UNIVERSITÀ DEGLI STUDI DI PADOVA

Facoltà di Ingegneria
Corso di Laurea in Ingegneria dell'Informazione

Tesi di Laurea Triennale

Inferenza statistica per Hidden Markov Models

Relatore:
Prof. Lorenzo Finesso

Laureando:
Marco Ruzza

Anno Accademico 2010-11

Alla mia famiglia,
che mi è sempre stata vicina
nel momento del bisogno.

Indice

Elenco delle figure

Capitolo 1

Introduzione

Molte volte in ambito ingegneristico ci si trova a dover far fronte ad un problema come il seguente. Un processo del mondo reale produce una sequenza di simboli osservabili, che possono essere a valori discreti (uscite di esperimenti di lancio di una moneta, lettere da un alfabeto finito, ...) o continui (campioni del parlato, vettori di autocorrelazione, ...): ciò che si chiede è di costruire un modello del segnale che spieghi e caratterizzi l'occorrenza dei simboli osservati, così da poter essere utilizzato per l'identificazione e il riconoscimento di altre sequenze di osservazioni.

La teoria dei segnali e dei sistemi ci insegna che per affrontare un tale problema si devono prendere alcune fondamentali decisioni, quale, per esempio, la forma del modello: lineare o non-lineare, tempo-variante o tempo-invariante, deterministico o stocastico.

I modelli di sistemi lineari tempo-invarianti, che modellano i simboli osservati come l'uscita di un sistema lineare a parametri costanti eccitato da un opportuno ingresso, si sono dimostrati utili per una grandissima varietà di applicazioni. Molti segnali del mondo reale, tuttavia, non possono essere significativamente modellati senza considerare una variazione temporale dei parametri prima accennati. Un tale problema può essere affrontato attraverso il seguente approccio.

Molti tra i segnali fisici che nel loro complesso necessitano di modelli tempo-varianti possono essere ciononostante modellati da un sistema lineare tempo-invariante se considerati in un intervallo di tempo sufficientemente piccolo: la natura tempo-variante del processo può essere vista come una diretta concatenazione di questi brevi intervalli di tempo, ciascuno dei quali singolarmente rappresentato da un modello di sistema tempo invari-

ante. Ciascuno di questi intervalli di tempo di osservazione viene visto come un'unità con una durata prestabilita, che in molti sistemi fisici si determina in modo empirico.

In molti processi, ovviamente, non ci si aspetta che le proprietà del processo cambino sincronicamente con la durata dell'analisi di ogni unità, né che si osservino drastici cambiamenti da un'unità alla successiva, se non in particolari casi. In molti casi, infatti, ciò che si osserva è un cambiamento di comportamento sequenziale: le proprietà del processo solitamente si mantengono per un certo periodo di tempo e successivamente cambiano, gradualmente o rapidamente, in un altro insieme di proprietà. Una rappresentazione efficiente può allora essere ottenuta utilizzando un modello di intervallo di tempo comune per ogni parte stabile del segnale, con l'aggiunta di una qualche caratterizzazione di come un tale periodo evolve verso il successivo.

Ecco quindi che per modellare un mondo mutevole si usa una variabile aleatoria per ogni aspetto del suo stato in ogni intervallo temporale. Le relazioni tra queste variabili descrivono l'evoluzione dello stato. Il processo di cambiamento può essere visto come una serie di fotografie, ognuna delle quali descrive lo stato in un particolare istante. Ogni fotografia contiene un insieme di variabili aleatorie, alcune osservabili e altre no. Per semplicità presumeremo che in ogni intervallo di tempo sia osservabile lo stesso sottoinsieme di variabili e assumeremo che:

- i cambiamenti sono causati da un processo stazionario, ovvero che i cambiamenti stessi sono regolati da leggi immutabili nel tempo;
- lo stato corrente dipende soltanto da una storia finita di stati precedenti; studieremo per semplicità la dipendenza di uno stato dal solo stato precedente.

Da queste assunzioni nasce l'idea degli Hidden Markov Models (HMM), processi doppiamente stocastici con un processo stocastico sottostante che non è osservabile (*hidden*, nascosto), ma che può solo essere osservato attraverso un altro insieme di processi stocastici che producono la sequenza di simboli osservati.

Gli Hidden Markov Models furono originariamente introdotti nella letteratura statistica nel lontano 1957. In seguito furono usati con parziale successo in varie applicazioni nel mondo dell'ingegneria a partire dalla fine

degli anni '70. Alcune di questa applicazioni riguardano lo speech processing e la codifica di sorgente. Recentemente, gli HMM sono stati utilizzati anche in alcuni problemi di biologia computazionale, quali l'identificazione dei geni di un organismo dal suo DNA e la classificazione di proteine in un piccolo numero di famiglie.

Data la presenza nella letteratura di diverse definizioni di HMM, nel capitolo 2 si considereranno le tre definizioni prevalenti, dimostrandone la completa equivalenza dal punto di vista del potere espressivo.

In seguito nel capitolo 3 si analizzeranno i problemi di interesse per questi modelli stocastici.

Infine nel capitolo 4 verranno offerti alcuni esempi di problemi ingegneristici affrontati attraverso gli HMM.

Capitolo 2

HMM: tre definizioni, un unico concetto

In questo capitolo prenderemo in considerazione le tre diverse definizioni di Hidden Markov Model che sono prevalenti in letteratura, e mostreremo che sono tutte equivalenti tra loro in termini di potenza espressiva. In altre parole, se un processo stocastico stazionario su un alfabeto finito ha uno dei tre qualsiasi tipi di HMM, allora esso ha tutti e tre i tipi di HMM. Tuttavia, la dimensione dello spazio degli stati è in generale differente nei tre tipi di HMM. Sotto questo aspetto, la definizione di 'processo di Markov congiunto' è la più economica in termini di dimensioni dello spazio degli stati, mentre la definizione di 'funzione deterministica di un processo di Markov' è la più onerosa.

L'espressione 'il processo stocastico ha un HMM', utilizzata in precedenza e nel seguito, significa che 'il processo stocastico può essere rappresentato attraverso un Hidden Markov Model'.

2.1 Le tre definizioni

La prima di questa definizioni è piuttosto popolare nell'ambito della statistica.

Definizione 2.1.1. *Sia* $\{Y_t\}$ *un processo stocastico stazionario con valori in un insieme finito* $\mathcal{Y}$. *Allora* $\{Y_t\}$ *ha un* HMM del tipo 'funzione deterministica di una catena di Markov' *se esistono un processo di Markov* $\{X_t\}$

con valori in un insieme finito $\mathcal{X} = \{1, ..., N\}$*, detto spazio degli stati, e una funzione* $f : \mathcal{X} \to \mathcal{Y}$ *tali che* $Y_t = f(X_t)$*.*

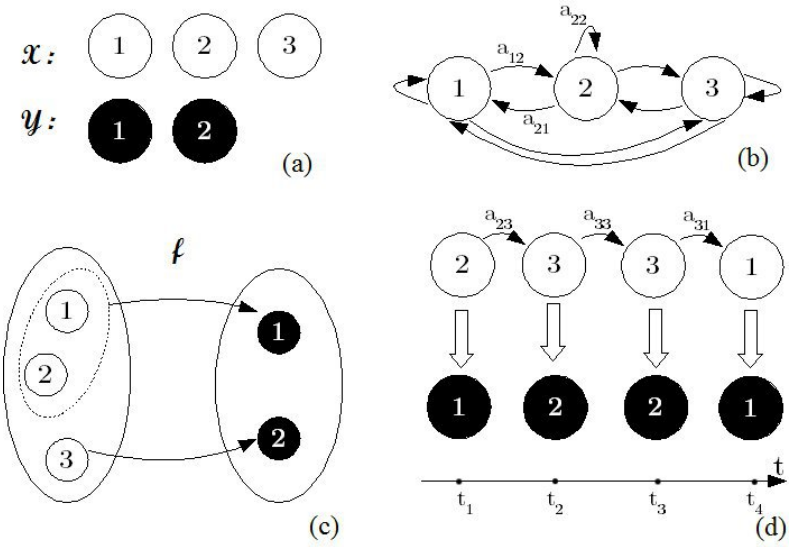

Figura 2.1: HMM del tipo 'funzione deterministica di una catena di Markov'. Nel dettaglio: (a) spazio degli stati $\mathcal{X}$ e spazio dei simboli osservabili $\mathcal{Y}$; (b) natura markoviana del processo nascosto; (c) generazione di natura deterministica dei simboli osservabili a partire dagli stati nascosti; (d) una particolare realizzazione dell'HMM in questione.

Si noti che l'esistenza di $\mathcal{X}$ ed f non è sempre verificata perchè si vuole uno spazio degli stati *finito*. Se è permesso uno spazio degli stati di cardinalità infinita, allora si può sempre costruire un HMM (del tipo 'funzione deterministica di una catena di Markov').

La definizione che segue è invece molto popolare nell'ambito dell'ingegneria.

