ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E SISTEMAS DIGITAIS

PCS2428 – Inteligência Artificial

Reconhecimento de Fala

Professora Anna Helena Reali Costa

Grupo 9

Fábio Belotti Colombo	5178770
Fabio Sussumu Komori	5173865
Tiago Bello Torres	5175526

Conteúdo

Introdução ao Reconhecimento da Fala

O reconhecimento da fala automático consiste em gerar, a partir de um sinal acústico, a representação correspondente que um computador seja capaz de entender. A complexidade do reconhecedor pode variar. Quando os primeiros reconhecedores foram desenvolvidos, eram capazes de reconhecer apenas algumas vogais, consoantes ou dígitos.

Atualmente existem reconhecedores de fala contínuos, que não requerem que o usuário faça uma pausa entre uma palavra ou outra. Também existem reconhecedores mais simples; capazes de reconhecer números ou algumas poucas palavras.

O interesse em máquinas capazes de "falar" se iniciou muito cedo. Isso se deve em parte, ao fato da fala ser nosso principal meio de comunicação. Em 1881, Alexander Graham Bell fundou, junto com um primo, uma empresa para fornecer máquinas capazes de gravar e reproduzir sons em ambientes de trabalho. A idéia na época era permitir que gerentes pudessem ditar textos para uma máquina e que suas secretárias pudessem, posteriormente, digitar os textos ditados.

O interesse em reconhecer a fala teve seus primeiros resultados na década de 1950. As primeiras tentativas eram focadas na acústica da voz e no reconhecimento da distribuição da potência do som pelo espectro do som. Em 1952 Davis, Biddulph, and Balashek do Bell Laboratories desenvolveram um circuito capaz de reconhecer os dígitos de um a nove e o digito zero ("oh"). Fry and Denes, da University College na Inglaterra foram os primeiros a utilizar a análise estatística para o reconhecimento da fala. Eles utilizaram informações estatísticas sofre as seqüências de fonemas permitidos na língua inglesa para melhorar o reconhecimento de fonemas dentro de palavras.

As décadas de 1960 e 1970 marcaram o aparecimento do Modelo Oculto de Markov em reconhecedores de fala. Devido às suas características e os bons resultados nesta aplicação, o seu uso se expandiu na década de 1980 e até hoje é o modelo mais utilizado pelos reconhecedores da fala modernos. A seguir, explicaremos como o Modelo Oculto de Markov é utilizado em reconhecedores de fala modernos.

Modelo Oculto de Markov

O Modelo Oculto de Markov (MOM) é uma variação do modelo de Markov clássico. O modelo de Markov pode ser usado para descrever processos de Markov. De maneira simples, um processo de Markov é um processo que não exibe memória. Isto é, o estado no tempo T depende apenas do estado no tempo T-1. O estado do sistema nos instantes t < T-1 não influencia o sistema no tempo T.

Imagine um sistema que pode ser descrito por um número N de estados S = {S_1, S_2, S_3, ..., S_N}. O modelo muda de estado em intervalos de tempo discretos e fixos. Os instantes de tempo em que ocorrem as mudanças de estado podem ser denominados t = 1, 2, 3, ..., e o tempo t do estado atual é definido como qt. Caso esse sistema seja um processo de Markov, temos:

$$P[q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_k, ...] = P[q_t = S_j | q_{t-1} = S_i]$$

Considerando que as transições de estado são independentes do tempo, temos que:

$$a_{ij} = P[q_t = S_j | q_{t-1} = S_i]$$

onde $1 \leq i, j \leq N$. Como os valores a_{ij} são probabilidades, também devemos ter:

$$a_{ij} \geq 0$$

$$\sum_{j=1}^{N} a_{ij} = 1$$

O modelo de Markov descrito pelos valores acima pode ser considerado um modelo de Markov observável. Sabemos a todo instante o estado atual do sistema. Abaixo temos um exemplo concreto de tal sistema. Imagine um sistema simples para modelar uma rodada de dado. Os seis possíveis estados são os possíveis valores do dado: 1, 2, 3, 4, 5 e 6. A taxa de transição entre os estados também é fácil de definir: 1/6 para todas as transições de estado. Abaixo temos uma imagem do modelo.

