16. Марковски вериги и вероятностни автомати. Скрити Марковски модели за разпознаване на говорни сигнали. Понятие за кепстър на говорен сигнал [35,42,43,28].

(Виж ПР-част1 – страници 66-74, LectureNotes_Besnik Chapter XIX, http://www2.it.lut.fi/kurssit/04-05/010586001/lectures - 9 лекция)

Марковски вериги и вероятностни автомати

Има модели за класификция, които се тъй наречените контекстно зависими модели за класификация. При тях последователност от решения е необходима, не както при другите просто едно решение и всяко едно решение зависи от останалите. Приложението на тези модели за класификация е когато имаме процес, който бива следен в множество определени моменти във времето. (СММ – често използвани за разпознаване на ръкопис и говор)

Вероятностни автомати

Вероятностния автомат(ВА) е недетериминиран краен автомат(НДА), при който

ВА=(Q,∑,P,v,F) са съответно:
Q – множеството от състоянията на автомата
∑ - азбуката на автомата
P – p: Q x ∑ → P(Q)
v – множество от начални състояние
F – финални състояния

Като често пъти P се представя чрез вероятностна матрица за състоянията(вероятност за преход от едно в друго), а F като вероятностна матрица за финално състояние(указва конкретни финални състояния, вероятност за финално състояние).

Марковски вериги(модели)

Марковските вериги са базирани на вероятностни автомати и движенията между състоянията.

Определение за Марковска верига:
Нека различните състояния на автомата бъдат - w_i.
Нека състоянието в конкретен момент е w(t), а последователността от състояния в рамките на наблюдението:

$$\Omega = (w(1), w(2), \dots, w(N))$$

Марковският модел от първи ред се описва чрез вероятността за преминаване от едно състояние в друго:

$$p\left(w_j(t+1)\middle|w_i(t)\right) = a_{ij}$$

което както се забелязва не зависи от времето t. Важно е да отбележим, че вероятностите не са симетрични и може да имаме преход към същото състояние. Марковските модели от по-висок ред се описват с вероятността от n състояния да преминем в състоянието wi.

Приложение при класификация:
Нека θ е множеството от всички парамери - a_{ij}. Тогава вероятността този модел да генерира определена верига е:

$$p(\Omega|\theta) = \prod_{t=0}^{N-1} p\big(w(t+1)\big|w(t)\big)$$

(Пример: $\Omega = (w_1, w_4, w_1, w_2, w_4) \rightarrow p(\Omega|\theta) = a_{14}a_{41}a_{12}a_{24}$))
Генерирайки множество Марковски вериги ние можем да отркрием тази с най-голяма вероятност за определена последователност от действия и по този начин да направим класификацията.
(Пример: Всяко състояние може да отговаря на фонема(говорно разпознаване) или буква(ръкопис), намирайки правилната последователност можем да разпознаваме думи.)

Скрити Марковски Модели(СММ)

При СММ състоянията са скрити – не могат да бъдат директно наблюдавани. Вместо това можем да правим независимо наблюдение на модела – x(t), за всеки един момент във времето. Така получаваме – X= (x(1), x(2), …, x(N)) – редица от наблюдения. Всяко наблюдение от своя страна зависи само от сегашното състояние и определено вероятностно разпределение:

$$p\left(x(t)\middle|w_j(t)\right) = p(x|w_j)$$

Сега имаме три основни проблема

- Оценка – ако имаме СММ (нейните параметри), да определим вероятността за наблюдение на определена редица от наблюдения X.
- Декодиране – ако имаме СММ и редица от наблюдения X, да определим най-добрата редица от скрити състояния довели до тези наблюдения
- Обучение – имайки знание за структурата на СММ(броя състояния) и множество от обучаващи наблюдения, да определим модела на параметрите(вероятностите на преминаването от едно състояние в друго и на наблюденията)

Оценка
Вероятността за множеството от наблюдения P(X) зависи от две неща – вероятността да се случи точно тази поредица от състояния и вероятността да наблюдаваме определена поредица състояния.

$$P(X) = \sum_{r=1}^{r_{max}} P(X|\Omega_r)P(\Omega_r)$$

където Ω_r е определена редица от състояния, а r_{max} е броя на възможните редици от състояния.
Нека разгледаме сега $P(X|\Omega_r)$:

$$P(X|\Omega) = \prod_{t=1}^{N} P(x(t)|w(t))$$

а за да намерим $P(\Omega_r)$ можем да използваме вероятността за Марковските модели:

$$P(\Omega) = \prod_{t=1}^{N} P\big(w(t)\big|w(t-1)\big)$$

и така за общата вероятност имаме:

$$P(X) = \sum_{r=1}^{r_{max}} \prod_{t=1}^{N} P\big(x(t)\big|w(t)\big)\, P\big(w(t)\big|w(t-1)\big)$$

За съжаление пресмятането на тази вероятност има сложност O(N*c^N), където с е броя на скритите състояния на СММ-то, което би било безсмислено да се прави напрактика. Но тъй като P(X) зависи само от настоящото и предишното състояние и настоящото наблюдение, то можем да направим рекурсивно изчисление(с помощта на методите на динамичното програмиране)