Definizione 2.1.2. *Sia* N *un numero intero finito. Allora* $\{Y_t\}$ *ha un* HMM del tipo 'funzione stocastica di una catena di Markov' *se la legge di* $\{Y_t\}$ *è la stessa del processo* $\{Z_t\}$ *ottenuto come specificato di seguito: esistono un intero* M *e una coppia di matrici* $A \in [0,1]^{N \times N}$ *e* $B \in [0,1]^{N \times M}$ *tali che valgono le proprietà:*

1. *A è stocastica per righe; in altre parole, ogni riga di A somma a uno.*

2. *B è stocastica per righe; in altre parole, ogni riga di B somma a uno.*

3. *Sia* $\pi \in [0,1]^N$ *un vettore stocastico riga tale che* $\pi = \pi A$. *Sia* $\{X_t\}$ *una catena di Markov con valori in* $\mathcal{X} = \{1, ..., N\}$ *con matrice di transizione tra gli stati* A *e distribuzione iniziale* π. Z_t *viene scelto in modo aleatorio da* M *secondo la legge* $\mathrm{P}(Z_t = u | X_t = j) = b_{ju}$.

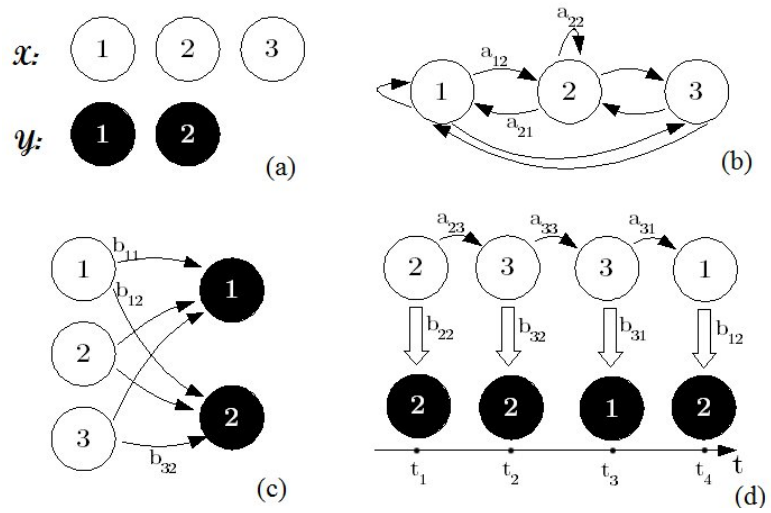

Figura 2.2: HMM del tipo 'funzione stocastica di una catena di Markov'. Nel dettaglio: (a) spazio degli stati $\mathcal{X}$ e spazio dei simboli osservabili $\mathcal{Y}$; (b) natura markoviana del processo nascosto; (c) generazione di natura aleatoria dei simboli osservabili a partire dagli stati nascosti; (d) particolare realizzazione dell'HMM in questione.

In questo caso A e B costituiscono la matrice di transizione tra gli stati e la matrice di output, rispettivamente, dell'HMM.

Infine introduciamo una definizione che si rivela molto utile dal punto di vista teorico ed è la più economica in termini di dimensioni dello spazio degli stati.

Definizione 2.1.3. *Sia* $\{Y_t\}$ *un processo stocastico stazionario sull'alfabeto finito* $\mathcal{Y} = \{1, ..., M\}$. *Il processo* $\{Y_t\}$ *ha un* HMM del tipo 'processo di Markov congiunto' *se esiste un altro processo stocastico stazionario* $\{X_t\}$ *su uno spazio degli stati finito* $\mathcal{X} = \{1, ..., N\}$ *tale che valgono le seguenti proprietà:*

1. *Il processo congiunto* $\{X_t, Y_t\}$ *è di Markov. Perciò*

$$\begin{aligned}\mathrm{P}(X_t = x_t, Y_t = y_t | X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1}, X_{t-2} = x_{t-2}, ...) = \\ = \mathrm{P}(X_t = x_t, Y_t = y_t | X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1}).\end{aligned}$$

2. *Inoltre, è vero che*

$$\begin{aligned}\mathrm{P}(X_t = x_t, Y_t = y_t | X_{t-1} = x_{t-1}, Y_{t-1} = y_{t-1}) = \\ = \mathrm{P}(X_t = x_t, Y_t = y_t | X_{t-1} = x_{t-1}) \triangleq \\ \triangleq m^{(y_t)}_{x_{t-1}x_t}.\end{aligned}$$

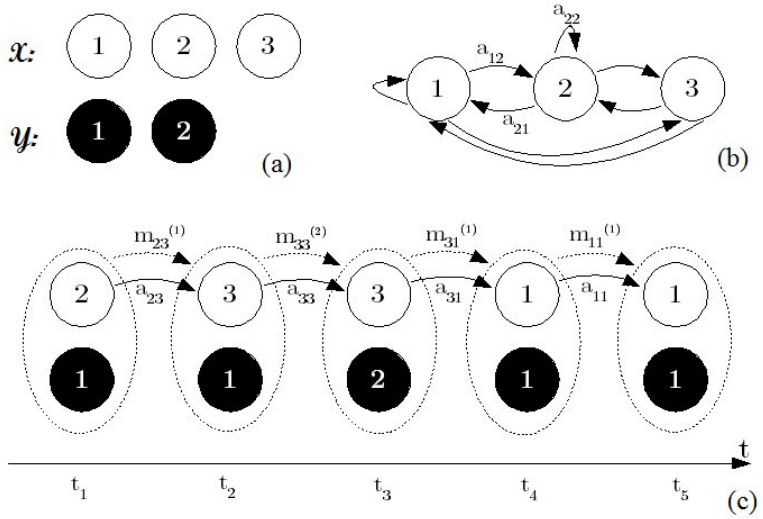

Figura 2.3: HMM del tipo 'processo di Markov congiunto'. Nel dettaglio: (a) spazio degli stati $\mathcal{X}$ e spazio dei simboli osservabili $\mathcal{Y}$; (b) natura markoviana del processo nascosto; (c) visione delle transizioni da stato a stato del processo nascosto e del processo congiunto.

Dalla definizione è chiaro che

$$\begin{aligned}\mathrm{P}(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, ...) = \\ = \mathrm{P}(X_t = x_t | X_{t-1} = x_{t-1}).\end{aligned}$$

In altre parole, $\{X_t\}$ è esso stesso un processo di Markov.

A questo punto è possibile definire le matrici $M^{(u)}$, $u \in \mathcal{Y}$, di dimensione $N \times N$, aventi come elementi proprio le probabilità $m^{(u)}_{ij}$. Definendo ora

$$a_{ij} \triangleq \sum_{u \in \mathcal{Y}} m^{(u)}_{ij} \qquad \forall i, j.$$

è chiaro che la matrice $N \times N$ avente come elementi i termini a_{ij} costituisce proprio la matrice di transizione tra stati A del processo di Markov nascosto $\{X_t\}$.

2.2 Equivalenza tra le definizioni

Dimostreremo ora che i tre modelli visti in precedenza sono equivalenti.

Lemma 2.2.1. *Le seguenti affermazioni sono equivalenti:*

1. *Il processo $\{Y_t\}$ ha un HMM del tipo 'funzione deterministica di una catena di Markov'.*
2. *Il processo $\{Y_t\}$ ha un HMM del tipo 'funzione stocastica di una catena di Markov'.*
3. *Il processo $\{Y_t\}$ ha un HMM del tipo 'processo di Markov congiunto'.*

Dimostrazione. 1. $\Rightarrow$ 2. Chiaramente ogni funzione deterministica di una catena di Markov è anche una funzione 'stocastica' della stessa catena di Markov, con ogni elemento di B uguale a zero o a uno. Per la precisione, poiché $\mathcal{X}$ e $\mathcal{Y}$ sono insiemi finiti, la funzione f induce semplicemente una partizione dello spazio degli stati $\mathcal{X}$ in M sottoinsiemi $\mathcal{X}_1, ..., \mathcal{X}_M$ dove $\mathcal{X}_u \triangleq \{j \in \mathcal{X} : f(j) = u\}$. Quindi due stati in $\mathcal{X}_u$ sono indistinguibili attraverso il processo di misurazione $\{Y_t\}$. Poniamo ora $b_{ju} = 1$ se $j \in \mathcal{X}_u$ e zero altrimenti.

2. $\Rightarrow$ 3. Se $\{Y_t\}$ è modellato come un HMM rappresentabile da una funzione stocastica del processo di Markov con $\{X_t\}$ nel ruolo di catena di Markov sottostante, allora il processo congiunto $\{(X_t, Y_t)\}$ è di Markov. Infatti, se definiamo $(X_t, Y_t) \in \mathcal{X} \times \mathcal{Y}$, allora dalle condizioni dell'HMM ne consegue che

$$\mathrm{P}\left[(X_{t+1}, Y_{t+1}) = (j, u) \,|\, (X_t, Y_t) = (i, v)\right] = a_{ij} b_{ju}$$

Definiamo ora

$$M^{(u)} := [a_{ij} b_{ju}] \in [0, 1]^{N \times N}.$$

Allora il processo $\{(X_t, Y_t)\}$ è di Markov, con matrice di transizione tra stati data da

$$\begin{bmatrix} M^{(1)} & M^{(2)} & \dots & M^{(M)} \\ \vdots & \vdots & \vdots & \vdots \\ M^{(1)} & M^{(2)} & \dots & M^{(M)} \end{bmatrix}.$$

Infine, si noti che $\mathrm{P}[(X_{t+1}, Y_{t+1}) = (j, u)]$ dipende solo da X_t ma non da Y_t. Perciò il processo congiunto $\{(X_t, Y_t)\}$ soddisfa tutte le condizioni richieste per un modello HMM della tipologia 'processo di Markov congiunto'.

