بسمه تعالی

دانشکده برق و کامپیوتر
دانشگاه صنعتی اصفهان

مدل مخفی مارکوف

گزارش سمینار درس شناسائی آماری الگو

ترم دوم ۸۶-۸۵

مدرس: آقای دکتر جواد عسگری

ارائه دهنده: نرجس خاتون حبیبی

تابستان ۸۶

فهرست

۱- مقدمه ۳
۲- تاریخچه ۳
۳- شرح مدل ۳
۳-۱- مدلهای مارکوف ۳
۳-۲- مدل های پنهان مارکوف ٦
۳-۳- سه مساله اساسی مدل های پنهان مارکوف ۱۰
۳-۳-۱- حل مساله ۱ ..... ۱۰
۳-۳-۲- حل مساله ۲ ..... ۱۲
۳-۳-۳- حل مساله ۳ ..... ۱۵
۴- کاربردهای مدل در شناسائی آماری الگو ۱۹
۴-۱- کاربرد مدل در شناسائی گفتار ۱۹
۴-۱-۱- مدل کانال نویزی ۲۰
۵- نتیجه گیری ۲۲
۶- مراجع ۲۳

۱- مقدمه

مدل های مخفی مارکوف ابتدا در اواخر دهه ۱۹۶۰ و اوایل دهه ۱۹۷۰ معرفی و مورد مطالعه قرار گرفت. روش‌های آماری مدل سازی مارکوف پنهان به طور روزافزونی در سال‌های اخیر مورد توجه قرار گرفته اند. برای این امر دو دلیل عمده را می توان برشمرد. اولاً این مدل‌ها از نظر ساختمان ریاضی بسیار غنی هستند و بنابراین می‌توانند مبنای نظری برای استفاده در محدوده وسیعی از کاربردها را تشکیل دهند. ثانیا مدل‌ها، اگر به طور صحیحی به کار برده شوند، در عمل برای کاربردهای مهم نتیجه مطلوبی خواهند داشت.

در این نوشتار به بررسی اجمالی این مدل می پردازیم. ساختار مطالب به صورت زیر است:
در بخش ۲ تاریخچه کوتاهی از مدل ارائه شده است. بخش ۳ مدل مارکوف و مدل مخفی مارکوف را شرح می دهد. در بخش ۴ کاربرد مدل در شناسائی گفتار بررسی شده است و در پایان نتیجه گیری و مراجع استفاده شده در جمع آوری تحقیق قرار دارد.

۲- تاریخچه

مدل های مخفی مارکوف اولین بار در یک سری از مقالات آماری توسط Leonard E. Buam و نویسندگان دیگر در دهه ۱۹۶۰مطرح گردید. اولین کاربرد آن در شناسایی گفتار بود که در دهه ۱۹۷۰شروع گردید. در نیمه دوم ۱۹۸۰برای آنالیز رشته های بیولوژیکی بخصوص DNA از آن استفاده شد و از آن زمان به عنوان یکی از روش های پرکاربرد در زمینه بیوانفورماتیک درنظر گرفته می شود.

۳- شرح مدل

۳-۱- مدلهای مارکوف

خروجی یک فرآیند در جهان واقعی به شکل یک سیگنال پیوسته یا گسسته مشاهده می شود. یک مساله حیاتی در علوم، ساختن مدل هایی برای این سیگنال های واقعی است . مدل سازی یک سیگنال مزایای فراوانی به همراه دارد . اولا، مدل، پایه ای برای توصیف تئوری سیگنال فراهم می کند که می تواند برای پردازش سیگنال استفاده شود تا خروجی خواص مطلوبی داشته باشد. ثانیا، مدل می تواند اطلاعات بسیار مفیدی درباره منبع سیگنال بدهد، بدون اینکه احتیاجی به خود منبع باشد . نهایتا و از همه مهمتر، مدل ها می توانند در عمل به خوبی کار کنند و امکان تحقق سیستم های عملی مهمی را فراهم می آورند.

بسته به نوع و خواص سیگنال، راه های مختلفی برای مدل کردن یک سیگنال وجود دارد . به طور کلی، یک سیگنال می تواند معین یا نامعین (تصادفی یا آماری) باشد . مدل های معین از

بعضی خواص شناخته شده سیگنال استفاده می کنند و مقادیر پارامترهای مدل را تخمین می زنند . در طرف دیگر، در مدل های آماری، یک فرآیند تصادفی، سیگنال را توصیف می کند . برای کاربردهایی نظیر تشخیص گفتار یا دست خط که با نویز و عدم قطعیت همراه هستند، مدل های آماری از کارایی بهتری برخوردارند. مدل های پنهان مارکوف (Rabiner, 1989) که **منابع مارکوف** یا **توابع آماری زنجیره های مارکوف** نامیده می شوند، در تئوری مخابرات یکی از پرکاربردترین مدل های آماری هستند.

دسته مهمی از فرآیند ها ی تصادفی، فرآیندهای مارکوف است که دارای خواصی است که مطالعه ریاضی آن ها را امکان پذیر می کند. در جهان واقعی، معمولا مطلوب است که یک دنباله متغیرهای تصادفی وابسته را - که مقدار هر متغیر به عنصر (یا عناصر) قبلی در دنباله بستگی دارد - بررسی کنیم . در یک فرآیند مارکوف مقدار متغیر تصادفی جاری برای پیش بینی مقدار متغیرهای تصادفی آینده کافی است . به بیان دیگر، وقتی عنصر جاری را داشته باشیم، عناصر آینده از عناصر گذشته مستقل شرطی اند . فرض کنید $X = (X_1, \ldots, X_T)$ دنباله متغیرهای تصادفی باشد که مقادیری از فضای محدود $S = \{s_1, \ldots, s_N\}$ می گیرند .خواص مارکوف با دو رابطه زیر بیان می شوند:

$$P(X_{t+1} = s_k \mid X_1, X_2, \ldots, X_t) = P(X_{t+1} = s_k \mid X_t) \qquad (۱)$$

$$P(X_{t+1} = s_k \mid X_t) = P(X_2 = s_k \mid X_1) \qquad (۲)$$

خاصیت دوم را تغییر ناپذیری با زمان می نامیم. اگر دنباله X هر دو خاصیت مارکوف را داشته باشد، آن را یک **زنجیره مارکوف** می نامیم.

یک زنجیره مارکوف را می توان با بردار تصادفی وضعیت اولیه Π و ماتریس تصادفی انتقال A به طور کامل توصیف کرد:

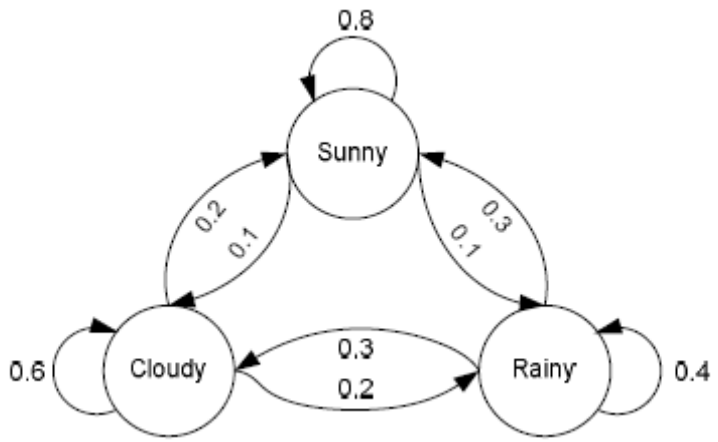
$$\pi_i = P(X_1 = s_i) \qquad (۳)$$

$$a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i) \qquad (۴)$$

که $\pi_i \geq 0, \forall i$ و $\sum_{i=1}^{N} \pi_i = 1$ و $a_{ij} \geq 0, \forall i, j$ و $\sum_{j=1}^{N} a_{ij} = 1, \forall i$.