Нека π_i е вероятността да започнем с начално състояние w_i, $\pi_i = P(w_i(1))$.
Сега да дефинираме $\alpha_j(t)$ като вероятността за наблюдение на частичната редица (x(1), ... , x(t)) и стигането и до състояние w_j в момента t. На базата на това можем да дефинираме рекурсивна формула за $\alpha_j(t)$:

$$\alpha_j(t) = \sum_{i=1}^{c} \alpha_i(t-1)\, a_{ij} P(x(t)|w_j)$$

Така получаваме така наречения Forward алгоритъм като: $P(X) = \sum_{j=1}^{c} \alpha_j(N)$.
Сложността му е доста по приемлива - O(N*c^2).

И така ако имаме много СММ-та и неизвестен клас редици той може да бъде класифициран като продуциран от редицата с най-висока вероятност. Ако е необходимо да завършваме в конкретно състояние w_i то вероятността да завършим в него си е $\alpha_i(N)$.

Декодиране
Идеята от дадена редица от наблюдения да открием най-подходящата редица от скрити състояния.

- Изпробване на всички варианти е невъзможно твърде много са.
- Има два подхода – локални максимуми на вероятностите и алгоритъм на Витерби(Viterbi)

Метода на локалните максимуми на всяка времева стъпка t избира w_i с най-висока вероятност - $\alpha_i(t)$. Алгоритъма не е много надежден, защото може да даде навилидни преходи във Марковската верига.

Алгоритъм на Витерби

С други думи това е алгоритъм за намиране на максималното правдоподобие на редица от състояния чрез динамично програмиране.
Идеята е на всяка стъпка да се изчисли най-вероятния път за всички състояния.
Вместо да работим с вероятности би било по-лесно да се работи с отрицателни логаритмични вероятности, което променя проблема от намиране на максимално вероятна редица до намиране на тази с минимална цена, за функция на цената имаме:

$$\delta_j(t) = -\ln P(w_j(t))$$

Така проблема се свежда до:

$$min_{j_1,j_2,\dots j_N}\left(\sum_{t=1}^{N}\delta_{j_t}(t)\right)$$

Което напрактика означава на всяка времева стъпка да се изчисли най-краткия път до всяко състояние. (Сложност - O(N*c^2)).

Обучение
Опростяване на модела би се получил, ако имаме обучаващо множество.

Нека имаме крайно множество от възможни нааблюдения. Можем без ограничение на общността да си мислим, че всяко състояние връща символ, а всяко наблюдение е точно този символ. Нека {x1, x2, …} е множеството от тези символи.
Сега можем да дефинираме вероятността като:

$$b_{jk} = P(x_k(t)|w_j(t))$$

Идеята сега е да намерим a_{ij} и b_{jk} за СММ чрез обучаващото множество.

Не е открит все още метод за намиране на глобален екстрмум, но има добро решение за локален екстрмум чрез алгоритъма на Баум-Уелч (Baum-Welch).

Алгоритъм на Баум-Уелч
Баум-Уелч алгоритъма има сходство с Forward алгоритъма и е базиран на максималното подобие и максимално очакване.
Също както при Forward алгоритъма имахме оценка на проблема, можем да направим същата оценка на базата на Backward алгоритъм, който е отговорен за изчислението на backward променливите $\beta_i(t)$. Като $\beta_i(t)$ е вероятността, ако модела е в състояние $w_i(t)$, да бъде генериран остатъка от наблюдаваната редица.

$$\beta_i(t) = P(x(t+1), x(t+2), \dots, x(N)|w_i(t))$$

За да изчислим $\beta_i(t)$ ефикасно е необходимо както и при $\alpha_j(t)$ да използваме рекурсивен подход.

$$\beta_i(t) = \sum_{j=1}^{c}\beta_j(t+1)\, a_{ij} P(x(t+1)|w_j)$$

При начално условие - $\beta_i(N) = 1$, за $\forall i$.

Сега ще дефинираме две помощни променливи - $\xi_{ij}(t)$ и $\gamma_i(t)$. Първата е вероятността да сме в състояние i в момента t и в състояние j в момента t+1:

$$\xi_{ij}(t) = P\big(w_i(t), w_j(t+1)\big|X, \theta\big)$$
$$\xi_{ij}(t) = \frac{P\big(w_i(t), w_j(t+1), X\big|\theta\big)}{P(X|\theta)}$$
$$\xi_{ij}(t) = \frac{\alpha_i(t)a_{ij}\beta_j(t+1)P(x(t+1)|w_j)}{\sum_{k=1}^{c}\sum_{l=1}^{c}\alpha_k(t)a_{kl}\beta_l(t+1)P(x(t+1)|w_l)}$$

а $\gamma_i(t)$ е вероятността да се намираме в състояние i в момента t с дадената редица от наблюдения и модел:

$$\gamma_i(t) = P(w_i(t)|X, \theta) = \frac{\alpha_i(t)\beta_i(t)}{\sum_{j=1}^{c}\alpha_j(t)\beta_j(t)}$$

