Some open issues/research trends

1. Model selection
   - how many states?
   - which topology?
2. Extending standard models
   - modifying dependencies or components
3. Injecting discriminative skills into HMM

48

Some open issues/research trends

1. Model selection
   - how many states?
   - which topology?
2. Extending standard models
   - modifying dependencies or components
3. Injecting discriminative skills into HMM

49

Model selection

- The problem of determining the HMM structure:
  - not a new problem, but still a not completely solved issue

1. Choosing the number of states: the "standard" model selection problem
2. Choosing the topology: forcing the absence or the presence of connections

50

Model selection problem 1: selecting the number of states

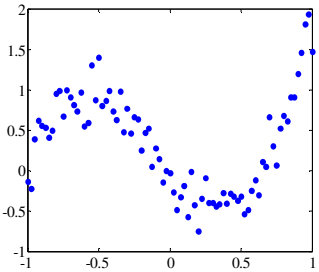
- Number of states: prevents overtraining
- The problem could be addressed using standard model selection approaches

...let's understand the concept with a toy example

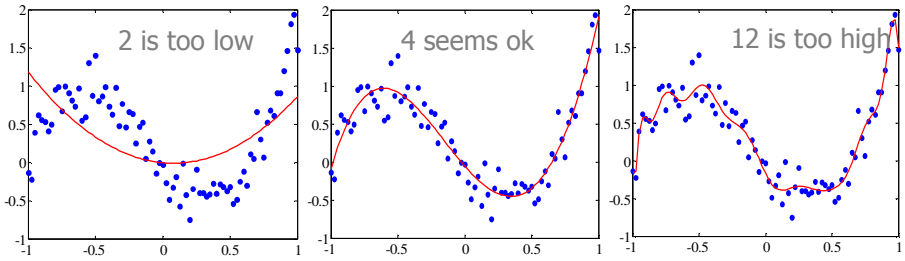
51

What is model selection?

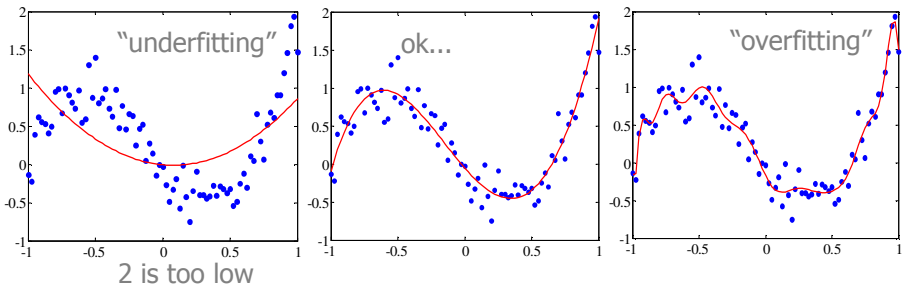
Toy example: some experimental data to which we want to fit a polynomial.



The model selection question is**:** which order?



What is model selection?



Model selection goal:

how to identify the underlying trend of the data, ignoring the noise?

53

Model selection: solutions

- Typical solution (usable for many probabilistic models)
  - train several models with different orders k
  - choose the one maximizing an “optimality” criterion

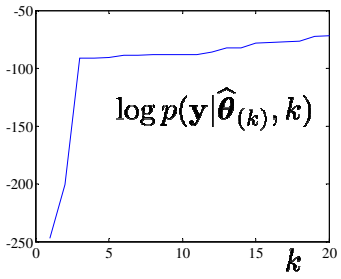
  Which “optimality” criterion?

- First naive solution: maximizing likelihood of data w.r.t. model

54

Maximizing Log Likelihood

- Problem: Log Likelihood is <u>not</u> decreasing when augmenting the order



Not applicable criterion!

55

Alternative: penalized likelihood

- Idea: find a compromise between fitting accuracy and simplicity of the model
- Insert a "penalty term" which grows with the order of the model and discourages highly complex models

$$K_{best} = \arg\max_k \left(LL(y|\theta_k) - C(k) \right)$$

complexity penalty

Examples: BIC, MDL, MML, AIC, ...

56

Alternative: penalized likelihood

- Example: Bayesian inference criterion (BIC) [Schwartz, 1978]

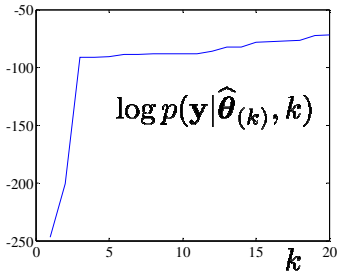
$$k_{best} = \arg\max_k \left\{ LL(y \mid \theta_k) - \frac{k}{2}\log(n) \right\}$$

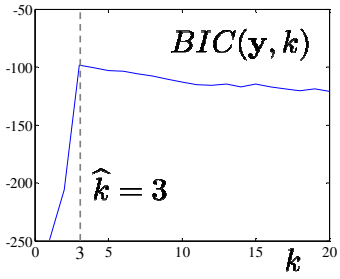
increases with k

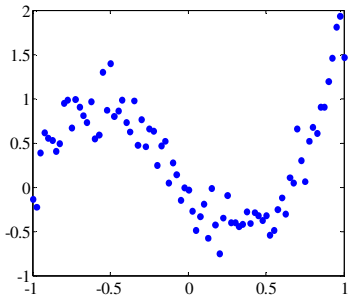
decreases with k (penalizes larger k)