برای روشن کردن این مطالب، مثالی درباره پیش بینی وضع هوا را در نظر بگیرید که در آن با استفاده از تاریخچه مشاهدات وضع هوا در گذشته می خواهیم هوای فردا را حدس بزنیم . برای

سادگی فرض می کنیم هوا سه حالت بیشتر نداشته باشد : خورشیدی، ابری و بارانی و وضعیت هوا در طول یک روز یکسان است؛ یعنی تغییر حالتی در وسط روز اتفاق نمی افتد . اگر فرض کنیم زنجیره وضعیت هوا در روزهای متوالی مارکوف است (که البته در جهان واقعی فرض درستی نیست)، آن گاه ماشین حالت محدود شکل ۱ با احتمالات انتقال (تغییر وضعیت) دلخواهی که روی پیکان ها داده شده است این زنجیره مارکوف را نشان می دهد . توجه کنید که مجموع احتمالات پیکان های خروجی در هر یک از سه حالت ۱ است. از شکل ۱ واضح است که مدل مارکوف را می توان به عنوان یک ماشین حالت محدود نامعین (تصادفی) به شمار آورد که در آن احتمالات روی پیکانهای تغییر وضعیت قرار گرفته اند.



فرض کنید s_3 = Rainy و s_2 = Cloudy ،s_1 = Sunny باشد و در اولین روز هوا خورشیدی باشد.
آن گاه:

$\Pi = (1.0, 0.0, 0.0)$

$$A = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.6 & 0.2 \\ 0.3 & 0.3 & 0.4 \end{bmatrix}$$

محاسبه احتمال یک دنباله از وضعیت ها $X_1, \ldots, X_K$ برای یک زنجیره مارکوف به سادگی با رابطه زیر انجام می شود:

$$
\begin{aligned}
P(X_1, \ldots, X_K) &= P(X_1)\, P(X_2 \mid X_1)\, P(X_3 \mid X_1, X_2) \ldots P(X_K \mid X_1, \ldots, X_{K-1}) \\
&= P(X_1)\, P(X_2 \mid X_1)\, P(X_3 \mid X_2) \ldots P(X_K \mid X_{K-1}) \\
&= \pi_{X_1} \prod_{t=1}^{K-1} a_{X_t X_{t+1}}
\end{aligned} \tag{۵}
$$

بنابراین در مثال بالا احتمال اینکه هوا در هفت روز آینده به ترتیب خورشیدی، خورشیدی، بارانی، بارانی، خورشیدی، ابری و خورشیدی باشد یا در واقع احتمال $O = s_1, s_1, s_3, s_3, s_1, s_2, s_1$ می تواند به شکل زیر محاسبه شود :

$$
\begin{aligned}
P(O \mid \text{Model}) &= \pi_{s_1} P(s_1 \mid s_1)\, P(s_1 \mid s_1)\, P(s_3 \mid s_1)\, P(s_3 \mid s_3)\, P(s_1 \mid s_3)\, P(s_2 \mid s_1)\, P(s_1 \mid s_2) \\
&= \pi_1\, a_{11}\, a_{11}\, a_{13}\, a_{33}\, a_{31}\, a_{12}\, a_{21} \\
&= 1.0\ (0.8)\ (0.8)\ (0.1)\ (0.4)\ (0.3)\ (0.1)\ (0.2) \\
&= 1.536 \times 10^{-4}
\end{aligned} \tag{۶}
$$

به طور کلی وقتی از مدل های مارکوف صحبت می کنیم، منظور ما مدل های مارکوف مرتبه اول است که در آن تاریخچه ای به طول ۱ (یعنی یک عنصر قبلی) برای پیش بینی رفتار آینده استفاده می شود. اما گاهی اوقات برای پیش بینی حالت های آینده تاریخچه بزرگتری لازم است.

در یک مدل مارکوف مرتبه n، برای پیش بینی حالت بعدی، از n حالت قبلی استفاده می شود. اما همواره می توان با تغییر شکل نمایش فضای حالت، هر مدل مارکوف مرتبه n را به مدل مارکوف مرتبه یک تبدیل کرد. بنابراین از نظر تئوری، فرض مارکوف مرتبه اول، محدود کننده نیست.

۳-۲- مدل های پنهان مارکوف

مدل های پنهان مارکوف (Hidden Markov Models) که به اختصار HMM نامیده می شوند، یکی از قوی ترین ابزارها برای پردازش سیگنال ها می باشند. انواع مختلف HMM علیرغم محدودیت هایی که دارند، هنوز پراستفاده ترین تکنیک در سیستم های مدرن بازشناسی گفتار و تشخیص متون هستند . مدل پنهان مارکوف، کل الگوی ورودی را به عنوان یک بردار ویژگی تکی مدل نمی کند، بلکه رابطه بین بخش های متوالی یک الگو را استخراج می کند، زیرا هر بخش نسبت به کل ورودی کوچکتر و بنابراین مدل سازی آن ساده تر است.

یک HMM را در واقع می توان یک ماشین حالت محدود (Finite State Machine) احتمالاتی به حساب آورد که هر حالت با یک تابع تصادفی مرتبط است . فرض می شود که در یک دوره گسسته از زمان t ، مدل در یک حالت است و با یک تابع تصادفی از آن حالت خروجی ای (مشاهده ای) تولید می کند . بر مبنای تابع احتمال انتقال حالت جاری، مدل مارکوف در زمان t+1

تغییر حالت می دهد . دنباله حالت هایی که مدل از آن می گذرد معمولا پنهان است، و تنها یک تابع احتمالاتی از آن آشکار است، که مشاهدات تولید شده به وسیله تابع تصادفی مربوط به حالت ها است، به همین دلیل در نام گذاری این مدل ها از صفت **پنهان** استفاده شده است. یک HMM را در واقع می توان یک فرآیند تصادفی به طور ناتمام (جزئی) مشاهده شده در نظر آورد. یک HMM با عناصر زیر توصیف می شود:

(۷) N: تعداد حالتهای مدل

(۸) $S = \{s_1, s_2, \ldots, s_N\}$: مجموعه حالتها

(۹) $\prod = \{\pi_i = P(s_i \text{ at } t = 1)\}$: احتمالات حالت اولیه

(۱۰) $A = \{a_{ij} = P(s_j \text{ at } t+1 \mid s_i \text{ at } t)\}$: احتمالات تغییر حالت

(۱۱) M: تعداد علائم قابل مشاهده (تولید شده)