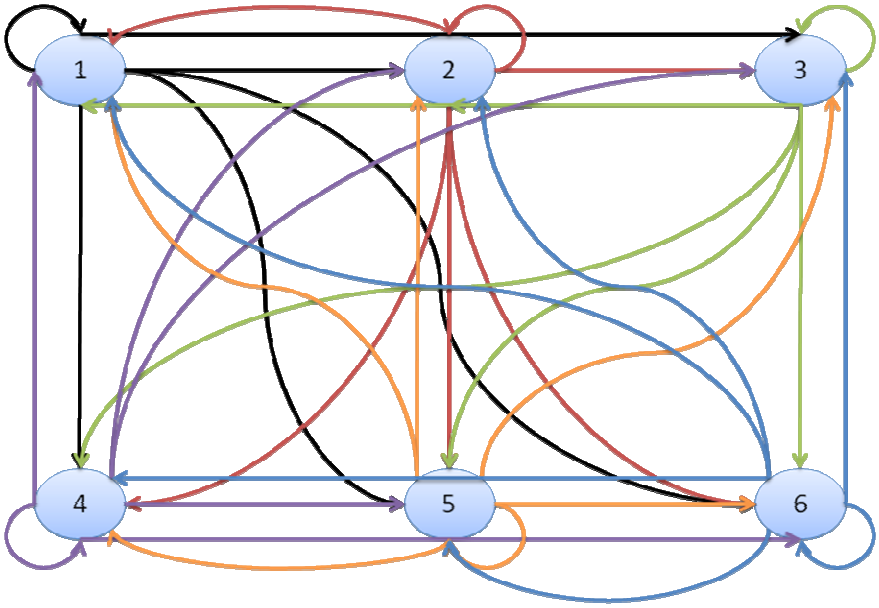

Figura 1: Exemplo de uma cadeia de Markov para modelar o rolar de um dado.

Em muitos sistemas que encontramos habitualmente, não temos como saber o estado atual do sistema. Esses sistemas apresentam apenas evidências do estado atual. O MOM busca modelar sistemas esses sistemas não observáveis.

Como exemplo simples, podemos imaginar uma pessoa que remove bolas de urnas atrás de uma cortina. A cada rodada, essa pessoa escolhe uma urna aleatoriamente e remove uma bola colorida da urna, também aleatoriamente. A pessoa que remove as bolas diz a cor da bola e a coloca novamente na urna de onde a tirou. Não podemos enxergar o que ocorre atrás da cortina, mas gostaríamos de poder modelar esse sistema para ter uma boa estimativa do que está ocorrendo. Podemos descrever esse sistema através de MOM.

Por simplicidade, vamos considerar que existem duas urnas (Urna 1 e Urna 2) com bolas de duas cores diferentes (pretas e brancas). Vamos supor que a probabilidade de uma bola branca ser retirada da Urna 1 é de 0,8 e a de uma bola preta ser retirada é de 0,2. Na Urna 2, a probabilidade de se retira uma bola branca é de 0,6 e de se retirar uma bola preta é de 0,4. A probabilidade de se escolher as urnas é a mesma.

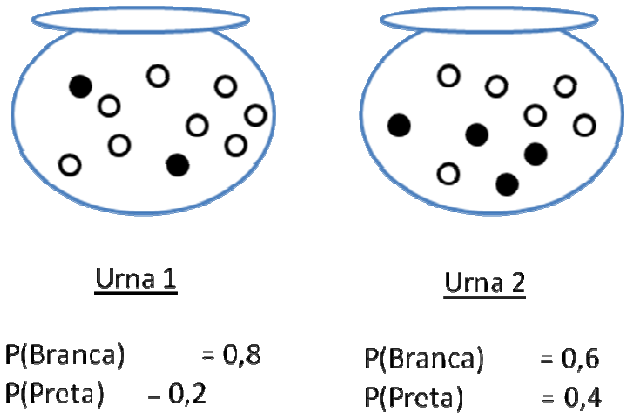

Figura 2: Exemplo das urnas.

Esse sistema pode ser modelo por um MOM com dois estados. Cada estado representa a urna de qual a bola foi tirada. Assim se a bola for tirada da Urna 1, significa que foi realizada uma transição para o estado 1. Se uma bola foi retirada da Urna 2, uma transição é feita para o estado 2. É importante lembrar que não sabemos de qual urna a bola foi retirada, sabemos apenas a cor da bola retirada. A taxa de transição entre os estados também é conhecida. Dado que a escolha das urnas é eqüiprovável, a taxa de transição do estado um para o estado 2 é de 0,5, e a taxa de transição do estado 2 para o estado 1 também é de 0,5. Como não conhecemos o estado do sistema, também devemos definir uma probabilidade para o estado inicial. Novamente, como a seleção das urnas é eqüiprovável, podemos definir a probabilidade do estado inicial ser o estado 1, que é igual ao do estado 2, como 0,5.

Resta definir mais uma probabilidade. Para cada estado (urna), temos uma probabilidade de uma bola branca ou preta ser retirada. Essa probabilidade é chamada de distribuição de probabilidade dos símbolos de observação. No caso acima, temos dois símbolos de observação: bolas brancas e bolas pretas. No estado 1, a probabilidade de se observar uma bola preta é de 0,2 e de se observar uma bola branca de 0,8. Já no estado 2, a probabilidade de se observar uma bola branca é de 0,6 e a de se observar uma bola preta, de 0,4. Na figura abaixo temos um diagrama do MOM descrito acima.