3. $\Rightarrow$ 1. Sia $\{X_t\}$ un processo di Markov tale per cui il processo congiunto $\{(X_t, Y_t)\}$ sia anch'esso di Markov. Allora chiaramente $\{Y_t\} = f\{(X_t, Y_t)\}$ per una funzione f opportuna, corrispondente alla proiezione su $\{Y_t\}$. Perciò esso è anche un HMM rappresentabile da una funzione deterministica di una catena di Markov.

□

2.3 Elementi, notazione e meccanismo di un HMM

Tra le definizioni presentate nelle sezioni precedenti, d'ora in poi considereremo la seconda definizione, che descrive un HMM come una funzione stocastica di una catena di Markov. Nel seguito verranno quindi esposti nel dettaglio gli elementi e il meccanismo di questa tipologia di HMM particolarmente utilizzata nel mondo dell'ingegneria.

Un HMM del tipo 'funzione stocastica di una catena di Markov' è costituito da diversi elementi; la notazione utilizzata sarà la seguente:

- T : lunghezza della sequenza di osservazioni (numero totale di cicli di clock).
- N : numero di stati (nascosti) nel modello.
- M : numero di distinti simboli osservabili.
- $\mathcal{X} = \{1, 2, ..., N\}$: spazio degli stati nascosti.
- $\mathcal{Y} = \{1, 2, ..., M\}$: spazio delle possibili osservazioni.
- $\pi = (\pi_1, \pi_2, ..., \pi_N)$: vettore probabilità degli stati iniziali.
- $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix} \in \mathbb{R}^{N \times N}$: matrice di transizione tra gli stati, con $a_{ij} = \mathrm{P}(X_{t+1} = j | X_t = i)$.
- $B = \begin{bmatrix} ba_{11} & b_{12} & \dots & b_{1M} \\ \vdots & \vdots & \vdots & \vdots \\ b_{N1} & b_{N2} & \dots & b_{NM} \end{bmatrix} \in \mathbb{R}^{N \times M}$: matrice di output, con

$b_{ij} = \mathrm{P}\,(Y_t = j | X_t = i)$.

- X_k : variabile aleatoria rappresentante lo stato in cui il sistema si trova al k-esimo ciclo di clock.
- Y_k : variabile aleatoria rappresentante il simbolo osservato al k-esimo ciclo di clock.
- x_k : numero reale indicante lo specifico valore assunto dalla variabile aleatoria X_k al k-esimo ciclo di clock; ovviamente $x_k \in \mathcal{X} \quad \forall k : 1 \leq k \leq T$.
- y_k : numero reale indicante lo specifico valore assunto dalla variabile aleatoria Y_k al k-esimo ciclo di clock; ovviamente $y_k \in \mathcal{Y} \quad \forall k : 1 \leq k \leq T$.

Per questo tipo di modello faremo le seguenti assunzioni:

1. Gli stati nel modello sono presenti in un numero finito, che denoteremo con N. Non definiremo rigorosamente cosa sono gli stati, ma diremo semplicemente che all'interno di uno stato il segnale presenta alcune proprietà misurabili e distintive.
2. A ogni ciclo di clock, t, si entra in un nuovo stato sulla base di una distribuzione delle probabilità di transizione che dipende solamente dallo stato subito precedente (*proprietà di Markov*). Si noti che la transizione può essere tale che il processo rimane nello stato precedente. Con la notazione introdotta, si ha che:

$$\begin{aligned} &\mathrm{P}(X_t = x_t | X_{t-1} = x_{t-1}, X_{t-2} = x_{t-2}, ...) = \\ &= \mathrm{P}(X_t = x_t | X_{t-1} = x_{t-1}). \end{aligned}$$

3. In seguito ad una transizione viene prodotto un simbolo di output osservabile in accordo a una distribuzione di probabilità che dipende solamente dallo stato attuale. Tale distribuzione di probabilità rimane fissa per un determinato stato, per cui ci sono N distribuzioni delle probabilità di osservazione che rappresentano ovviamente variabili aleatorie

o processi stocastici. Formalmente:

$$\begin{aligned}&\mathrm{P}(Y_1 = y_1, Y_2 = y_2, ..., Y_n = y_n | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = \\ &= \mathrm{P}(Y_1 = y_1 | X_1 = x_1)\mathrm{P}(Y_2 = y_2 | X_2 = x_2)...\mathrm{P}(Y_n = y_n | X_n = x_n) = \\ &= \prod_{i=1}^{n} \mathrm{P}(Y_i = y_i | X_i = x_i).\end{aligned}$$

In altre parole, la sequenza $\{Y_n\}$ è *condizionatamente* indipendente dato $\{X_n\}$.

Si noti come la terza assunzione permetta di instaurare la seguente equivalenza:

$$\begin{aligned}m_{ij}^{(u)} &\triangleq \mathrm{P}(Y_{t+1} = u, X_{t+1} = j | X_t = i) = \\ &= \mathrm{P}(Y_{t+1} = u | X_{t+1} = j, X_t = i)\mathrm{P}(X_{t+1} = j | X_t = i) = \\ &= [\text{assunzione 3}] = \\ &= \mathrm{P}(Y_{t+1} = u | X_{t+1} = j)\mathrm{P}(X_{t+1} = j | X_t = i) = \\ &= a_{ij} b_{ju}.\end{aligned}$$

Utilizzando il modello, una sequenza di T osservazioni $\{Y_1 = y_1, ..., Y_T = y_T\}$, esprimibile nella forma compatta $\left\{Y_1^T = y_1^T\right\}$, è generata nel modo seguente:

1. Si sceglie uno stato iniziale, x_1, secondo la distribuzione iniziale π ;
2. Si pone $t = 1$;
3. L'uscita y_t si ottiene in accordo con $b_{x_t y_t}$, la distribuzione di probabilità delle osservazioni per lo stato x_t ;
4. La transizione allo stato x_{t+1} all'istante $t + 1$ è legata ad $a_{x_t x_{t+1}}$, la distribuzione di probabilità di transizioni tra stati per lo stato x_t ;
5. Si pone $t = t+1$; si ritorna al punto 3 se $t < T$; altrimenti si termina la procedura.

Nel seguito spesso si utilizzerà la notazione compatta $\lambda = (A, B, \pi)$ per rappresentare un HMM. La specificazione di un HMM coinvolge la scelta di un numero di stati, N, e di un numero di simboli discreti M: la scelta fatta in sede di spiegazione della notazione di etichettare gli stati con dei numeri, avendo quindi a che fare con un $\mathcal{X} = \{1, 2, ..., N\}$ piuttosto che con un $\mathcal{X} = \{x_1, x_2, ..., x_N\}$, si giustifica quindi col fatto che l'interesse è

per il *numero* di stati e non per la *natura* degli stati, e che un'espressione dello spazio di stato del secondo tipo avrebbe inutilmente appesantito la notazione.

Capitolo 3

Problemi collegati agli HMM

3.1 I tre problemi per gli HMM

Dato un HMM descritto come nell'ultima parte del precedente capitolo, esistono tre problemi di interesse che devono essere risolti per il modello per risultare utile nelle applicazioni del mondo reale. Questi problemi sono i seguenti:

1. Trovare la probabilità di una sequenza osservata dato un HMM (*valutazione*).

2. Trovare la sequenza di stati nascosti che più probabilmente ha generato una sequenza osservata (*decodifica*).

3. Identificare i parametri (A, B, π) di un HMM data una sequenza di osservazioni (*addestramento*).

Il problema di valutazione più essere tradotto nel seguente modo: data una sequenza di osservazioni $\left\{Y_1^T = y_1^T\right\}$ e il modello $\lambda = (A, B, \pi)$, si chiede di calcolare $\mathrm{P}\left(Y_1^T = y_1^T | \lambda\right)$. Visto in un altro modo, dato il modello e una sequenza di osservazioni si chiede di valutare il modello. L' ultimo punto di vista risulta molto utile: se si pensa al caso in cui si ha a che fare con molti modelli in competizione, la soluzione al problema 1 consente di scegliere il modello che si accorda nel miglior modo con le osservazioni. Per calcolare la probabilità di una sequenza di osservazioni dato un particolare HMM, e quindi scegliere il più probabile HMM, possono essere utilizzati in modo efficiente vari algoritmi ricorsivi, quali il forward, il backward e il forward-backward.