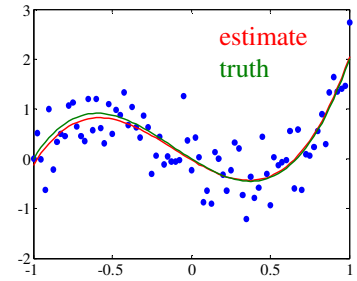
57

Back to the polynomial toy example









Some more HMM-oriented solutions

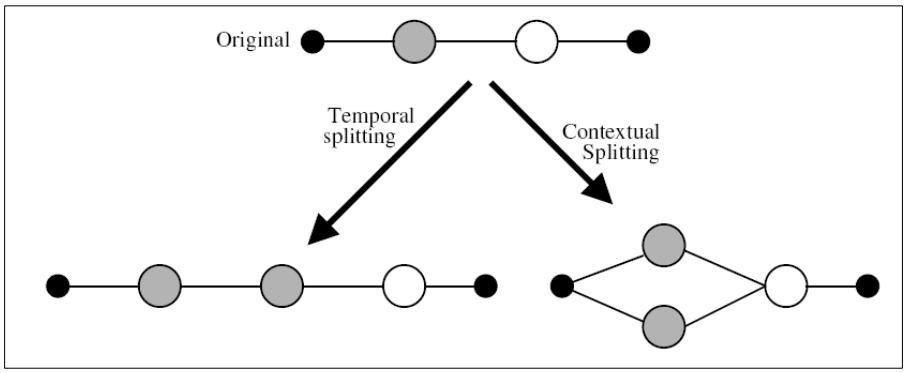
- Application driven model selection: states have some physical meaning [Hannaford,Lee IJRR 91]
- Split and merge approaches: starting from an inappropriate but simple model, the correct model is determined by successively applying a splitting (or merging) operation

  [Ikeda 93] [Singer,Ostendorf ICASSP 96] [Takami, Sagayama ICASSP 92]

59

Some more HMM-oriented solutions

- Application driven model selection:



[Ikeda 93] [Singer,Ostendorf ICASSP 96] [Takami, Sagayama ICASSP 92]

60

Some more HMM-oriented solutions

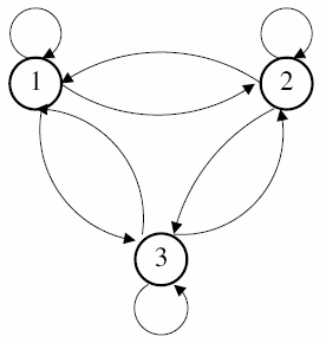
- Bayesian Model Selection: employing methods from Bayesian Estimation Theory.
  - most famous is [Stolcke,Omohundro NIPS93], where a merging state strategy was implemented in order to maximize the posterior probability $P(M_i|X)$ of the model M_i, given the data X
- Sequential Pruning Model Selection [Bicego,Murino,Figueiredo PRL03]: perform a "decreasing" learning, at each iteration the least important state is pruned

61

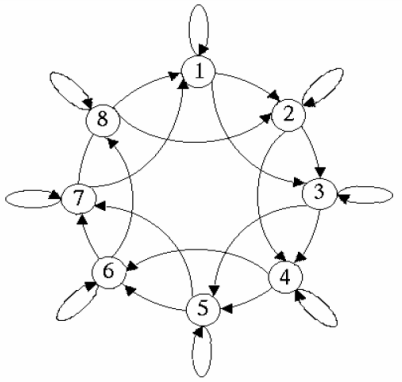
Model selection problem 2: selecting the best topology

- Problem: forcing the absence or the presence of connections
- Typical ad-hoc solutions
  - ergodic HMM (no contraints)
  - left to right HMM (for speech)
  - circular HMM (for shape recognition)

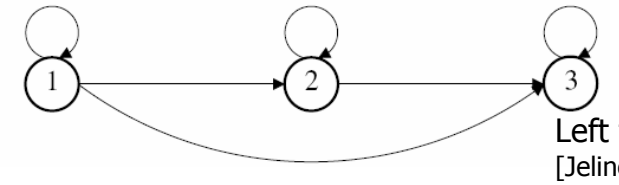
62



standard ergodic HMM



circular HMM [Arica,Yarman-Vural ICPR00]



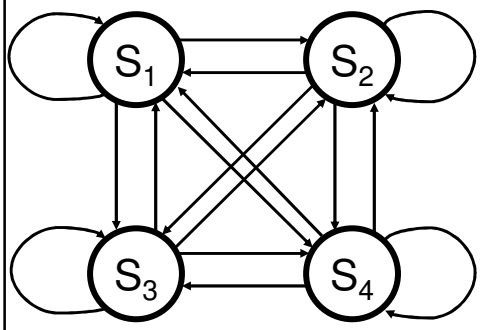
Left to right HMM
[Jelinek, Proc. IEEE 1976]

63

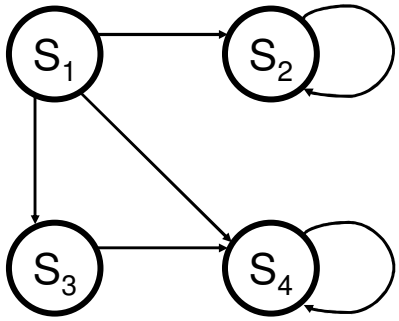
One data-driven solution

[Bicego, Cristani, Murino, ICIAP07]

Sparse HMM: a HMM with a sparse topology (irrelevant or redundant components are *exactly* 0)



Fully connected model: all transitions are present



Sparse model: many transition probabilities are zero (no connections)

64

Sparse HMM

Sparseness is highly desirable:

- It produces a structural simplification of the model, disregarding unimportant parameters
- A sparse model distills the information of all the training data providing only high representative parameters.
- Sparseness is related to generalization ability (Support Vector Machines)

65

Some open issues/research trends

1. Model selection
   - how many states?
   - which topology?
2. Extending standard models
   - modifying dependencies or components
3. Injecting discriminative skills into HMM

66

Extending standard models (1)

First extension:

adding novel dependencies between components, in order to model different behaviours

Examples:

- Input/Output HMM
- Factorial HMM
- Coupled HMM
- ...