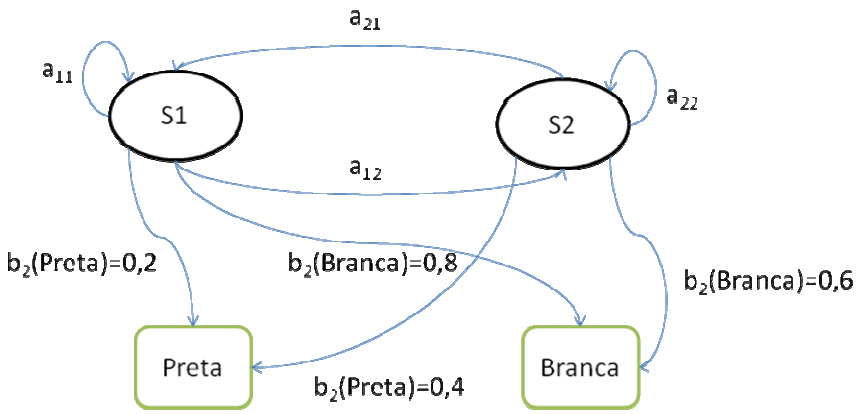

Figura 3: Diagrama de um MOM para as urnas.

Com o diagrama acima, podemos calcular, por exemplo, a probabilidade de ocorrer a observação O = preta, preta, branca, branca, branca, preta, preta. Talvez até mais interessante, podemos tentar definir a seqüência de estados que tem a maior probabilidade de gerar a observação acima. Poderíamos também tentar encontrar os parâmetros melhores para o nosso modelo de tal maneira que a probabilidade da observação acima ocorrer seja máxima. Esses três problemas estão relacionados com os três problemas básicos para um MOM.

No contexto do reconhecimento da fala, a segunda resposta é a que nos interessa. Em termos gerais, os estados são como as letras das palavras. As transições entre os estados refletem a probabilidade de se encontrar uma letra após a outra em uma palavra da língua sendo reconhecida, i.e. a taxa de transição do estado que representa "e" para o estado que representa "t" está ligado à probabilidade de se encontrar palavras onde existe uma letra e seguida de t, como "deter". A taxa de geração de símbolos está ligada à probabilidade de uma letra gerar um determinado som. Dependendo da palavra ou até da posição da letra em uma palavra, o seu som muda. Essa probabilidade reflete a probabilidade da letra ter aquele som na língua.

Para definir formalmente um MOM, precisamos dos seguintes valores:

- Um número N de estados representados aqui por S = {S_1, S_2, S_3, ..., S_N}.
- Um número M de símbolos distintos de observação, que serão representados por V = {v_1, v_2, v_3, ..., v_M}.
- As probabilidades de transição de estado A = {a_{ij}} onde:
  $a_{ij} = P[q_t = S_j | q_{t-1} = S_i]$, $1 \le i, j \le N$.
- A distribuição de probabilidade dos símbolos de observação para cada estado. A probabilidade discreta do símbolo k ser observado no estado j no tempo t é dada por:
  $b_j(k) = P[v_k \mid q_t = S_j]$, $\begin{matrix} 1 \le j \le N \\ 1 \le k \le M \end{matrix}$.
- A distribuição de estados iniciais π = {π_i} onde:
  $\pi = p[q_1 = S_i]$, $1 \le i \le N$.

Daremos agora também uma descrição mais formal dos problemas básicos para um MOM. Para tanto, considere um MOM descrito pelos valores listados acima. Os problemas básicos para um MOM são:

1. Dado uma seqüência de observações O = $O_1O_2O_3...O_T$, como calcular de maneira eficiente a probabilidade de se observar O dado o modelo, i.e. .
2. Dado uma seqüência de observações O = $O_1O_2O_3...O_T$, como escolher a seqüência de estados Q = $q_1q_2q_3...q_T$ que seja ótima de acordo com alguma definição sensata, i.e., qual a melhor explicação para a observação?
3. Como ajustar os parâmetros de $Modelo$ de maneira a maximizar $P(O \mid Modelo)$. Isto é, como escolher os parâmetros de $Modelo$ de maneira que o modelo seja a melhor representação possível do que foi observado.

A solução destes três problemas ajuda no desenvolvimento e operação de um reconhecedor de fala. A solução do primeiro permite selecionar o melhor modelo entre diversos disponíveis para uma observação. A solução do terceiro problema permite aperfeiçoar os parâmetros do modelo para certa observação. As observações utilizadas nesses casos são denominadas os dados de treinamento, por serem utilizadas para treinar, ou aperfeiçoar, o reconhecedor. Esses problemas podem ser solucionados pelo algoritmo *forward-backward*.

O segundo problema necessita, antes de uma solução, de uma definição para a seqüência de estados ótima. Quando definimos a seqüência de estados ótima como sendo a seqüência que maximiza $P(Q \mid O, Modelo)$. Isto é equivalente a maximizar $P(Q, O \mid Modelo)$. O algoritmo de Viterbi encontra essa seqüência e será descrito adiante.

Existem algumas modificações que podem ser feitas ao MOM para facilitar o seu uso em reconhecedores de fala. Entre elas podemos citar a modificação para aceitar sinais contínuos como entrada.

