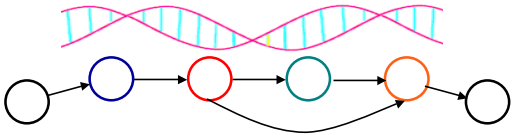
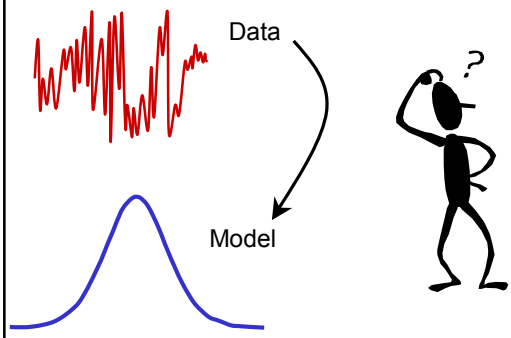


Introduction to Hidden Markov Models

Hagit Shatkay, Celera

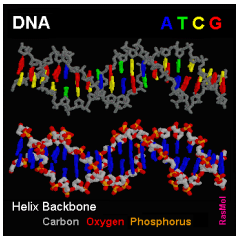
Tübingen, Sept. 2002

Model Fitting



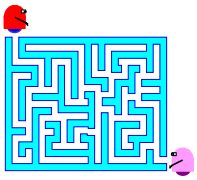
The Many Facets of HMMs...

Speech Recognition



DNA/Protein Analysis

Medical Decision Making



Robot Navigation and Planning

Overview

- The Components of HMMs
- Evaluating a sequence WRT an HMM (Problem 1)
- Fitting a sequence to an HMM (Problem 2)
- Fitting an HMM to sequences (Problem 3)
- Issues, Extensions, Applications
- Conclusion

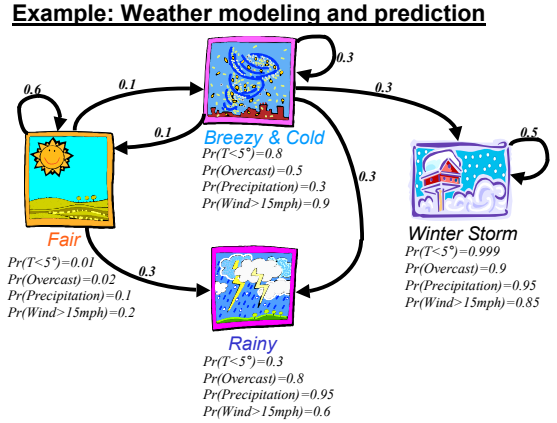
HMMs: The Basics

What are HMMS?

Models that are:

- **Stochastic** (probability-based)
- **Generative**
  Provide a putative production process for generating data.
- **Satisfying the Markov Property**
  The *present* state *summarizes* the past.
  F*uture events* depend only on the *current situation* –
  not on the preceding ones.

Example: Weather modeling and prediction



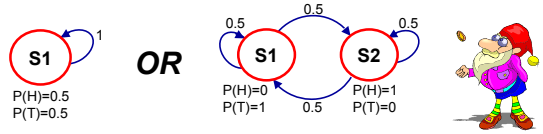
The Building Blocks of HMMs

An HMM is a tuple: $\lambda = < S, V, A, B, \pi >$

- **States** $(S = s_1 , . . . , s_N)$ – Hidden
- **Observations** $(V = v_1 , . . . , v_M)$

Parameters:

- **A:** Transition matrix $A_{ij} = \Pr(q_{t+1} = s_j \mid q_t = s_i)$ $\sum_{j=1}^{N} A_{ij} = 1$
- **B:** Observation matrix $B_{ik} = \Pr(o_t = v_k \mid q_t = s_i)$ $\sum_{k=1}^{M} B_{ik} = 1$
- **π:** Initial distribution $\pi_i = \Pr(q_1 = s_i)$ $\sum_{j=1}^{N} \pi_i = 1$



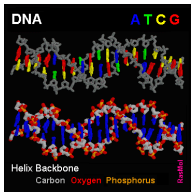
Examples Revisited

States: Possible phonemes in a word
Observations: Uttered Phonemes
Transitions: Phoneme order in a word

Speech Recognition

States: Positions for nucleotides deletion/matching/insertion
Observations: Nucleotides
Transitions: Nucleotides order in the DNA

States for modeling purposes.

