 3 & 0/5 \\ 0/5 & 3 \end{pmatrix}, \quad m = [16, 0]^T$$

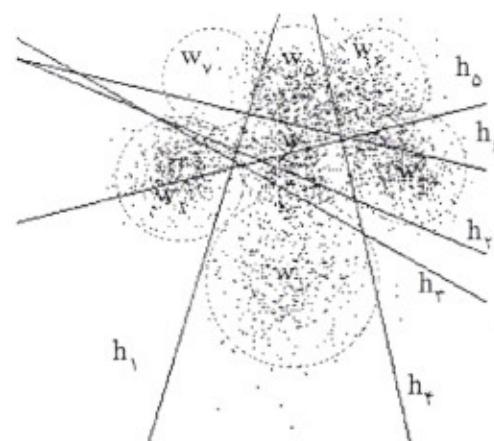
شکل (۱۰) نتیجه یک بار اجرای الگوریتم (۹) برای حل مسئله نمونه ۲ با پارامترهای $\alpha = K = 3$ و $\beta = 1$ و با درنظر گرفتن یک خط تعابیر در هر گره و تصریح آن با استفاده از آنماتونهای $SL_R - 1$ با پارامتر بادگیری $= 2000$ $W =$ را نشان می‌دهد. کار هر گره میانی، به ترتیب از چپ به راست، دقت تابع تعابیر اولیه حدس زده شده، دقت تابع تعابیر اصلاح شده و تعداد مرحله مورد نیاز برای اصلاح ثابت، نشان داده شده است. دقت جداسازی کلیه کلاس‌های توسط خطوط اولیه حدس زده شده، بود که پس از اصلاح خطوط، این دقت به $584/54$ ٪ افزایش یافت. شکل (۱۰-۱۱) وضعیت اولیه خطوط و شکل (۱۱-۱۲) وضعیت نهایی خطوط تعابیر را نشان می‌دهد.



شکل ۱۰



شکل ۱۱



شکل ۱۱

۶- جداسازی آواهای فارسی

در این بخش، کاربرد روش‌های ارائه شده در بخش‌های قبلی برای جداسازی آواهای گفتاری در زبان فارسی، مورد بررسی قرار می‌گیرد. به دلیل ویژگیهای خاصی که در سیگنال صحبت وجود دارد، پردازش سیگنال صحبت و تشخیص صدا پکی از مسائل بسیار پیچیده، محاسبه محسوب می‌گردد. وجود این ویژگیها باعث بوجود آمدن تکنیکهای ویژه‌ای در تشخیص سیگنال صحبت شده است [۱۷] [۱۸].

در گفتار فرآیندی بسیار پیچیده، محسوب می‌گردد، بطوریکه علاوه بر تشخیص جنبه‌های سیگنال صدا (مثلًا تشخیص آواهای تکنیکهای مختلف از هوش مصنوعی و

زبان طبیعی نیز باید مورد استفاده قرار گیرد. روش‌های تشخیص سیگنال صحبت به دو دسته کلی تقسیم می‌شوند که عبارتند از: روش‌های پارامتریک و روش‌های غیر پارامتریک. در روش‌های غیر پارامتریک، معمولاً نتیجه تجسس روی سیگنال صحبت، جهت آنالیز مورد استفاده قرار می‌گیرد. تبدیل فوریه یکی از این نوع تبدیلات می‌باشد. در روش‌های پارامتریک، سیگنال صحبت توسط یک مدل پارامتریک مدل می‌شود، پارامترهای مدل مورد نظر محاسبه گردیده و نتیجه آن جهت آنالیز گفتار یکار برده می‌شود. در این فصل از روش‌های پارامتریک و از آنالیز MFCC استفاده گردیده است.

محرای گفتار توسط چندین لوله صوتی باز و سته که با طولها و قطرهای مختلف به یکدیگر متصل شده باشد پتوانند تخمین از شکل مجرای گفتار انسان در هنگام تلفظ آرایی مورد نظر را حاصل نمایند. تعداد نولهای درنظر گرفته شده در مدل، باید بقدری باشد که پتواند دقت موردنظر را ایجاد کند و با توجه به کاربرد و دقت موردنظر این تعداد من تواند متغیر باشد. در مدل استفاده شده در این بخش، از ۱۲ لوله صوتی برای مدل کردن مجرای گفتار استفاده شده است.

