



## به دست آوردن پارامترهای مدل مخفی مارکوف به روش تکاملی

مریم تجویدی  
دانشگاه صنعتی خواجه نصیر الدین طوسی  
دانشکده مهندسی برق  
EMail: M\_Tajvidi@yahoo.com

رکسانا اخباری  
دانشگاه صنعتی خواجه نصیر الدین طوسی  
دانشکده مهندسی برق  
EMail: Fiora@safineh.net

امیر مسعود فرهمند  
دانشگاه صنعتی خواجه نصیر الدین طوسی  
دانشکده مهندسی برق  
EMail: SoloGen@ieee.org

**چکیده:** HMM به عنوان مدلی احتمالی کاربردهای بسیار گسترده‌ای چون استفاده در تشخیص صوت دارد. اما روش متداول تعلیم آن (روش Baum-Welch) چون بسیاری از روش‌های جستجوی محلی دیگر، در اکستریم‌های محلی گیر می‌افتد. از طرف دیگر روش‌های تکاملی تا حد زیادی از این مشکل به دورند. لذا در این مقاله چندین روش مختلف برای تکامل HMM پیشنهاد شده است که با روش متداول B-W مقایسه شده‌اند. نتایج حاصل بیان‌گر برتری مطلق روش‌های تکاملی است.

**کلمات کلیدی:** روش‌های تکاملی، مدل مخفی مارکوف (HMM)

### ۱- مقدمه

HMM (Hidden Markov Model) به عنوان ابزاری برای مدل کردن فرآیندهای اتفاقی کاربرد بسیار گسترده‌ای دارد. این مدل احتمالی به دلیل قابلیت‌های بالای خود در مسائلی چون تشخیص گفتار-که ذاتاً فرآیندی تصادفی است- بسیار به کار می‌رود، به طوری که اکنون به عنوان یکی از روش‌های مناسب تشخیص گفتار و همچنین مسائلی شبیه به آن، چون تشخیص هویت گوینده می‌باشد. [1] علاوه بر این، امروزه کاربرد‌های دیگری چون مدل کردن کانال‌های مخابراتی ناشناخته و تشخیص بافت در تصویر نیز پیدا کرده است. [2] همان‌طور که در بخش‌های بعدی خواهید دید، مساله اساسی در استفاده از HMM، در کاربردی چون تشخیص گفتار، بدست آوردن مدلی است که به خوبی بتواند نسبت به داده‌های اتفاقی وروردی، واکنش نشان بدهد و در صورتی که داده‌ها از همان منبعی باشند (که در اینجا، منبع می‌تواند چیزی مثل سیستم صوتی انسان در هنگام ادای حرف یا کلمه‌ای خاص باشد) باشند که با آن آموزش دیده است، این امر را مشخص کند. روش‌های مختلفی برای به دست آوردن پارامترهای مدل وجود دارد، که بدون شک الگوریتم تکراری Baum-Welch (کارکرد بر اساس Maximum-Likelihood) معروف‌ترین آنهاست. از دیگر روش‌ها می‌توان به روش‌های عمومی مبتنی بر گرادیان اشاره کرد. [1]، [3] اما بزرگترین عیب این دو روش این است که تنها یک جواب محلی را پیدا می‌کند و هیچ تضمینی برای ساختن مدل اکستریم وجود ندارد. (و اغلب مواقع نیز به آن نمی‌رسد). برای حل این مشکل می‌توان از روش‌های تکاملی استفاده کرد، که روش‌هایی هستند که در حل مسائل بهینه‌سازی به خوبی جواب داده‌اند. [4] هدف ما در این مقاله بررسی تاثیر بکارگیری روش‌های تکاملی برای حل مساله بدست آوردن پارامترهای HMM است. در اینجا چندین روش مختلف برای تکامل مدل، پیشنهاد شده است. ساختار این مقاله از این پس به این صورت می‌باشد:

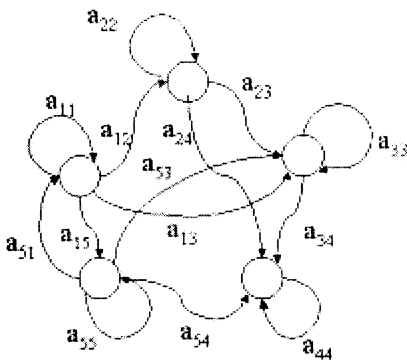
ابتدا در بخش (۲)، مرور مختصری بر HMM می‌کنیم، ایده آن، را بیان نموده و فرمول بندی مورد استفاده در این مقاله را بیان می‌کنیم. سپس در بخش (۳)، به طور خلاصه درباره روش های تکاملی خواهیم نوشت. در بخش (۴) که قسمت اصلی این مقاله است، چندین روش مختلف برای بکارگیری روش های تکاملی در بدست آوردن پارامترهای مدل، پیشنهاد می‌کنیم و پس از آن در بخش (۵) این روش ها را با هم و همچنین با روش کلاسیک Baum-Welch مقایسه می‌کنیم. در نهایت نتایج پیشنهاداتمان را در بخش (۶) خواهید دید.