Il problema di decodifica consiste nel trovare la sequenza di stati più probabile date certe osservazioni: data una sequenza di osservazioni $\{Y_1^T = y_1^T\}$ si chiede di scegliere una sequenza di stati $\{X_1^T = x_1^T\}$ che è in un certo senso ottimale. Questo è un tipico problema di stima. Solitamente si usa un criterio di ottimizzazione per risolvere questo problema nel miglior modo possibile. Sfortunatamente, come vedremo, ci sono molti possibili criteri di ottimizzazione che possono essere imposti e quindi la decisione del modello è in forte collegamento con l'uso che si vuol fare della sequenza di stati nascosti. Un tipico uso della sequenza di stati nascosti è di comprendere la struttura del modello, e di ottenere medie statistiche, comportamento, etc. all'interno dei singoli stati. Per determinare la più probabile sequenza di stati nascosti data una sequenza di osservazioni si utilizza l'algoritmo di Viterbi.

Il problema di identificazione consiste nell'aggiustare i parametri del modello $\lambda = (A, B, \pi)$ al fine di massimizzare $\mathrm{P}\big(Y_1^T = y_1^T|\lambda\big)$. Esso è cruciale per moltissime applicazioni e, senza dubbio, il più difficile da risolvere. Qualora le matrici A e B non siano direttamente (empiricamente) misurabili, come molto spesso avviene nel caso di applicazioni reali, si utilizza l'algoritmo forward-backward.

3.2 Problema di valutazione

L'obiettivo del problema di valutazione è di calcolare la probabilità di una sequenza di osservazioni dato un Hidden Markov Model; si vuole cioè determinare $\mathrm{P}\big(Y_1^T = y_1^T \big|\, \lambda\big)$ con ovviamente i parametri del modello (A, B, π) noti.

3.2.1 Calcolo diretto

Il modo più diretto per raggiungere tale obiettivo consiste nel numerare tutte le possibili sequenze di stati di lunghezza T (il numero di osservazioni). Per ogni singola sequenza $\{X_1 = x_1, X_2 = x_2, ..., X_T = x_T\}$, che in forma compatta può essere indicata come $\{X_1^T = x_1^T\}$, la probabilità di osservare una sequenza $\{Y_1^T = y_1^T\}$ è $\mathrm{P}\big(Y_1^T = y_1^T|X_1^T = x_1^T, \lambda\big)$, dove

$$\mathrm{P}\left(Y_1^T = y_1^T|X_1^T = x_1^T, \lambda\right) = b_{x_1 y_1} b_{x_2 y_2} ... b_{x_T y_T}$$

La probabilità di una tale sequenza di stati $X_1^T = x_1^T$, d'altra parte, è data da

$$\mathrm{P}\left(X_1^T = x_1^T|\lambda\right) = \pi_{x_1} a_{x_1 x_2} a_{x_2 x_3} ... b_{x_{T-1} x_T}$$

La probabilità congiunta di $Y_1^T = y_1^T$ e $X_1^T = x_1^T$, cioè la probabilità che $Y_1^T = y_1^T$ e $X_1^T = x_1^T$ abbiano luogo contemporaneamente, è data semplicemente dal prodotto tra i due termini sopra: $\mathrm{P}\big(Y_1^T = y_1^T, X_1^T = x_1^T|\lambda\big) = \mathrm{P}\big(Y_1^T = y_1^T|X_1^T = x_1^T, \lambda\big)\mathrm{P}\big(X_1^T = x_1^T|\lambda\big)$. La probabilità di $Y_1^T = y_1^T$ si ottiene quindi sommando questa probabilità congiunta su tutte le possibili sequenze di stati:

$$\begin{aligned}\mathrm{P}\left(Y_1^T = y_1^T|\lambda\right) &= \sum_{\forall x_1^T} \mathrm{P}\left(Y_1^T = y_1^T|X_1^T = x_1^T, \lambda\right) \mathrm{P}\left(X_1^T = x_1^T|\lambda\right) \\ &= \sum_{\forall x_1, x_2, ..., x_T} \pi_{x_1} b_{x_1 y_1} a_{x_1 x_2} b_{x_2 y_2} ... a_{x_{T-1} x_T} b_{x_T y_T}\end{aligned}$$

L'interpretazione del calcolo mostrato è la seguente. Inizialmente (all'istante $t = 1$) ci si trova nello stato x_1 con probabilità π_{x_1}, e viene generato il simbolo y_1 con probabilità $b_{x_1 y_1}$. In seguito si compie la transizione allo stato x_2 con probabilità $a_{x_1 x_2}$ e viene generato il simbolo y_2 con probabilità $b_{x_2 y_2}$. Questo processo continua fino a quando avviene l'ultima transizione dallo stato x_{T-1} allo stato x_T con probabilità $a_{x_{T-1} x_T}$ e si genera il simbolo y_T con probabilità $b_{x_T y_T}$. Una breve riflessione dovrebbe convincere il lettore che il calcolo di $\mathrm{P}\big(Y_1^T = y_1^t|\lambda\big)$ attraverso il calcolo diretto appena trattato richiede un numero di calcoli dell'ordine di $2T \cdot N^T$, poiché ad ogni istante $t = 1, 2, ..., T$ ci sono N stati a cui è possibile andare e per ogni addendo della sommatoria sono richiesti circa $2T$ calcoli. (Per la precisione, occorrono $(2T - 1))N^T$ moltiplicazioni e $N^T - 1$ addizioni.) Una richiesta del genere è computazionalmente inattuabile, anche per piccoli valori di N e T; per esempio con $N = 5$ e $T = 100$ si arriva a un ordine di $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ calcoli! Chiaramente è necessaria una procedura più efficiente in grado di risolvere il problema 1. Una tale procedura esiste ed è nota col nome di forward-procedure.

3.2.2 Forward-procedure

In questa sezione verrà analizzata la forward-procedure (letteralmente 'procedura in avanti'), che permette di calcolare la probabilità di una sequenza osservata in modo ricorsivo.

Si consideri la variabile-forward $\alpha_t(x_t)$ definita come:

$$\alpha_t(x_t) \triangleq \mathrm{P}\left(Y_1^t = y_1^t, X_t = x_t|\lambda\right)$$

rappresentante cioè la probabilità della parziale sequenza di osservazioni (fino all'istante t) e dello stato x_t, dato il modello λ. A questo punto si può risolvere il problema attraverso $\alpha_t(x_t)$ in modo induttivo con la seguente procedura:

1. $\alpha_1(x_1) = \pi_{x_1} b_{x_1 y_1} \quad 1 \leq x_1 \leq N$
2. per $t = 1, 2..., T-1 \quad 1 \leq x_{t+1} \leq N$

$$\boxed{\alpha_{t+1}(x_{t+1}) = \left(\sum_{x_t=1}^{N} \alpha_t(x_t) a_{x_t x_{t+1}}\right) b_{x_{t+1} y_{t+1}}}$$

3. infine, $\mathrm{P}\left(Y_1^T = y_1^T|\lambda\right) = \sum_{x_T=1}^{N} \alpha_T(x_T)$.

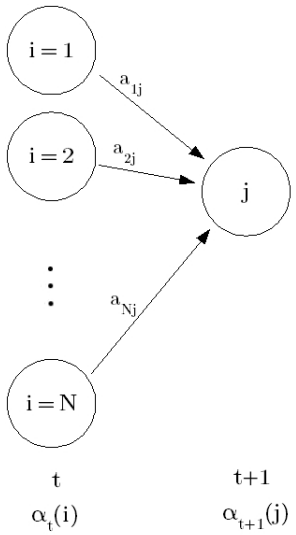

Figura 3.1: Sequenza di operazioni richieste per il computo della variabile-forward $\alpha_{t+1}(j)$.

Il passo 1 da inizio alle probabilità-forward con la probabilità congiunta dello stato $X_1 = x_1$ e dell'osservazione iniziale $Y_1 = y_1$.
Il passo 2 consiste nel tener conto di come lo stato x_{t+1} possa essere raggiunto dagli N possibili stati x_t, $x_t = 1, 2, ..., N$, all'istante t. Poiché $\alpha_t(x_t)$

è la probabilità congiunta di osservare la sequenza $Y_1^t = y_1^t$ e di trovarsi nell'istante t nello stato x_t, allora il prodotto $\alpha_t(x_t)a_{x_t x_{t+1}}$ è la probabilità congiunta di osservare la sequenza $Y_1^t = y_1^t$ e di trovarsi nell'istante $t+1$ nello stato x_{t+1} giungendo dallo stato x_t. Sommando questo prodotto su tutti gli N possibili stati x_t all'istante t, con $x_t = 1, 2, ..., N$, si ottiene la probabilità di trovarsi in x_{t+1} all'istante $t+1$ e, congiuntamente, di aver osservato la parziale sequenza di osservazioni suddetta (probabilità totale). Fatto ciò e noto x_{t+1}, è facile constatare che $\alpha_{t+1}(x_{t+1})$ si ottiene moltiplicando la somma appena vista per la probabilità di osservare $Y_{t+1} = y_{t+1}$ dato lo stato $X_{t+1} = x_{t+1}$, vale a dire $b_{x_{t+1}y_{t+1}}$.
Infine il passo 3 permette di calcolare $\mathrm{P}\big(Y_1^T = y_1^T|\lambda\big)$ attraverso la semplice somma delle variabili-forward finali $\alpha_T(x_T)$. Esso consiste infatti nell'applicazione del teorema della probabilità totale: poiché gli eventi $X_T = 1, X_T = 2, ..., X_T = N$ sono disgiunti e formano una partizione dello spazio campionario, si ha che:

$$\mathrm{P}(Y_1^T = y_1^T|\lambda) = \ = \sum_{i=1}^{N} \mathrm{P}(Y_1^T = y_1^T, X_T = i|\lambda)$$

dove appunto $\mathrm{P}(Y_1^T = y_1^T, X_T = i|\lambda)$ costituisce $\alpha_T(i)$.