آزمایش‌های النجام شده در این بخش برای جداسازی ۱۵ آرای مختلف /a/, /e/, /i/, /o/, /u/, /v/, /z/, /zh/ طراحی و پیاده‌سازی شده‌اند. هر آواز بصورت کالسی مشکل از مجموعه‌ای از نمونه‌های آموزشی و تست و بصورت بردارهای ویژگی در فضای R^{12} درنظر گرفته شده و مجموعه کلیه بردارهای ویژگی استخراج شده از نمونه‌های مدل پایه دادهای مسئله را تشکیل می‌دهد.

در مجموع از ۱۵ فرد مختلف با سن و جنس‌های مختلف و با درجه تحصیلات مختلف خویسته شد که هریک از آواهای فوق را بصورت عادی و به مدت یک ثانیه تلفظ نمایند و صدای هر فرد بصورت یک فایل kHz/s با فرمت wave می‌دانند. هر آواز بصورت مونو با استفاده از یک میکروفون سلفی $600\ \Omega$ امپی از طریق ورودی Mic کارت صوتی در رایله ذخیره گردید. در قدم بعد، آواهای مختلف موجود در صدای ضبط شده افراد مختلف بصورت دستی جدا گردید و مجموعه شکل موجود مربوط به هر آواز بصورت یک فایل wave ذخیره گردید. در این قدم، مدت نیم ثانیه از آرای تلفظ شده توسط هر فرد جدا گردید، این نیم ثانیه از قسمت‌های وسطی هر آرای تلفظ شده جدا گردید.

در مرحله بعد، مجموعه بردارهای ویژگی هر آواز انجام آنالیز MFCC روی فایلهای بدست آمده از بخش، قبل استخراج گردید. هر بردار ویژگی بصورت ضرایب آنالیز MFCC با ۱۲ مقادیر ویژگی درنظر گرفته شد. ضرایب MFCC روی پنجه‌های $WAVE$ مربوط به هر آواز محاسبه گردید و بین هر دو پنجه متوازن، روی هماندادگی $10 \times \frac{0.05s}{20ms} = 0.5s$ میلی‌ثانیه درنظر گرفته شد. به این ترتیب از هر آرای تلفظ شده توسط هر فرد تعداد $25 \times \frac{0.5s}{20ms} = 125$ نمونه بردار ویژگی و مجموعاً برای هر آواز حدود ۴۰۰ بردار ویژگی استخراج گردید. در پایگاه اطلاعاتی، بردارهای ویژگی مربوط به آواهای هر فرد به دنبال پکدیگر و بردارهای افراد مختلف پشت سر یکدیگر ذخیره گردید.

مجموعه بردارهای ویژگی هر آواز به دو مجموعه آموزشی و تست تقسیم گردید. 30% میلی‌ثانیه از فایل $WAVE$ مربوط به هر آواز درنظر گرفته شد. به این ترتیب عمل پادگیری و تست روى صدای افراد مختلف النجام می‌گيرد. لیست کامل نمونه‌های آزمایش شده را می‌توانید در مرجع [۱۹] ببینید.

۶-۱- دسته‌بندی دودویی

در این بخش برای هر در کلاس موجود در حوزه مسئله یک تابع تمايز در نظر گرفته می‌شود. به این ترتیب اگر M کلاس الگو در حوزه مسئله وجود داشته باشد، برای جداسازی هر کلاس از سایر کلاسها، $1 - M$ تابع تمايز درنظر گرفته می‌شود و مجموعاً تعداد $\frac{M(M-1)}{2}$ تابع تمايز تخمین زده می‌شود.