(۱۲) $V = \{v_1, v_2, \ldots, v_M\}$: مجموعه علائم قابل مشاهده

(۱۳) $B = \{b_i(v_k) = P(v_k \text{ at } t \mid s_i \text{ at } t\}$: احتمالات تولید (انتشار) علائم قابل مشاهده

(۱۴) O_t: علامت مشاهده شده در زمان t

(۱۵) T: طول دنباله مشاهدات

(۱۶) $\lambda = (A, B, \prod)$: نماد خلاصه برای مدل پنهان مارکوف

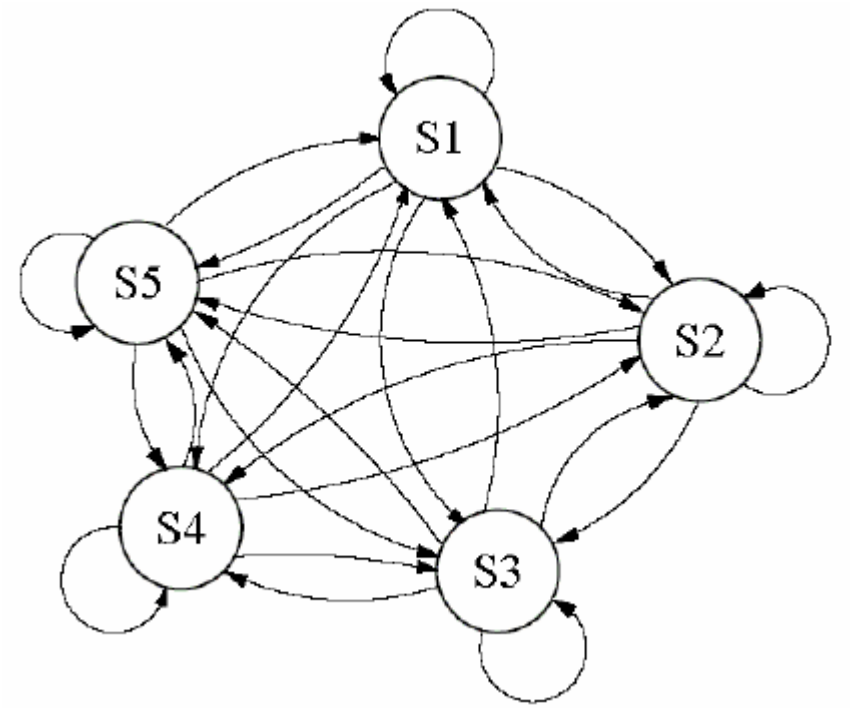
واضح است که بر احتمالات Π ، A و B سه قید وجود دارد:

$$\sum_{i=1}^{N} \pi_i = 1 \text{ ،} \quad \sum_{j=1}^{N} a_{ij} = 1, \ \forall i \quad \text{و} \quad \sum_{k=1}^{M} b_i(v_k) = 1, \ \forall i \text{ .}$$

ساختار ماتریس A توپولوژی HMM را تعیین می کند. اگر در مدل داشته باشیم:

$$a_{ij} \neq 0 \ \ \forall i, j$$

یعنی هر حالت مدل از هر حالت دیگر با یک گذر قابل رسیدن باشد، مدل **کاملا متصل** (ergodic) نامیده می شود. (شکل زیر)



یک توپولوژی دیگر که در کاربردهای بازشناسی گفتار و متن بسیار پراستفاده است، توپولوژی چپ به راست (LR) یا بکیس (Bakis) می باشد که در آن حالت های با شماره پایین تر، مشاهدات نخستین را تولید می کنند. ترتیب زمانی در HMM های چپ به راست با قرار دادن صفرهای ساختاری در مدل به شکل قیدهای

$a_{ij} = 0,\ i > j$ و $\prod = \{1, 0, ..., 0\}$

اعمال می شود، که یعنی مدل از اولین حالت (حالت با کمترین شماره و در واقع سمت چپ ترین حالت) شروع می کند و در هر حالت تغییر فقط می تواند به حالت های با شماره بالاتر صورت گیرد. به عنوان یک محدودیت دیگر، بیشتر اوقات در HMMهای چپ به راست، اندازه پرش های رو به جلو در هر حالت محدود می شود و به این ترتیب از تغییر حالت زیاد جلوگیری می شود، یعنی برای یک Δ ثابت $a_{ij} = 0, j > i + \Delta$ در نظر گرفته می شود.

مثال زیر به درک کاربرد مدل های پنهان مارکوف کمک می کند . فرض کنید یک نفر برای مدتی در اتاقی زندانی است و می خواهد از وضعیت هوای بیرون اطلاع داشته باشد . تنها اطلاعاتی که او از دنیای بیرون دارد این است که آیا کسی که هر روز برای او وعده غذایش را می آورد با چتر

وارد می شود یا نه؛ بنابراین برای مشاهده حمل چتر داریم V = {True, False}. برای سادگی فرض می شود که هوا تنها سه وضعیت دا رد: آفتابی، ابری و بارانی، و یک روز معادل یک بازه زمانی است، یعنی وضعیت هوا در طول روز تغییر نمی کند . فرض کنید احتمال حمل چتر در یک روز به شرط این که هوا آفتابی باشد 0.1 باشد، همچنین احتمال حمل چتر به شرط ابری بودن 0.3 و احتمال حمل چتر به شرط بارانی بودن 0.7 باشد . هدف این است که فرد از مشاهدات (حمل کردن یا نکردن چتر) درباره وضعیت هوای بیرون (که از او پنهان است) نتیجه ای بگیرد. فرض کنید w_i وضعیت هوا در روز i باشد و متغیر بولی u_i به این معنی باشد که در آن روز چتر مشاهده می شود یا خیر. با استفاده از قانون بیز داریم:

$$P(w_1,\ldots,w_n \mid u_1,\ldots u_n) = \frac{P(u_1,\ldots u_n \mid w_1,\ldots,w_n)P(w_1,\ldots w_n)}{P(u_1,\ldots,u_n)} \qquad (۱۷)$$

احتمال$P(w_1, \ldots,w_n)$ معادل مدل مارکوف مثال قبل است، و $P(u_1, \ldots,u_n)$ احتمال از پیش معلوم دیدن دنباله ای خاص از رخدادهای حمل کردن یا نکردن چتر است . اگر فرض شود که به ازای همه i ها، به شرط w_i، u_i از هر w_j و u_j به ازای هر $j \neq i$ مستقل باشد، آن گاه احتمال $P(u_1,\ldots,u_n \mid w_1,\ldots,w_n)$ این گونه محاسبه می شود:

$$\prod_{i=1}^{n} P(u_i \mid w_i)$$

در مثال پیش بینی وضع هوا (و خیلی از مسائل دیگر) می توان از احتمال پیشین $P(u_1, \ldots,u_n)$ صرف نظر کرد زیرا از وضع هوا مستقل است . بر مبنای فرض مارکوف مرتبه اول، معیار محتمل بودن (Likelihood) که با احتمال متناسب است، این گونه محاسبه می شود:

$$P(w_1,\ldots, w_n \mid u_1, \ldots, u_n) \propto$$

$$L(w_1,\ldots, w_n \mid u_1, \ldots, u_n) = P(u_1,\ldots,u_n \mid w_1,\ldots, w_n)\, P(w_1, \ldots, w_n)$$

$$= \prod_{i=1}^{n} P(u_i \mid w_i) \prod_{i=1}^{n} P(w_i \mid w_{i-1}) \qquad (۱۸)$$

فرض کنید روز حبس شدن فرد آفتابی بوده باشد، و روز بعد مسئول غذا با چتر وارد شده باشد. مطلوب است پیش بینی وضعیت هوای روز بعد.