Se si vuole esaminare la complessità di computazione richiesta dalla procedura si vede che per il calcolo di $\alpha_t(x_t)$, per $1 \leq t \leq T$ e $1 \leq x_t \leq N$, è richiesto un numero di calcoli dell'ordine di N^2T, contro i $2TN^T$ richiesti dal calcolo diretto. (Per la precisione, occorrono $N(N+1)(T-1)+N$ moltiplicazioni e $N(N-1)(T-1)$ addizioni.) Per $N = 5, T = 100$ occorrono circa 3000 computazioni per la forward-procedure contro i 10^{72} per il calcolo diretto, un risparmio di circa 69 ordini di grandezza!

3.2.3 Backward-procedure

La probabilità di una sequenza osservata può essere calcolata anche mediante la backward-procedure (letteralmente 'procedura all'indietro'), un modo analogo, e per certi versi duale, alla forward-procedure appena vista.

Definiamo la variabile $\beta_t(x_t)$:

$$\beta_t(x_t) \triangleq \mathrm{P}(Y_{t+1}^T = y_{t+1}^T|X_t = x_t, \lambda)$$

rappresentante la probabilità di una sequenza di osservazioni parziale dall'istante $t+1$ fino alla fine, dati lo stato x_t all'istante t e il modello λ. Si può trovare $\beta_t(x_t)$ per ogni istante e per ogni stato in modo induttivo:

1. $\beta_T(x_T) = 1 \quad 1 \leq x_T \leq N$
2. per $t = T-1, T-2, ..., 1 \quad 1 \leq x_{t+1} \leq N$

$$\boxed{\beta_t(x_t) = \sum_{x_{t+1}=1}^{N} a_{x_t x_{t+1}} b_{x_{t+1} y_{t+1}}}$$

3. infine, $\mathrm{P}\big(Y_1^T = y_1^T|\lambda\big) = \sum_{x_1=1}^{N}(\beta_1(x_1)\pi_{x_1})$.

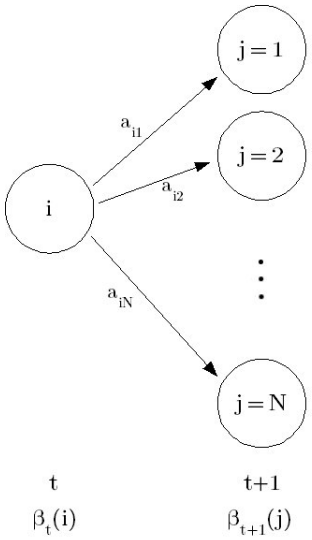

Figura 3.2: Sequenza di operazioni richieste per il computo della variabile-backward $\beta_t(i)$.

Il passo 1 definisce in modo arbitrario $\beta_T(x_T)$ qualsiasi sia x_T.
Il passo 2 mostra che per essere nello stato x_t all'istante t e per tener conto del resto della sequenza di osservazioni si deve compiere una transizione ad ognuno degli N possibili stati all'istante $t+1$, tener conto del simbolo osservato y_{t+1} all'istante $t+1$ e infine considerare il resto della sequenza di osservazioni.
Il passo 3 consiste nell'applicazione del teorema della probabilità totale: poiché gli eventi $X_1 = 1, X_1 = 2, ..., X_1 = N$ sono disgiunti e formano una partizione dello spazio campionario, si ha che:

$$\mathrm{P}(Y_1^T = y_1^T|\lambda) =$$
$$= \mathrm{P}(Y_1^T = y_1^T, X_1 = 1|\lambda) + \mathrm{P}(Y_1^T = y_1^T, X_1 = 2|\lambda) + ... + \mathrm{P}(Y_1^T = y_1^T, X_1 = N|\lambda) =$$
$$= \mathrm{P}(Y_1^T = y_1^T|X_1 = 1, \lambda)\mathrm{P}(X_1 = 1|\lambda) + ... + \mathrm{P}(Y_1^T = y_1^T|X_1 = N, \lambda)\mathrm{P}(X_1 = N|\lambda)$$

dove appunto $\mathrm{P}(Y_1^T = y_1^T | X_1 = i, \lambda)\mathrm{P}(X_1 = i|\lambda)$ costituisce $\beta_1(i)$ e i termini $P(X_1 = i|\lambda)$ sono rappresentati da π_i.

Ancora una volta il calcolo di $\beta_t(x_t)$ per $1 \leq t \leq T$ e $1 \leq x_t \leq N$ richiede un numero di calcoli dell'ordine di N^2T.

3.2.4 Forward-backward procedure

Infine il problema di valutazione può essere risolto con una procedura che utilizza entrambe le variabili forward e backward precedentemente definite.

L'idea è di utilizzare ancora una volta il teorema della probabilità totale, applicato sui valori assunti dalla variabile X_t al generico istante t. Supponiamo dunque che il processo di Markov nascosto assuma nell'istante t il valore x_t. Si ha che:

$$\begin{aligned}
&\mathrm{P}(Y_1 = y_1, Y_2 = y_2, ..., Y_T = y_T, X_t = x_t|\lambda) = \\
&= \mathrm{P}(Y_1^t = y_1^t, Y_{t+1}^T = y_{t+1}^T, X_t = x_t|\lambda) = \\
&= \mathrm{P}((Y_1^t = y_1^t, X_t = x_t), Y_{t+1}^T = y_{t+1}^T|\lambda) = \\
&= \mathrm{P}(Y_1^t = y_1^t, X_t = x_t|\lambda)\mathrm{P}(Y_{t+1}^T = y_{t+1}^T|Y_1^t = y_1^t, X_t = x_t, \lambda) = \\
&= \mathrm{P}(Y_1^t = y_1^t, X_t = x_t|\lambda)\mathrm{P}(Y_{t+1}^T = y_{t+1}^T|X_t = x_t, \lambda)
\end{aligned}$$

dove a questo punto il primo fattore coincide con $\alpha_t(x_t)$ e il secondo con $\beta_t(x_t)$.

Per la probabilità totale:

$$\begin{aligned}
&\mathrm{P}(Y_1^T = y_1^T|\lambda) = \\
&= \sum_{x_t=1}^{N} \mathrm{P}(Y_1^T = y_1^T, X_t = x_t|\lambda) = \\
&= \sum_{x_t=1}^{N} [\alpha_t(x_t)\beta_t(x_t)]
\end{aligned}$$

3.3 Problema di decodifica

Il problema di decodifica consiste nel trovare la più probabile sequenza di stati nascosti: preso un HMM, si vuole determinare da una sequenza di osservazioni la sequenza di stati nascosti sottostante che più probabilmente potrebbe averla generata.

Si può trovare la sequenza più probabile di stati nascosti elencando tutte le possibili sequenze di stati nascosti e trovando la probabilità della sequenza

osservata per ogni combinazione. La sequenza più probabile di stati nascosti $X_1^T = \overline{x_1^T}$ sarà tale che:

$$\overline{x_1^T} = \underset{x_1,x_2,...,x_T}{\arg\max}\ \mathrm{P}(Y_1^T = y_1^T | X_1^T = x_1^T, \lambda)$$

Questa via è praticabile, ma trovare la sequenza più probabile calcolando in modo esaustivo ogni combinazione è computazionalmente molto dispendioso. Come nella forward-procedure, si può sfruttare la tempo invarianza delle probabilità per ridurre la complessità del calcolo.

3.3.1 Riduzione della complessità attraverso la ricorsione

Vediamo ora come trovare in modo ricorsivo la più probabile sequenza di stati nascosti data una sequenza di osservazioni e un HMM.

Lo scopo è di trovare la sequenza di stati ottimale associata ad una data sequenza di osservazioni. Poiché vi sono molti possibili criteri di ottimizzazione, esistono molteplici modi per risolvere tale questione. Uno dei possibili criteri di ottimizzazione è di scegliere gli stati x_t al variare dell'istante t che sono singolarmente i più probabili.