اگر $i = 1 \dots M$, $j = 1 \dots M-1$ باشند، $f_{ij} = \begin{cases} > 0 & \text{if } x \in w_i \\ < 0 & \text{if } x \in w_j \end{cases}$. برای دسته‌بندی یک بردار الگوی ورودی، این بردار توسط کلیه توابع تمايز دسته‌بندی می‌شود. به این ترتیب به بردار ورودی تعداد $\frac{M(M-1)}{2}$ برچسب نسبت داده می‌شود که پیانگر تعلق داشتن بردار ورودی به یک کلاس الگو توسط هر تابع تمايز از دید هریک از توابع تمايز می‌باشد. در حالت ایده‌آل، بردار ورودی x به کلاس w_i تعلق دارد اگر $f_{ij} > 0$ باشد، باشد ملاکی برای مقایسه نتیجه دسته‌بندی توابع تمايز تعریف گردد. این ملاک می‌تواند تعداد نسبت دهن الگو به هر کلاس درنظر گرفته شود، بدین معنی که بردار x که توسط پیشترین تعداد توابع تمايز به کلاس w_i نسبت داده شود، به عنوان عضوی از کلاس w_i درنظر گرفته می‌شود.

برای حل مسئله دسته‌بندی آواهای صوتی، از آنجا که ۱۵ آرای مختلف مورد بررسی قرار می‌گیرند، تعداد $105 = \frac{15 \times 14}{2}$ تابع تمايز با استفاده از تابع f_{ij} [۱]

تخمین زده شد. جدول (۴) درصد دسته‌بندی مجموعه نمونه‌های تست توسط توابع تمايز بدست آمده را نشان می‌دهد.

هر خانه از جدول (۴)، درصد دسته‌بندی مجموعه نمونه مربوط به هر کلاس مشخص شده توسط سطر مربوط به آن خانه، در کلاس مشخص شده با برچسب متنون آن خانه را نشان می‌دهد. به این ترتیب در صدای افراد مختلف تابع تمايز شده توسط خانه‌های مربوط به افراد اصلی، درصد دسته‌بندی افراد مختلف مربوط به هر کلاس آواز می‌باشد. در حالت ایده‌آل، عناصر فلتر اصلی باید ۱۰۰٪ و عناصر سایر خانه‌ها باید صفر باشند.

توابع تمايز خطی اولیه بدست آمده، با استفاده از روش پیشنهاد شده، در بخش (۳)، بوسیله پارسی آناتامونهای پادگیر با پاسخ یکسان اصلاح گردید. برای اصلاح توابع $W = 25$ و $K = 1$ درنظر گرفته شد. به این ترتیب برای هر آناتامون 2 عمل تعریف می‌گردد. برای اصلاح از آناتامونهای I_R با پارامتر پادگیری $SL_R = 1000$ استفاده گردید.

اصلاح توابع تمايز شده با پیشنهاد شده دقت جداسازی، به عنوان تابع شمايز بهینه درنظر گرفته شد.

پیارا مترهای $K = 50$ = ای‌پارادیگر مورد اصلاح قرار گرفتند و در قدم بعد توابع خطی اصلاح شده‌ای که دقت کمتر از ۹۰٪ نسبت داده شد، با درجه ۲ ارتقا داده شدند و بوسیله پارسی آناتامونهای پادگیری $SL_R = 1000$ ضرایب ای‌پارادیگر تغییر گردید. به این ترتیب با توجه به ۴ مرحله النجام شده، در

تعیین توابع تمايز (مرحله اول حدس اولیه با استفاده از تابع فیشر، مرحله درم الگوریتم ۲ و مرحله سوم و چهارم، تابع اصلاح‌های ذکر شده) برای جداسازی دو کلاس w_i و w_j حداقل ۴ تابع تمايز بدست آمد و نتیجه که حداقل دقت جداسازی را برای جداسازی دو کلاس w_i و w_j از خود نشان داد، به عنوان تابع تمايز نهایی انتخاب گردید. جدول (۵) درصد دسته‌بندی نمونه‌های آواهای مختلف توسط توابع تعیین شده را نشان می‌دهد.