Сега вече може да дефинираме процеса на Баум-Уелч за преоценка на праметрите на СММ с цел максимизиране на $P(X|\theta)$. И така избирайки начален модел θ = (a, b, π), можем да преизчислим стойностите за α и β и след това за $\gamma_i(t)$ и $\xi_{ij}(t)$. След можем да преизчислим и a_{ij} и b_{jk}:

$$\pi_i = \gamma_i(1)$$
$$a_{ij} = \frac{\sum_{t=1}^{N}\xi_{ij}(t)}{\sum_{t=1}^{N}\gamma_i(t)}, \text{за } \forall i, j$$
$$b_{jk} = \frac{\sum_{t|x(t)=x_k}\gamma_i(t)}{\sum_{t=1}^{N}\gamma_i(t)}, \text{за } \forall j, k$$

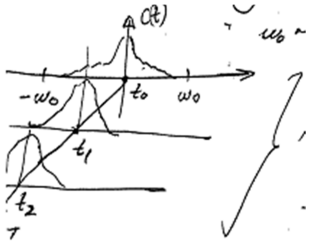
Понятие за кепстър на говорен сигнал

Как да представим говорния сигнал?

Можем да считаме, че в някакъв интервал Δt говорния сигнал е почти постоянен. И нека използваме прозорец Δt, с който да обходим сигнала с цел извличане на фрейми от говорния сигнал.

Нека C(t) е функцията на звука по времето. Като $C(t) = S(t) \circledast H(t)$ – където това е конволюция .

1) Разлагаме C(t) по Фурие и получаваме спектър с $w_0 \sim \frac{1}{\Delta t}$, а $\Delta t \sim 100ms$ от минимума (10Hz) ще имаме една синусоида а за максимума няколко.



2) Разглеждаме даден фрейм:

$$\widetilde{C_i}(w) = F(C_i(t))_{(w)} = F(S_i \circledast H_i)_{(w)} = F(S_i)F(H_i)_{(w)}$$

Взимаме мощностния спектър (слагаме модул на всички) – чрез експерименти е доказано, че фазата не е толкова важна колкото аплитудата.

$$\widetilde{C_i}(w) = |F(S_i)||F(H_i)|_{(w)}$$

3) Правим филтриране по Мел с цел да приведем сигнал в подобен на този, който чува мозъка.

$$\widetilde{C_i}(w) = C_i(w)M(w)$$

4) Логаритмуваме получения сигнал:

$$\tilde{C}(w) = Log\ \widetilde{S_i}(w) + Log\ \widetilde{H_i}(w)$$

5) Правим дискретна косинус трансформация по Фурие (DCT)

$$C(\tau) = DCT(\tilde{C}(w))_\tau = S(\tau) + H(\tau)$$

Горната функция се нарича кепстрално пространство с:

- Cepster - $C(\tau), S(\tau), H(\tau)$
- τ – quefrencies(анаграма на честотата на звъка)
- $S(\tau)$ – сигнал от глатиса
- $H(\tau)$ – сигнал от устната кухина

6) Сляпа деконволюция:

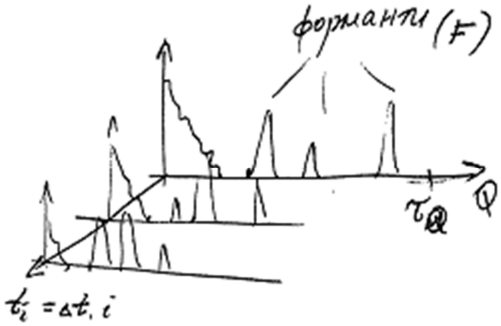
Искаме да разделим $H(\tau)$ от $S(\tau)$:

$$\bar{C}(\tau) = \frac{1}{T}\sum_{t=0}^{T} c(\tau, t) = \frac{1}{T}\sum s(\tau, t) + \frac{1}{T}\sum h(\tau, t) = \bar{s}(\tau) + \bar{h}(\tau)$$

Като извадим средните стойности от началния сигнал получаваме само влиянието на звуковата част.

$$Q \approx 50 \div 100$$

оказва се, че $\bar{h}(\tau)$ е почти константа.



⇨ $\bar{C}(\tau, t) = C(\tau, t) - \bar{C}(\tau)$

⇨ $\bar{C}(\tau, t) = S(\tau, t) - \bar{S}(\tau) + \varepsilon(h)$

Като $\varepsilon(h)$ е пренебрежимо малко.

СММ при разпознаване на говорен сигнал

Сега остава само да кажем, че на СММ подаваме кепстри(Oi) а резултат е последователност от кепстри – разпозната дума(O=(O1,O2,…,ON)).

1) Обучение – когато се подаде дума(O), СММ се настроива така, че P(O) да е максимално. Използваме Баум-Уелч.
2) Разпознаване – чрез стъпка едно генерираме база от СММ оптимално настроени към съответните думи, като тази база се съхранява под формата на речник.