In tale direzione risulta utile definire la variabile $\gamma_t(x_t)$:

$$\gamma_t(x_t) \triangleq \mathrm{P}(X_t = x_t | Y_1^T = y_1^T, \lambda)$$

indicante la probabilità di trovarsi nello stato x_t all'istante t, data l'intera sequenza di osservazioni e il modello λ. Una breve riflessione porta a verificare che $\gamma_t(x_t)$ è banalmente esprimibile in funzione della variabile forward $\alpha_\cdot(\cdot)$ e della variabile backward $\beta_\cdot(\cdot)$ nel seguente modo:

$$\boxed{\gamma_t(x_t) = \frac{\alpha_t(x_t)\beta_t(x_t)}{\mathrm{P}(Y_1^T = y_1^T|\lambda)}}$$

poiché $\alpha_t(x_t)$ tiene conto delle osservazioni $Y_1 = y_1, Y_2 = y_2, ..., Y_t = y_t$ e dello stato $X_t = x_t$, mentre $\beta_t(x_t)$ considera $Y_{t+1} = y_{t+1}, Y_{t+2} = y_{t+2}, ..., Y_T = y_T$ dato lo stato $X_t = x_t$. Il denominatore $\mathrm{P}(Y_1^T = y_1^T|\lambda)$ costituisce semplicemente il fattore di normalizzazione, rendendo $\gamma_t(x_t)$ una probabilità condizionata, soddisfacente quindi $\sum_{x_t=1}^{N} \gamma_t(x_t) = 1$.

Utilizzando $\gamma_t(x_t)$, lo stato individualmente più probabile $\overline{x_t}$ all'istante t é:

$$\boxed{\overline{x_t} = \underset{x_t=1,2,...,N}{\arg\max}\ [\gamma_t(x_t)] \qquad 1 \leq t \leq T}$$

Potrebbero tuttavia esserci dei problemi con il criterio e la soluzione visti sopra. Quando ci sono transizioni non permesse, ovvero $a_{x_t x_{t+1}} = 0$ per qualche x_t e x_{t+1}, la sequenza di stati ottenuta potrebbe essere infatti una sequenza di stati impossibile. La soluzione proposta infatti determina semplicemente lo stato più probabile di volta in volta senza considerare la struttura globale del traliccio, gli stati raggiungibili da uno stato, e la lunghezza della sequenza di osservazioni. Comunque è ancora utile dal momento che nella pratica simili situazioni indesiderabili non sempre avvengono.

Lo svantaggio dell'approccio trattato è quindi la necessità di vincoli globali di qualche tipo sulla sequenza di stati ottimale trovata. Quasi banalmente, un criterio di ottimizzazione di questo tipo è trovare il miglior cammino singolo con la più alta probabilità, ovvero massimizzare $\mathrm{P}(Y_1^T = y_1^T, X_1^T = x_1^T|\lambda)$. Una tecnica formale per trovare questa miglior sequenza di stati singola esiste ed è nota col nome di algoritmo di Viterbi.

3.3.2 Algoritmo di Viterbi

I passi formali nell'algoritmo di Viterbi per trovare la singola sequenza di stati migliore sono i seguenti:

1. Inizializzazione.

$$\delta_1(x_1) = \pi_{x_1} b_{x_1 y_1} \qquad 1 \leq x_1 \leq N$$
$$\psi_1(x_1) = 0$$

2. Ricorsione.

$$\text{Per} \quad 2 \leq t \leq T \quad \text{e} \quad 1 \leq x_t \leq N$$
$$\delta_t(x_t) = \max_{1 \leq x_{t-1} \leq N} [\delta_{t-1}(x_{t-1}) a_{x_{t-1} x_t}] b_{x_t y_t}$$
$$\psi_t(x_t) = \arg\max_{1 \leq x_{t-1} \leq N} [\delta_{t-1}(x_{t-1}) a_{x_{t-1} x_t}]$$

3. Fine.

$$\overline{\mathrm{P}} = \max_{1 \leq x_T \leq N} [\delta_T(x_T)]$$
$$\overline{x_T} = \arg\max_{1 \leq x_T \leq N} [\delta_T(x_T)]$$

4. Cammino all'indietro.

$$\text{Per} \quad t = T-1, T-2, ..., 1$$
$$\overline{x_t} = \psi_{t+1}(\overline{x_{t+1}})$$

L'algoritmo di Viterbi è simile, a parte i passi di cammino all'indietro, alla procedura forward-backward; la differenza più sostanziale riguarda la massimizzazione, non più la somma, sugli stati precedenti. Ancora una volta una struttura a traliccio implementa in modo efficiente la computazione.

3.4 Problema di addestramento

Il terzo problema consiste nell'aggiustare i parametri del modello (A, B, π) al fine di massimizzare $\mathrm{P}(Y_1^T = y_1^T|\lambda)$, la probabilità di una sequenza di osservazioni dato il modello. Questo è senza dubbio il problema più complesso dei tre che abbiamo trattato. Non vi è alcuna via nota per risolvere un problema di massima verosimiglianza in modo analitico. Il risultato è tuttavia ottenibile mediante metodi di discesa del gradiente ottenendo quindi un minimo (massimo) locale e non globale. Molto spesso la computazione del gradiente è piuttosto difficoltosa, per cui la discesa avviene mediante algoritmi approssimati come il *random direction descent* o il *line search*. L'algoritmo più utilizzato, che comunque fornisce massimi locali, è basato su *Expectation-Maximization* o semplicemente *EM*, ed è il Baum-Welch.

Prima di affrontare l'algoritmo di Baum-Welch, definiamo $\xi_t(x_t, x_{t+1})$ come la probabilità di essere nello stato x_t al tempo t e nello stato x_{t+1} al tempo $t+1$, dati modello ed osservazioni; formalmente:

$$\xi_t(x_t, x_{t+1}) \triangleq \mathrm{P}(X_t = x_t, X_{t+1} = x_{t+1}|Y_1^T = y_1^T, \lambda).$$

Possiamo esprimere questa probabilità come:

$$\boxed{\xi_t(x_t, x_{t+1}) = \frac{\alpha_t(x_t) a_{x_t x_{t+1}} b_{x_{t+1} y_{t+1}} \beta_{t+1}(x_{t+1})}{\mathrm{P}(Y_1^T = y_1^T, \lambda)}}$$

dove $\alpha_t(x_t)$ tiene conto delle prime t osservazioni terminando allo stato x_t al tempo t, il termine $a_{x_t x_{t+1}} b_{x_{t+1} y_{t+1}}$ coincide con la transizione allo stato x_{t+1} all'istante $t+1$ e dell'occorrenza del simbolo osservabile y_{t+1}, e $\beta_{t+1}(x_{t+1})$ tiene conto della rimanente parte della sequenza di osservazioni. Il denominatore $\mathrm{P}(Y_1^T = y_1^T|\lambda) = \sum_{x_t} \sum_{x_{t+1}} \alpha_t(x_t) a_{x_t x_{t+1}} b_{x_{t+1} y_{t+1}} \beta_{t+1}(x_{t+1})$ serve per normalizzare $\xi_t(x_t, x_{t+1})$.

Ricordando la definizione di $\gamma_t(x_t)$ come la probabilità di essere nello stato x_t al tempo t, possiamo scrivere $\gamma_t(x_t)$ in funzione di $\xi_t(x_t, x_{t+1})$

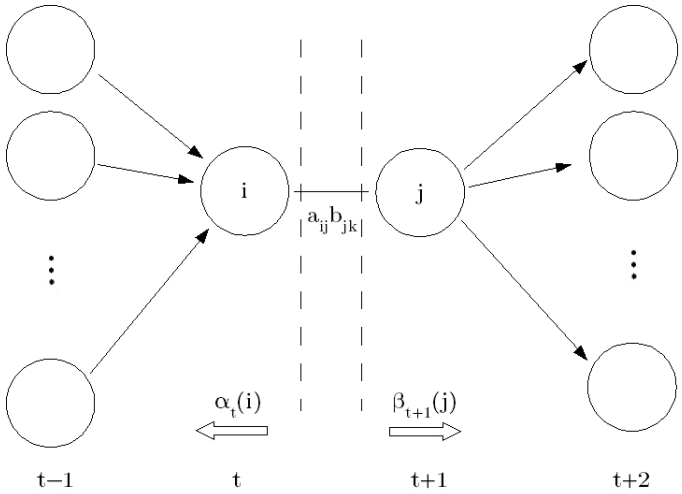

Figura 3.3: Sequenza di operazioni richieste per il computo della variabile $\xi_t(i,j)$.

sommando $\xi_t(x_t, x_{t+1})$ tra tutti i possibili x_{t+1}:

$$\gamma_t(x_t) = \sum_{x_{t+1}=1}^{N} \xi_t(x_t, x_{t+1})$$

Se si somma $\gamma_t(i)$ sopra l'indice di tempo t si ottiene una quantità interpretabile con il numero atteso di volte (nel tempo) che lo stato i viene visitato o, equivalentemente, il numero atteso di transizioni fatte dallo stato i nella somma se si esclude l'ultimo istante, T. Analogamente, sommando $\xi_t(i,j)$ su t (da $t=1$ a $t=T$) si ottiene il numero atteso di transizioni dallo stato i allo stato j. In formule:

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{valore atteso di transizioni fatte da } i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{valore atteso di transizioni dallo stato } i \text{ allo stato } j$$

Utilizzando queste formule e il concetto di conteggio delle occorrenze di un evento si può utilizzare il metodo di Baum-Welch per ottenere una stima migliore dei parametri dell'HMM.