جدول ۴

	/a/	/aa/	/e/	/f/	/i/	/kh/	/l/	/o/	/t/	/s/	/sh/	/u/	/v/	/z/	/zh/
/a/	۶۶	۰	۱	۱	۰	۰	۱	۱۱	۱۵	۰	۱	۱	۲	۱	۰
/aa/	۰	۸۸	۰	۰	۰	۳	۰	۰	۰	۰	۰	۰	۰	۰	۹
/e/	۴	۰	۲۵	۲	۹	۰	۳	۳	۵۲	۰	۰	۱	۰	۰	۰
/f/	۰	۰	۰	۸۰	۰	۰	۰	۱	۰	۰	۰	۹	۵	۰	۰
/i/	۱	۰	۵	۰	۰	۰	۲۲	۰	۱۱	۰	۱	۰	۸	۰	۰
/kh/	۰	۱۸	۰	۰	۰	۶۹	۰	۰	۰	۴	۰	۱	۰	۸	۰
/l/	۵	۰	۰	۰	۲۰	۰	۳۳	۴	۱۲	۰	۰	۱۸	۸	۰	۰
/o/	۳	۰	۲۰	۰	۰	۰	۱	۷۰	۲	۰	۰	۰	۰	۰	۰
/t/	۸	۰	۱۶	۲	۹	۰	۱۵	۸	۱۰	۰	۰	۱۶	۷	۲	۰
/s/	۰	۰	۰	۱۰	۱	۰	۰	۳	۱	۷۶	۰	۰	۶	۴	۰
/sh/	۰	۳	۰	۰	۰	۲۰	۰	۰	۰	۷۵	۰	۰	۰	۲	۰
/u/	۰	۰	۴	۰	۱۲	۰	۱۷	۲	۱۱	۰	۰	۴۹	۰	۰	۰
/v/	۱	۰	۶	۱۶	۹	۰	۱۷	۹	۲	۰	۰	۳	۱۹	۱۱	۰
/z/	۰	۰	۲۵	۹	۰	۰	۱	۰	۱۱	۰	۰	۱	۲۶	۲۲	۳
/zh/	۰	۲۴	۵	۰	۰	۱۱	۰	۴	۰	۰	۱۰	۲	۰	۰	۳۸

جدول ۵

	/a/	/aa/	/e/	/f/	/i/	/kh/	/l/	/o/	/t/	/s/	/sh/	/u/	/v/	/z/	/zh/
/a/	۷۷	۰	۱	۱	۰	۰	۰	۰	۱۸	۰	۱	۲	۲	۰	۰
/aa/	۰	۹۲	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۰	۳
/e/	۳	۰	۴۱	۵	۴	۰	۱	۰	۴۲	۰	۰	۳	۰	۰	۰
/f/	۰	۰	۰	۸۹	۰	۰	۰	۰	۰	۵	۰	۰	۰	۵	۰
/i/	۰	۰	۹	۰	۲۳	۰	۱۶	۰	۱۸	۰	۱	۲	۹	۲	۰
/kh/	۰	۳	۰	۰	۰	۸۰	۰	۰	۰	۰	۵	۰	۰	۰	۷
/l/	۰	۰	۰	۰	۱۱	۰	۲۶	۰	۲۸	۰	۰	۲۵	۸	۳	۰
/o/	۰	۰	۱۵	۰	۰	۰	۰	۷۷	۲	۰	۰	۵	۰	۰	۰
/r/	۰	۰	۹	۳	۳	۰	۵	۱۲	۲۵	۰	۰	۸	۹	۲	۰
/s/	۰	۰	۰	۱۰	۰	۰	۰	۰	۰	۹۰	۰	۰	۰	۰	۰
/sh/	۰	۱	۰	۰	۰	۲۱	۰	۰	۰	۰	۷۷	۰	۰	۰	۱
/u/	۰	۰	۲	۰	۶	۰	۳	۰	۲۶	۰	۰	۵۹	۵	۰	۰
/v/	۰	۰	۴	۲۱	۴	۰	۰	۰	۲۰	۱	۰	۷	۲۶	۱۱	۰
/z/	۰	۰	۷	۸	۰	۰	۱	۰	۵	۰	۰	۶	۱۲	۵۷	۰
/zh/	۰	۲	۰	۱	۳	۰	۲	۰	۰	۱۳	۴	۰	۰	۷۲	۰

از آنجا که اعضای قطر اصلی، بیانگر درصد دستبندی صحیح برای هر کلاس آوا می باشد و از آنجا که تعداد نمونه های آموزشی و تست آواهای مختلف تقریباً یکسان می باشد، میانگین قطر اصلی هر جدول را می توان به عنوان درصد دقت دستبندی کل سیستم دستبندی درنظر گرفت. در این صورت، دقت جدول(۵) برابر ۶۳/۶۷٪ است.

