Bayesian Regularization of Hidden Markov Models with an Application to Bioinformatics

Dirk Husmeier
Biomathematics and Statistics Scotland (BioSS)
SCRI, Dundee DD2 5DA, United Kingdom

Email: dirk@bioss.sari.ac.uk

This paper discusses a Bayesian approach to regularizing hidden Markov models and demonstrates an application of this scheme to Bioinformatics.

1 Introduction

Hidden Markov models (HMMs) are close relatives of neural networks (NNs) in many respects. Both models are composed of a network of *visible* and *hidden* nodes connected by edges, which can be described in the more general framework of probabilistic graphical models [5]. The forward-backward algorithm for training HMMs [10] shows striking resemblance to the backpropagation algorithm for NNs. Finally, many applications, especially in signal processing [2] and bioinformatics [1], employ hybrid schemes which combine HMMs and NNs. This article focuses on a problem common to both models. For sparse data, the classical training algorithms (mentioned above), which derive from a maximum likelihood (ML) approach, are sub-optimal due to overfitting. Following [6], much research on NNs in the last few years has explored Bayesian methods as a possible remedy in this respect. The objective of this article is to take a similar route and test if the generalization performance of HMMs can be improved by Bayesian free energy minimization.

2 Classical HMMs

Consider a vector of observations, $\mathbf{y}_t$, where t represents some ordered label, e.g., time in signal processing, or the site of a DNA strand in bioinformatics (see below). Corre-

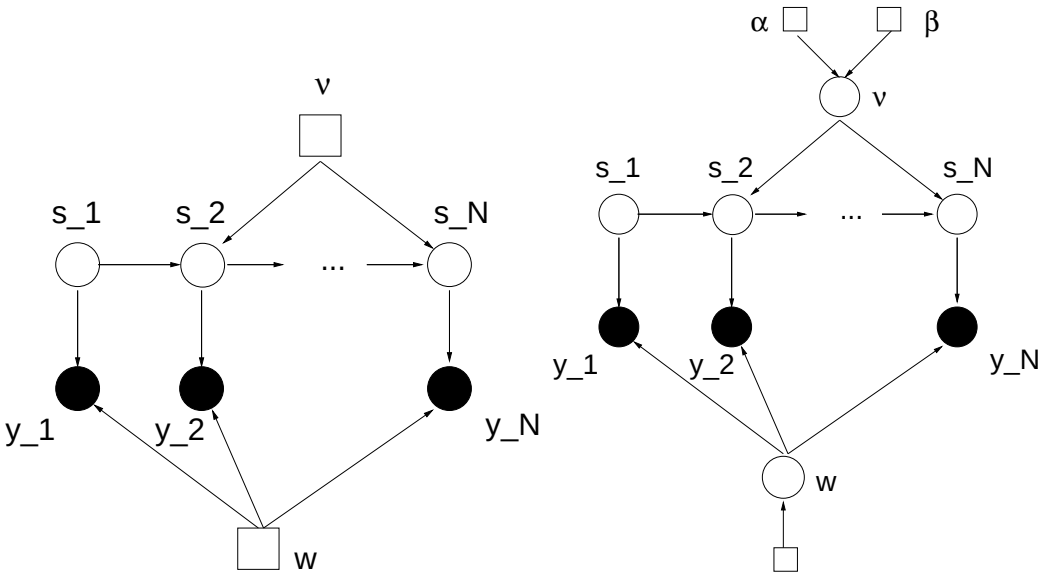

Figure 1: *Left:* HMM drawn as a probabilistic graphical model, where nodes represent random variables and arcs indicate conditional dependencies between these nodes. *Empty circles* show hidden states, *filled circles* represent vectors of observations. The parameters of the model are shown as squares. *Right:* A Bayesian approach to parameter adaptation. The parameters of the model, ν and $\mathbf{w}$, are themselves treated as random variables, whose prior distribution depends on further so-called hyperparameters (shown as squares).