3.4.1 Metodo di Baum-Welch

Le formule per ottenere iterativamente una stima migliore dei parametri $\lambda' = (A', B', \pi')$ partendo da una precedente stima $\lambda = (A, B, \pi)$ sono:

1.
$$\begin{aligned}\pi'_i &= \text{probabilità di essere nello stato } i \text{ all'istante } t = 1 \\ &= \gamma_1(i)\end{aligned}$$

2.
$$\begin{aligned}a'_{ij} &= \frac{\text{valore atteso di transizioni dallo stato } i \text{ allo stato } j}{\text{valore atteso di transizioni dallo stato } i} \\ &= \sum_{t=1}^{T-1} \xi_t(i,j) \Big/ \sum_{t=1}^{T-1} \gamma_t(i)\end{aligned}$$

3.
$$\begin{aligned}b'_{jk} &= \frac{\text{valore atteso di volte nello stato } j \text{ e simbolo osservato } k}{\text{valore atteso di volte nello stato } j} \\ &= \sum_{t=1\ ,\ Y_t=k}^{T} \gamma_t(j) \Big/ \sum_{t=1}^{T} \gamma_t(j)\end{aligned}$$

A questo punto delle due l'una:

- Il modello iniziale $\lambda = (A, B, \pi)$ definisce un punto critico della funzione di verosimiglianza, nel qual caso $\lambda' = \lambda$.
- Il modello λ' è più probabile, nel senso che $\mathrm{P}(Y_1^T = y_1^T|\lambda') > \mathrm{P}(Y_1^T = y_1^T|\lambda)$. In altre parole, il nuovo modello $\lambda' = (A', B', \pi')$ ha con maggior probabilità prodotto la sequenza di osservazioni.

In conclusione, se si usa iterativamente λ' al posto di λ e si esegue una nuova stima, si può migliorare la probabilità delle osservazioni a partire dal modello fino al raggiungimento di un certo punto limite. Il risultato ottenuto sarà la miglior stima del modello.

Capitolo 4

Esempi di applicazione degli HMM

4.1 Riconoscimento vocale

Il primo esempio si rifà al lavoro di Rabiner e Juang [1]: gli HMM vengono utilizzati per costruire un riconoscitore di singole parole. Utilizzando una notazione simile a quella vista nei capitoli precedenti, supponiamo di avere un vocabolario di N parole da riconoscere. A disposizione si hanno un training set di M simboli osservabili per ogni parola (pronunciata da una o più persone) e un test set indipendente.[1] Per poter avere un riconoscimento vocale si eseguono i seguenti punti:

1. Si costruisce un HMM per ogni parola nel vocabolario. Si utilizzano le osservazioni dall'insieme di M simboli per stimare i parametri ottimi per ogni parola, ottenendo così il modello λ^n per l' n-esima parola del vocabolario, $1 \leq n \leq N$.

2. Per ogni parola sconosciuta del test set, caratterizzata da una sequenza di osservazioni $Y_1 = y_1, Y_2 = y_2, ..., Y_T = y_T$, e per ogni modello di parola, λ^n, si calcola $P_n = \mathrm{P}(Y_1^T = y_1^T | \lambda^n)$ secondo la procedura vista nel capitolo 3.

3. Si sceglie la parola il cui modello ha probabilità maggiore:

$$\overline{n} = \arg\max_{1 \leq n \leq N} P_n$$

[1] Il training set è l'insieme di esempi utilizzati per addestrare il sistema, mentre il test set è l'insieme di esempi utilizzati per valutare le prestazioni del sistema.

Struttura della catena di Markov Per il riconoscimento della parola nel caso in cui gli istanti di inizio e fine pronuncia siano approssimativamente noti risulta vantaggioso utilizzare un HMM con matrice di transizione tra gli stati non completa.

Ciò è reso possibile dal fatto che per la pronuncia delle singole parole la natura progressiva della sequenza di stati è abbastanza univoca e il numero di stati necessari per ogni modello di parola è solitamente maneggevole. Se l'obiettivo fosse quello di modellare un segnale vocale corrispondente ad un'intera conversazione tali semplificazioni sul modello potrebbero non essere adeguate.

Facendo corrispondere molto semplicemente gli stati ai suoni presenti nella parola, è chiaro che la natura sequenziale del processo di cambiamento dei suoni pronunciati ritrova una perfetta corrispondenza nella struttura markoviana attraverso cui si cerca di modellarli.

Informazione di durata Poiché gli HMM permettono di eseguire una vera e propria segmentazione (utilizzando la soluzione al problema 2), al fine di riconoscere la parola risulta molto utile l'analisi della permanenza, in termini di tempo, in un certo stato. Essendo approssimativamente noti gli istanti iniziale e finale della pronuncia della parola, per ogni modello di parola tale informazione di durata è spesso rappresentata in una forma normalizzata: $\mathrm{P}_j(I|T)$ equivale alla probabilità di essere nello stato j per esattamente gli I/T della parola, dove T è il numero di frame nella parola e I il numero di frame spesi nello stato j.

4.2 Computer Vision

Oltre che nel campo della *speech recognition* gli HMM risultano molto utili anche in diverse applicazioni di *Computer Vision*. L'approccio basato sugli HMM fornisce un'intrinseca sequenzialità temporale alla rappresentazione del sistema e si è dimostrato efficace, per esempio, per il riconoscimento di modelli spaziali 1- o 2-D: la flessibilità data dall'addestramento permette al modello di apprendere le dinamiche che sottendono effettivamente i dati piuttosto che quelle immaginate dal programmatore.

Una possibile applicazione potrebbe essere il riconoscimento della posizione di un viso posto di fronte ad una webcam, come descritto nell'elabo-

rato di Belardinelli [5], che prende spunto dal lavoro di Rao, Schon, Meltzoff [6].

Segmentazione Il primo problema da affrontare è l'identificazione della zona dei frame contenente il viso da seguire. Per poter individuare in maniera automatica questa zona, isolandola dallo sfondo e da altri elementi presenti sulla scena, si procede alla segmentazione dei frame di interesse: questa operazione separa gruppi di oggetti in partizioni tali che gli elementi all'interno di un cluster siano il più vicini possibile gli uni agli altri e il più lontano possibile dagli elementi degli altri cluster.

Metrica per i movimenti Una volta individuato l'oggetto da seguire ci si pone il problema di introdurre una metrica adatta alla descrizione degli spostamenti, al fine di ottenere dei dati che rappresentino in maniera consistente le dinamiche oggetto dell'apprendimento. Si può quindi scegliere di constatare gli spostamenti relativi dell'oggetto segmentato ad intervalli regolari in ogni video.

Modellazione con HMM A questo punto si possono individuare gli elementi costitutivi di un HMM rispetto al sistema in esame: gli *stati* sono costituiti da un insieme di movimenti della testa discretizzati, per esempio: fermo, destra, destra-in alto, alto, sinistra-in alto, sinistra, sinistra-in basso, in basso, destra-in basso; le *osservazioni* possono essere costituite da una coppia di simboli dell'alfabeto $\{+, -, =\}$ indicanti la variazione rispetto all'asse x e rispetto all'asse y rispetto al frame precedente; la *matrice di transizione* potrebbe essere inizializzata privilegiando lo stato di immobilità e le traslazioni verso destra e verso sinistra, ritenute gli stati più frequenti; la *matrice di output* parimenti può essere inizializzata assegnando, per esempio, una probabilità più alta alla coppia $(-, =)$, corrispondente ad una'avvenuta traslazione verso sinistra, qualora lo stato che l'ha prodotta sia appunto quello corrispondente alla posizione 'a sinistra'; *distribuzione di probabilità a priori degli stati* si può assumere, per esempio, con una componente più elevata per una zona centrale e con componenti più basse per gli angoli.

Addestramento Infine la fase di addestramento consiste in una prima operazione di stima delle matrici delle transizioni e di output, per esempio con il calcolo del Maximum Likelihood delle stesse partendo dalle matrici

viste sopra ed basandosi sulla sequenza di emissioni (derivate dal vettore di spostamenti prodotto dalla segmentazione) e sulla sequenza di stati effettiva (desunti per ispezione visiva dalla sequenza di frame), e, come seconda operazione, la fase di vero addestramento delle matrici in base a un training set che si suppone dato, utilizzando per esempio uno stimatore Maximum Likelihood dei parametri del modello che fa uso dell'algoritmo di Baum-Welch e fornendo come parametri iniziali le matrici trovate con l'operazione di stima.

4.3 Biologia

Gli HMM applicati al *data mining* [2] sono spesso usati nella bioinformatica. Nel seguito prenderemo in considerazione l'attività di riconoscimento e confronto di sequenze proteiche, ben descritto nell'elaborato di Marin [3].

Una proteina è descritta da una sequenza di amminoacidi, che costituiscono un alfabeto di simboli finito. Tuttavia, da una parte non siamo certi che le sequenze di amminoacidi che studiamo non contengano errori, dall'altra accade talvolta che un amminoacido, su due proteine con la medesima funzione, possa prendere il posto di un altro.