67

Preliminary note: the Bayesian Network formalism

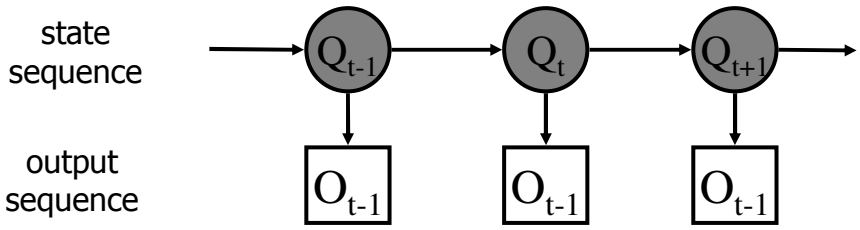
Bayes Net: graph where nodes represent variables and edges represent causality

hidden variable

observable variable

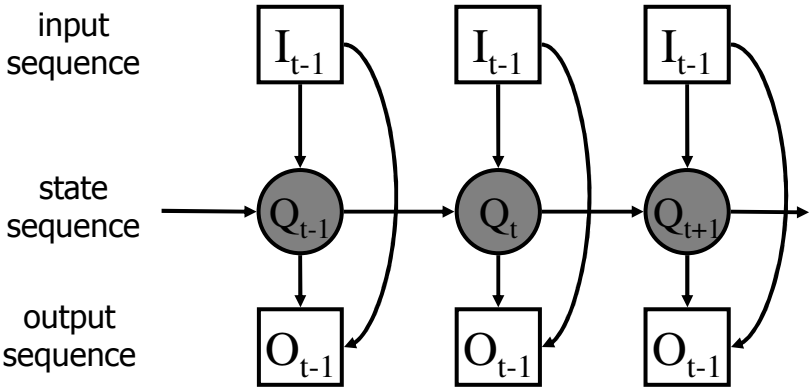
causal dependency

EX.: HMM



68

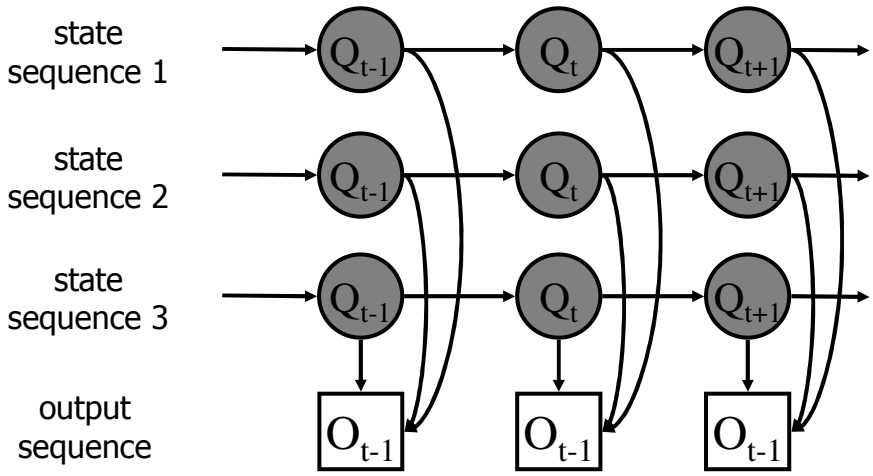
Input-Output HMM: HMM where transitions and emissions are conditional on another sequence (the input sequence)



EX.: finance, the input sequence is a leading market index

69

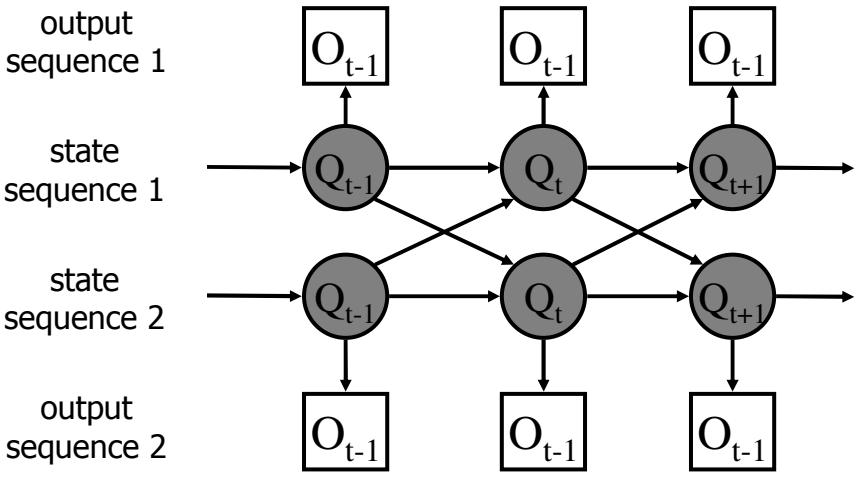
Factorial HMM: more than one state-chain influencing the output



Ex.: speech recognition, where time series generated from several independent sources.

70

Coupled HMMs: two interacting HMMs



Ex.: video surveillance, for modelling complex actions like interacting processes

71

Extending standard models (2)

Second extension:

employing as emission probabilities (namely functions modelling output symbols) complex and effective techniques (classifier, distributions,...)