Apesar de ser possível quantizar esses sinais através de *codebooks* e continuar a utilizar distribuições de probabilidade discretas, é mais interessante utilizar uma distribuição de probabilidade contínua. O cálculo da probabilidade de observação de um sinal representado pelo vetor O passa a ser feito seguindo a formula abaixo.

$$b_j(O) = \sum_{m=1}^{M} \left(c_{jm} \cdot N\left(O, \mu_{jm}, U_{jm}\right)\right)$$

O vetor O é uma variável aleatória contínua composta por M coeficientes. Cada coeficiente é em si uma variável aleatória contínua com uma distribuição normal definida pela média μj_m e covariância U_{jm}. Essa abordagem produz resultados melhores, mas torna o cálculo mais complexo. Pois temos agora definições de várias distribuições gaussianas para cada estado, ao invés das probabilidades discretas.

Em alguns modelos também pode ser útil associar as evidências a uma transição e não a um estado. Nesses casos, uma transição nula pode simplificar os modelos. Como mostrado

abaixo. O reconhecimento da palavra “Pouco”, que possui variações na pronúncia, é simplificado, pois pode ser representado em um modelo único que permite a “remoção” do som correspondente à letra “u” do sinal.

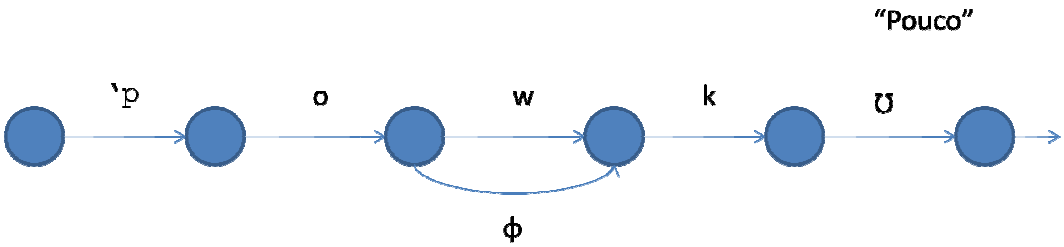

Figura 4: Reconhecimento da palavra "pouco".

Outra modificação que melhora o modelo é a modificação do tempo de duração de um estado em um MOM. O modelo descrito acima tem um tempo de duração de estado exponencial, dado pela formula abaixo.

$$p_i(d) = (a_{ii})^{d-1}(1-a_{ii})$$

Esse modelo é inapropriado para sinais físicos, pois sua duração não é exponencial. Utiliza-se então uma duração explicita dos estados, definindo:

- $a_{ii} = 0$
- $p_i(d)$ = densidade de duração

Essa modificação produz resultados melhores, mas requer modificações nos algoritmos *forward-backward* e de Viterbi. Essa solução também é mais custosa computacionalmente.

Fone e Fonema

Como mencionamos anteriormente, um reconhecedor da fala pode ter diversos graus de complexidade, reconhecendo desde dígitos ou palavras até frases inteiras enunciadas de maneira natural. Para reconhecedores mais simples, pode-se usar como elemento básico as palavras que se deseja reconhecer. No entanto, isso se torna proibitivo para modelos mais complexos, onde se desejar reconhecer um amplo vocabulário.

Isso se deve em parte à necessidade de se treinar o reconhecedor da fala. Para realizar o treinamento, é preciso ter diversas pronúncias de uma mesma palavra. Para sistemas grandes, isso pode ser muito difícil de gerar. Uma opção para esse problema é quebrar as palavras em pedaços menores e utilizar esses pedaços como blocos básicos. Assim, seria necessário ter apenas varais pronúncias desses blocos básicos, e não de todas as suas possíveis combinações.

Esses blocos básicos podem ser, por exemplo, sílabas, fonemas ou fones. Um fonema é a menor unidade sonora de uma língua que ainda é capaz de estabelecer contraste de significado entre palavras. A fonologia de uma língua define, entre outras coisas, os fonemas da língua. O português tem 33 ou 34 fonemas, dependendo do dialeto. Os fonemas ajudam a

identificar como uma palavra é pronunciada e são escritos entre / /. Nos dicionários, os fonemas são geralmente listados junto aos significados das palavras.

Um fone é um segmento da fala que possui características ou percepções distintivas. É um evento físico que não está ligado à fonologia da língua. Tanto o fonema quanto o fone são bastante utilizados por reconhecedores da fala. A unidade básica usada depende, novamente, do reconhecedor.

Modelo Acústico e Modelo de Linguagem

O problema da inferência probabilística no reconhecimento da fala pode ser descrito da seguinte maneira. O reconhecedor captura um sinal acústico *Sinal*. Este sinal é capturado de um meio ruidoso. Essas fontes de ruído podem ser, literalmente, fontes de ruído, como um caminhão ou avião passando ou música no fundo. O microfone utilizado, os cabos de conexão e fontes eletromagnéticas próximas, bem como o próprio conversor A/D também podem inserir ruído na linha. Em fim, o sinal captado, mesmo que o usuário tente, não será garantidamente o mesmo para uma mesma palavra.