## ۲- Hidden Markov Model:

هدف مدل سازی، پیش بینی خروجی فرآیندهای طبیعی می باشد. خروجی یک فرآیند طبیعی، ممکن است پیوسته یا گسسته باشد. منبع این سیگنال ممکن است، ایستا (Stationary) یا نا ایستا (Non-Stationary) باشد و همچنین سیگنال مشاهده شده، ممکن است ناشی از یک منبع تنها یا چند منبع هم زمان باشد. در هر حال، هدف ما بدست آوردن مدلی است که خروجی این فرآیند طبیعی را مدل کند و تا حد ممکن شبیه آن باشد.

مدل های مختلفی می توان پیشنهاد داد. می توان مدل Deterministic برای فرآیند ایجاد کرد و یا اینکه از مدل های اتفاقی (Stochastic) برای این کار استفاده کرد. هدف این مقاله، بررسی مدلی اتفاقی است به نام HMM، که گسترشی از Markov Model یا Markov Chain محسوب می شود.

از دیگر مدل های اتفاقی می توان به مدل گاوسی، پواسون و همچنین تلفیق گاوسی (Gaussian Mixture) اشاره کرد.



شکل ۱- Markov Chain

## ۲-۱- فرآیند مارکوفی و HMM:

در دیاگرام حالت شکل (۱)، هر State بیانگر یک خروجی سیستم می باشد و انتقال (Transition) بین این state ها، که با احتمال مشخص شده روی یال متصل کننده دو state مشخص می شود، باعث تولید خروجی های جدید می شود. به این مدل، Markov Model یا Markov Chain می گوئیم که کاربرد نسبتاً گسترده ای مخصوصاً در مخابرات دارد. اگر  $q_t$  را state سیستم در زمان  $t$  بدانیم و  $S_i$  را بیانگر state  $i$  ام:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, \dots] = P[q_t = S_j | q_{t-1} = S_i] \quad (1)$$

که  $a_{ij}$  بیانگر انتقال بین state  $i$  و  $j$  است. از این فرمول بندی مشخص می شود که در مدل مارکوف، به state قبلی اش مربوط است.

حال فرض کنید، به جای اینکه هر state، خود به تنهایی بیانگر خروجی سیستم باشد (به عبارت دیگر باعث تولید Observation شود)، بیانگر توزیع احتمالی فرآیند دیگری باشد که آن مشخص کننده خروجی سیستم است. یعنی اتصال بین دو state تنها توزیع احتمالی به وجود آمدن خروجی ها را تغییر بدهد. به این مدل HMM می گویند. [1]

## ۲-۲- فرمول بندی HMM:

در این قسمت به فرمول بندی HMM می پردازیم. فرض کنید، مدلی دارای  $N$  state مختلف باشد و در هر state یکی از  $M$  سمبل موجود به طور تصادفی انتخاب می شود. (در این جا HMM گسسته است) تعریف می کنیم:

$$A_{ij} = P[q_{t+1} = S_j | q_t = S_i] \quad 1 \leq i, j \leq N \quad (2)$$

که این احتمال انتقال بین دو state را مشخص می کند.

$$b_j(k) = P[V_k \text{ at } t | q_t = S_j] \quad 1 \leq j \leq N, 1 \leq k \leq M \quad (3)$$

که  $V_k$  سمبل انتخابی می باشد که از مجموعه  $M$  تایی انتخاب شده است.

$$\pi_i = P[q_1 = S_i] \quad 1 \leq i \leq N \quad (4)$$

که بیانگر احتمالات آغازین است.

همچنین باید داشته باشیم :

$$\begin{aligned} \sum_j a_{ij} &= 1 & (5) \\ \sum_k b(k) &= 1 & (6) \\ \sum_i \pi_i &= 1 & (7) \end{aligned}$$

با دانستن  $\lambda = (A, B, \Pi)$  می توان خروجی مدل یعنی  $O = O_1 O_2 \dots O_T$  را به صورت زیر به دست آورد:

(۱) با توجه به توزیع احتمال  $\Pi$ ، یکی از state ها را انتخاب می کنیم.  $(q_1 = s_i)$

(۲) قرار می دهیم :  $t=1$ .

(۳) با توجه به توزیع  $B$ ، یکی از  $M$  سمبل را انتخاب می کنیم.  $O_t = V_k$

(۴) با احتمال مشخص شده توسط ماتریس  $A$ ، از state  $i$  به state جدید  $j$  می رویم و قرار می دهیم :  $t=t+1$ .

(۵) تا هنگامی که  $t \leq T$ ، از مرحله ۲ دوباره تکرار می کنیم.