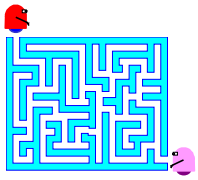


DNA/Protein Analysis

States: Patient's (varying) condition
Observations: Instrument Readings
Transitions: Changes due to treatment

Medical Decision Making

States: Robot's position
Observations: Sensors Readings
Transitions: Changes due to movement



Robot Navigation and Planning

A physical notion of states.

HMMs: The Three Problems

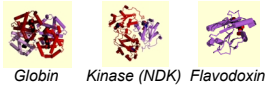
Problem 1: Given a *model* λ and a sequence of observations $O=O_1,\ldots,O_T$, find **O**'s probability under λ, $\Pr(O|\lambda)$.

Problem 2: Given a *model* λ and a sequence of observations $O=O_1,\ldots,O_T$, find the *best state sequence* $Q=q_1,\ldots,q_T$ *explaining* it.

Problem 3: Given a sequence of observations $O=O_1,\ldots,O_T$, find the *best model* λ that could have *generated* it.

Example: Modeling Protein Families

Different protein families (e.g. *Globin, Flavodoxin, Kinase*), have different *characteristic* sequences. Families are represented as HMMs.



Globin *Kinase (NDK)* *Flavodoxin*

(http://www.sanger.ac.uk/cgi-bin/Pfam/)

Observations ➡ 20 Amino acids *(Glu, Gly, Arg,...)*

States ➡ Anchor points for typical AA emition, insertion and deletion

Problem 3: Given multiple aligned sequences, learn the family HMM

Problem 2: Given a family HMM and a sequence, find the best alignment.

Problem 1: Given a family HMM and a protein sequence, calculate how likely the protein is to be in the family.

Basic Tools

Def. of Conditional Probability: $\Pr(X|Y) = \frac{\Pr(X,Y)}{\Pr(Y)}$

Bayes Rule: $\Pr(X|Y) = \frac{\Pr(Y|X) \cdot \Pr(X)}{\Pr(Y)}$

Chain Rule of Conditional Probability:

$\Pr(X_1, X_2, ..., X_n) = \Pr(X_1)\ \Pr(X_2|X_1)...\Pr(X_n|\ X_1, ..., X_{n-1})$

The Markov Property (An *assumption* – not a fact!):

$\Pr(q_{t+1}=s_j \mid q_t=s_i) = \Pr(q_{t+1}=s_j \mid q_1=s_{i1}, q_2=s_{i2}, ..., q_t= s_{it}=s_i)$

[Rabiner&Juang86, Rabiner89] The ultimate *Introduction to HMMs*, and *application in NLP (speech recognition)*

[Charniak93] and references therein. HMMs in *NLP*

[Leek97, Ray&Craven01] HMMs in *NLP, Infomation Extraction* from *Biomedical text*

[Hauskrecht&Fraser98] HMMs in *medical decision making*

[Simmons&Koenig95, Koenig&Simmons96, Shatkay&Kaelbling97,Shatkay&Kaelbling02] HMMs for *robot navigation*

[Churchill89, Krogh et al 94a, Krogh et al 94b, Eddy 98, Burge97, Durbin et al 98] and references therein. HMMs in *computational biology*.

Problem 1

Pr(Sequence|Model)

Pr(Sequence|Model)

Given:

$\boldsymbol{O} = O_1,\ldots,O_T$ A sequence of observations

$\boldsymbol{\lambda} = \langle \mathbf{S, V, A, B, \pi} \rangle$ An HMM

Calculate:

The probability of $\boldsymbol{O}$ to be generated under the model $\boldsymbol{\lambda}$, $\Pr(\mathbf{O}|\boldsymbol{\lambda})$

Example Application:

Given a protein sequence, P, and several possible protein families ($\boldsymbol{\lambda}_1 \ldots \boldsymbol{\lambda}_L$), find the most likely family of P.