Database mining e classificazione Se prendiamo un modello HMM addestrato possiamo utilizzarlo per trovare in un database sequenze simili a quelle utilizzate per l'addestramento: poiché tutte le proteine di una stessa famiglia sono associate ad una sequenza di amminoacidi simile, risulta possibile addestrare un modello su un training set di sequenze note appartenenti ad una stessa famiglia e quindi ricercare in una sequenza di amminoacidi (o un suo frammento) la codifica di proteine appartenenti alla stessa famiglia con un certo grado di probabilità.

Per quanto riguarda la classificazione è possibile addestrare un modello per ogni famiglia di proteine, e quindi associare alla sequenza di amminoacidi una probabilità di appartenenza ad una certa famiglia. In questo caso il problema è quello di disporre di un training set sufficientemente ampio da rendere significativa questa operazione. Si supponga di avere N classi di oggetti all'interno di un database, con N incognito. Si ha una rappresentanza sufficiente come training set per M classi di oggetti; a questo punto si

[2] Per data mining si intende il processo di estrazione di un sapere o di una conoscenza a partire da grandi quantità di dati (attraverso metodi automatici o semi-automatici) e l'utilizzazione industriale o operativa di questo sapere.

effettua l'addestramento di M modelli, ciascuno con il training set associato ad una delle classi note. Risulta ora possibile sia classificare le sequenze di amminoacidi presenti nel database in base agli M modelli noti, sia raggruppare le rimanenti sequenze non classificate in base agli score (normalizzati) ottenuti; l'idea è che se due sequenze di amminoacidi ottengono punteggi simili sugli stessi modelli, allora è probabile che appartengano ad una stessa famiglia. Naturalmente maggiore è il numero M di classi noti nel training set, migliori saranno i risultati derivanti dall'applicazione di questa tecnica.

Vediamo ora due esempi specifici.

HMM per il DNA Il primo esempio è tratto dal lavoro di Churcill [7]: il DNA è una sequenza di basi che possono essere etichettate con le lettere A,C,G,T. Per alcune analisi di sequenziamento si è osservato che i segmenti alternano sezioni in cui sono presenti prevalentemente la basi A-T (stato 1) e sezioni in cui sono presenti prevalentemente la basi G-C (stato 2). Le uscite sono dunque sequenze di singole basi azotate, mentre ogni stato è un insieme di basi azotate in sequenza. Il problema consiste nel capire quando ci si trova, analizzata una sequenza, nello stato 1 o 2. Churchill ha sviluppato sulla base di osservazioni sperimentali, l'HMM di figura 4.1, costruito sulla base di addestramento ed informazioni biologiche.

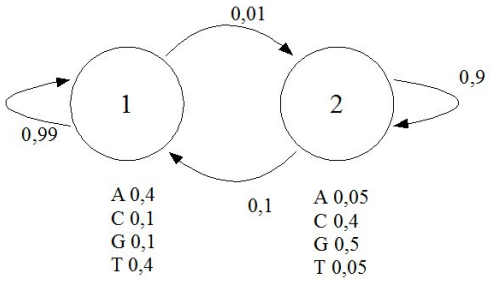

Figura 4.1: Catena di Markov presente nell'HMM per il riconoscimento del DNA di Churchill.

In questo caso l'HMM è stato utile perchè ha permesso di intrecciare facilmente le informazioni ottenute dall'addestramento a quelle note per altri studi propri della biologia.

Gestione di inserzioni o cancellazioni Il secondo esempio evidenza come gli HMM diventino utili in quei casi in cui si vuole avere uno stret-

to controllo sulle penalità da assegnare alle inserzioni o alle cancellazioni nel riconoscimento della sequenza. Intuitivamente con gli HMM riusciamo a trattare variazioni diverse sui pattern in modo diverso, e questo li rende estremamente flessibili. Una proteina è codificata da amminoacidi che costituiscono un alfabeto di venti simboli: per ragioni biologiche, ma anche per errori nel campionamento, due proteine con medesima funzione possono differire nella loro sequenza per:

- sostituzione di un amminoacido con uno simile (tabelle di sostituzione danno il grado di similarità degli amminoacidi);
- inserzione di un amminoacido;
- eliminazione di un amminoacido.

Naturalmente più dissimilarità presentano due sequenze, meno probabilità vi è che appartengano alla stessa famiglia di proteine. Una tecnica consiste nel definire per ogni Matching anche uno stato di cancellazione e uno di inserzione. Per ogni transizione verso questi stati si definiscono probabilità basse; inoltre per gli stati di Matching e di inserzione si definiscono le probabilità di emissione di amminoacidi per quella famiglia di proteine. Per ogni prolungamento della sequenza da riconoscere si aggiungono al modello 49 parametri (20 probabilità di emissione per lo stato M, 20 per lo stato D, più 9 probabilità di transizione).

Questo modello è molto flessibile: ad esempio è noto che le inserzioni all'inizio della sequenza sono meno gravi di quelle centrali (potrebbe trattarsi di un problema sperimentale di allineamento), e questo può essere modellato evitando di penalizzare con basse probabilità le transizioni verso lo stato 0.

Capitolo 5

Conclusioni

I principali vantaggi dell'impiego degli HMM in task relativi al data mining sono legati da una parte al robusto impianto statistico-teorico che sostiene gli algoritmi forniti, dall'altro alla relativa efficienza degli stessi. Inoltre questi modelli si rivelano utili in molte applicazioni poiché:

- effettuano matching di sequenze che differiscono per qualche inserimento o cancellazione, con valutazione della relativa penalità;
- trattano agevolmente sequenze di lunghezza variabile;
- possono svolgere compiti di allineamento, data mining/classificazione, analisi strutturale e pattern discovery.

I principali limiti nell'applicazione degli HMM sono essenzialmente due. In primo luogo i parametri da stimare crescono velocemente con il numero di stati del modello iniziale, e con l'alfabeto in input. Questo oltre a portare problemi relativi al tempo di calcolo, introduce anche serie difficoltà negli algoritmi di training dovute al moltiplicarsi di massimi locali. Il secondo problema è che le probabilità di transizione non dipendono dalla funzione di emissione, ma solo dallo stato nascosto. Una classica correlazione che non può essere colta dagli HMM è la seguente: supponiamo che lo stato $X_t = i$ nel caso in cui emetta $Y_t = j$ sia sovente seguito dallo stato $X_{t+1} = k$ con emissione $Y_{t+1} = l$ e che, analogamente per altre due coppie stato-uscita, lo stato $X_{t'} = i'$ nel caso in cui emetta $Y_{t'} = j'$ sia sovente seguito dallo stato $X_{t'+1} = k'$ con emissione $Y_{t'+1} = l'$: purtroppo questo chiaramente esula dalle capacità rappresentative degli HMM fin qui approfonditi, in quanto viene contro l' assunzione fatta nel capitolo 2 a pagina 10 di dipendenza di

un'uscita dal solo stato attuale. Ciò impedisce di scrivere $m_{ij}^{(u)}$ attraverso il prodotto $a_{ij}b_{ju}$; tuttavia adottando le probabilità $m_{ij}^{(u)}$ l'idea base degli algoritmi di risoluzione dei vari problemi rimane ancora valida.

Bibliografia

[1] L.R. Rabiner e B.H. Juang, *An introduction to hidden Markov models*, IEEE ASSp Mag., vol. 3, n.1, pp. 4-16, 1986.

[2] M. Vidyasagar, *The Complete Realization Problem for Hidden Markov Models: A Survey and Some New Results*, pp. 10-14, 2009.

[3] A. Marin, *Hidden Markov Models e applicazioni al Data Mining*, pp. 4-11, 2006, Università Ca' Foscari di Venezia.

[4] S.J. Russell, P. Norvig, *Intelligenza artificiale. Un approccio moderno.*, seconda edizione Pearson, pp. 173-191, 2005.

[5] A. Belardinelli, *Apprendimento di movimenti della testa tramite Hidden Markov Model*, pp. 80-90, 2006, Università degli studi di Roma 'La Sapienza'.

[6] R.P.N. Rao, A.P. Shon, A.N. Meltzoff, *A Bayesian Model of Imitation in Infants and Robots*, in 'Imitation in Robots, Humans, and Animals: Behavoiural, Social and Communicative Dimensions', K.Dautenham and C.Nehaniv (eds), Cambridge University Press, 2004.

[7] G.A. Churchill, *Stochastic models for heterogeneus dna sequences*, Bull Math Biol 51, pp. 79-94, 1989.

Ringraziamenti

Desidero porgere i miei ringraziamenti al prof. Lorenzo Finesso per il tempo dedicatomi e per l'interesse che ha saputo trasmettermi per questa materia.

Dato che terminato il triennio ognuno prenderà una diversa strada, voglio ringraziare Davide, Colla e Frack per gli indimenticabili momenti passati insieme in questi anni. Questa prima parte del cammino universitario sarebbe stata per me molto più dura (e noiosa) se non avessi avuto il loro aiuto e la loro amicizia.

Infine ringrazio la mia famiglia: i miei genitori per essermi stati sempre vicini e mio fratello Andrea per il punto di riferimento che mi offre.

Marco Ruzza