## ۲-۳- سه مساله اساسی :

برای استفاده از HMM به عنوان مدلی احتمالی، نیاز به حل ۳ مساله زیر داریم: [1]

(۱) ورودی های  $O = O_1 O_2 \dots O_T$ ، (هر  $O_k$  یک سمبل است) و  $\lambda = (A, B, \pi)$  را داریم. چگونه  $P(O | \lambda)$  را حساب کنیم ؟

(۲) ورودی های  $O = O_1 O_2 \dots O_T$  و  $\lambda = (A, B, \pi)$  را داریم. چگونه ترکیب  $Q = q_1 q_2 \dots q_T$  (دنباله state ها) را به دست بیآوریم به

طوری که به بهترین نحو توصیف گر انتقال بین state به ازای این دنباله ورودی باشد؟

(۳) با داشتن  $O = O_1 O_2 \dots O_T$  به عنوان داده های بدست آمده از منبع، چگونه می توان پارامترهای مدل

$\lambda = (A, B, \pi)$  را طوری انتخاب کرد که مدل بیشترین شباهت را (ماکزیمم  $P(O | \lambda)$ ) نسبت به  $O$  داشته باشد.

این ۳ مساله اساسی مطرح شونده در HMM می باشد. برای روشن شدن معنی هر کدام از این ها، سیستمی برای تشخیص صوت را در نظر می گیریم :

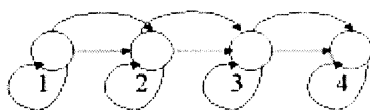
فرض می کنیم این سیستم قرار است  $W$  کلمه مختلف را تشخیص دهد. در نتیجه ما  $W$  مدل مختلف نیاز داریم. هر کدام از این مدل ها باید به کلمه ای که برای آن آموزش می بینند بیشترین شباهت (یا ماکزیمم  $P(O | \lambda)$ ) را نشان دهند. در ابتدا می بایست هر کدام از این مدل ها را با داده های بدست آمده از نمونه گیری صوتی آموزش دهیم. (مساله ۳) سپس برای تشخیص یک کلمه ناشناس، باید آن را به هر کدام از این  $W$  مدل مختلف وارد کرد و هر مدلی که  $P(O | \lambda)$  بیشتری داشت، به عنوان مدل توصیف گر آن کلمه مشخص می شود. (مساله ۱) از مساله ۲ برای بررسی دقیق تر چگونگی عملکرد مدل و بهبود آن، استفاده می شود

برای حل این مسائل روش های گوناگونی وجود دارد که البته بیشترین اختلاف در هنگام حل مساله ۳ بروز می کند. برای حل مساله ۱ از روشی موسوم به Forward-Bakward استفاده می شود. برای حل مساله ۲، معمولاً از الگوریتم Viterbi استفاده می شود. روش معمول برای حل مساله ۳، استفاده از روشی موسوم به Baum-Welch می باشد که یک روش EM (Expectation-Modification) بر اساس ML (Maximum Likelihood) است. البته روش های مبتنی بر گرادینان نیز وجود دارد که جواب های قابل مقایسه ای با روش قبلی می دهد. علاوه بر اینها به تازگی از روش های تکاملی نیز استفاده می شود که هدف اصلی این مقاله است. [5]، [6]، [7]

## ۲-۴- دیگر مسائل :

هدف ما در این مقاله ارائه توصیف نسبتاً کاملی از HMM نمی باشد و این بخش، تنها آشنایی دهنده ای با HMM است. برای همین تنها به اشاره ای به بعضی از دیگر صورت بندی ها و مسائل اکتفا می شود.

در قسمت قبل، فرض بر این بود که خروجی سیستم گسسته باشد. در عمل سیستمی که بیشتر به کار می رود، صورت بندی پیوسته HMM است که به جای  $b_i(k)$  گسسته، صورتی پیوسته قرار می گیرد. به علاوه، این مدل می تواند شکل های



شکل ۲- Left-Right HMM

مختلفی هم داشته باشد. مدلی که در آن هر state به تمامی state های دیگر متصل باشد، مدل Ergodic نام دارد. شکل متداول استفاده شونده در تشخیص گفتار و صوت، به صورتی مشابه با شکل (۲) است که به مدل Left-Right(Bakis) معروف است. همچنین از دیگر مسائل مطرح، می توان به اضافه کردن قابلیت تاخیر در مدل اشاره کرد (در این state ضرایب انتقال به خود برای تمامی state ها یعنی همان  $a_{ii}$  ها مساوی صفر قرار داده می شوند و چگالی استمرار (Duration Density) به طور صریح مشخص می شود. به این مفهوم که انتقال از یک state به state ی جدید تنها پس از اینکه سیستم به تعداد دفعات مناسبی در همان state باقی ماند و خروجی تولید کرد، صورت می گیرد.) که این نیز باعث بهتر عمل کردن سیستم های تشخیص گفتار می شود. [8] در نهایت، این که می توان مدل های دیگری غیر از ML مانند MMI(Maximum Mutual Information) استفاده کرد که البته باعث صورت بندی متفاوتی برای الگوریتم یادگیری (الگوریتم Baum-Welch به طور خاص) می شود. [9]