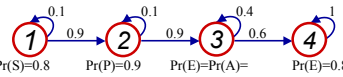
$$\underset{\boldsymbol{\lambda}_i}{argmax}[Pr(P|\boldsymbol{\lambda}_i)]$$

Calculating $\Pr(O|\lambda)$

Example

Sequence: TPEE A (small) protein motif

HMM, λ:



$\boldsymbol{\pi}_1 = 1$

Pr(S)=0.8	Pr(P)=0.9	Pr(E)=Pr(A)=	Pr(E)=0.8
Pr(T)=0.2	Pr(E)=0.1	Pr(S)=Pr(G)=0.25	Pr(Q)=Pr(D)=0.1

If we knew that the generating state sequence is: *1234*

$Pr(\boldsymbol{O}=\text{TPEE}|\boldsymbol{\lambda},\boldsymbol{Q}=1234) =$

$Pr(O_1=\text{T}|\boldsymbol{\lambda}, q_1=1)\cdot Pr(O_2=\text{P}|\boldsymbol{\lambda}, q_2=2)\cdot Pr(O_3=\text{E}|\boldsymbol{\lambda}, q_3=3)\cdot Pr(O_4=\text{E}|\boldsymbol{\lambda}, q_4=4) =$

$0.2\cdot 0.9\cdot 0.25\cdot 0.8 = 0.036$

No explicit state sequence ➡ *Marginalize*:

$$Pr(\boldsymbol{O}=\text{TPEE}|\boldsymbol{\lambda}) = \sum_{q_1q_2q_3q_4} Pr(\boldsymbol{O}=\text{TPEE}, \boldsymbol{Q} = q_1q_2q_3q_4|\boldsymbol{\lambda}) =$$

$$\sum_{q_1q_2q_3q_4} Pr(\boldsymbol{O}=\text{TPEE}|\boldsymbol{Q}=q_1q_2q_3q_4,\boldsymbol{\lambda})\cdot Pr(\boldsymbol{Q}=q_1q_2q_3q_4|\boldsymbol{\lambda}) = \sum_{q_1q_2q_3q_4} \boldsymbol{\pi}_{q_1}\cdot\prod_{i=1}^{4} B_{q_i o_i}\prod_{j=1}^{3} A_{q_j q_{j+1}}$$

Calculating Pr(O|λ) (Cont.)

In the general case

$$Pr(\mathbf{O}|\boldsymbol{\lambda}) = \sum_{Q=q_1\ldots q_T} Pr(\mathbf{O} \mid \mathbf{Q},\boldsymbol{\lambda}) \cdot Pr(\mathbf{Q} \mid \boldsymbol{\lambda}) = \sum_{Q=q_1\ldots q_T} \boldsymbol{\pi}_{q_1} \cdot \prod_{i=1}^{T} B_{q_i o_i} \cdot \prod_{j=1}^{T-1} A_{q_j q_{j+1}}$$

Computation time

- ☹ N^T State-Sequences $\boldsymbol{Q}$ (N states, T time steps)
- ☹ $2T$ products per sequence
- ☹ $O(TN^T)$

Not Feasible

Calculating Pr(O|λ) (Cont.)

Solution:

An alternative approach, using ***Dynamic Programming***

Idea: Sum over *all final-states* rather than over *all state-sequences*:

$$Pr(\mathbf{O} = o_1,\ldots,o_T \mid \boldsymbol{\lambda}) = \sum_{i=1}^{N} Pr(\mathbf{O} = o_1,\ldots,o_T,\ q_T = s_i \mid \boldsymbol{\lambda})$$

Making it work:

- Define, for any time $t \le T$: $\boldsymbol{\alpha}_t(i) = Pr(o_1,\ldots,o_t,\ q_t = s_i \mid \boldsymbol{\lambda})$.
- Initialization: $\boldsymbol{\alpha}_1(i) = \boldsymbol{\pi}_i \cdot B_{i\,o_1}$
- Recursion: $\boldsymbol{\alpha}_{t+1}(j) = \sum_{i=1}^{N} \boldsymbol{\alpha}_t(i) \cdot A_{ij} \cdot B_{j\,o_{t+1}}$
- Termination: $Pr(\mathbf{O} = o_1,\ldots,o_T \mid \boldsymbol{\lambda}) = \sum_{i=1}^{N} \boldsymbol{\alpha}_T(i)$

Calculating Pr(O|λ) (last!)

