

ALLINEAMENTO DI SEQUENZE BIOLOGICHE TRAMITE MODELLI DI MARKOV NASCOSTI

Andrea Fedeli, Sauro Menchetti
Università di Firenze
Facoltà di Ingegneria
anfed@ats.it

1. INTRODUZIONE

La bioinformatica sta emergendo come una disciplina alla frontiera tra la biologia e l'informatica: in questo ambiente, caratterizzato dalla presenza di grandi quantità di dati rumorosi e dall'assenza di teorie generali per la loro analisi, l'idea fondamentale è di *imparare la teoria automaticamente dai dati*. Infatti, la maggior parte dei dati biologici riguardanti gli acidi nucleici e le proteine sono sotto forma di sequenze biologiche. Una caratteristica fondamentale di tali sequenze è che possono essere descritte come *stringhe di simboli* scelti da un *alfabeto* di piccole dimensioni. Il rumore presente in ambito biologico si ripercuote in questo modello e gli algoritmi di addestramento ne devono tenere conto. Pertanto, tra i vari paradigmi di apprendimento, l'ambiente di lavoro probabilistico appare essere il più promettente per questa tipologia di dati fornendo una solida base teorica ai metodi di apprendimento automatico. In questo contributo sarà presentata una tecnica basata su un ambiente bayesiano per l'allineamento delle sequenze.

2. L'ALLINEAMENTO

L'allineamento è un metodo per riuscire a trovare caratteristiche funzionali o strutturali comuni ad un insieme di sequenze biologiche: lo scopo è di determinare quando le similarità sono sufficientemente grandi, in modo da inferire una certa somiglianza strutturale o funzionale delle sequenze. La misura di quanto due sequenze sono allineate non è la stessa attraverso l'intero dominio delle sequenze e le corrispondenze prodotte dagli algoritmi di allineamento dipendono da vari fattori. I classici algoritmi di allineamento sono basati sulla

programmazione dinamica [2,3]: la complessità di tale tecnica, detto K il numero di sequenze da allineare ed N la lunghezza media delle sequenze, è dell'ordine di $O(N^K)$. Gli algoritmi qui illustrati hanno invece una complessità di $O(KN^2)$. La programmazione dinamica utilizza una matrice di sostituzione per misurare l'allineamento delle sequenze che stabilisce un insieme di costi s_{ij} per la sostituzione di una lettera dell'alfabeto con un'altra. Date quindi due sequenze X_1, \dots, X_N e Y_1, \dots, Y_M che si sono evolute nel tempo, l'allineamento consiste nel minimizzare un certo costo indotto dalla matrice di sostituzione. Saranno qui illustrate altre tecniche per misurare l'allineamento.

3. IL MODELLO DI MARKOV

Un modello di Markov nascosto o HMM [1,2] discreto del 1° ordine è definito da:

- un insieme finito di stati S
- un alfabeto discreto di simboli A
- una matrice delle probabilità di transizione $T = (P(i | j))$
- una matrice delle probabilità di emissione $E = (P(X | i))$

dove $P(i | j)$ indica la probabilità di transizione dal nodo j verso il nodo i e $P(X | i)$ indica la probabilità che il nodo i emetta il simbolo X . La scelta di un'architettura per l'HMM dipende fortemente dal problema. Nel caso di sequenze biologiche, l'aspetto lineare delle sequenze è ben rappresentato dalla cosiddetta architettura left-right [1].

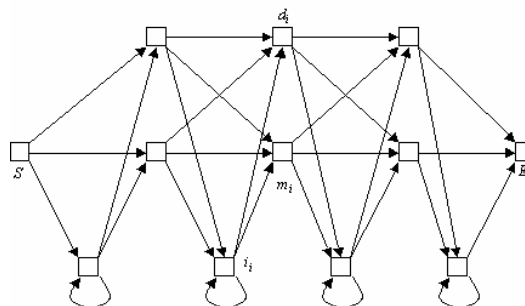


Figura 1: L'architettura lineare standard

Oltre ai due stati *Start* ed *End*, ci sono altri tre tipi di stati chiamati rispettivamente *main*, *insert* e *delete*: i nodi *main* ed *insert* emettono un simbolo, mentre i *delete* sono muti.