Seja *Palavras* uma variável aleatória que varai sobre todas as seqüências possíveis de palavras que poderiam ser articulados. Para inferir o que foi dito, devemos encontrar a seqüencia de palavras *Palavras* que maximiza a probabilidade:

$$P(Palavras \mid Sinal)$$

Aplicando a regra de Bayes, obtemos:

$$P(Palavras \mid Sinal) = \alpha P(Sinal \mid Palavras) P(Palavras)$$

Na expressão acima, o termo $P(Sinal \mid Palavras)$ é o modelo acústico. Esse modelo descreve os sons das palavras. O modelo acústico define também que duas palavras têm o mesmo som. Se $P(Sinal \mid Palavras)$ tem o mesmo valor para um dado sinal quando *Palavras* é “sela” ou “cela”, essencialmente estamos definindo que sela e cela possuem a mesma pronúncia.

A probabilidade $P(Palavras)$ é chamada de modelo de linguagem. Esse modelo especifica as probabilidades *a priori* de se encontrar uma determinada seqüência de palavras. Através desse modelo, podemos definir que “cela trancada” é mais provável que “sela trancada”. A primeira frase é mais comumente encontrada e, portanto, sua probabilidade *a priori* será maior que a da segunda frase.

Esses dois modelos são importantes, pois ajudam a definir o resultado da inferência. Quanto melhor os parâmetros desses modelos, maior será a taxa de acerto do reconhecedor da fala.

Sons da Fala: Processo

O processo de reconhecimento de fala se inicia a partir das ondas sonoras emitidas por uma determinada pessoa, correspondendo à variação de pressão que se propaga no ar. Tal pressão é captada por um microfone (um conversor de sinais analógicos em digitais) e amostrada a uma taxa de 8 a 16 kHz (para músicas de alta qualidade, geralmente a taxa de amostragem é de 44 kHz, mas, para voz, a faixa citada é suficiente para obter uma boa qualidade, uma vez que a voz humana possui uma banda de freqüência menor). Os sinais digitais obtidos do microfone são quantizados numa quantidade de bits que varia de 8 a 12 bits.

Como a taxa de mudanças no sinal de voz é baixa (em torno de 100 Hz), é necessário efetuar um processo de simplificação do sinal recebido. Considerando-se taxas de amostragem em torno de 8 a 16 kHz e quantização de 8 a 12 bits, tem-se um volume de dados correspondente a, aproximadamente, 0,5MB/min. Tal volume de dados é excessivamente grande e ineficiente para servir de entrada para um algoritmo de reconhecimento de fala. Por esse motivo, o sinal digital é resumido em frames, correspondendo a um período de, aproximadamente, 10 ms, nos quais não existem grandes alterações no formato da onda sonora. Cada frame, por sua vez, é analisado e categorizado em características (*features*). Num sistema real, existem por volta de 10 a 100 características que refletem o formato de onda, possibilitando uma modelagem de fonemas mais simplificada. A seguir, é apresentado um esquema dos processos anteriormente descritos:

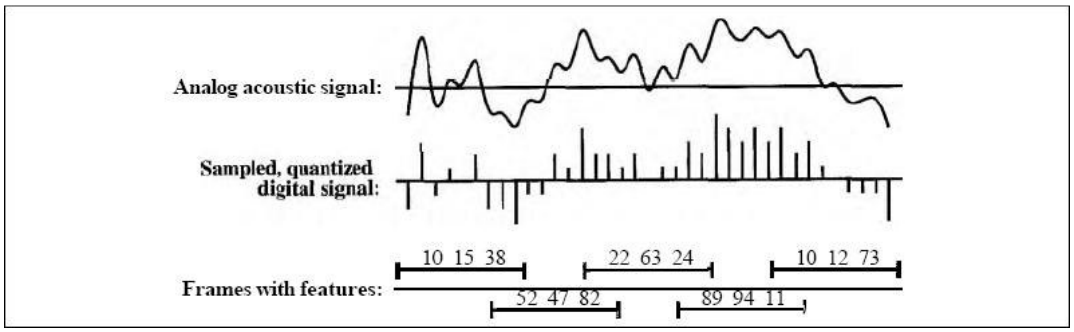

Figura 5: Sinal analógico, sinal amplificado e os frames com as características.

A implementação desta simplificação gera, ainda, alguns problemas. O principal é o seguinte: tendo uma quantidade N de características, cada uma delas, geralmente, com 256 possíveis valores, o frame possui 256^N possíveis valores. Montar uma tabela simples relatando P(característica|fonema) torna-se impraticável, devido ao número excessivo de possibilidades. Desta forma, existem duas abordagens que contornam este problema:

- Quantização vetorial: o espaço n-dimensional representado pelos 256 valores das N características é dividido em 256 regiões (por exemplo, C1-C256), passando de 256^N valores para somente 256. Tal abordagem é simplista demais e, por isso, em sistemas de larga escala não tem sido mais utilizada;
- Distribuição gaussiana (normal): uma alternativa é representar P(característica|fonema) como uma distribuição normal, caracterizada por sua média e variância. Deste modo, trata-se de modelar a distribuição de probabilidades de

forma estatística e contínua, o que se mostra mais próximo da realidade do que a solução anterior (discreta).