### ۳- روش های تکاملی :

بسیاری از مسائل طراحی در مهندسی، در اصل مساله ای بهینه سازی هستند. در این مسائل می بایست پارامترهای مساله را طوری تغییر داد تا به پاسخ بهینه-که توسط نوع مساله مشخص می شود- رسید. روش های مختلفی برای این کار وجود دارد که از مهم ترین و عمومی ترین آنها روش های مبتنی بر گرادینان هستند. اما عیب بزرگشان گیر افتادن در اکسترمم های محلی است. برای حل این مشکل روش های مختلفی وجود دارد که معروف ترین آنها روش های تکاملی ای هستند که امروز بسیار پر کاربرد گشته اند. [4] البته درباره چرایی و چگونگی عملکرد بهینه آنها در اینجا صحبتی نمی کنیم. [10]، [11]، [12]، [13]

### ۴- روش پیشنهادی:

در این قسمت چندین روش و صورت بندی برای تکامل HMM پیشنهاد می گردد. تجزیه و تحلیل دقیق این روش ها در اینجا صورت نمی گیرد زیرا در اکثر موارد نمی توان تئوری های توصیف گری چون Schema را که توصیف کننده رفتار تکاملی در GA است بکار برد و کاربرد آنها تنها محدود به state های بسیار ساده می باشد و بیشتر جنبه کیفی پیدا می کند تا کمی. در هر صورت فرض بر این است که تکامل HMM نیز چون بسیاری از مسائل مهندسی، مساله گمراه کننده ای (Deceptive) نباشد و یک سری روش های ساده برای Selection و همچنین CrossOver, Mutation جواب های خوبی بدهند و برتر بودن نسبی یک روش نسبت به بقیه با چند آزمایش تا حدی مشخص شود.

### ۴-۱- Fitness و Selection:

در این مساله، ما سه پارامتر  $A, B, \Pi$  داریم که به ترتیب برداری  $N$  بعدی، ماتریس  $N^2$  بعدی و ماتریسی  $NM$  بعدی هستند. برای ذخیره سازی آنها، از صورت بندی متداول در ES (و یا Real-Global GA) استفاده می کنیم و در نتیجه مجموعه پارامترهایمان، فضای جستجوی  $N+NM+N^2$  بعدی ای تشکیل می دهد. تنها نکته ای که لازم به ذکر هست، این است که در اکثر موارد برای اعمال اپراتورهای تکاملی روی هر عنصر جمعیتی (Individual)، هر کدام از این دسته پارامترها را به طور خاص در نظر می گیریم و با عناصر  $A$ ، مانند عناصر  $B$  رفتار نمی کنیم. این نکته بعداً مشخص خواهد شد.

برای ساختن جمعیت اولیه، از توزیع یکنواخت (بین ۰ و ۱) استفاده می کنیم و بعد به توجه به شروط  $(\Delta)$  تا  $(\gamma)$ ، آن ها را نرمال می کنیم. سپس تابع  $P(O|\lambda_i)$  را به ازای  $O_{Train}$  و  $\lambda_i$  های متعلق به جمعیت حساب می کنیم. بدیهی است که هدف ما، ماکزیمم کردن این تابع است. ما در اینجا همانند اکثر پیاده سازی های روش Forward-Backward، از  $\log(P(O|\lambda_i))$  استفاده می کنیم. اما با این وجود تفاوت مقدار  $\log(P(O|\lambda_i))$  های مختلف چندان زیاد نیست. (مثلاً بین ۵۰- و ۵۵- قرار دارند) و برای زیاد کردن این تفاوت و بالاتر بودن تفاوت بین یک عنصر خوب و بد از Ranking زیر برای Fitness استفاده می کنیم :

$$P_i = \text{abs}[P(O_{Train}|\lambda_i)] \quad 1 \leq i \leq \text{Population Size} \quad (8)$$

$$P = 1/N * \sum_i P_i, \quad \sigma_p^2 = 1/N * \sum_i (P_i - P)^2$$

$$P_{\min} = \min(P_i)$$

$$\text{Fitness}(P_i) = (P - (P_{\min} - \sigma_p^2) + K)^{-1}$$

که با تعیین  $K$  می توان میزان حداکثر Fitness را به طور تقریبی مشخص کرد. در پیاده سازی های ما  $K=0.1$  در نظر گرفته شده است. برای Selection از ترکیبی Deterministic, Stochastic استفاده می کنیم. کسری از بهترین اعضا به طور مستقیم به نسل بعد وارد می شوند و بقیه با برگزاری یک Selection تورنمنتی برای نسل بعد انتخاب می شوند.

#### ۴-۲-۱- اپراتورهای تکاملی

در این مقاله  $\gamma$  نوع پارامتر تکاملی پیشنهاد شده است که  $\gamma$  تای آنها از نوع Mutation و بقیه از نوع CrossOver هستند. با توجه به اینکه فرم نگهداری ژن ها در کوموزوم، به صورت حقیقی است، قدرت مانورمان نسبت به فرم متداول در GA بسیار بیشتر می شود. می توان گفت که پیاده سازی بیش از آنکه شبیه به GA باشد، به ES شباهت دارد.