در ابتدا با این فرض که روز بعد آفتابی بوده باشد معیار محتمل بودن را حساب می کنیم:

$$L(w_2 = Sunny \mid w_1 = Sunny, u_2 = True) = P(u_2 = True \mid w_2 = Sunny)\ .$$

$$P(w_2 = Sunny \mid w_1 = Sunny) = 0.1\ (0.8) = 0.08 \qquad (۱۹)$$

سپس با این فرض که روز بعد ابری بوده باشد معیار محتمل بودن را حساب می کنیم:

$$L(w_2 = Cloudy \mid w_1 = Sunny, u_2 = True) = P(u_2 = True \mid w_2 = Cloudy)\ .$$

$$P(w_2 = Cloudy \mid w_1 = Sunny) = 0.3\ (0.1) = 0.03 \qquad (۲۰)$$

و سرانجام با این فرض که روز بعد بارانی بوده باشد معیار محتمل بودن را حساب می کنیم:

$$L(w_2 = Rainy \mid w_1 = Sunny, u_2 = True) = P(u_2 = True \mid w_2 = Rainy)\ .$$

$$P(w_2 = Rainy \mid w_1 = Sunny) = 0.7\ (0.1) = 0.07 \qquad (۲۱)$$

بنابراین پر احتمال ترین حالت این است که روز بعد آفتابی بوده باشد.

۳-۳- سه مساله اساسی مدل های پنهان مارکوف

در کاربردهای HMM نیازمند حل حداقل یکی از سه مساله زیر هستیم :

مساله ۱. اگر مدل $\lambda = (A, B, \Pi)$ را داشته باشیم، چطور می توانیم احتمال $P(O \mid \lambda)$ یعنی احتمال وقوع دنباله مشاهده $O = O_1, O_2, ..., O_T$ به شرط مدل را به شکل موثری حساب کنیم؟

مساله ۲. اگر مدل λ و بردار مشاهده O را داشته باشیم، چطور دنباله حالت $S = s_1, s_2, ..., s_T$ را انتخاب کنیم که $P(O, S \mid \lambda)$ یعنی احتمال مشترک دنباله مشاهده $O = O_1, O_2, ..., O_T$ و دنباله حالت S به شرط مدل، ماکزیمم شود؟ به بیان دیگر، هدف این است که دنباله حالتی S پیدا کنیم که مشاهدات را به بهترین نحو توصیف کند.

مساله ۳. اگر بردار مشاهده O را داشته باشیم، چطور پارامترهای مدل $\lambda = (A, B, \Pi)$ را تنظیم کنیم که احتمال $P(O \mid \lambda)$ یا $P(O, S \mid \lambda)$ ماکزیمم شود. به بیان دیگر، هدف پیدا کردن (یا در واقع آموزش) مدلی است که داده های مشاهده شده را به بهترین نحو توصیف کند.

۳-۳-۱- حل مساله ۱

در این مساله هدف محاسبه احتمالی است که مدل λ دنباله مشاهده O را تولید کند . ساده ترین روش برای محاسبه $P(O \mid \lambda)$ پیدا کردن $P(O \mid S, \lambda)$ برای یک دنباله حالت ثابت S و ضرب آن در $P(S \mid \lambda)$ و جمع برای تمام دنباله های حالت ممکن به طول T است:

$$P(O \mid \lambda) = \sum_S P(O \mid S, \lambda).P(S \mid \lambda) \qquad (۲۲)$$

چون داریم:

$P(O \mid S,\lambda) = b_{s_1}(O_1)b_{s_2}(O_2)..b_{s_T}(O_T)$ و $P(S \mid \lambda) = \pi_{s_1} a_{s_1 s_2} a_{s_2 s_3} ... a_{s_{T-1} s_T}$

معادله ۲۲ می تواند این گونه نوشته شود:

$$P(O \mid \lambda) = \sum_S \pi_{s_1} b_{s_1}(O_1) a_{s_1 s_2} b_{s_2}(O_2) ... a_{s_{T-1} s_T} b_{s_T}(O_T) \qquad (۲۳)$$

محاسبه احتمال با استفاده از رابطه (۲۳) عملی نیست، زیرا N^T دنباله حالت وجود دارد و بنابراین $(2T-1)N^T$ عمل ضرب و $N^T - 1$ عمل جمع لازم است. بنابراین استفاده از یک روش عملی و کارا ضروری است . خوشبختانه دو روش برای این کار وجود دارد : الگوریتم رو به جلو (Forward) و الگوریتم رو به عقب (Backward) که شرح آن ها در زیر می آید.

الگوریتم رو به جلو برای هر حالت s متغیر رو به جلو $\alpha_t(s)$ را که به شکل زیر تعریف می شود محاسبه می کند:

$$\alpha_t(s) = P(O_1, O_2, ..., O_t, s_t = s \mid \lambda) \qquad (۲۴)$$

که احتمال دیدن جزئی دنباله مشاهده تا زمان t و بودن در حالت s به شرط مدل λ روال سه مرحله ای زیر $\alpha_t(s)$ را برای تمام حالات و زمانها حساب میکند:

۱- مقدار دهی نخستین:

$$\alpha_1(s) = \pi_s b_s(O_1),\ 1 \le s \le N \qquad (۲۵)$$

۲- استقرا:

$$\alpha_{t+1}(r) = [\sum_{s=1}^{N} \alpha_t(s) a_{sr}] b_r(O_{t+1}),\ 1 \le r \le N, 1 \le t \le T-1 \qquad (۲۶)$$

این احتمال رو به جلوی قرار گرفتن در حالت r در زمان t+1 بر مبنای احتمال مشترک متغیرهای رو به جلوی همه حالت ها در زمان t و احتمالات انتقال از هر یک از آنها به حالت r است . به این دلیل می توان احتمال را به این شکل محاسبه کرد که از هر یک از N حالت در زمان t می توان به طور مستقل (در زمان t+1) به حالت r (با احتمال a_{sr}) رسید.

۳- خاتمه:

(۲۷)
$$P(O \mid \lambda) = \sum_{s=1}^{N} \alpha_T(s)$$

محاسبه متغیرهای رو به جلو برای تمام حالتها و در تمام زمانها به N(N-1)(T-1)+(N-1) عمل جمع و N+N(N+1)(T-1) عمل ضرب احتیاج دارد که از مرتبه N^2T است، در مقایسه با مرتبه محاسبات در روش مستقیم که TN^T بود.