3.1. ALGORITMI FONDAMENTALI

La verosimiglianza per una sequenza $S = S_1, \dots, S_t, \dots, S_T$ rispetto ad un certo HMM $M = M(w)$ dove w è l'insieme dei parametri si indicherà con $P(S | w)$. La probabilità di trovare un percorso le cui emissioni coincidano con quelle della sequenza osservata è data da:

$$P(S, \pi | w) = \prod_{Start}^{End} P(j | i) \prod_{t=1}^T P(S_t | i)$$

La verosimiglianza può essere calcolata come:

$$P(S | w) = \sum_{\pi} P(S, \pi | w)$$

Questa espressione non costituisce un metodo di calcolo computazionalmente efficiente per calcolare la verosimiglianza, in quanto il numero di percorsi in un'architettura è tipicamente esponenziale.

3.1.1. Algoritmi Forward e Backward

Definiamo innanzitutto la probabilità congiunta $\alpha(i, t)$ di trovarsi nello stato i avendo osservato i primi $t - 1$ simboli della sequenza, dati i parametri del modello, nel seguente modo:

$$\alpha(i, t) = P(N_t = i, S_1, \dots, S_{t-1} | w)$$

dove $N_t = i$ indica l'evento di trovarsi nel nodo i al tempo t , mentre S_1, \dots, S_{t-1} indicano i simboli della sequenza al tempo $1, \dots, t - 1$. Si può osservare che:

$$P(S | w) = \alpha(End, T + 1)$$

Il generico elemento $\alpha(i, t)$ può essere calcolato ricorsivamente nel seguente modo, distinguendo i nodi di emissione da quelli di cancellazione:

$$\sum_{j \in G(i)} \alpha(j, t) P(i | j) \quad \text{se } i \in D$$

$$\sum_{j \in G(i)} \alpha(j, t-1) P(i | j) P(S_t | i) \quad \text{se } i \in E$$

dove E indica l'insieme di tutti i nodi di emissione, D quello dei nodi di cancellazione, $G(i)$ l'insieme dei genitori del nodo i . Definiamo adesso la probabilità $\beta(i, t)$ di aver osservato i simboli della sequenza da S_t a S_T dato lo stato

i e dati i parametri del modello, nel seguente modo:

$$\beta(i, t) = P(S_t, \dots, S_T | N_t = i, w)$$

Il generico elemento $\beta(i, t)$ può essere calcolato ricorsivamente nel seguente modo:

$$\sum_{j \in F(i) \cap E} \beta(j, t+1) P(j | i) P(S_t | j) + \sum_{j \in F(i) \cap D} \beta(j, t) P(j | i)$$

dove $F(i)$ indica l'insieme dei figli del nodo i . La complessità di tali algoritmi, detta T la lunghezza della sequenza corrente e supponendo che la lunghezza media delle sequenze sia uguale alla lunghezza del modello, è $O(N^2)$ in tempo e spazio per ogni sequenza.

3.1.2. Utilizzo degli alfa e dei beta

Noti tutti gli $\alpha(i, t)$ e tutti i $\beta(i, t)$, possiamo calcolare la probabilità $\gamma(i, t) = P(N_t = i | S)$ di trovarsi in un certo nodo i al tempo t , data una sequenza S . È utile anche calcolare la probabilità $\gamma(j, i, t)$ della transizione dal nodo i al nodo j al tempo t . Utilizzando questi coefficienti, possiamo calcolare alcuni parametri utili per gli algoritmi di apprendimento:

1. la probabilità $f(i) = P(i | S)$ che la sequenza passi attraverso il nodo i : tale probabilità si ottiene marginalizzando $\gamma(i, t)$ su t ;
2. la probabilità $f(i, X) = P(i, X | S)$ che un simbolo X sia emesso dal nodo i (se i è un nodo di emissione) data la sequenza S ;
3. la probabilità $f(j, i) = P(N_t = j, N_{t-1} = i | S)$ che la sequenza attraversi un arco: tale probabilità si ottiene marginalizzando $\gamma(j, i, t)$ su t .

3.1.3. Calcolo del percorso più probabile e dell'allineamento

Il percorso più probabile di una sequenza S lungo il grafo viene individuato cercando qual è il nodo più probabile ad ogni istante di tempo t . Tale nodo sarà sempre un nodo di emissione e mai un nodo di cancellazione. Un modo per calcolare quale sia il nodo più probabile al tempo t è quello di scegliere quello che massimizza $\gamma(i, t)$, fissato t . Un altro metodo è quello di utilizzare l'algoritmo di Viterbi. Iniziamo col definire le seguenti variabili:

$$\delta(i, t) = \max_{\pi(i,t)} P(\pi(i, t) | w)$$

dove $\pi(i, t)$ rappresenta la parte iniziale di una sequenza S_1, \dots, S_t che termina nello stato i . Si ha che $\delta(i, t)$ rappresenta la probabilità associata al percorso più probabile dei primi t simboli della sequenza S che termina nello stato i . Queste variabili possono essere aggiornate usando un meccanismo di propagazione simile all'algoritmo forward, dove le sommatorie sono sostituite con delle massimizzazioni. Il percorso più probabile trovato in precedenza non è un vero e proprio "percorso". Si può allora procedere nel seguente modo: si scandiscono tutti i nodi al tempo t tenendo traccia del più probabile per il quale esiste un cammino con il nodo al tempo $t - 1$. A questo punto, possiamo calcolare l'allineamento che consiste soltanto nell'evidenziare i simboli emessi da nodi di tipo main.

3.2. ALGORITMI DI APPRENDIMENTO

Gli algoritmi di addestramento implementati per l'addestramento dell'HMM sono:

- l'algoritmo di discesa lungo il gradiente;
- l'algoritmo EM;
- l'algoritmo di Viterbi.

Nel caso in cui le sequenze di allineamento siano più di una, queste possono essere considerate indipendenti e la verosimiglianza totale è data dal prodotto delle verosimiglianze di tutte le sequenze.

3.2.1. Discesa lungo il gradiente

Una possibile realizzazione dell'algoritmo di discesa lungo il gradiente può essere quella di massimizzare la verosimiglianza. È utile introdurre una riparametrizzazione del modello, dove w_{iX} e w_{ji} sono le nuove variabili [1]. Le formule di aggiornamento dell'algoritmo di discesa lungo il gradiente sono quindi le seguenti:

$$\Delta w_{iX} = \eta (f(i, X) - f(i)P(X|i))$$

$$\Delta w_{ji} = \eta (f(j, i) - f(i)P(j|i))$$

dove η indica il tasso di apprendimento o learning rate. La complessità di questo algoritmo è dell'ordine di $O(N)$. Concludendo,

l'addestramento tramite l'algoritmo di discesa lungo il gradiente di M sequenze richiede un tempo dell'ordine di $O(MN^2)$, lineare quindi nel numero delle sequenze.

3.2.2. EM

Questo algoritmo si propone di adattare il modello all'insieme di sequenze considerato, massimizzando una certa funzione di energia. È un algoritmo di tipo batch in quanto esegue l'aggiornamento ad ogni epoca, dopo aver osservato tutte le sequenze. Il passo di aggiornamento dell'algoritmo comporta la sostituzione di certe probabilità con la loro stima ottenuta massimizzando l'energia libera [1] del modello:

$$P(X|i) = f(X, i) / f(i)$$

$$P(j|i) = f(j, i) / f(i)$$

dove X è un simbolo della sequenza. La complessità dell'algoritmo EM è la stessa di quello di discesa lungo il gradiente.

3.2.3. Viterbi

Il precedente algoritmo di discesa lungo il gradiente utilizza tutti i percorsi relativi ad una sequenza per l'aggiornamento. L'idea generale dell'algoritmo di Viterbi è sostituire i calcoli relativi a tutti i percorsi con altri che riguardano solo un piccolo numero di percorsi, tipicamente uno solo [1]. In questa versione dell'algoritmo, si ignorano tutti i percorsi tranne il più probabile: viene dunque sfruttata solo parte dell'informazione disponibile. Apportando le opportune modifiche, l'algoritmo di aggiornamento è lo stesso del gradiente, ma dovremmo verificare un peggiore allineamento.

4. RISULTATI E TEST

Per poter misurare quanto è allineato un insieme di sequenze, definiamo innanzitutto delle metriche che quantifichino il concetto di allineamento. Confronteremo poi i vari algoritmi di apprendimento su un insieme di sequenze di proteine, con particolare riguardo al problema della generalizzazione.

4.1. MISURE DEGLI ALLINEAMENTI

È interessante chiedersi se non si possa utilizzare una misura o un insieme di misure che prescindano dalla natura delle sequenze trattate: d'altra parte chiunque ha un'idea di allineamento e quindi vogliamo capire come tradurre in numero questo concetto. Per questo, abbiamo introdotto alcuni criteri per pesare le caratteristiche risultanti dall'allineamento e definito la misura come la media della somma di tali pesi.