Sons da Fala: Refinamento

Cada fone, dependendo do interlocutor, tem uma duração que varia de 5 a 10 *frames*. Porém, nesse período de tempo, o fone não apresenta um comportamento sonoro estável e sofre variações na sua pronúncia e no som gerado. Por esse motivo, o modelo oculto de Markov geralmente sofre modificações no sentido de se aproximar à realidade. Desse modo, a pronúncia de um determinado fone é modelada em três fases: começo, meio e fim. Como exemplo, temos o fone [t] da língua inglesa:

- Começo silencioso
- "Pequena explosão" no meio
- "Assovio" no fim

Outro refinamento utilizado é considerar o contexto no qual o fone é pronunciado. Desta forma, o estado no modelo oculto de Markov representa o fone em si, porém, com os fones imediatamente anterior e sucessor, uma vez que o cérebro humano, na hora da fala, "agrupa" os sons e processa os fones de maneira totalmente isolada. Este modelo é o que se chama de "trifonético".

Com esses dois melhoramentos expostos, o número de estados possíveis numa cadeia oculta de Markov passa de N para $3N^3$, adicionando maior complexidade para o sistema, ao mesmo tempo em que melhora a execução do reconhecedor de fala.

Palavras

Na fase de avaliação de palavras, o reconhecimento se baseia em modelos flexíveis que suportam certos desvios na fala (como dialetos, regionalismos e outros desvios do cotidiano) de forma probabilística. Exemplos disso são apresentados a seguir:

- Assimilação: /me'nino/ = [me'ninʊ] ou [mi'ninʊ]
- Fortalecimento: /ra'pas/ = [ra'pas] ou [ra'pays]
- Enfraquecimento: /abay'ʃar/ = [abay'ʃar] ou [aba'ʃar]

Na língua inglesa, por exemplo, tem-se a palavra "tomato", que, seguindo variações dialéticas e de articulação, pode assumir as seguintes pronúncias:

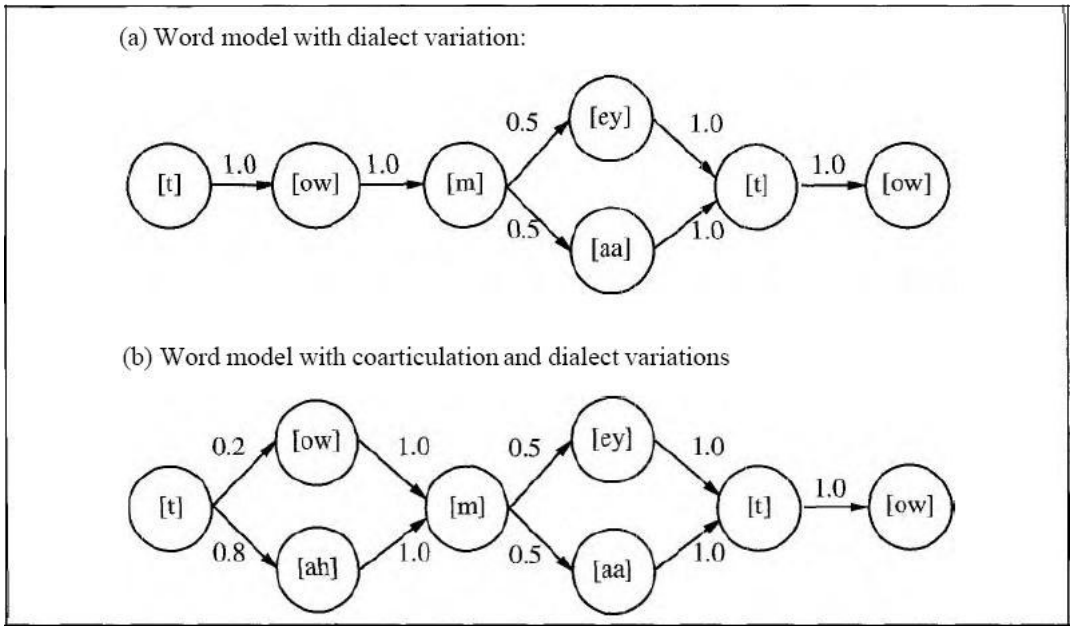

Figura 6: Modelo de palavras com variação de dialeto (a) e com variação de dialeto e articulação (b).

Sentenças

Dificuldades do Reconhecimento de Sentenças

Em uma primeira análise, o processo de reconhecer fala contínua pode parecer equivalente ao processo de reconhecer palavras em seqüência. A idéia apresentada é falsa por dois motivos:

1. A seqüência de palavras mais prováveis é diferente da seqüência mais provável de palavras. Em outras palavras, analisar a probabilidade de que uma palavra isolada tenha sido pronunciada é menos eficiente do que analisar a probabilidade de que essa mesma palavra tenha aparecido dentro do contexto da sentença. Por exemplo, a frase "I have a gun" é certamente mais provável do que a frase "I have a gub", mesmo que a análise isolada da última palavra tenha revelado que "gub" era a alternativa mais provável.
2. Durante a análise da fala contínua, existe o problema de segmentação, isso é, decidir onde é o fim de uma palavra e o início da próxima. Tal problema surge devido ao fato de que a maioria dos falantes não faz pausas claras entre palavras.