lations between these observations are assumed to be indirect via unobserved, discrete *hidden states* $s_t \in \{1, \ldots, K\}$. The model is characterized by two sets of conditional independence relations [10]: (1) The observation at time or site t, $\mathbf{y}_t$, is independent of all other observations and states given s_t. (2) The state at t, s_t, is independent of the previous sequence of states given s_{t-1}. Using these independence relations, the joint probability for the sequence of states and observations can be factored as

$$P(\mathbf{Y}, \mathbf{s}) = P(\mathbf{y}_1, \ldots, \mathbf{y}_N, s_1, \ldots, s_N) = P(s_1) \prod_{t=2}^{N} P(s_t|s_{t-1}) \prod_{t=1}^{N} P(\mathbf{y}_t|s_t) \qquad (1)$$

The corresponding graphical model is shown in Figure 1. In general, the *transition probabilities* $P(s_t|s_{t-1})$ and the *emission probabilities* $P(\mathbf{y}_t|s_t)$ depend on some model parameters ν and $\mathbf{w}$. A standard training scheme is to optimize these parameters so as to maximize the likelihood $L = P(\mathbf{Y}|\nu, \mathbf{w}) = \sum_{\mathbf{s}} P(\mathbf{Y}, \mathbf{s}|\nu, \mathbf{w})$. This can be accomplished with the forward-backward and the Baum-Welch algorithm, as described in [10].

3 A Bayesian approach

3.1 General outline of the methodology

A possible shortcoming of the maximum likelihood (ML) method is its susceptibility to overfitting when the data are sparse. We therefore adopt a Bayesian approach, which has proven to improve the generalization performance of neural networks [6], [9], and treat the parameters ν and $\mathbf{w}$ as random variables. Starting from some prior distribution $P(\nu, \mathbf{w})$, the ultimate objective of learning is to determine the posterior distribution $P(\mathbf{s}, \mathbf{w}, \nu|\mathbf{Y})$. As this is analytically intractable, we need to make certain approximations. We can sample from the posterior distribution numerically with *Markov chain Monte Carlo* (MCMC). A good overview of this method can be found in [4]. Alternatively, we can assume a simpler, analytically tractable distribution $Q(\mathbf{s}, \mathbf{w}, \nu)$ and adapt its parameters so as to minimize the Kullback-Leibler divergence between Q and the true posterior. This so-called method of *variational free energy minimization* has become very popular in the context of learning in graphical models [5]. A first application to HMMs can be found in [7].

We will constrain our approximating distribution to be separable, such that[1]

$$Q(\mathbf{s}, \mathbf{w}, \nu) \quad = \quad Q(\mathbf{s})Q(\mathbf{w})Q(\nu) \tag{2}$$

The closeness of this distribution to the true posterior is measured by the Kullback-Leibler divergence

$$F(Q) \quad = \quad \sum_{\mathbf{s}} \int d\mathbf{w} \int d\nu Q(\mathbf{s}, \mathbf{w}, \nu) \ln \left[\frac{Q(\mathbf{s}, \mathbf{w}, \nu)}{P(\mathbf{Y}, \mathbf{s}, \mathbf{w}, \nu)} \right] \tag{3}$$

where

$$P(\mathbf{Y}, \mathbf{s}, \mathbf{w}, \nu) \quad = \quad \prod_{t=2}^{N} P(s_t|s_{t-1}, \nu) \prod_{t=1}^{N} P(\mathbf{y}_t|s_t, \mathbf{w}) P(s_1) P(\nu) P(\mathbf{w}) \tag{4}$$

and we have introduced $P(\nu)$ and $P(\mathbf{w})$ as priors on ν and $\mathbf{w}$. Our objective is to optimize $Q(\mathbf{s}, \mathbf{w}, \nu)$ so as to minimize F. This can be accomplished in an iterative scheme, where each of the marginal distributions $Q(\mathbf{s})$, $Q(\mathbf{w})$, $Q(\nu)$ is optimized separately while keeping the remaining two distributions fixed.

[1] We use the convention that different arguments indicate different probability distributions.

3.2 Explicit form of the update scheme

We here assume a specific form for the transition probabilities, first suggested in [8]:

$$P(s_t|s_{t-1}) \quad = \quad \nu\delta(s_t, s_{t-1}) + \frac{1-\nu}{K-1}[1-\delta(s_t, s_{t-1})] \tag{5}$$

This choice is motivated by the phylogenetic problem of detecting recombinations in DNA sequence alignments, where $1-\nu$ can be interpreted as a recombination probability (see below). Since ν is a binomial random variable, the natural conjugate prior is a beta distribution:

$$P(\nu) \; = \; D(\nu|\alpha,\beta) \; := \; \frac{1}{Z}\nu^{\alpha-1}(1-\nu)^{\beta-1} \tag{6}$$

where $\alpha, \beta > 0$ are hyperparameters that determine the mean and the dispersion of the distribution, while Z is a normalization factor. The resulting model is depicted on the right of Figure 1.