توضیح ۱: در دستبندی دودویی، برای مقایسه نتایج دستبندی نوعی تابع تمايز مختلف برای سیستم انتخاب یافته تعریف گردد. سیستم انتخابی که در این فصل استفاده گردید، یک سیستم ایستاده می باشد (به عنوان مثال، نسبت دهنی بردار نمونه به کلاس) که توسط پیشترین تعداد نوعی تمايز بردار نمونه به آن نسبت داده شده بالا). شاید بتوان با استفاده از یک سیستم یادگیر یا در این قسمت، نتایج دقیقتری بدست آورد.

توضیح ۲: توسط دستبندی دودویی می توان ملاکی برای تشابه ساختاری دو آوا (کلاس الگو) تعریف نمود. در صورتی که تشابه یک کلاس با خودش را فرض نماییم و اگر $t_{ij} \geq t_{ij} + t_{ii}$ عضو ۴۳ام جدول دستبندی باشد، مقدار کمی تشابه (t_{ij}) را می توان بصورت $t_{ij} = \frac{t_{ij}}{t_{ij} + t_{ii}}$ تعریف نمود.

توضیح ۳: برای شناسایی کلمه از طریق آشناسی، می توان سیگنال ورودی را به فریمها بیان نموده و آواها را با درنظر گرفتن محدودیتهایی که به حوزه کاربرد موردنظر منوط می شود، بصورت نویزی مشخص نمود. به این ترتیب یک رشته آواز نویزی بدست می آید. با داشتن یک سیستم استخراج برای شناسایی رشته های نویزی، می توان کلمه موردنظر را شناسایی نمود. مرجع [۲۰] پک الگوریتم شناسایی رشته کلمات نویزی با استفاده از آناتامونهای یادگیر ساختار ثابت را ارائه نموده است.

۷ - خلاصه

در این مقاله روش‌های ترکیبی مبتنی بر آناتامونهای یادگیر و روش‌های کلاسیک، برای دسته‌بندی الگوها ارائه شد. در اولین روش، ابتدا یکتابع خطی (غیر دقیق) با استفاده از روش‌های کلاسیک برای جذب‌آسازی یک کلاس از سایر کلاسها تخمین زده می‌شود و سپس پارامترهای تابع خطی بدست آمده بوسیله بازی آناتامونهای یادگیر تصحیح می‌گردد. در قسمت بعد، روش برای افزایش درجه تابع تخمین زده شده به منظور افزایش دقت تابع تعابیر، ارائه گردید. در بخش چهارم الگوریتم دسته‌بندی قطعه به قضمه تطبیقی «پیشنهاد گردید. در این روش، پوش توزیع هر کلاس الگو بوسیله مجموعه‌ای از خطوط توسط روش‌های کلاسیک تغیرپذیده می‌شود. بعد از حذف خطوط غیر ضروری (خطوطی که با حداف آنها، دقت دسته‌بندی تغییر نمی‌کند) با احتی افزایش می‌یابد) خطوط باقی مانده بوسیله آناتامونهای یادگیر اصلاح می‌گردند. در بخش پنجم، الگوریتم «دسته‌بندی درختی» پیشنهاد گردیده است. در این الگوریتم، ابتدا با توجه به میزان جدایی پذیری کلاسها از الگوی موجود در حوزه کاربرد مورد نظر، یک ساختار درختی از کلاسها ایجاد می‌گردد و سپس توابع تعابیر جدا کننده در هر گره از ساختار درختی توسط روش‌های که پیشتر به آنها اشاره گردید، تخمین زده می‌شود. کاربرد روش‌های پیشنهادی در مسئله جذب‌آسازی آوارهای گفتاری فارسی نیز بررسی شد.