الگوریتم رو به عقب کاملا شبیه به الگوریتم رو به جلو است با این تفاوت که در این جا متغیر رو به عقب $\beta_t(s)$ که در زیر تعریف می شود برای هر حالت در جهت عکس محاسبه می شود:

(۲۸)
$$\beta_t(s) = P(O_{t+1}, O_{t+2}, \ldots, O_T \mid s_t = s, \lambda)$$

این احتمال دنباله مشاهده از t+1 تا T به شرط مدل λ و قرار گرفتن در حالت s در زما ن t است.

مشابه $\alpha_t(s)$، $\beta_t(s)$ هم می تواند برای تمام حالات و زمانها با یک الگوریتم سه مرحله ای محاسبه شود:

۱- مقدار دهی نخستین:

(۲۹)
$$\beta_T(s) = 1, \ 1 \le s \le N$$

۲- استقرا:

(۳۰)
$$\beta_t(s) = \sum_{r=1}^{N} a_{sr} b_r(O_{t+1}) \beta_{t+1}(r), \ 1 \le s \le N, t = T-1, T-2, \ldots, 1$$

۳- خاتمه:

(۳۱)
$$P(O \mid \lambda) = \sum_{s=1}^{N} \pi_s b_s(O_1) \beta_1(s)$$

محاسبه $P(O \mid \lambda)$ با استفاده از متغیرهای رو به عقب نیز شامل محاسباتی از مرتبه N^2T است.

۳-۳-۲- حل مساله ۲

در این مساله می خواهیم پراحتمال ترین دنباله حالت (یعنی قسمت پنهان مدل) متناظر با یک دنباله مشاهده را پیدا کنیم . الگوریتم معروف **ویتربی** (Viterbi, 1967) یک الگوریتم برنامه نویسی پویا (Dynamic Programming) برای پیدا کردن این دنباله (مسیر) بهینه است.

این الگوریتم بهترین دنباله حالت را در هر لحظه از زمان برای هر یک از N حالت نگه می دارد و در نهایت بهترین مسیر را برای هر یک از N حالت به عنوان حالتی که آخرین مشاهده در آن صورت گرفته است محاسبه می کند ، و از بین این مسیرها، مسیر دارای بیشترین احتمال را به عنوان مسیر بهینه کلی انتخاب می نماید.

الگوریتم چهار مرحله ای ویتربی همان استراتژی الگوریتم رو به جلو را دنبال می کند، اما با این تفاوت که عمل جمع با ماکزیمم (مینیمم) جایگزین می شود . انتخاب ماکزیمم یا مینیمم بستگی به این دارد که معیار بهینگی چه انتخاب شده باشد (بدیهی است که اگر معیار انتخاب شده احتمال باشد، باید ماکزیمم شود و اگر هزینه باشد باید مینیمم شود). برای دنباله مشاهده $O = O_1, O_2, \ldots$ و مدل λ، الگوریتم به شکل زیر بیان می شود:

۱- مقدار دهی نخستین:

$$\delta_1(s) = \pi_s b_s(O_1) \qquad (۳۲)$$

$$\psi_1(s) = 0,\ 1 \le s \le N \qquad (۳۳)$$

که $\delta t(s)$ نشان دهنده وزنهای انباشته شده است وقتی که مدل در زمان که t در حالت s قرار دارد، و $\psi t(s)$ نمایانگر حالتی است که در زمان t-1 پائین ترین هزینه (بالاترین احتمال) مربوط به انتقال حالت به s در زمانt را داشته است.

۲- استقرا:

$$\psi_t(s) = \arg\max_{1\le r\le N}[\delta_{t-1}(r)a_{rs}],\ 1 \le s \le N, 2 \le t \le T \qquad (۳۵)$$

$$\delta_t(s) = \max_{1\le r\le N}[\delta_{t-1}(r)a_{rs}]b_s(O_t) \qquad (۳۴)$$

۳- خاتمه:

$$P^* = \max_{1\le s\le N}[\delta_T(s)] \qquad (۳۶)$$

$$q_T^* = \arg\max_{1\le s\le N}[\delta_T(s)] \qquad (۳۷)$$

۴- برگشت به عقب برای پیدا کردن مسیر بهینه:

$$q_t^* = \psi_{t+1}(q_{t+1}^*),\ t = T-1, T-2, \ldots, 1 \qquad (۳۸)$$

و حالا $Q^* = \{q_1^*,...,q_T^*\}$ دنباله حالت بهینه است و P^* احتمال مشترک دنباله مشاهده O و دنباله حالت بهینه Q^* است.

شبیه به الگوریتم های رو به جلو و رو به عقب، پیچیدگی محاسباتی الگوریتم ویتربی از مرتبه N^2T می باشد.

توجه به این نکته ضروری است که پیاده سازی مستقیم الگوریتم بالا می تواند باعث به وجود آمدن مشکل پاریز (underflow) شود، زیرا احتمالاتی که حساب می شوند شامل ضرب اعداد کوچک است که به سرعت باعث می شود محدوده اعداد تولید شده از محدوده پیش بینی شده اعداد ممیز شناور کامپیوتر خارج شود . برای حل این مشکل، الگوریتم ویتربی طوری تغییر داده می شود تا به جای احتمال با لگاریتم احتمال کار کند . این تکنیک نه تنها مشکل پاریز را حل می کند، بلکه روند محاسبات را نیز سرعت می بخشد، زیرا عمل جمع خیلی سریع تر از ضرب است. باید توجه کرد که الگوریتم ویتربی یک الگوریتم زمان اجراست نه یک الگوریتم آموزش که معمولا به شکل آفلاین انجام می شود ، بنابراین پیاده سازی سریع این الگوریتم بسیار مطلوب و ضروری است. نسخه کارا و عملی الگوریتم ویتربی در زیر می آید:

۰- پیش پردازش

$$\tilde{\pi}_s = \log(\pi_s),\ 1 \le s \le N \tag{۳۹}$$

$$\tilde{a}_{rs} = \log(a_{rs}),\ 1 \le r, s \le N \tag{۴۰}$$

$$\tilde{b}_s(O_t) = \log(b_s(O_t)),\ 1 \le s \le N, 1 \le t \le T \tag{۴۱}$$

۱- مقدار دهی نخستین:

$$\tilde{\delta}_1(s) = \tilde{\pi}_s + \tilde{b}_s(O_1) \tag{۴۲}$$

$$\psi_1(s) = 0,\ 1 \le s \le N \tag{۴۳}$$

۲- استقرا:

$$\tilde{\delta}_t(s) = \max_{1 \le r \le N}[\tilde{\delta}_{t-1}(r) + \tilde{a}_{rs}] + \tilde{b}_s(O_t) \tag{۴۴}$$

$$\psi_t(s) = \arg\max_{1 \le r \le N}[\tilde{\delta}_{t-1}(r) + \tilde{a}_{rs}],\ 1 \le s \le N, 2 \le t \le T \tag{۴۵}$$

۳- خاتمه:

$$P^* = \max_{1\le s\le N}[\tilde{\delta}_T(s)] \quad (۴۶)$$

$$q_T^* = \arg\max_{1\le s\le N}[\tilde{\delta}_T(s)] \quad (۴۷)$$

۴- برگشت به عقب برای پیدا کردن مسیر بهینه:

$$q_t^* = \psi_{t+1}(q_{t+1}^*),\ t = T-1, T-2, \ldots, 1 \quad (۴۸)$$

و حالا $Q^* = \{q_1^*, \ldots, q_T^*\}$ دنباله حالت بهینه است و $\exp(P^*)$ احتمال مشترک دنباله مشاهده O و دنباله حالت بهینه Q^* است.