Misura A. Una prima possibilità è di contare in ogni colonna come si ripartiscono in percentuale i vari simboli dell'alfabeto. In questo modo possiamo dividere in due classi le colonne prodotte dell'allineamento: quelle allineate e quelle non allineate, dove si considerano allineate tutte quelle colonne in cui esiste un simbolo X che appare con una percentuale molto elevata, per esempio il 95%. Questo perché alla comune esperienza appare ragionevole chiamare 'allineata' una colonna in cui compare sempre lo stesso simbolo. In genere si dirà che una colonna è allineata al tot% se esiste un simbolo che compare almeno con quella frequenza.

Misura B. La precedente suddivisione è tuttavia abbastanza drastica: si potrebbe pensare che esistano casi in cui questo numero è sempre 0 perché le colonne sono allineate al 94%. Perciò appare naturale definire una seconda misura che tenga conto di questa sfumatura. Perciò è utile considerare un criterio che a colonne meno allineate faccia corrispondere pesi più piccoli fino ad una certa soglia minima di allineamento sotto la quale il peso è nullo. Nei nostri esperimenti abbiamo adottato come soglia minima l'80% ed assegnato i seguenti valori: 0,25 se la colonna è allineata ad un valore compreso tra 80% e 90%; 0,5 se è compreso tra 90% e 95%; 1 se è compreso tra 95% e 100%. Sotto lo 80% possiamo ragionevolmente affermare che la colonna non è affatto allineata in quanto su 5 simboli ce ne è uno differente.

Misura C. Un'altra possibilità per misurare l'allineamento è di fare riferimento ad una sequenza reale se nota oppure ad una verosimile, così definita rispetto ad un insieme di se

quenze allineate in colonne: il simbolo i -esimo della sequenza più probabile è il simbolo più probabile della colonna i -esima (non va confuso questo con il percorso più probabile). A questo punto, avendo un parametro di confronto, definiamo la seguente misura: ogni simbolo della colonna i -esima contribuisce con 1 se è differente dal simbolo omologo della sequenza più probabile, con 0 altrimenti. Quindi se tutte le sequenze sono uguali, questa misura vale 0, se le sequenze sono casuali sarà massima. In particolare la media di questo valore, se i simboli sono M , tutti equiprobabili, su K sequenze di lunghezza N sarà con probabilità 1, quando K tende a infinito, $(1-1/M)N$, ovvero dividendo per il numero di colonne in modo da avere un valore normalizzato $(1-1/M)$.

Misura D. Questa misura tuttavia non fa differenza tra colonne allineate e non allineate: infatti non dovrebbe avere molta influenza il fatto che sia diverso un simbolo in una colonna allineata solo al 20%. Quindi si potrebbe procedere ad una misura simile alla precedente che tuttavia pesi l'allineamento delle colonne utilizzando un criterio analogo a quello della misura B. In particolare, ogni simbolo contribuisce alla misura con 8 se la colonna è allineata tra il 97% ed il 100%; 4 se lo è tra il 95% ed il 97%; 2 se lo è tra il 90% ed il 95%; 1 se lo è tra il 90% e lo 80%; 0,5 se lo è tra lo 80% ed il 60%. Questa misura vale 0 sia per le colonne perfettamente allineate sia per quelle che non lo sono affatto: questo può fornire una utile indicazione. Infatti questa misura man mano che l'algoritmo produce allineamenti migliori tenderà prima a salire, in quanto cominciano ad esserci alcune colonne allineate, poi a diminuire. Si dovrà prestare attenzione al fatto che questa diminuzione non indichi un overfitting.

Per ognuna di queste misure è inoltre interessante considerare, oltre al valore medio, anche il valore massimo ed il numero di sequenze comprese tra la $(\text{media} + \text{massimo}) / 2$ ed il massimo che rappresentano il numero di sequenze che non sono state allineate, qualora la differenza tra massimo e media sia rilevante; altrimenti è un numero abbastanza casuale. È

importante osservare come nessuna di queste misure possa funzionare da sola:

- il numero di colonne allineate, anche pesato, non ci dice nulla sull'allineamento delle singole sequenze;
- il numero di simboli differenti dalla sequenza più probabile non ci dice nulla su quanto sia lo scostamento dalle colonne allineate;
- la misura dello scostamento dalle colonne allineate, in quanto vale 0 sia per sequenze perfettamente allineate sia per sequenze poco allineate non ci dice se quel numero è piccolo perché ci sono poche colonne allineate o perché l'allineamento è buono.