Examples:

- Neural Networks
  [Bourlard, Wellekens, TPAMI 90],...
- Another HMM (to compose Hierarchical HMMs)
  [Fine, Singer, Tishby, ML 98]
  [Bicego, Grosso, Tistarelli, IVC 09]

72

Extending standard models (2)

Examples:

- Kernel Machines, such as SVM
- Factor analysis
  [Rosti, Gales, ICASSP 02]
- Generalized Gaussian Distributions
  [Bicego, Gonzalez-Jimenez, Alba-Castro, Grosso, ICPR 08]
- ...

73

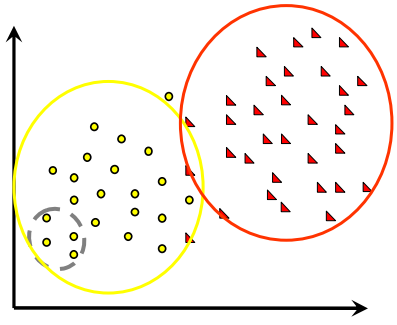
Some open issues/research trends

1. Model selection
   - how many states?
   - which topology?
2. Extending standard models
   - modifying dependencies or components
3. Injecting discriminative skills into HMM

74

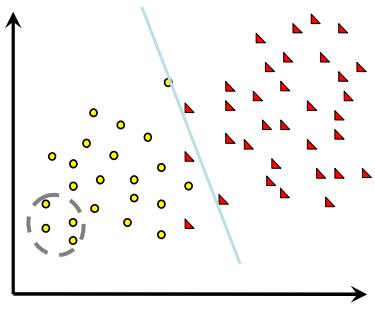
The general problem: generative vs discriminative modelling

Generative: one model for each class/group (e.g. HMM)



generative are better in describing classes

Discriminative: just model how to separate classes (e.g. SVM)



discriminative are better in solving the problem

75

Injecting discriminative information into HMM

- HMM are generative models, could be improved injecting discriminative information (information from other classes)
- Two ways:
  - inject discriminative information in the training phase
  - inject discriminative information in the classification phase

76

Discriminative training

- Standard HMM training is blind (no information from other classes is used)
- IDEA: training HMMs with discriminative criteria, i.e. considering also other classes' information
- Two popular examples:
  - maximum mutual information (MMI)

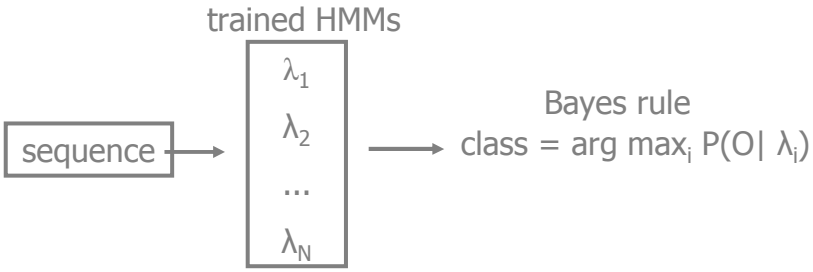
  [Bahl, Brown, de Souza, Mercer, ICASSP 00]
    - maximize likelihood for the objects in the class while minimizing the likelihood for the other obejcts
  - minimum Bayes risk (MBR)

  [Kaiser, Horvat, Kacic, ICSLP 00]

77

Discriminative classification

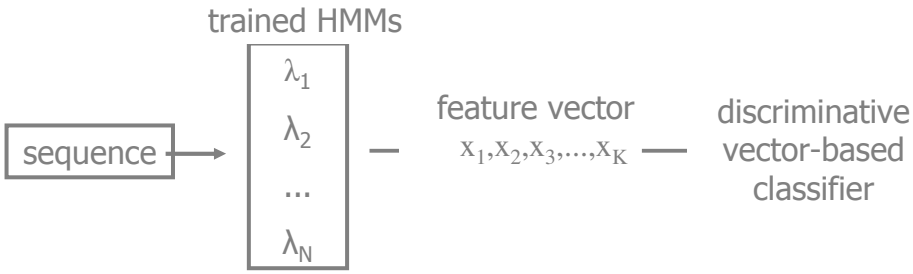
Standard HMM classification: train one HMM per class and apply the Bayes rule



78

Discriminative classification

Idea of discriminative classification: using trained HMMs to derive a feature space, where a discriminative classifiers is trained



79

Discriminative classification

- Kind of features:
  - the gradient of the Log Likelihood (or other related quantities):
    - this is the well known Fisher Kernel:

    [Jaakkola, Haussler, NIPS 99]

  - the log likelihood itself (or other quantities directly computable from the posterior probability)
    - using “score spaces” [Smith, PhD 03]
    - using the “dissimilarity-based representation” paradigm [Bicego, Murino, Figueiredo PR 04], [Bicego, Pekalska, Duin MCS 07]

80

HMM in bioinformatics

(Lucidi A. Perina)

Hidden Markov Models

- In bioinformatica le HMM spesso non si utilizzano in modo generale ma si costruiscono architetture “ad hoc”
  - Classificazione introni/esoni
  - Costruzione architettura
  - Allineamento di sequenze
  - Identificazione di geni
  - …

Classificazione con HMM

- Le HMM consentono di maneggiare sequenze di lunghezza variabile
- In bioinformatica molte sequenze hanno lunghezza variabile
  - Geni
  - Proteine
  - Introni esoni e regioni sul DNA in genere
- La classificazione tramite probabilità risolvendo il problema della valutazione

Classificazione con HMM: introni/esoni

- **I** insieme degli introni, **E** insieme degli esoni
- Addestro due HMM
  - *HMM_in* utilizzando **I** come training
  - *HMM_es* con **E** come training
- Valuto la classe di una nuova osservazione **x** risolvendo il problema della valutazione e calcolando…
  - P(**x** | *HMM_in*)
  - P(**x** | *HMM_es*)
- …e prendendo la classe che mi dà probabilità massima