$\boldsymbol{\alpha}_t(i) = Pr(o_1,\ldots,o_t\ ,\ q_t = s_i \mid \lambda)$: Known as the ***Forward probability.***

Computation time

- ☺ *Calculating each* $\alpha_{t+1}(j)$ *: N Summands*
- ☺ N states, T time points ➡NT such summations.
- ☺ 2 products per summand
- ☺ $O(TN^2)$

Efficient Computation!

Problem 2

Find Best States for Observations

Best States for Observations

Given:

$\boldsymbol{O} = O_1,\ldots,O_T$ A sequence of observations

$\boldsymbol{\lambda} = < \mathbf{S, V, A, B, \pi} >$ An HMM

Find:

The sequence of states, $\boldsymbol{Q} = q_1,\ldots,q_T$, that generated $\boldsymbol{O}$ under the model $\boldsymbol{\lambda}$

Example Application:

Given a protein sequence, P, and an HMM model for a family (a *profile*), find the best *alignment* of P with the profile.

What is the "*Best*" State Sequence?

Options:

- Optimize the *expected* number of *correct states*.
  The state at time *t*, q_t, *is* s_i *that maximizes* $Pr(q_t = s_i | O, \lambda)$.
  The resulting sequence $\boldsymbol{Q} = q_1,\ldots,q_T$ may *not be a* ***valid*** *one*...
- Optimize the whole *state-sequence probability*, $Pr(\mathbf{Q}|O,\lambda)$.
  Equivalent to: $\boldsymbol{Q}^* = \underset{\mathbf{Q}}{argmax}[Pr(\mathbf{Q},O|\ \lambda)]$

  Efficiently done using the ***Viterbi Algorithm***.

The Viterbi Algorithm

Dynamic Programming (again)

$Q^* = \underset{Q}{argmax}[Pr(Q,O|\lambda)]$, without *explicit expansion* of all N^T state sequences.

- Define, for any time $t \le T$:

$$\delta_t(i) = \max_{q_1,\ldots,q_{t-1}}[Pr(q_1,\ldots,q_{t-1},q_t = s_i, o_1,\ldots,o_t \mid \lambda)].$$

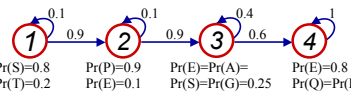
- Initialization: $\delta_1(i) = \pi_i \cdot B_{io_1}$ $\qquad \psi_1(i)=0$
- Recursion: $\delta_{t+1}(j) = \max_i[\delta_t(i)\cdot A_{ij}]\cdot B_{j\,o_{t+1}}$ $\qquad \psi_{t+1}(j) = \underset{i}{argmax}[\delta_t(i)A_{ij}]$
- Termination: $P^* = \max_Q[Pr(Q,O|\lambda)] = \max_i[\delta_T(i)]$

$$Q^* = q_1^*, q_2^*, \ldots, q_T^* \qquad q_T^* = \underset{i}{argmax}[\delta_T(i)], q_{t-1}^* = \psi_t(q_t^*)$$

Example

Sequence: TPSS

HMM, λ:



$\pi_1 = 1$

Pr(S)=0.8	Pr(P)=0.9	Pr(E)=Pr(A)=	Pr(E)=0.8
Pr(T)=0.2	Pr(E)=0.1	Pr(S)=Pr(G)=0.25	Pr(Q)=Pr(D)=0.1

Find the State Sequence Q^*:

T: $\delta_1(1)=0.2 \quad \delta_1(2)=\delta_1(3)=\delta_1(4)=0$ $\qquad \psi_1(i)=0$

P: $\delta_2(2)=0.2\cdot 0.9\cdot 0.9=0.162 \quad \delta_2(1)=\delta_2(3)=\delta_2(4)=0$ $\qquad \psi_2(2)=1$