Modelo de Linguagem

Um modelo de linguagem é responsável por estabelecer a probabilidade de cada uma das seqüências de palavras possíveis. Considere a sentença $w_1 w_2 w_3 \dots w_n$, formada por n palavras em seqüência. A probabilidade de essa sentença ocorrer é dada pela regra da cadeia:

$$P(w_1 w_2 w_3 \dots w_n) = P(w_1)P(w_2|w_1)P(w_3|w_2) \dots P(w_n|w_1 w_2 w_3 \dots w_{n-1})$$

Considerando que os termos acima são difíceis de calcular, surge a necessidade de realizar aproximações, como o modelo do bigrama e o modelo do trigrama.

O **modelo do bigrama** determina que a probabilidade de uma palavra dentro da sentença é condicionada apenas à palavra anterior (modelo de Markov de primeira ordem):

$$P(w_i|w_1w_2w_3 \dots w_{i-1}) = P(w_i|w_{i-1})$$

Esse modelo é fácil de ser treinado, bastando uma contagem do número de palavras em uma base textos reais. Por exemplo, se a palavra "eu" aparecer 1000 vezes em um texto, mas for seguida da palavra "tenho" em apenas 37 ocorrências, então a probabilidade de que "tenho" apareça após "eu" vale 37/1000.

Uma aproximação mais complexa é o **modelo do trigrama**, que determina que a probabilidade de uma palavra dentro da sentença é condicionada às duas palavras anteriores (modelo de Markov de segunda ordem):

$$P(w_i|w_1w_2w_3 \dots w_{i-1}) = P(w_i|w_{i-1}w_{i-2})$$

Tal modelo é capaz de realizar julgamentos de probabilidade mais sofisticados, mas ainda não possui a flexibilidade de modelos que consideram a gramática. Assim, o modelo do trigrama seria capaz de avaliar que "o homem tenho" é uma sentença improvável (erro gramatical), mas seria incapaz de fazer um julgamento semelhante para a frase "o homem de chapéu amarelo tenho".

Na discussão a seguir, será considerado apenas o modelo do bigrama.

Modelo Oculto de Markov Combinado

Para que o sistema reconheça sentenças corretamente, é necessário combinar os modelos de palavra (modelos de pronúncia e de fones) com o modelo de linguagem. Para isso, será construído um modelo oculto de Markov combinado, em que cada estado é rotulado com três informações:

1. Fone atual;
2. Estado do fone (início, meio ou fim);
3. Palavra atual.

Figura 7: Exemplo de um estado no modelo de Markov combinado.

Nesse modelo, três tipos de transição são possíveis:

1. Entre estados de um mesmo fone;
2. Entre fones de uma mesma palavra;
3. Entre palavras (probabilidade dada pelo modelo do bigrama).

Se houver w palavras possíveis, cada uma com uma média de p fonemas, e cada fonema for modelado com três estados, então o modelo de Markov combinado possuirá $3wp$ estados.

Algoritmo de Viterbi (solução para o segundo problema)

Depois de construído o modelo de Markov combinado, basta utilizar o algoritmo de Viterbi para determinar a seqüência de estados mais prováveis. Quando essa seqüência for conhecida, basta ler as palavras nos rótulos dos estados para determinar a sentença. Considerando que o algoritmo de Viterbi leva em conta todas as seqüências de palavras possíveis e todas as fronteiras possíveis entre palavras, soluciona-se o problema se segmentação (segundo problema apresentado).

Genericamente, o algoritmo de Viterbi depende de algumas considerações importantes, todas condizentes com um modelo de Markov de primeira ordem:

- Os eventos observáveis e ocultos devem estar em seqüência do ponto de vista temporal.
- As seqüências de eventos observáveis e ocultos devem estar sincronizadas, de forma que um evento observado deve corresponder a um único evento oculto.
- O cálculo da seqüência oculta mais provável até o instante t deve depender apenas do evento observado no instante t e da seqüência mais provável no instante $t-1$.

O algoritmo executa as seguintes etapas:

- Processa os nós no sentido crescente de seqüência, calculando a probabilidade do caminho mais provável a cada etapa. Notar que as probabilidades na etapa t dependem apenas das probabilidades da etapa $t-1$.
- Ao final do processamento, terão sido encontradas as seqüências mais prováveis chegando a cada um dos nós finais.
- Através da análise das probabilidades, é imediato escolher qual das seqüências é a mais provável de forma global.
- Por fim, tal seqüencia pode ser reconstruída percorrendo, a partir do estado final, os apontadores que indicam o estado anterior na seqüencia mais provável. Esses apontadores são atualizados pelo algoritmo durante o processamento.