#### ۴-۲-۱- اپراتور جهش ۱:

به احتمال  $P_m$ ، هر کدام از عناصر ماتریس  $A, B, \Pi$  مقدارشان با مقدار اتفاقی (با توزیع یکنواخت) بین  $0$  و  $M_{max}$  عوض می شود. به طور ساده می توان گفت که این شبیه ترین نوع Mutation ای است که به Mutation متداول در GA داریم و با اینکه صورت بندی حقیقی دارد، به نظر می رسد که نحوه عملکردشان تا حد زیادی شبیه به هم باشد. احتمال  $P_m$  (که جدا از احتمال انتخاب شدن این نوع Mutation است)، به طور تجربی مساوی  $0.05$  انتخاب شده است. مقادیر بیشتر باعث تغییر زیاد در ماتریس می شوند و Fitness آن به طور ناگهانی کاهش پیدا می کند و مقادیر کم هم باعث کند شدن بیش از حد فرآیند می شوند. مقدار  $m_{max}$  برای هر کدام از ماتریس های  $A, B, \Pi$  تفاوت دارد و باید متناسب با  $M, N$  انتخاب شود تا حداکثر معقول یک ضریب را در بر بگیرد. (مثلا از آنجایی که در یک ماتریس  $\Pi$  با  $5$  عنصر، وجود درایه ای با مقدار  $0.9$  معقول نیست) (مگر در حالات خاص) این مقدار کوچک تر و در حد  $0.5$  انتخاب می شود). نکته ای که لازم به ذکر است، این است که پس از اجرای هر اپراتور تکاملی، ماتریس ها بایست نرمال شوند تا در شرایط صدق کنند.

#### ۴-۲-۲- اپراتور جهش ۲:

این اپراتور نیز مشابه با اپراتور قبلی است، با این تفاوت که تغییرات هر درایه ماتریس متناسب با مقدار آن درایه است و در نتیجه تغییرات شدید در درایه ای که باید کوچک باشد به وقوع نمی پیوندد. در این روش داریم:

$$\Delta a_{ij} = \alpha a_{ij} \quad 0 < \alpha < m_{max} \quad (9)$$

می توان انتظار داشت که این روش فضای جستجو را نسبت به روش قبلی آرام تر می پیماید ولی در عوض احتمال خراب شدن (degrade) یک عنصر کمتر است.

#### ۴-۲-۳- اپراتور Simple Crossover

این روش، ساده ترین حالت برای انجام عمل Crossover می باشد. این نحوه شکل دهی کروموزوم که در [5] و [6] برای مدل Left-Right (یا Bakis) جهت تشخیص صوت به کار رفته و نتایج خوبی هم داده است به صورت زیر می باشد:

فرض می کنیم همه سطرهای ماتریس  $A$  و  $B$  را پشت سر هم به صورت برداری  $N^2$  و  $NM$  بعدی مرتب کرده ایم. برای انجام عمل Crossover از Simple 1-Point Crossover برای ماتریس  $A$  و  $\pi$  و Simple 2-Points Crossover برای ماتریس  $B$  استفاده می کنیم.

#### ۴-۲-۴- اپراتور Interpolation Crossover (BLX- $\alpha$ )

در این روش، مکان جدید نقطه ای ایجاد شده در فضای جستجو ( $N^2$  بعدی برای  $A$  و  $NM$  بعدی برای  $B$  و  $N$  بعدی برای  $\pi$ )، مکانی بین دو بردار  $V_1$  و  $V_2$  می باشند که هر کدام از این ها بیان گر پارامترهای  $\pi, A, B$  مدل می باشند. برای ایجاد کروموزوم

$$\begin{aligned} A' &= \alpha \cdot A_1 + (1 - \alpha) \cdot A_2 \\ A'' &= (1 - \alpha) \cdot A_1 + \alpha \cdot A_2 \end{aligned} \quad (10)$$

چنین رابطه ای برای  $B$  و  $\pi$  نیز صادق است. در رابطه ی بالا،  $\alpha$  عددی است اتفاقی بین  $0$  تا  $1$  که البته در این پیاده سازی،  $\alpha$  را توزیع نرمالی با واریانس کوچک تر از  $1$  و متوسط صفر گرفته ایم. دلیل این کار این است که اولاً تغییرات حول کروموزوم اصلی کوچک باشد (کوچک بودن واریانس) و هم چنین کروموزوم حاصل بتواند بیرون از فضای درون یابی شده نیز باشد (با منفی شدن

مقدار  $\alpha$  و انجام عمل برون‌یابی) که این عمل یکی از راه‌های گسترش قابلیت جستجوی کل فضای پارامتری (exploration) می‌باشد.

ایده‌ی استفاده از این نوع عمل‌گر این است که فرض می‌کنیم مجموعه پارامترهای دو کروموزم در دو سوی قله‌ای از تابع Fitness قرار گرفته‌اند و با این عمل نتیجه‌ی حاصل به سمت قله حرکت می‌کند.