4.2. ESEMPIO DI ALLINEAMENTO

```

AAACTGTTGGGCCCC      A A A CTgTTG G G C CC C
AAACTTTGTTGGGCCCC    A A A CT TTGttG G C CC C
AAACTTTGGGCCACC      A A A CT TTG G G C CA Cc
AACTTTGGGCCCC        A Ac T TTG G G C CC C
AAACTTTGTTGGGCCCC    A A A CT TTGttG GGC CC C
AAACTTTGAAGCCACC     A A A CT TTG GaaG C CA Cc
AATGACTTTGGGCCCC     A AtgA CT TTG G G C CC C
AAACTTTGGGCCCC       A A A CT TTG G G C CC C
AAACTTTGACGGCCCC     A A A CT TTGacG G C CC C
AAACTTTGGGCCCC       A Ac A CT TTG G G C CC C
AACTTTGGCCCC         A Ac T TTG G C CC C
AAACTTTGGGCGGCC      A A A CT TTG G G Cggcc C
AGAACTTTGGGCCCC      AgA A CT TTG G G C CC C
AAACTTTGGGCCATGCC    A A A CT TTG G G C CATgcc
AAACTATTGGGCCCC      A A A CTaTTG G G C CC C
AAACTTTGGGCCCC       A A A CT TTG G G C CC C
AAAGCTTTGGGCCCC      A A AgCT TTG G G C CC C

AAACTTTGGGCCCC      A A A CT TTG G G C CC C
AAACTTTGGGCCCC      A A A CT TTG G G C CC C
AAACTTTGGGCCCC      A A AtCT T G G G C CC C
AAACTTTGGGCCCC      A A A CT TTG G G C CC C

```

Le sequenze, anche se utilizzano l'alfabeto del DNA, non sono biologiche. L'esempio è stato costruito ad hoc per testare gli algoritmi illustrati e si può notare come sia una variazione di un tema comune: vediamo che il processo di allineamento ha diviso la parte comune dal "rumore".

Osserviamo il comportamento delle misure introdotte su questo particolare insieme, considerando insieme addestrati con una parte dell'insieme totale delle sequenze.

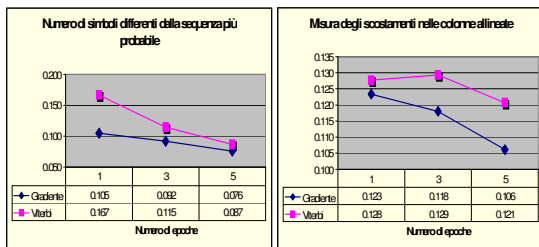


Figura 2: Andamento delle misure C e D

4.3. TEST SULLE PROTEINE

Sono stati effettuati dei test per misurare la bontà degli allineamenti, secondo le misure appena definite. Gli addestramenti impiegano solo una parte, da 1/3 a 2/3, dell'insieme di sequenze in modo da poter verificare la capacità di generalizzazione in quanto l'allineamento poi è stato sempre calcolato su tutte le sequenze. In questo modo, si esegue una sorta di validazione incrociata. Abbiamo utilizzato un file di 150 immunoglobuline di diversa provenienza biologica: queste possiedono, per motivi chimici e fisici, almeno due colonne che devono allinearsi.

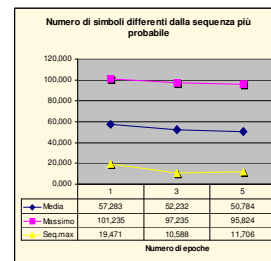


Figura 3: Andamento della misura C

Notiamo che nel complesso c'è una leggera diminuzione di questa misura sia in media sia nel suo valore massimo.

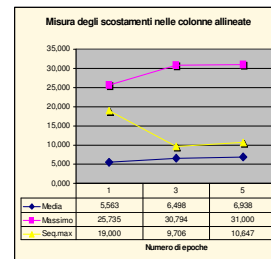


Figura 4: Andamento della misura D

Notiamo che la misura cresce tendendo a stabilizzarsi dopo 5 epoche nella media: questo potrebbe essere il valore massimo.

Quindi d'ora innanzi la misura può solo migliorare o andare in sovraddestramento. In questo caso, dalla conoscenza delle proprietà biologiche, sappiamo che si avrà un sovraddestramento.

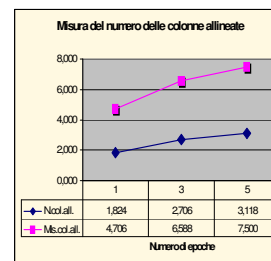


Figura 5: Andamento delle misure A e B

Il grafico è concorde con gli altri: questo sottolinea la coerenza delle misure adottate e la loro forte correlazione.