Classificazione con HMM

- Metodi ibridi HMM/SVM hanno mostrato le migliori performance
- Addestro una HMM
- Estraggo informazioni da questa e le uso in modo discriminativo (come input per una Support Vector Machine)
  - Fisher kernel
  - Top kernel
  - Product probability kernel

Pair hidden Markov models

Problema dell'allineamento di sequenze

- **INPUT:** Due sequenze su di un alfabeto ∑ (4 nucleotidi or 20 aminoacidi):

  ATGTTAT e **ATCGTAC**

- **OUTPUT:** Inserendo '-' e shiftando le due sequenze devo produrre due sequenze allineate della stessa lunghezza

  A T – G T T A T
  A T C G T – A C

“Punteggio”

- Possibile dare un punteggio ad un allineamento
  - Mis-match sono penalizzati da $-\mu$
  - Un inserimento di ‘-’ penalizzato da $-\sigma$
  - I Match sono premiati con $+1$
- Il punteggio risultante è quindi

  $\#matches - \mu(\#mismatches) - \sigma\ (\#'-')$

```
A T – G T T A T
A T C G T – A C
```

$5 - \mu - 2\sigma$

Scoring Matrix

	A	R	N	K
A	5	-2	-1	-1
R	-	7	-1	3
N	-	-	7	0
K	-	-	-	6

AKRANR
KAAANK

-1 + (-1) + (-2) + 5 + 7 + 3 = 11

- Penalità e Premi per match e mis-match possono dipendere anche la particolare aminoacido (nucleotide)
- Si tiene conto dell’evidenza biologica
- K – R sono entrambi carichi → Non cambia la funzione di una proteina

Altre penalizzazioni…

- In natura spesso gli indels (‘-’) appaiono raggruppati (affine indels)
- Posso penalizzare di più allineamenti in cui ho tanti singoli indels

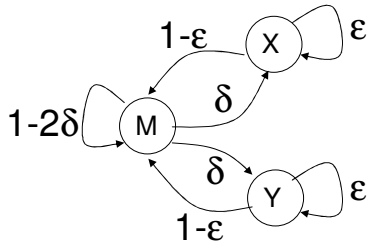
Utilizzando un normale punteggio questi 2 allineamenti sono equiprobabili

Allineamento di sequenze tramite HMM

- Le PAIR-HMM risolvono questo compito incorporando tutte le idee viste in precendeza
  - Penalizzazioni per mismatch e indels – Affine indels
- **Stati nascosti**
  - Match (M)
  - Insertion in *x* (X)
  - Insertion in *y* (Y)
- **Simboli osservati**
  - Match (M): {(*a*,*b*)| *a*,*b* in ∑ }.
  - Insertion in *x* (X): {(*a*,-)| *a* in ∑ }.
  - Insertion in *y* (Y): {(-,*a*)| *a* in ∑ }.

Pair - hidden Markov models

- **Architettura**



Probabilità di emissione

M: $P_{xi,yj}$
X: q_{xi}
Y: q_{yj}

Punteggio di un allineamento utilizzando una pair-HMM

1. In base ad un HMM ad ogni allineamento può essere assegnato un punteggio
2. Ogni "observation symbol" dell'HMM è una coppia ordinata di 2 lettere o di una lettera e un gap
3. Gli stati nascosti della HMM devono rappresentare un sistema di punteggio che rifletta un modello evoluzionario
4. Prob. di emissione e di transizioni danno la probabilità di allineamento di ogni coppia di simboli
5. Date due sequenze cerchiamo l'allineamento più probabile

Matrici di emissione e transizione

Probabilità di transizione

δ = probabilità del 1st '-'
ε = probabilità di estendere '-'

	M	X	Y
M	1-2δ	δ	δ
X	1- ε	ε	0
Y	1- ε	0	ε

Probabilità di emissione

i. Match: (*a,b*) con p_{ab} – solo per lo stato M
ii. Insertion in *x*: (*a,*-) con q_a – solo per lo stato X
iii. Insertion in *y*: (-,*a*).con q_a - solo per lo stato Y

Assegnamento del punteggio

- Date ***x*** (lunghezza m) *ed* ***y*** *(*lunghezza n)
  - Molti possibili allineamenti
  - **Corrispondenti a diversi percorsi sugli stati (di lunghezza compresa tra max{*m,n*} ed *m+n*)**
- Dati i parametri della pair-HMM posso calcolare un punteggio ad ogni allineamento
- Cerco l'allineamento che massimizza il punteggio
  - **Ovvero cerco il cammino di Viterbi!**

Profile HMM

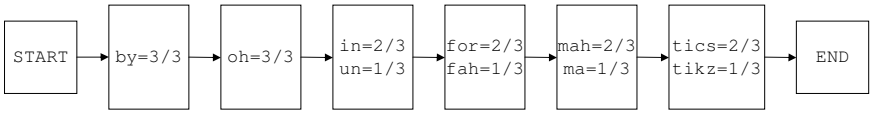
Costruzione di un architettura

- Vogliamo creare una HMM per generare la pronuncia della parola “Bioinformatics”
  - Supponiamo di avere a disposizione 3 esempi

Northern England	by	oh	in	for	ma	tics
London	by	oh	in	for	mah	tics
German	by	oh	un	fah	mah	tikz

Markov Chains

- Le catene di Markov non hanno stati nascosti



- Northern English:
  - "by-oh-in-for-ma-tics"
  - 3/3 * 3/3 * 2/3 * 2/3 * 1/3 * 2/3 = 72/729 ~ 0.1
- German:
  - "by-oh-un-fah-mah-tikz"
  - 3/3 * 3/3 * 1/3 * 1/3 * 2/3 * 1/3 = 18/729 ~ 0.02
- New:
  - "by-oh-un-for-ma-tikz"
  - 3/3 * 3/3 * 1/3 * 2/3 * 1/3 * 1/3 = 18/729 ~ 0.02