First, we consider $F(Q)$ as a functional of $Q(\nu)$ with $Q(\mathbf{s})$ and $Q(\mathbf{w})$ fixed. Introducing the definitions

$$\Psi \; = \; \sum_{\mathbf{s}} \sum_{t=2}^{N} Q(\mathbf{s})\delta(s_t, s_{t-1}) \qquad \tilde{\alpha} \; = \; \alpha + \Psi \qquad \tilde{\beta} \; = \; \beta + N - 1 - \Psi \tag{7}$$

and dropping terms that are independent of ν, this gives

$$\begin{aligned} F[Q(\nu)] \quad &= \quad \int d\nu Q(\nu) \left[\ln Q(\nu) - \sum_{\mathbf{s}} Q(\mathbf{s}) \sum_{t=2}^{N} \ln P(s_t|s_{t-1},\nu) \; - \; \ln D(\nu|\alpha,\beta) \right] \\ &= \quad \int d\nu Q(\nu) \left[\ln Q(\nu) - \Psi \ln \nu - (N-1-\Psi)\ln(1-\nu) - \ln D(\nu|\alpha,\beta) \right] \\ &= \quad \int d\nu Q(\nu) \ln \left[\frac{Q(\nu)}{D(\nu|\tilde{\alpha},\tilde{\beta})} \right] + C \end{aligned} \tag{8}$$

This expression is minimized for

$$Q(\nu) \quad = \quad D(\nu|\tilde{\alpha},\tilde{\beta}) \tag{9}$$

where $D(.)$ denotes the beta distribution, defined in (6). Second, we have to optimize $Q(\mathbf{w})$ with $Q(\mathbf{s})$ and $Q(\nu)$ fixed. At the current stage this has not been made explicit, though, and we rather assume a simple delta distribution located at the ML estimate $\hat{\mathbf{w}}$: $Q(\mathbf{w}) = \delta(\mathbf{w} - \hat{\mathbf{w}})$. Finally, the optimal distribution over the hidden states s_t for

fixed $Q(\mathbf{w})$ and $Q(\nu)$ is given, after some algebra[2], by:

$$Q(\mathbf{s}) \quad = \quad \frac{1}{Z_{\mathbf{s}}} P(s_1) \prod_t a(s_{t+1}, s_t) \prod_t P(\mathbf{y}_t | s_t, \hat{\mathbf{w}}) \tag{10}$$

in which $Z_{\mathbf{s}}$ is a normalization constant, and where we have defined:

$$\ln a(s_t, s_{t-1}) = \delta(s_t, s_{t-1}) \Upsilon(\tilde{\alpha}) - \Upsilon(\tilde{\alpha} + \tilde{\beta}) + [1 - \delta(s_t, s_{t-1})] \Big(\Upsilon(\tilde{\beta}) - \ln(K-1) \Big) \tag{11}$$

$$\Upsilon(x) = \frac{d}{dx} \ln \Gamma(x) = \frac{d}{dx} \ln \int_0^\infty u^{x-1} \exp(-u) du \tag{12}$$

Note that, after a sufficient number of iterations, $Q(\mathbf{s})$ converges to the optimal posterior distribution $P(\mathbf{s}|\mathbf{Y})$, and the most likely state sequence is given by maximizing eqn.(10).