Decodificador A* (solução para o primeiro problema)

Na prática, a probabilidade de uma **seqüência de palavras** é a soma das probabilidades de todas as **seqüências de estados** que são consistentes com a seqüência de palavras. Assim, é possível que a expressão “a back” seja reconhecida por 10 seqüências diferentes de estados (variações de pronúncia, por exemplo), cada uma com probabilidade 0,03. Por outro lado, é possível que a expressão “aback” seja reconhecida por apenas uma seqüência de estados (pronúncia única), cuja probabilidade é 0,20. O algoritmo de Viterbi escolheria “aback”, já que 0,20 é maior do que 0,03. Entretanto, “a back” é mais provável, já que a soma das probabilidades de todas as seqüências de estados que reconhecem a expressão é 0,30.

Para solucionar essa questão (segundo problema apresentado), utiliza-se um decodificador A*, que realiza uma busca A* para encontrar a seqüência mais provável de palavras. Considera-se um grafo em que cada nó representa uma palavra. Assim, os sucessores de um nó são todas as palavras possíveis que podem seguir a palavra rotulada nesse nó.

Considere que se deseja encontrar o caminho mais provável entre os nós w_1 e w_n. No momento da análise de um nó genérico w_k, pode-se estimar o custo do caminho como $f(w_k) = g(w_k) + h(w_k)$, em que:

- $g(w_k) = \sum_{i=1}^{k} -\log P(w_i|w_{i-1})$
- $h(w_k)$ é uma função heurística que estima o custo entre w_k e w_n.

Com a definição apresentada, o problema de encontrar o caminho mais curto torna-se exatamente equivalente ao problema de encontrar a seqüência mais provável de palavras. Uma inconveniência do método apresentado é a dificuldade de se encontrar uma heurística $h(w_k)$ interessante do ponto de vista prático.

Construindo um Reconhecedor de Fala

A qualidade do reconhecedor depende exclusivamente da qualidade de seus componentes:

- Modelo de linguagem;
- Modelo de pronúncia de palavras;
- Modelo de fones;
- Algoritmo de processamento de sinal.

O modelo de linguagem já foi discutido e o algoritmo de processamento de sinal está fora do escopo do texto. Devem-se analisar, portanto, os modelos de pronúncia e fones.

O modelo de pronúncia de palavras é normalmente feito à mão (dicionários de pronúncia) e o modelo de fones é constante. Resta, portanto, entender como as probabilidades entre fones são obtidas, já que há potencialmente milhões de parâmetros. A maneira mais fácil de obter as probabilidades é analisando grandes volumes de gravações reais em áudio e utilizando uma abordagem de contagem semelhante à do modelo do bigrama. Fica em aberto, entretanto, a tediosa tarefa de ouvir cada uma das palavras e associá-las a fones (outra tarefa manual e sujeita a erros). Uma alternativa é o **algoritmo de maximização da expectativa**, que é capaz de "aprender" as probabilidades do modelo oculto de Markov sem a necessidade de marcações manuais.

Os sistemas de reconhecimento de voz modernos utilizam volumes de dados e poder computacional muito grandes para treinar seus modelos. Sistemas de reconhecimento de palavras isoladas com vocabulário de milhares de palavras têm precisão de 99%. Entretanto, a maioria dos sistemas de reconhecimentos de fala contínua possui precisão da ordem de 60-80%. Deve-se enfatizar também que ruídos diversos diminuem significativamente as precisões apresentadas.

Conclusão

Em suma, verifica-se que o modelo oculto de Markov se mostra eficaz no problema de reconhecimento de fala, já que o modelo é capaz de abranger diversas formas de pronúncia de maneira probabilística. De fato, a tarefa de reconhecimento de voz seria inviável em um sistema baseado somente em regras simples.

Bibliografia

1. “Curso do MIT sobre reconhecimento de fala”, http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-345Automatic-Speech-RecognitionSpring2003/CourseHome, acessado em 01/11/2008.
2. “Wikipedia – Hidden Markov Model”, http://en.wikipedia.org/wiki/Hidden_Markov_Model, acessado em 01/11/2008.
3. “Wikipedia – Speech recognition”, http://en.wikipedia.org/wiki/Speech_recognition, acessado em 30/10/2008.
4. “Wikipedia, Viterbi algorithm”, http://en.wikipedia.org/wiki/Viterbi_algorithm, acessado em 31/10/2008.
5. B.H. Juang e L. R. Rabiner “Automatic Speech Recognition – A Brief History of the Technology Development”, http://www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/354_LALI-ASRHistory-final-10-8.pdf, acessado em 04/11/2008.
6. C. A. Ynogutti “Reconhecimento de Fala Contínua Usando Modelos Ocultos de Markov”, http://www.decom.fee.unicamp.br/lpdf/teses_pdf/Tese-Doutorado-Carlos_Alberto_Ynoguti.pdf, acessado em 04/11/2008.
7. L. R. Rabiner “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition” , Proceedings of the IEEE, Vol. 77, Nº 2, 1989.
8. S. J. Russel e P. Norvig “Inteligência Artificial”, 2º edição, Elsevier Editora, Rio de Janeiro, RJ, Brasil.