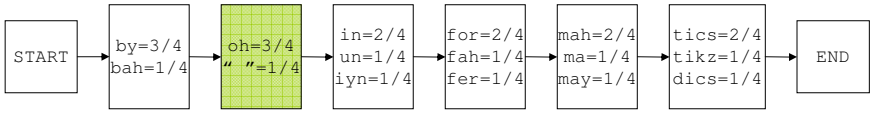
Costruzione di un architettura

- Aggiungiamo un nuovo esempio

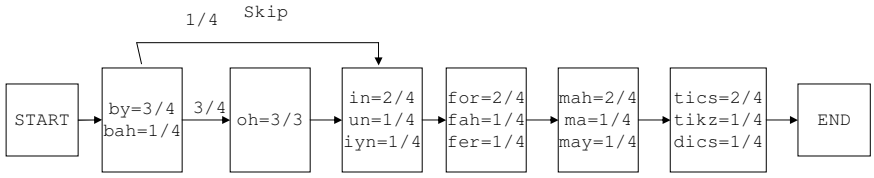
Northern England	by	oh	in	for	ma	tics
London	by	oh	in	for	mah	tics
German	by	oh	un	fah	mah	tikz
Texan	**bah**	-	**iyn**	**fer**	**may**	**dics**

Costruzione di un architettura

- Abbiamo 2 possibilità
  - Inserire un osservazione “vuota”

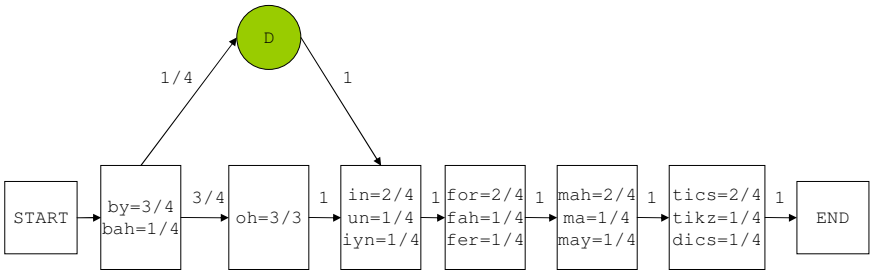


  - Inserire un “salto”



Costruzione di un architettura

- Più formalmente possiamo introdurre uno stato “DELETE” che salta un osservazione



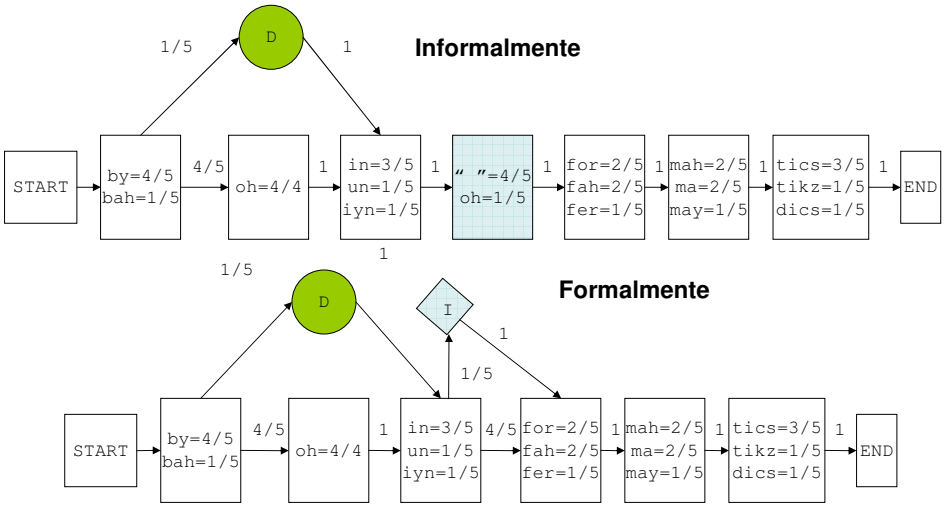
Costruzione di un architettura

- Introduciamo la pronuncia italiana

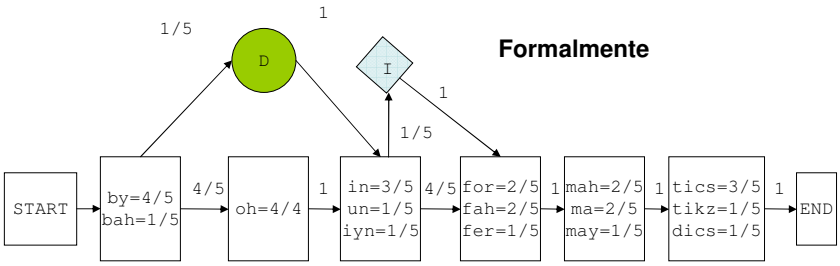
Northern England	by	oh	in	-	for	ma	tics
London	by	oh	in	-	for	mah	tics
German	by	oh	un	-	fah	mah	tikz
Texan	bah	-	iyn	-	fer	may	dics
Italian	**by**	**oh**	**in**	**oh**	**fah**	**ma**	**tics**

- A differenza di prima ora dobbiamo INSERIRE un nuovo stato

Costruzione di un architettura



Costruzione di un architettura



- Modelliamo l' "oh" (dell'italiano) addizionale come un inserimento casuale
- Modelliamo la mancanza dell' "oh" (nel texano) come un cancellamento casuale