S: $\delta_3(3)=0.162\cdot 0.9\cdot 0.25=0.03645 \quad \delta_3(1)=\delta_3(2)=\delta_3(4)=0$ $\qquad \psi_3(3)=2$

S: $\delta_4(3)=0.03645\cdot 0.4\cdot 0.25=\mathbf{0.003645} \quad \delta_3(1)=\delta_3(2)=\delta_3(4)=0$ $\qquad \psi_4(3)=3 \quad q_4^*=3$

Problem 3

Learning an HMM from Data

Best Model for Observations

Given:

$O = O_1,\ldots,O_T$ Sequence(s) of observations

Implicit also:

The set of possible observation values, $V = v_1, \ldots, v_M$

The number of states in the model, N.

Find:

The model $\lambda = \langle S, V, A, B, \pi \rangle$ that generated O

Example Application:

Given multiple protein sequences, $P_1,\ldots,P_K$, from a protein family, find an HMM for the family (a *profile*).

Finding an HMM

Example

Sequences:

```
V G A – – H
V – – – – N
V E A – – D
V N A – – N
I A G A D N
1 2 3     4
```

A (small) Globin motif

(Modified from [Durbin et al 98])

Count-based estimates:

Match-state M_1

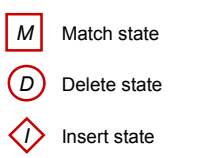
$$A_{12} = \Pr(q_{t+1}=M_2 \mid q_t=M_1) \approx \frac{(\text{\# of transitions from } M_1 \text{ to } M_2)}{(\text{\# of transitions from } M_1)} = 4/5$$

$$B_{1V} = \Pr(o_t=V \mid q_t=M_1) \approx \frac{(\text{\# of times V observed in } M_1)}{(\text{\# of visits to } M_1)} = 4/5$$

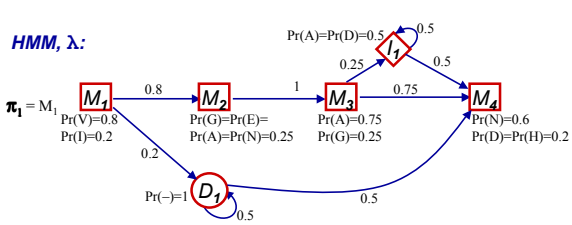
Example (Cont.)

Sequences:

```
V G A – – H
V – – – – N
V E A – – D
V N A – – N
I A G A D N
1 2 3     4
```



HMM, λ:



Formal & General

Given:

$\boldsymbol{O} = O_1,\ldots,O_T$ Sequence(s) of observations

The number of states in the model, N.

Find:

Model $\boldsymbol{\lambda} = \langle \mathbf{S, V, A, B, \pi} \rangle$, maximizing the likelihood $Pr(\boldsymbol{O}|\lambda)$

Use *Expected* Counts:

- Pick an initial transition and observation model.
- Iterate:
  - **E** ✶ Use current model and observations to compute a distribution on state sequences.
  - **M** ✶ Use distribution and observations to estimate a ***new*** transition and observation model, based on ***expected*** frequencies.

Increases $Pr(\boldsymbol{O}|\lambda)$ at each iteration, until convergence.

Baum-Welch Algorithm [Baum et al 71, Rabiner89]

- Receives number of states.
- Picks an initial model.
- Updates iteratively:

$\pi_i \leftarrow$ Expected frequency of s_i at time 0;

$$A_{ij} \leftarrow \frac{E(\text{\# of trans. from } s_i \text{ to } s_j)}{E(\text{\# of trans. from } s_i)};$$

$$B_{ik} \leftarrow \frac{E(\text{\# of times in } s_i \text{ observing } o_k)}{E(\text{\# of times in } s_i)}.$$

Baum-Welch Algorithm *(details)*

Dynamic programming *for calculating:*

✶ $\alpha_t(i) = \Pr(O_1,\ldots, O_t, q_t = s_i|\boldsymbol{\lambda})$ ***Forward***

✶ $\beta_t(i) = \Pr(O_{t+1},\ldots, O_T|q_t = s_i, \boldsymbol{\lambda})$ ***Backward***