۸ - منابع

- [1] Fukunaga, An Introduction to Statistical Pattern Recognition, New York, Academic Press, 1972.
- [2] Pierre A. Devijver and Joseph Kittler, Pattern Recognition Theory and Applications, NATO ASI Series, Series F : Vol. 30, 1986.
- [3] Andrew G. Barto and P. Anandan, "Pattern Recognition Stochastic Learning Automata", IEEE Trans. Sys. Man and Cybem., Vol. SMC-15, No. 3, May/January 1985.
- [4] Mandayam A. L. Thatachar and P. S. Sastry, "Learning Optimal Discriminant Functions Through a Cooperative Game of Automata", IEEE Trans. Sys. Man and Cybem., Vol. SMC - 17, No. 1, January/February 1987.
- [5] M. A. L. Thatachar and V. V. Phananskar, "A feedforward Network of Learning Automata for Pattern Recognition", in Proc. Int. Joint. Conf. Neural Networks, Singapore, Nov. 1991.
- [6] M. A. L. Thatachar and V. V. Phananskar, "Convergence of Teams and Hierarchies of Learning Automata in Connectionist Systems", IEEE Trans. Sys. Man and Cybern., Vol. 25, No 11, Nov. 1995.
- [7] V. V. Phananskar and M. A. L. Thatachar, "Global Convergence of Teams of Learning Automata", in Symp. Intell. Syst., Bangalore, Dec. 1991.
- [8] M. A. L. Thatachar and V. V. Phananskar, "Learning Global Maximum with Parameterized Learning Automata", IEEE Trans. Neural Networks, Vol. 6, No. 2, March 1995.
- [9] Kumpan S. Narendra and M. A. L. Thatachar, Learning Automata An Introduction, New Jersey, Prentice Hall, 1989.
- [10] K. Najim and A. S. Poznyak, Learning Automata Theory and Applications, Elsevier Science Ltd, 1994.
- [11] P. Mars, J.R. Chen and R. Nombiar, Leaming Algorithms Theory and Applications in Signal Processing, Control and Communications, CRC Press, New York, 1998.
- [12] S. Lakshimavarhan, Learning Algorithms Theory and Applications, New York, Springer - Verlag, 1981.
- [13] J. T. Tou and R. C. Gonzalez, Pattern Recognition Principals, Reading, MA : Addison-Wesley, 1974.
- [14] M. R. Meybodi and S. Lakshimavarhan, "On a class of Learning Algorithms which have a symmetric Behavior under Success and Failure", Springer - Verlag Lecture Notes in Statistics, PP. 145-155, 1984.
- [15] M. R. Meybodi, "Results on Strongly Absolutely Expedient Learning Automata", Proceedings of Inference Conference 86, ed. D.R. Moates and R. Butrick (Athens, Ohio : Ohio University Press, 1987), pp. 197-209.
- [16] N. Baba and H. Handa, "Utilization of Hierarchical Structure Stochastic Automata for the Back Propagation Method with Momentum", proc. of IEEE ICNN-95, pp. 389-393, 1995.
- [17] John R. Deller, John G. Proakis, and John H. L. Hansen, Discrete Time Processing of Speech Signal, Mc Millan, New York, 1993.
- [18] Sadaoiki Furui, Digital Speech Processing, Synthesis and Recognition, Marsel Dekker inc., New York and Basel, 1989.
- [19] G. R. Rezaei, Study the Performance of Learning Automata in Pattern Recognition, Msc. Thesis, Computer Engineering Department, Amir Kabir University of Technology, 1999. (in Farsi)
- [20] B. John Oemmen and Edward V. de St. Croix, "String Taxonomy Using Learning Automata", IEEE Trans. on Sys., Man and Cybem, Vol. 27, No. 3, April 1997.
- [21] G. R. Rezaei and M.R. Meybodi, "Automatic Determination of Discriminant Functions for Pattern Recognition", Proceedings of Fourth International Conference of Computer Society of Iran, pp. 298-309, 1998 (in Farsi).
- [22] G. R. Rezaei and M.R. Meybodi, "Determining Discriminant Functions for Pattern Classification Using Hierarchical Learning Automata", Proceedings of Seventh International Conference on Electrical Engineering of Iran, pp. 9-16, 1999 (in Farsi).