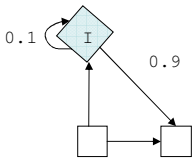
Costruzione di un architettura

- Il metodo formale è il più indicato perché non è necessario modellare la 4° e la 7° colonna

1	bi	oh	in		for	ma	tics	
2	bi	oh	in		for	mah	tics	
3	bah	oh	in		fer	ma	tikz	
4	bi	ou	in		fuh	maar	tics	
5	bi	oh	un		for	ma	tikz	
6	bah	-	iyn		fer	may	dics	
7	bi	oh	in		for	ma	tics	
8	bi	oh	in	oh	fah	may	tics	
9	bi	oh	un		far	mah	tikz	
10	be	oh	iyn		fer	ma	dics	
11	bi	ou	in		for	mah	tic	ah
12	bi	-	un		fah	mah	tics	

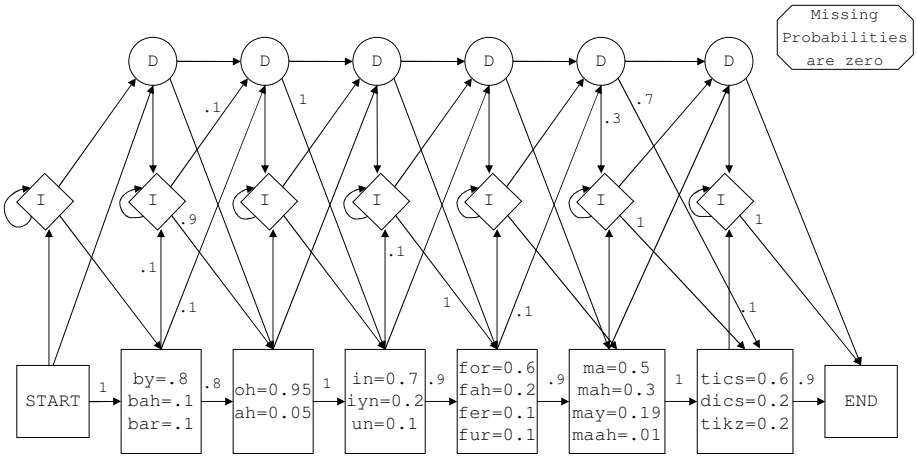
Costruzione di un architettura

- L'utilizzo degli stati è superiore al metodo informale
  - Più formale
  - Più capacità di generalizzare
- Inserimenti e cancellazioni possono avvenire in continuazione. Ad esempio…



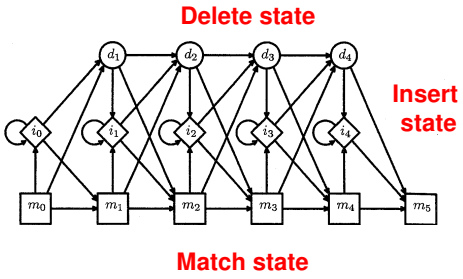
- Sono gli stati nascosti dell'HMM che in un architettura ad hoc hanno significati ben definiti

Costruzione di un architettura



Profile HMM

- Ho quindi tre stati nascosti
  - Insert
  - Delete
  - "Pronunce" → Match



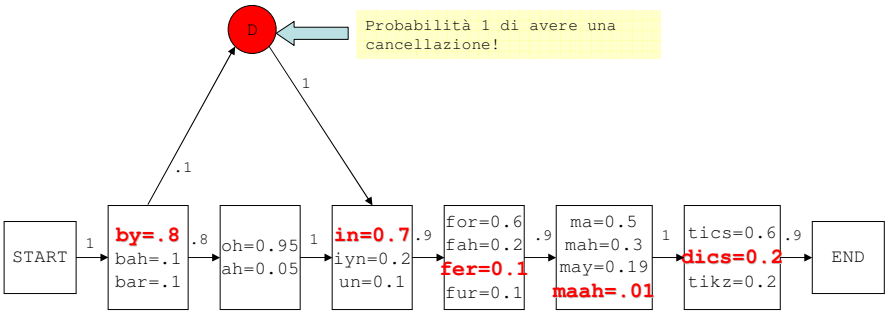
Allineamento multiplo di sequenze

- Una HMM con i tre stati match-delete-insert è detta "profile HMM"
- L'alfabeto della HMM consiste nei 20 simboli degli aminoacidi e nel simbolo delete ("-")
- Gli stati "delete" emettono solamente "–" con probabilità 1
- Ogni stato "insert" o "match" emette uno dei 20 aminoacidi ma non il simbolo " – ".

Allineamento multiplo di sequenze

- Utilizzando questa architettura ed l'algoritmo di Viterbi si possono allineare sequenze
- Prendiamo la pronuncia di bioinformatica in nordirlandese "by-in-fer-maah-dics" e vediamo due percorsi diversi sull'architettura appena creata

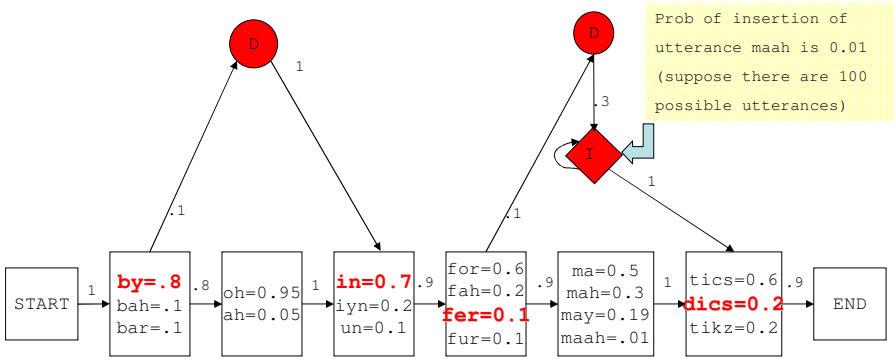
Viterbi path 1



- Path 1

P = (1*.8)*(.1*1)*(1*0.7)*(0.9*0.1)*(0.9*0.01)* (1*0.2)*(0.9) = 8.2 x 10-6

Viterbi path 2



- Path 2:

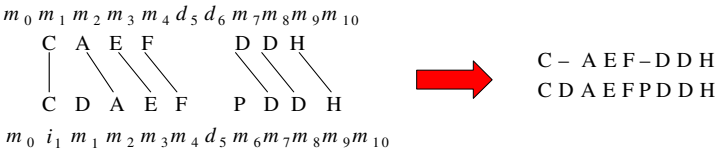
P = (1*.8)*(0.1*1)*(1*0.7)*(0.9*0.1)*(0.1*1)*(0.3*.01)* (1*0.2)*(0.9) = 2.7 x 10^{-7}

Allineamento multiplo di sequenze

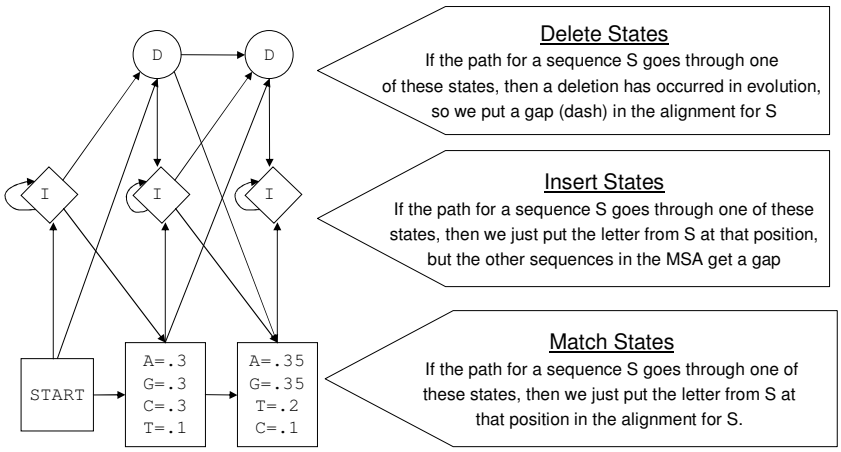
- Data una famiglia di sequenze, per allinearle sono necessari i seguenti passi
  1. Addestrare la HMM con l'insieme di sequenze (Baum-Welch)
     - Scegliendo prima la lunghezza del modello
  2. Calcolare la sequenza di stati più probabile per ogni sequenza (Viterbi path)
     - Due simboli sono allineati se entrambi producono lo stesso match state nei cammini di Viterbi
  3. Aggiungere delete ed inserimenti appropriatamente

Allineamento multiplo di sequenze: esempio

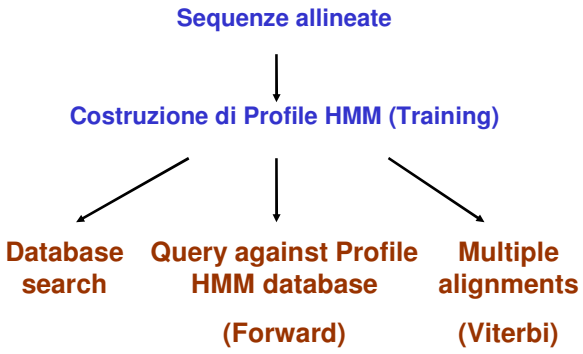
- Si considerino CAEFDDH e CDAEFPDDH
- Supponiamo che il nostro modello abbia lunghezza 10 e che i due cammini di viterbi siano
  - **M0, M1, M2, M3, M4, D5, D6, M7, M8, M9, M10**
  - **M0, I1, M1, M2, M3, M4, D5, M6, M7, M8, M9, M10**
- L'allineamento è calcolato allineando posizioni generate dallo stesso match state



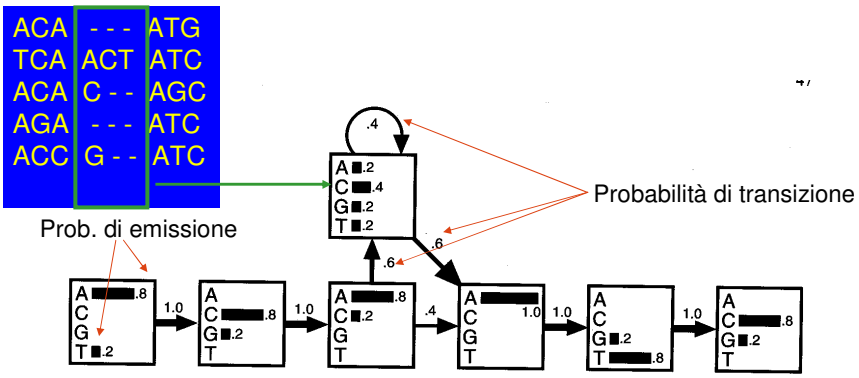
Da Viterbi all'allineamento



Profile HMM: Overview



1- Learning da allineamenti esistenti



- **Topologia** → (Lunghezza del modello) solitamente è la media delle lunghezze delle sequenze
- Addestrare una pHMM vuol dire creare un profilo

2- Learning da sequenze non allineate

INIT: Parti da un modello casuale di lunghezza uguale alla lunghezza media delle sequenze

1- Allinea le sequenze

2- Calcola le probabilità di transizione ed emissione utilizzando gli allineamenti appena calcolati

3- Ripeti il procedimento fino a che il modello smette di cambiare

By-product: Produce allineamenti multipli!

118