✶ $\gamma_t(i) = \Pr(q_t = s_i|\mathbf{O},\boldsymbol{\lambda}) = \frac{Pr(q_t = s_i, O|\lambda)}{Pr(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$

✶ $\xi_t(i,j) = \Pr(q_t = s_i, q_{t+1} = s_j|\mathbf{O},\boldsymbol{\lambda}) = \frac{Pr(q_t = s_i, q_{t+1} = s_j, O|\lambda)}{Pr(O|\lambda)} =$

$$= \frac{\alpha_t(i)A_{ij}B_{j,O_{t+1}}\beta_{t+1}(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}$$

Sum $\gamma_t(i)$'s and $\xi_t(i,j)$'s to obtain ***expected counts***.

Baum-Welch Algorithm *(last)*

General Practical Issues:

- Reaches local maxima
- Strongly depends on initial conditions
- May require a lot of data and many iterations

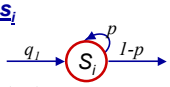
[Rabiner&Juang86, Rabiner89] A comprehensive introduction.
[Baum et al 70] and references therein. Baum's original results.
[Durbin et al 98] and references therein. Applications in computational biology.

Issues, Extensions, Applications

State Duration, *Semi-Markov* Models

Staying d time steps in the same state, s_i

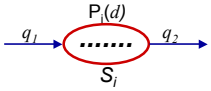
Standard HMM:



Pr(d consecutive time steps in state Si) = $p^{(d-1)} \cdot (1-p)$

Geometric distribution over duration in a state.

Supporting *other* distributions:



$\lambda = < S, V, A, B, \pi > \cup \{ P_1, ..., P_N \}$

P_i : A probability distribution over duration d, at state i.

$P_i(d)$ = Pr(*staying d time steps in state* Si), for $d \leq D$

$P_i(d)$ can be a ***continuous*** density function (e.g. Gaussian).

Multi-Dimensional Data

Observations may be *structured* (see weather example)

Standard observation sequence: $O=O_1,\ldots,O_T$

$O_j \in V=\{v_1, \ldots, v_M\}$ V is assumed to be a set of *atomic* values.

Multi-dimensional observations:

$O_j \in \vec{V} = \{\vec{v}_1, \ldots, \vec{v}_M\}$ $\vec{v}_i = \langle v_i^1\ v_i^2\ \cdots,\ v_i^K \rangle$

$\vec{V}$ is a set of ***k-dimensional vectors.***

The components $v_i^1\ v_i^2, \cdots,\ v_i^K$ are (typically) assumed to be *conditionally independent* given the state

Multi-Dimensional Data (cont.)

Applications:

Twinscan: [Korf et al 01]

- Gene structure prediction over a target DNA sequence, ***D***
- HMM is used for modeling regions in the DNA
- 2-dimensional observations: *<Nucleotide, Conservation tag>*.

  Nucleotide: $\{A,C,G,T\}$ *Conservation tag*: $\{\,.\,,\ |\ ,\ :\,\}$

  Conservation tag represents alignment of ***D*** with an informant sequence.

Robotics: [Cassndra et al 96, Shatkay&Kaelbling97]

- Observations represent the robot's view in each state.
- They are factored into the view in each cardinal direction: *front*, *left* and *right (3-dimensional)*.

Multi-Dimensional *States*: *Factorial HMMs* [Ghahramani and Jordan 97]

Other Topics

- Pseudo-counts and priors (Never say *Never*...)
- Other methods for learning HMMs:
  - *Bayesian Model Merging* [Stolcke&Omohundro93,94]
  - *Viterbi training* [Durbin et al 98]
  - *Optimizing other measures* [Rabiner 89]
- Comparing HMMs [Rabiner 89]
- Choosing an initial model [Rabiner 89]
- Constraining HMM with domain knowledge and data. [Shatkay&Kaelbling02]

Conclusion

- G*enerative probabilistic* models. Useful for modeling sequences with variations and/or noise.
- Efficient ways exist to relate and align sequences with families (HMMs).
- Generally: Require a lot of data and domain specific knowledge to construct.
- *Versatile, flexible* and *general*. Support extensions, special cases, and a wide variety of applications.

