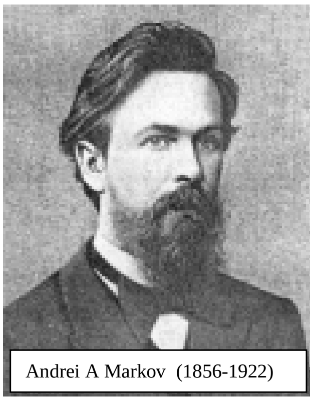
Introduction

Programming hidden Markov models

Chris Bystroff
Department of Biology, Rensselaer Polytechnic Institute, Troy, NJ, USA



Andrei A Markov (1856-1922)

Since their formulation by Andrei Markov in 1906 [7], Markov chains (MC) and hidden Markov models (HMM) have found a place in diverse fields of science and engineering, from speech recognition to weather prediction to protein sequence alignment. Wherever a data set can be expressed as a string of discrete symbols, and when the data has a common source or common underlying principle, then for those cases a hidden Markov model may be designed to extract that underlying principle.

A Markov chain is a directed graph where the nodes are Markov states and the edges are directed state-state transitions. A Markov state is said to 'emit' a symbol, which is unique to that state. A "hidden" Markov model (also sometimes called a "state space" or "latent Markov" model) differs from a MC in that each state emits one of a set of symbols drawn from a distribution; different hidden states may emit the same symbol, and non-emitting states are possible. The term 'hidden' is used because the symbol sequence alone does not tell us the state sequence directly. Instead, the latter must be inferred. In a HMM the states have a meaning all their own, separate from the meaning of the symbols they emit. For example, in a HMM composed of states that emit temperature readings, the states themselves may represent precipitation readings, or wind direction, or seasons, or all of the above. HMM states are classifiers of the symbols in the data string(s), their types and their contexts.

Algorithms for computing the probabilistic fit between a data string, or a set of strings, and a HMM have long-since been worked out. The groundbreaking work of L.E. Baum in the 1960's led to the expectation-maximization (EM) method for locally optimizing HMM parameters. In 1967, Andrew Viterbi wrote a general algorithm for finding the optimal state pathway given a sequence [8]. Lawrence Rabiner's highly-cited 1989 tutorial [6] outlined the "Three Basic Problems" for HMMs (see box), and brought these techniques within reach of scientists not traditionally trained in probability theory, including even biologists (who then became known as "bioinformaticists"). Several good books on the subject are now available [1–5].

But the *algorithmology* of HMMs still has many unsolved problems, some of which are addressed in the current special issue. For example, the space of all possible directed graphs of size Q may be very, very large, far too large to be searched exhaustively. How do we find the optimal graph connectivity in a data-driven manner? How do we, simultaneously, define the number of states and initialize their emission probabilities, also in a data driven manner? One theoretical approach is presented in this issue (see the article by Li and Biswas). A related problem is to determine the Markov chain order, given only the data. In a first-order chain, the transition probability depends only on the current state; in a second order model it depends also on the previous two states, and so on. An answer is given in this issue (see the article by Boys and Henderson). Also, how do we impose constraints from domain-specific knowledge on the HMM topology? In this issue we present the special cases of psychological data and speech recognition (see articles by Visser et al., and by Abdulla, respectively).

The Three Basic Problems for HMMs [6]

Problem 1: Given the observation sequence $O = O_1O_2 \dots O_T$, and a model λ, how do we efficiently compute $P(O|\lambda)$?

ISSN 1058-9244/02/$8.00 © 2002 – IOS Press. All rights reserved

Problem 2: Given the observation sequence O, and the model λ, how do we choose a corresponding state sequence $Q = q_1q_2 \dots q_T$, which is optimal in some meaningful sense?

Problem 3: How do we adjust the model parameters λ to maximize $P(O|\lambda)$?

Outstanding problems remain. Given a HMM, how do we sum all of the self-avoiding pathways in a computationally efficient manner? How do we incorporate non-local covariance? How do we selectively prune the graph or grow new edges without overfitting? Once a topology is defined, the optimal parameters may be found using EM, but only if they don't differ too much from their initial values. How do we overcome this local optimum problem?

Once a problem is solved for one domain, it is solved for many, and in some cases (Baum-Welsh and Viterbi algorithms for example) it is completely general. By sharing our thoughts and standardizing our language across fields we can avoid "re-inventing the wheel" and thereby make faster progress in building useful probabilistic models. Presented here is one small contribution toward that overall goal.

References

[1] P. Baldi and S. Brunak, *Bioinformatics: the machine learning approach,* (2nd ed.), MIT Press, Cambridge, Mass, 2001.

[2] H. Bunke and T. Caelli, Hidden Markov models: applications in computer vision, *International journal of pattern recognition and artificial intelligence,* World Scientific, River Edge, NJ, 2001.

[3] R. Durbin, S. Eddy, A. Krogh and G. Mitchison, Biological sequence analysis, Cambridge University Press, Cambridge, 1998.

[4] R.J. Elliott, L. Aggoun and J.B. Moore, *Hidden Markov models: estimation and control,* Springer-Verlag, New York, 1995.

[5] I.L. MacDonald and W. Zucchini, *Hidden Markov and Other Models for Discrete-valued Time Series,* Monographs on Statistics and Applied Probability 70. 70 vols, Chapman & Hall, London, 1997.

[6] L.R. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, *Proc IEEE* **77** (1989), 257–286.

[7] O.B. Sheynin, A A Markov's Work on Probability, *Archive for History of Exact Science* **39** (1988), 337–377.

[8] A.J. Viterbi, Error bounds for convolutional codes and an asymptotically optimal decoding algorithm, *IEEE Trans. Informatic. Theory* **IT-13** (1967), 260–269.