#### ۴-۲-۵- اپراتور Row Replacing Crossover

این اولین نوع Crossoverی است که به ساختار HMM نیز نظر دارد و بر اساس ساختار ویژه‌ی آن عمل می‌کند. در این روش دو state مختلف از دو کروموزم متفاوت با هم تعویض می‌شوند. یعنی تمام درایه‌هایی از ماتریس انتقال که به یک state خاص وارد می‌شوند، از stateای متفاوت از ماتریس دیگری گرفته می‌شود و برعکس.

$$A' = A_1 \ \& \ A'' = A_2 \quad (11)$$

except :

$$a'_{ij_{new}} = a_{ik_2}$$

$$a''_{ik} = a_{ij_1}$$

فرض شده است که یک state با ورودی‌های اش می‌تواند بیان‌گر یک Building Block خوب باشد که لازم‌ه رسید به جواب مطلوب است.

#### ۴-۲-۶- اپراتور State Replacing Crossover I

در این شیوه Crossover، تعدادی از stateهای یک کروموزم با stateهای کروموزم دیگر مبادله می‌شود. به این معنا که با انتخاب اتفاقی  $1 \leq m \leq N$ ، تمام درایه‌های ماتریس انتقال بین stateهای کوچک‌تر از  $m$  و stateهای بزرگ‌تر و مساوی آن از کروموزم اول با معادل‌شان در کروموزم دوم مبادله می‌شوند. می‌توان این را تعمیمی از روش قبلی دانست زیر چندین state را در بر می‌گیرد. اما از طرف دیگر تفاوتی که دارد این است که هم انتقال‌های مشخص شونده با  $a(i,j)$  و همچنین  $a(j,i)$ ها را که شرط مورد نظر را داشته باشند، در بر می‌گیرد درحالی که روش قبلی تنها بر درایه‌های ماتریس انتقال یک جهت تاثیر می‌گذاشتند. هم‌چنین در این روش بر ماتریس احتمالات (B) تغییری داده نمی‌شود.

#### ۴-۲-۷- اپراتور State Replacing Crossover II

این شیوه نیز درست مشابه روش قبل است با این تفاوت که دو حد بالا و پایین برای مجموعه stateهایی که جابه‌جا می‌شوند گذاشته می‌شود.

### ۵- نتایج آزمایش:

در بخش قبلی روش‌های متفاوتی برای به کارگیری سیستم‌های تکاملی برای تعیین پارامترهای HMM پیشنهاد شد. همان‌طور که گفته شد برای بررسی دقیق و ریاضی تاثیر یک عمل (مانند Crossover، Mutation، Selection و هم‌چنین چگونگی اختصاص Fitness به تابع Cost) معمولاً بسیار پیچیده و غیرعملی است. لذا بهترین و ساده‌ترین روش مقایسه بین این‌ها، آزمایش روش‌های مختلف روی مساله‌های واقعی است.

ما برای این کار مدل آزمایشی زیر را پیشنهاد داده‌ایم که تا حد خوبی مشخص کننده‌ی کارکرد سیستم است:

HMM مدلی است که یک فرآیند مارکوفی (فرآیندی که دارای چندین state مختلف است و هر state تنها به state قبلی وابستگی دارد) را تخمین می‌زند و هر چه مدل به منبع مارکوفی تولیدکننده‌ی خروجی نزدیک‌تر باشد، مدل بهترین تقریب‌زده شده است. با وجود این که سیستم‌های طبیعی‌ای چون سیستم تولید گفتار در انسان واقعاً یک مدل مارکوفی نیست ولی در عمل با یک مدل مارکوفی می‌توان تا حد خوبی ساختار داخلی آن‌ها را مدل کرد. در نتیجه اگر سیستمی بتواند هر سیستم مارکوفی ناشناخته را مدل کند، می‌توان نتیجه گرفت که سیستم تعلیم‌دهنده‌ی خوبی برای آن مدل مارکوفی است.

در این مقاله ما تعدادی HMM اتفاقی و ناشناخته (برای سیستم تعلیم‌دهنده) به وجود می‌آوریم و دنباله‌ای از خروجی آن به دست می‌آوریم. هر کدام از این دنباله‌ها را به سیستم تعلیم‌دهنده‌ای که تنها از یکی از روش‌های ذکر شده استفاده می‌کند می‌دهیم و اجازه می‌دهیم تا مدل پیشنهادی خود را تعلیم دهد. سپس این مدل را با داده‌های دیگری که از منبع ناشناخته (برای

مدل تعلیم‌یافته) تولید شده است، مورد آزمایش قرار می‌دهیم. ( $P(O_{Test}|\lambda)$  را حساب می‌کنیم). نتیجه، مشخص کننده‌ی میزان خوب بودن مدل است. در جدول‌های زیر نتایج 7 مدل تکاملی مختلف پیشنهاد شده و همچنین مدل به دست آمده از طریق روش تکراری Baum-Welch را می‌بینید. در این سلسله آزمایش‌ها، پارامترهای زیر انتخاب شده است:

برای هر مجموعه پارامتر (N و M که به ترتیب مشخص کننده تعداد stateها و تعداد سمبل‌های مدل اتفافی‌اند) 5 آزمایش مختلف با 5 مدل متفاوت صورت گرفته است. تعداد مجموعه داده‌های تعلیم و همچنین تست از هر مدل منبع، 10 مجموعه بوده که هر کدام شامل 15 سمبل (به عبارت دیگر، شامل 15 observation) بوده‌اند. جمعیت سیستم تکاملی 20 عضو داشته است و  $P(Crossover)=0.6$  (در صورت انتخاب یک اپراتور از نوع Crossover) و  $P(Mutation)$  برابر 0.05 برای Mutation نوع اول و 0.08 برای نوع دوم انتخاب شده بود. حداکثر تعداد نسل‌های ممکن 30 نسل بود که البته در صورتی که شرط هم‌گرایی (تفاوت متوسط cost (که همان  $P(O_{Train}|\lambda)$  می‌باشد) در دو نسل متوالی کم‌تر از 0.01 باشد) برقرار می‌بود، تکامل در طی دو نسل پس از آن وضعیت پایان می‌یافت. این شرط باعث این می‌شد که در اکثر مواقع تعداد نسل‌های محاسبه شده کم‌تر از 20 بشود. برای الگوریتم Baum-Welch نیز حداکثر تکرار به 100 محدود شده بود که البته معمولاً در طی 40 تکرار سیستم هم‌گرا می‌شد. نتایج به دست آمده به خوبی بیان‌گر برتری روش‌های تکاملی نسبت به روش جستجوی محلی Baum-Welch می‌باشد. این تفاوت هر چقدر مدل پیچیده‌تر می‌شود (افزایش تعداد stateها و یا سمبل‌های قابل تولید) آشکارتر است به طوری که در موردی چون  $N=6$  و  $M=64$  مقادیر به دست آمده از روش دوم عملاً قابل استفاده نیست.

علاوه بر تفاوت معنادار در  $\log P(O|\lambda)$  بین دو گروه، ویژگی دیگری که مشخص شده است، تغییرات زیاد آن مقدار در روش Baum-Welch است که به خوبی بیان‌گر ویژگی جستجوی محلی آن می‌باشد. با توجه به این موارد می‌توان ادعا کرد که روش‌های تکاملی از نظر بهینه‌سازی، مدل بسیار بهتری را ارائه می‌دهند. گرچه عیب بزرگ آن‌ها این است که نسبت به روش‌های دیگر جستجو بسیار زمان‌بر می‌باشند به طوری که محاسبه نتایجی که می‌بینید چندین ساعت بر روی یک کامپیوتر شخصی طول کشید. (از ارائه اطلاعات کمی به دلیل عدم زمان‌بندی دقیق و بهینه نبودن کد برنامه صرف‌نظر می‌شود).

اما از طرف دیگر مقایسه‌ی بین روش‌های تکاملی چندان آسان نیست و تاویل دقیق آن‌ها نیاز به بررسی‌های سیستماتیک‌تری دارد. همچنین به دلیل این که بعضی از اپراتورهای تکاملی چون State Cut Crossover I,II تنها بر روی ماتریس A عمل می‌کنند، در نتیجه هیچ‌گونه جستجویی بر روی فضای B و  $\pi$  انجام نمی‌شود و لذا منطقی است که این روش از حداکثر قابلیت تغییر پارامترها برای به دست آوردن Building Block های مناسب استفاده نمی‌کند. گرچه با این وجود می‌بینیم که این دو روش به طور نسبی خوب کار می‌کنند (مخصوصاً در داده‌های تست) و این بیان‌گر قدرت بالای آن‌هاست که در صورت ترکیب با روش‌های دیگر می‌توانند به عنوان اپراتور بسیار مناسبی برای جستجو درآید.

N=4 M=16	Simple Cross.	InterPole Cross.	Row Replacing Cross.	State-Cut1 Cross.	State-Cut2 Cross.	Mut.1	Mut.2	Baum-Welch
TR.Mean	-39.347	-39.510	-38.280	-39.461	-39.678	-39.457	-38.063	-47.100
TR.Var	1.096	0.775	0.750	1.034	0.702	0.990	1.5372	6.3927
TE.Mean	-41.147	-42.382	-42.483	-41.597	-41.377	-41.997	-42.097	-52.841
TR.Var	1.160	2.152	2.522	1.233	1.246	1.852	1.556	9.861

N=4 M=32	Simple Cross.	InterPole Cross.	Row Replacing Cross.	State-Cut1 Cross.	State-Cut2 Cross.	Mut.1	Mut.2	Baum-Welch
TR.Mean	-50.191	-50.551	-48.739	-50.341	-49.878	-49.434	-48.920	-59.061
TR.Var	0.534	0.362	0.512	0.760	0.601	1.175	1.165	9.156
TE.Mean	-53.346	-52.887	-54.468	-52.012	-53.387	-54.746	-52.799	-58.309
TR.Var	1.446	1.089	2.203	1.737	1.838	5.551	2.526	9.592