شاید متوجه شده باشید که الگوریتم ویتربی تنها شامل عملیات ضرب است که با لگاریتم گیری این اعمال ضرب به سادگی تبدیل به جمع می شوند . اما الگوریتم های رو به جلو و رو به عقب شامل عملیات جمع هستند .در این جا اگر لازم باشد مشکل پاریز اعداد ممیز شناور می تواند با همان لگاریتم گیری حل شود، تنها لازم است log(x+y) حساب شود که با استفاده از تکنیک زیر قابل انجام است (Manning and Schutze, 1999):

```
if y - x > log big
    return y;
  else if x - y > log big
    return x;
  else
      return min(x,y) + log(exp(x - min(x,y)) + exp(y - min(x,y)));
```

که big یک ثابت به حد کافی بزرگ مثلا 10^{30} است.

۳-۳-۳- حل مساله ۳

بر مبنای اینکه چه احتمالی برای ماکزیمم کردن انتخاب شود، دو روش کلی برای تخمین پارامترهای مدل (آموزش) وجود دارد : الگوریتم Segmental K-Means (Rabiner and Juang, 1990) که پارامترهای مدل را تنظیم می کند تا $P(O, Q^* \mid \lambda)$ ماکزیمم شود، که Q^* دنباله حالت بهینه متناظر با دنباله مشاهده O است و الگوریتم Baum-Welch (Rabiner, 1989) که پارامترهای مدل را تنظیم می کند تا $P(O \mid \lambda)$ به یک ماکزیمم برسد. در اینجا $P(O \mid \lambda)$ می تواند جمع احتمالات $P(O, S \mid \lambda)$ روی تمام دنباله حالت های ممکن S تعبیر شود، یعنی این الگوریتم روی دنباله حالت خاصی متمرکز نمی شود . معمولا الگوریتم segmental k-mean برای آموزش ترجیح داده می شود زیرا در مقایسه با الگوریتم

Baum-Welch به محاسبات خیلی کمتری احتیاج دارد، و همچنین برای خیلی از کاربردهای مدلسازی و کدگشایی، معیار $P(O,Q^* \mid \lambda)$ کاملا طبیعی به نظر می رسد. در دو قسمت زیر شرح الگوریتم ها آورده می شود.

۳-۳-۳-۱- الگوریتم Segmental K-Means

مانند هر الگوریتم آموزش، الگوریتم Segmental K-Means نیز به تعدادی دنباله مشاهده (آموزشی) احتیاج دارد. فرض کنید به تعداد w از چنین دنباله هایی موجود باشد . هر دنباله $O = O_1,...,O_{T_i}$ شامل T_i بردار مشاهده است، بنابراین در مجموع $\sum_{i=1}^{w} T_i$ بردار مشاهده وجود دارد. اگر تنها یک دنباله طولانی موجود باشد، می توان آن را به تعداد اختیاری دنباله های کوچک تقسیم کرد و از آن ها برای آموزش استفاده نمود. فرض می شود هر علامت (Symbol) مشاهده یک بردار با اندازه (بعد) یک با بیشتر باشد و بدیهی است که بردارهای مشاهده باید دارای بعد یکسان باشند . همچنین در مدل های پنهان مارکوف گسسته که تا این جا مورد بحث قرار گرفته اند لازم است که این علامت ها متعلق به یک مجموعه محدود باشند. الگوریتم شامل مراحل زیر است:

۱- N بردار آموزشی $C_1,...,C_N$ را به طور تصادفی انتخاب کنید و هر یک از بردارهای آموزشی باقیمانده را به یکی از این N بردار که کمترین فاصله (مثلا اقلیدسی) را تا آن دارد منسوب کنید . بنابراین N دسته (خوشه) شکل می گیرد که هر کدام یک حالت (با شماره ای بین ۱ تا N) نامیده می شود. نماد $O_t \in s$ یعنی t امین علامت از یک دنباله مشاهده به حالت (دسته) s منسوب شده است . انتخاب اولیه دسته ها بر HMM نهایی تاثیری نمی گذارد، اما می تواند تعداد تکرارهای لازم برای آموزش مدل را تعیین کند . برای اینکه انتخاب اولیه دسته ها تا حد امکان توزیع پراکنده ای داشته باشد، یک استراتژی خوب برای وقتی که $w \geq N$ این است که C_1 اولین بردار مشاهده اولین دنباله انتخاب شود، C_2 دومین بردار مشاهده دومین دنباله انتخاب شود و ... (Dugad and Desai, 1996). این مرحله مقداردهی نخستین مناسبی برای کل روال آموزش آماده می کند.

۲- احتمالات نخستین و انتقال را به شکل زیر محاسبه کنید:

$$\hat{\pi}_s = \frac{\text{number of occurrences of } \{O_1 \in s\}}{\text{total number of occurrences of } O_1}, \ 1 \leq s \leq N \tag{۴۹}$$

$$\hat{a}_{rs} = \frac{\text{number of occurrences of } \{O_t \in r \text{ and } O_{t+1} \in s\}}{\text{total number of occurrences of } \{O_t \in r\}}, \ 1 \leq r,s \leq N, \ 1 \leq t \leq T_i - 1 \tag{۵۰}$$

۳- ماتریس میانگین و کوواریانس را به شکل زیر محاسبه کنید:

$$\hat{\mu}_s = \frac{1}{N_s}\sum_{O_t \in s} O_t,\ 1 \le s \le N \qquad (۵۱)$$

$$\hat{V}_s = \frac{1}{N_s}\sum_{O_t \in s} (O_t - \hat{\mu}_s)^T (O_t - \hat{\mu}_s),\ 1 \le s \le N \qquad (۵۲)$$

۴- توزیع احتمال را برای هر بردار مشاهده و برای هر حالت محاسبه کنید:

$$\hat{b}_s(O_t) = \frac{1}{(2\pi)^{\frac{D}{2}} \mid \hat{V}_s \mid^{\frac{1}{2}}} \exp(-\frac{1}{2}(O_t - \hat{\mu}_s)\hat{V}_s(O_t - \hat{\mu}_s)^T) \qquad (۵۳)$$

ثابت شده است که برای دسته وسیعی از توابع چگالی احتمال از جمله گاوسی، الگوریتم همگرا می شود .در این جا تابع گاوسی به طور اختیاری انتخاب شده است.

۵- با احتمالات جدید با استفاده از الگوریتم ویتربی برای هر دنباله آموزشی، دنباله حالت بهینه Q^* را پیدا کنید . اگر انتساب حالت قبلی یک بردار مشاهده متفاوت از حالت بهینه تخمین زده شده متناظر باشد آن را به حالت جدید منسوب کنید؛ یعنی اگر $q_t^* = s$ آن گاه O_t را به S منسوب کنید.