Si può osservare che i risultati sono omologhi a quelli ottenuti sulle sequenze costruite ad hoc con l'alfabeto del DNA. Questo è una conferma della bontà delle misure.

A questo punto è utile considerare i risultati differenziati per algoritmo di allineamento.

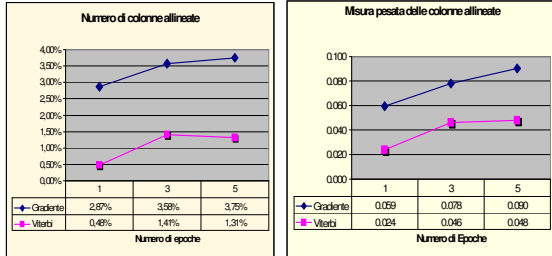


Figura 6: Andamento delle misure A e B

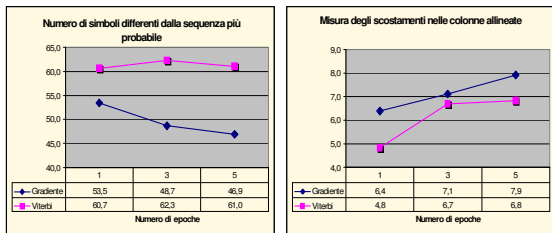


Figura 7: Andamento delle misure C e D

Notiamo che l'algoritmo di Viterbi per l'allineamento ha un comportamento anomalo, mentre quello del gradiente è molto più netto nei suoi andamenti. Questo è indicativo di come il primo riesca ad estrarre in modo peggiore informazioni dal modello poco addestrato, mentre ci riesce molto meglio il secondo.

5. CONCLUSIONI

Un ultimo confronto interessante è tra i vari algoritmi di addestramento. I seguenti grafici rendono l'idea del funzionamento dei tre algoritmi:

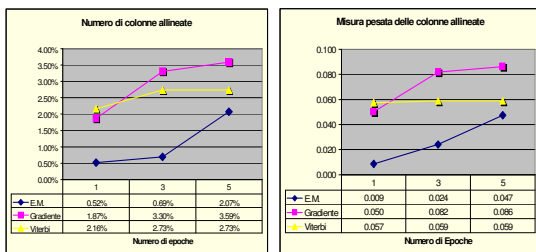


Figura 8: Andamento delle misure A e B

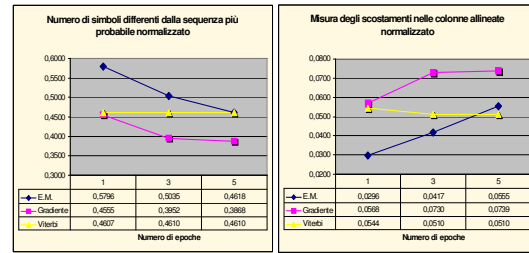


Figura 9: Andamento delle misure C e D

- L'algoritmo del gradiente sfrutta tutte le informazioni presenti per l'aggiornamento: è indubbiamente il migliore, come si evidenzia in tutte le misure.
- L'algoritmo di Viterbi ha un comportamento singolare: sembra indipendente dal numero di epoche e stabilizzato su valori inaccettabili: questo è dovuto ad irregolarità nei test. In accordo con la teoria, si verifica che non è un buon algoritmo per l'aggiornamento ed ha una scarsa capacità di generalizzare.
- L'algoritmo EM ha una non buona velocità di convergenza su poche epoche, mentre su un numero alto di epoche tende ad avviarsi verso gli stessi valori del gradiente. In realtà su un numero alto di epoche è ancora più veloce, ma tende ad adattarsi troppo all'insieme di addestramento e ad andare in sovraddestramento facilmente.

Questi risultati sono stati ottenuti attraverso un simulatore HMM da noi interamente sviluppato in C++ [3]. Infine, confrontando gli allineamenti ottenuti dal nostro simulatore con quelli prodotti da HMMpro 2.2 sviluppato dalla NetID Inc., abbiamo rilevato solo alcune differenze dovute alla non perfetta uguaglianza dei parametri e degli algoritmi.

BIBLIOGRAFIA

- [1] P.Baldi S.Brunak "Bioinformatics, The Machine Learning Approach" A Bradford Book, The Mit Press - Cambridge, Massachusetts London, England
- [2] S.J.Russel P.Norvig "Artificial Intelligene. A Modern Approach" Prentice Hall International
- [3] G.Ausiello e altri "Teoria e progetto di algoritmi fondamentali" Franco Angeli