N=6 M=32	Simple Cross.	InterPole Cross.	Row Replacing Cross.	State-Cut1 Cross.	State-Cut2 Cross.	Mut.1	Mut.2	Baum-Welch
TR.Mean	-50.366	-50.419	-48.549	-50.173	-50.465	-50.456	-48.910	-103.19
TR.Var	0.261	0.507	0.714	0.418	0.398	0.850	1.181	39.572
TE.Mean	-52.691	-51.810	-54.276	-52.852	-52.631	-53.618	-53.208	-168.67
TR.Var	0.841	0.726	2.296	0.656	1.244	3.216	0.991	48.08

N=4 M=16	Simple Cross.	InterPol Cross.	Row Replacing Cross.	State- Cut1 Cross.	State- Cut2 Cross.	Mut.1	Mut.2	Baum- Welch
TR Mean	-60.975	-60.976	-59.187	-60.612	-60.333	-59.968	-59.959	-191.421
TR Var.	0.372	0.302	0.633	0.266	0.257	1.519	0.735	129.295
TE. Mean	-62.452	-62.986	-62.236	-63.209	-62.618	-67.230	-64.620	-Inf
TE. Var	0.794	1.460	1.387	1.030	1.102	5.158	0.572	NaN

## ۶- نتیجه گیری و پیشنهادها:

در این مقاله ابتدا کمی درباره HMM و ایده‌ی نهفته در آن نوشتیم و سپس با بیان این که روش‌های تکاملی در بسیاری از مسایل بهینه‌سازی به خوبی جواب می‌دهند، چندین اپراتور متفاوت برای عملیات Crossover و Mutation برای HMM ارائه دادیم. در نهایت با مقایسه این روش‌ها با روش متداول Baum-Welch، برتری روش تکاملی را نشان دادیم. مراحل زیر برای ادامه‌ی کار پیشنهاد می‌شود که در حال بررسی و مطالعه درباره‌شان هستیم: بررسی شکل کلی تابع  $P(O|\lambda)$  در فضای پارامتری HMM و ارائه روش‌های جدید بر اساس ساختار ویژه‌ی آن‌ها، بررسی میزان کارکرد روش‌های تکاملی بر روی مسایل عملی‌ای چون تشخیص صوت و مقایسه با دیگر روش‌ها، پیاده‌سازی توزیعی سیستم تکاملی برای گسترش قابلیت exploitation و exploration [14]، پیاده‌سازی سیستمی هایپرید که در آن از روش‌های جستجوی محلی (مانند Baum-Welch) و جستجوی کلی (سیستم‌های تکاملی) با هم استفاده شده باشد. (سیستم‌های تکاملی لامارکی)

## ۷- مراجع:

- [1]-Rabiner, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," in Proceedings of The IEEE, vol. 77, no. 2, February 1989.
- [2]-Li, J., Najmi, A., Gray, R. M., "Image Classification by a Two Dimensional Hidden Markov Model," in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999.
- [3]-Galindo, P. L., "The Competitive Forward-Backward Algorithm (CFB)," in IEE Artificial Neural Networks, Conference Publication, 1995.
- [4]-Goldberg, D. E., Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing, 1989.
- [5]-Chau, C. W., Kwong, S., Diu, C.K., Fahrner, W.R., "Optimization of HMM by a Genetic Algorithm," in IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing, 1997.
- [6]-Kwong, S., Chau, C. W., "Analysis of Parallel Genetic Algorithms on HMM Based Speech Recognition System," in IEEE Trans. on Consumer Electronics, Nov 1997, Vol 43/4.
- [7]-Kwon, J. Lee, S., Shin, Ch., Jang, Y., Hong, S., "Signal Hybrid HMM-GA-MLP Classifier for Continuous EMG Classifier purpose," Engineering in Medicine Biology Society, 1998, Proceeding of the 20<sup>th</sup> Annual Inter. Conf. of the IEEE.
- [8]-Russel, M. J., Moore, R. K., "Explicit Modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition," 85, 1985
- [9]-Juang, B. H., Levinson, S. E., Sandhi, M. M., "Maximum Likelihood Estimation for Multivariate Mixture Observations of Markov Chains," IEEE Trans. on Information Theory, vol. IT-32, no.2, Mar 1986.
- [10]-Stephens, Ch., Walebroech, H., "Schemata Evolution and Building Blocks", MIT Evolutionary Computation, vol 7, no 2, 1999.
- [11]-Goldberg, D.E., "Real-Coded Genetic Algorithms, Virtual Alphabets, and Blocking," Illinois Genetic Algorithm Lab (IlligAL) report no. 90001
- [12]-Goldberg, D.E., Deb, K., Clark, J.H., "Genetic Algorithms, Noise, and the Sizing of Population," IlligAL report no. 91010
- [13]-Goldberg, D.E., Deb, K., Thierens, D., "Toward a Better Understanding of Mixing in Genetic Algorithms", IlligAL report no. 92009

[۱۴]-فرهمنده، ام.، میرمیرانی، ا.ع.، "سیستم‌های الگوریتم ژنتیکی توزیعی"، سومین کنفرانس دانشجویی مهندسی برق ایران، ۱۳۷۹.