۶- اگر در مرحله ۵ هر کدام از بردارهای مشاهده به حالتی جدید (یعنی حالتی مخالف با حالت انتساب یافته قبلی اش) منسوب شد، آن گاه مراحل ۲ تا ۶ باید تکرار شوند و در غیر این صورت الگوریتم متوقف می شود.

۳-۳-۳-۱- الگوریتم Baum-Welch

این الگوریتم جزء دسته الگوریتم های موسوم به ماکزیمم کردن توقع (EM) می باشد. الگوریتم EM روش بسیار پراستفاده ای برای یادگیری در مسائلی است که در آن ها متغیرهای مشاهده نشده (پنهان) وجود دارد. یک الگوریتم EM با تخمین زدن مکرر مقدار مورد انتظار (Expected Value) هر متغیر پنهان با استفاده از فرض های جاری و سپس محاسبه مجدد متحمل ترین فرض ها (Maximum Likelihood Hypothesis) با استفاده از مقادیر مورد انتظار متغیرها، محتمل ترین فرضیه ها را جستجو می کند . به بیان دیگر، در مرحله اول، فرض های جاری برای تخمین مقدار متغیرهای پنهان استفاده می شوند و در مرحله دوم مقادیر این متغیرها برای بهبود دادن فرضیه ها استفاده می شود. می توان اثبات کرد که چنین روندی نهایتا فرضیه هایی دارای بیشترین احتمال محلی را پیدا می کند، یعنی ممکن است الگوریتم EM همانند بسیاری از الگوریتم های تکراری ممکن است نتواند جواب بهینه سراسری را پیدا کند، اما همواره یک بهینه محلی را پیدا می نماید (Mitchell, 1997).

تخمین اولیه فرض ها (پارامترها)، یعنی ساخت HMM اولیه، می تواند با هر روشی انجام شود، اما یک تخمین اولیه معقول می تواند با استفاده از چهار مرحله اول الگوریتم Segmental K-Means به دست آید.

قبل از ارائه فرمول های اصلی باید چند مفهوم و نماد را معرفی کنیم. $\gamma_t(s) = P(s_t = s \mid O, \lambda)$ را در نظر بگیرید، که احتمال قرار داشتن در حالت s در زمان t به شرط مدل λ و دنباله مشاهده O است. با استفاده از قانون بیس داریم:

$$\gamma_s(t) = \frac{P(s_t = s, O \mid \lambda)}{P(O \mid \lambda)} = \frac{\alpha_t(s)\beta_t(s)}{P(O \mid \lambda)},\ 1 \le s \le N \qquad (۵۴)$$

که $\alpha_t(s)$ و $\beta_t(s)$ متغیرهای رو به جلو و رو به عقب هستند.

همچنین $\xi_t(r,s) = P(s_t = r, s_{t+1} = s \mid O, \lambda)$ را تعریف می کنیم که احتمال قرار داشتن در حالت r در زمان t و انتقال به حالت s در زمان t+1 به شرط مدل λ و دنباله مشاهده O است. با استفاده از قانون بیز و خاصیت کازال بودن زنجیره مارکوف (رابطه (۱)) می توان نشان داد:

$$\xi_t(r,s) = \frac{\alpha_t(r) a_{rs} b_s(O_{t+1}) \beta_{t+1}(s)}{P(O \mid \lambda)},\ 1 \le r, s \le N \qquad (۵۵)$$

اگر $\gamma_t(s)$ از t=1 تا T جمع زده شود، تعداد دفعات قابل انتظاری که حالت s دیده شده است به دست می آید و اگر تنها تا T-1 جمع زده شود، تعداد دفعات قابل انتظاری که از حالت s انتقالی انجام شده است به دست می آید. به طور مشابه، اگر $\xi_t(r,s)$ از t=1 تا T جمع زده شود، تعداد دفعات قابل انتظاری که انتقالی از حالت r به حالت s انجام گرفته است به دست می آید:

$$\sum_{t=1}^{T-1} \gamma_t(s) = \text{expected number of transitions from state } s,\ 1 \le s \le N \qquad (۵۶)$$

$$\sum_{t=1}^{T-1} \xi_t(r,s) = \text{expected number of transitions from state } r \text{ to state } s,\ 1 \le r,s \le N \qquad (۵۷)$$

رابطه بین $\gamma_t(s)$ و $\xi_t(r,s)$ می تواند با جمع کردن $\xi_t(r,s)$ روی s به دست آید:

$$\gamma_t(r) = \sum_{s=1}^{N} \xi_t(r,s),\ 1 \le r \le N \qquad (۵۸)$$

و حالا فرمول های تخمین مجدد Buam-Welch به شکل زیر تعریف می شوند:

$$\hat{\pi}_s = \gamma_1(s),\ 1 \le s \le N \tag{۵۹}$$

$$\hat{a}_{rs} = \sum_{t=1}^{T-1} \xi_t(r,s) \Big/ \sum_{t=1}^{T-1} \gamma_t(r),\ 1 \le r,s \le N \tag{۶۰}$$

$$\hat{b}_s(v_k) = \sum_{t=1, O_t=v_k}^{T} \gamma_t(s) \Big/ \sum_{t=1}^{T} \gamma_t(s),\ 1 \le s \le N \tag{۶۱}$$

فرمول تخمین مجدد برای Π_s به سادگی برابر است با احتمال قرار داشتن در حالت S در زمان ۱. فرمول تخمین مجدد برای a_{rs} برابر است با نسبت تعداد دفعات قابل انتظار انتقال حالت از r به S، به تعداد دفعات قابل انتظاری که از r انتقال حالتی انجام شده است. و سرانجام، فرمول تخمین مجدد برای $b_s(v_k)$ برابر است با نسبت تعداد دفعات قابل انتظار قرار داشتن در حالت S و مشاهده علامت v_k، به تعداد دفعات قابل انتظار قرار داشتن در حالت S.

۴- کاربردهای مدل در شناسائی آماری الگو

مدل مخفی ماکوف کاربردهای متعددی دارد که می توان نمونه های زیر را نام برد:

- شناسایی گفتار
- تشخیص حرکات بدن
- تشخیص کارکتر نوری (OCR)
- ترجمه ماشینی
- رمزشکنی (cryptanalysis)
- بیوانفورماتیک

در ادامه کاربرد مدل در شناسائی گفتار، به اختصار بررسی می شود.

۴-۱- کاربرد مدل در شناسائی گفتار

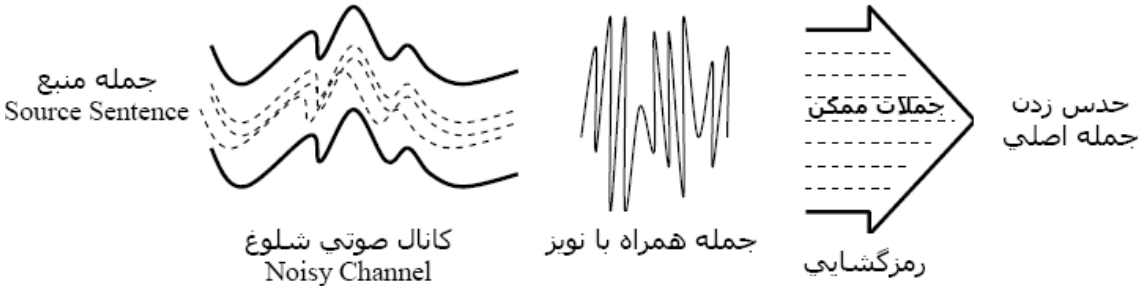
برقراری ارتباط گفتاری با کامپیوترها به جای استفاده از صفحه کلید و ماوس یکی از زمینه‌های تحقیقاتی مهم چند دهه‌ی اخیر بوده است و شرکت‌های بزرگی چون IBM، ALIT، Philips و Microsoft سالانه هزینه‌های هنگفتی را برای این منظور پرداخت کرده و می‌کنند.