Bibliography

[Baum et al 70] Baum L. E. (1970). "*A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains*", The Annals of Mathematical Statistics, 41, #1, pp. 164-171.

[Burge97] Burge C. (1997). "*Identification of Genes in Human Genomic DNA*", PhD Thesis, Stanford University, Stanford, CA.

[Cassandra et al 96] Cassandra A. R. , Kaelbling L. P. and Kurien J. A. (1996). "*Acting Under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation*", Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems.

[Charniak93] Charniak E. (1993). "*Statistical Language Learning*", MIT Press.

[Churchill89] Churchill G. A. (1989). "*Stochastic Models for Heterogeneous DNA Sequences*", Bulletin of Mathematical Biology, 51, #1, pp. 79-94.

[Durbin et al 98] Durbin R. et al (1998). "*Biological Sequence Analysis*", Cambridge U. Press.

[Eddy98] Eddy S. (1998). "*Profile Hidden Markov Models*", Bioinformatics, 14, pp. 755-763.

[Ghahramani&Jordan97] Gharahmani Z. and Jordan M. I. (1997). "*Factorial Hidden Markov Models*", Machine Learning, 29, pp. 1-31.

[Hauskrecht&Fraser98] Hauskrecht M. and Fraser H. (1998). *"Planning Medical Therapy Using Partially Observable Markov Decision Processes*", Proc. of the Ninth International Workshop on Principles of Diagnostics, pp. 182-189.

Bibliography (cont.)

[Koenig&Simmons 96] Koenig S. and Simmons R. (1996). "*Unsupervised Learning of Probabilistic Models for Robot Navigation*", Proc. of the IEEE Int. Conf. on Robotics and Automation.

[Korf et al 01] Korf I. et al (2001). "*Integrating genomic homology into gene structure prediction*" , Proc. of ISMB'01, pp. S140-S148.

[Krogh et al 94a] Krogh A. et al (1994). "*Hidden Markov Models in Computational Biology*", Journal of Molecular Biology, 235, pp. 1501-1531.

[Krogh et al 94b] Krogh A., Mian S.I. and Haussler D. (1994). "*A Hidden Markov Model that Finds Genes in E. Coli DNA*", Nucleic Acids Research, 22, pp. 4768-4778.

[Leek97] Leek T.R. (1997). "*Information Extraction Using Hidden Markov Models*", MSc thesis, Dept. of Computer Science, University of California, San Diego.

[Rabiner&Juang86] Rabiner L.R., Juang B.H. (1986). "*An Introduction to Hidden Markov Models*", IEEE ASSP Magazine, 3, #1, pp. 4-16.

[Rabiner89] Rabiner L.R. (1989). "*A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*", Proc. of the IEEE, 77, #2, pp. 257-285.

[Ray&Craven01] Ray S. and Craven M. (2001). "*Representing Sentence Structure in Hidden Markov Models for Information Extraction*", Proc. of IJCAI'01.

Bibliography (cont.)

[Shatkay&Kaelbling97] Shatkay H. and Kaelbling L. P. (1997). "*Learning Topological Maps with Weak Local Odometric Information*", Proc. of IJCAI'97.

[Shatkay&Kaelbling02] Shatkay H. and Kaelbling L. P. (2002). "*Learning Hidden Markov Models with Geometrical Constraints: Bridging the Topological-Geometrical Gap*", Journal of AI Research, 16, pp. 167-207.

[Simmons&Koenig95] Simmons R. and Koenig S. (1995). "*Probabilistic Navigation in Partially Observable Environments*", Proc. of IJCAI'95.

[Stolcke&Omohundro93] Stolcke A. and Omohundro S. M. (1993). "*Hidden Markov Model Induction by Bayesian Model Merging*", Advances in Neural Information Systems, 5, Morgan Kaufmann, pp. 11-18

[Stolcke&Omohundro94] Stolcke A. and Omohundro S. M. (1994). "*Best-first Model Merging for Hidden Markov Model Induction*", ICSI Technical Report, TR-94-003, International Computer Science Institute, Berkeley.