فناوری تشخیص گفتار به رایانه‌ای که توانایی دریافت صدا را دارد (برای مثال به یک میکروفن مجهز است) این قابلیت را می‌دهد که صحبت کاربر را متوجه شود. این فناوری در تبدیل گفتار به متن و یا به عنوان جایگزینی برای صفحه کلید یا ماوس برای وارد کردن دستورات مورد استفاده قرار می گیرد. سیستم های تشخیص دهنده گفتار انواع مختلفی دارند، بعضی قادرند گفتار پیوسته را

شناسایی نمایند، بعضی دیگر فقط می توانند گفتار گسسته (که بین کلمات سکوت وجود دارد) را تشخیص دهند. همچنین سیستم ها قادرند کلمات بیان شده توسط افراد مختلف و یا فقط توسط یک گوینده تشخیص دهند. بهر حال ایده آل ترین سیستم آن است که بتواند گفتار پیوسته غیر وابسته به گوینده را در محیط نویزی شناسایی نماید. این سیستم ها با بکار گیری روش های مختلف طبقه بندی و شناسایی الگو قادر به تشخیص کلمات هستند که البته برای افزایش دقت در شناسایی از یک فرهنگ لغات نیز در انتهای سیستم استفاده می شود. روشهایی مانند مدل مخفی مارکوف و شبکه های عصبی در بسیاری از سیستم های تشخیص گفتار مورد استفاده قرار می گیرند و در بخش‌های انتهایی سیستم از هوش مصنوعی کمک گرفته می شود. امروزه با داشتن میکروفن و کارت صوتی در کامپیوتر و بکار گیری نرم افزار تشخیص گفتار میتوان دستورات یا کلمات را به صورت صوتی به کامپیوتر وارد کرد. حتی در بعضی از گوشی های تلفن همراه از این سیستم ها جهت دریافت دستورات بصورت صوتی استفاده می شود.

۴-۱-۱- مدل کانال نویزی

سیستم های تشخیص صوت با ورودی صوتی شامل جمله همراه با نویز برخورد می کنند. برای کدگشایی این جملات ما همه جملات ممکن را در نظر می گیریم و برای هر کدام احتمال اینکه بتواند جمله اصلی را تولید کند محاسبه می کنیم، سپس جمله با بیشترین احتمال را انتخاب می کنیم.



تشخیص دهنده گفتار به وسیله جستجو از میان فضای بسیار بزرگی از جملات منبع کار می کند و جمله ای را که بیشترین احتمال تولید جمله نویزدار را دارد، انتخاب می نماید. برای انجام این کار ، باید قالب هایی که احتمال جملات را بیان می کنند، به عنوان رشته ای از لغات مجسم شوند (N-grams) و قالب هایی که احتمال مجسم کردن لغات را بیان می کنند، به عنوان یک رشته مشخص از صداها (HMMs).

عملی کردن مدل کانال نویزی که در شکل قبل آن را بیان کردیم نیاز به حل دو مسئله دارد . اول، انتخاب جمله ای که بیشترین شباهت را با ورودی همراه با نویزی داشته باشد . به دلیل اینکه گفتار

بسیار بی ثبات و تغییر پذیر است، ممکن است یک جمله ورودی صوتی واقعا با هیچ مدلی که ما برای آن داریم مطابقت نداشته باشد. همان گونه که در بخش قبل پیشنهاد کردیم، ما از احتمال بعنوان میزان اندازه گیری استفاده خواهیم کرد و نشان خواهیم داد که چگونه با ترکیب احتمالات مختلف برآورد کاملی برای احتمال یک توالی از جملات کاندید می توان بدست آورد.

دوم ، از آن جایی که مجموعه کل جملات یک زبان بسیار بزرگ است، ما به یک الگوریتم کارا احتیاج داریم که نیاز به جستجوی از میان همه جملات ممکن ندارد . یک رهیافت کارا در این مورد رمزگشائی Viterbi است.

هدف معماری کانال نویزی وابسته به احتمال برای تشخیص گفتار می تواند این گونه خلاصه شود: شبیه ترین جمله از میان همه جملات زبان L که از ورودی صوتی O گرفته شده چه چیزی است؟ می توان با ورودی صوتی O به عنوان توالی منحصر به فردی از مشاهدات رفتار کرد. (به عنوان مثال به وسیله قطعه قطعه کردن ورودی در هر ده میلی ثانیه و نمایش هر قطعه به وسیله مقداری از انرژی یا فرکانس آن قطعه):

$$O = o_1,...,o_n$$

به طور مشابه با جمله ای که بطور ساده از رشته ای از لغات تشکیل شده، رفتار خواهیم کرد.

$$W = w_1,...w_n$$

با توجه به موارد، ما به دنبال جمله ای هستیم که احتمال زیر را حداکثر نماید:

$$\hat{W} = \arg\max_{W \in L} P(W \mid O) = \arg\max_{W \in L} \frac{P(O \mid W).P(w)}{P(O)} = \arg\max_{W \in L} P(O \mid W).P(W)$$

P(W) که یک احتمال پیشین است، مدل زبانی و P(O|W) مدل صدائی نامیده می شود. مقدار P(W) را می توان به وسیله مدل های زبانی N-gram به دست آورد. P(O|W) براساس روش های گفته شده مدل مخفی مارکوف قابل محاسبه است.

۵- نتیجه گیری

مدل مخفی مارکوف شامل تعدادی ند به عنوان وضعیت های پنهان است که به وسیله لینک هائی که احتمالات شرطی عبور میان وضعیت ها را نشان می دهند، به هم متصلند. هر وضعیت پنهان، مجموعه ای از احتمالات ایجاد وضعیت های قابل مشاهده سیستم را دارد.

مدل مخفی مارکوف می تواند در مدل کردن توالی ها، به ویژه توالی های وابسته به متن، مانند فونم در گفتار، مفید می باشد.

احتمالات انتقال می تواند به طور تکراری از روی توالی های نمونه به وسیله الگوریتم forward-backward و یا Buam-Welch یاد گرفته شود.

طبقه بندی، به وسیله یافتن مدلی که با بیشترین احتمال یک توالی مشاهده شده را تولید می کند، انجام می شود.

۶- مراجع

- L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”, Pro. Of IEEE, Vol 77, February 1989

- R. O. Duda, P. E. Hart, D. G. Stork, “Pattern Classification”, 2th edition.

- Tutorial from University of Leeds:
http://www.comp.leeds.ac.uk/roger/HiddenMarkovModels/html_dev/main.html

- Wikipedia, the online encyclopedia:
http://en.wikipedia.org