4 Simulation experiments

4.1 A synthetic toy problem

Consider a rogue casino in which a fair die is occasionally replaced by a loaded one. The latter has probability 0.5 of a six and probability 0.1 for the numbers one to five. The visitor to the casino observes a sequence of rolls, $\mathbf{Y} = (\mathbf{y}_1, \ldots, \mathbf{y}_N)$, but he does not know which rolls used a loaded (L) die and which used a fair (F) one. The nature of the die in the tth roll is represented by the hidden state $s_t \in \{L, F\}$, and the objective is to infer the true sequence of states $\mathbf{s} = (s_1, \ldots, s_N)$ from the sequence of observations. We parameterized the transition probabilities according to eqn.(5) with a value of $\nu = 0.9$, and generated 20 training sets of length $N_{train} = 50$ from the true model. The emission parameters were kept at their correct values, while three different training schemes were applied for adapting the transition parameter (always starting from a neutral value of $\nu = 0.5$): (1) ML, (2) the Bayesian scheme with a fairly vague prior on ν ($\alpha = 2.5, \beta = 1.5$, corresponding to a mean of $\mu = 0.625$ and a standard deviation of $\sigma = 0.22$), (3) the Bayesian scheme using a more narrow prior with a higher probability mass in the region around the correct value ($\alpha = 17$, $\beta = 3$, corresponding to $\mu = 0.85$, $\sigma = 0.08$). The prediction performance of the 20 ensuing models was assessed on an independent test set of length $N_{gen} = 1000$. The results are shown in Figure 2 and suggest that both Bayesian training schemes lead to a significant improvement over results obtained with ML.

[2]See `http://www.bioss.sari.ac.uk/~dirk/` for a complete derivation.

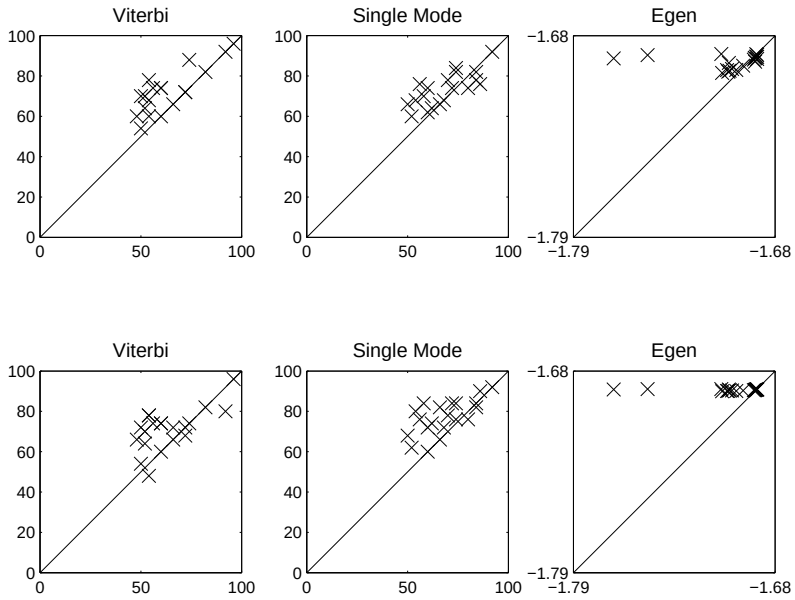

Figure 2: Comparison between training simulations with ML *(horizontal axis)* and the variational Bayesian approach *(vertical axis)*. Two different prior distributions were chosen. *Top row:* Informative prior, $\alpha = 17, \beta = 3$. *Bottom row:* Vague prior, $\alpha = 2.5, \beta = 1.5$. The methods were compared with three performance criteria. *Left column:* Percent correct classification of hidden states, as predicted by the mode of the joint distribution $P(\mathbf{s}|\mathbf{Y}) = P(s_1, \ldots, s_N|\mathbf{Y})$ (Viterbi path). *Middle column:* Percent correct classification based on the single-site mode, $P(s_t|\mathbf{y}_t)$. *Right column:* Generalization performance in terms of the normalized test-set log likelihood, $E_{gen} = \frac{1}{N} \ln P(\mathbf{Y})$. The crosses compare the performance of the ML *(horizontal axis)* and the Bayesian approaches *(vertical axis)* for the various training sets. The diagonal line indicates an equal performance of the two methods. Crosses above this line represent simulations for which the Bayesian approach outperforms the ML method.

4.2 Detection of recombination in DNA sequence alignment

The second study is related to the bioinformatics problem of estimating phylogenetic trees from DNA sequence alignments. A phylogenetic tree is a directed acyclic graph, whose topology conveys information about evolutionary relatedness between different species, whereas the edge lengths indicate evolutionary time in terms of the average number of mutations per site. Standard training methods aim to optimize the topology and the edge lengths by maximizing the likelihood of a given DNA sequence alignment [1], [3]. This approach, however, cannot deal with *recombination*, which is the exchange of DNA subsequences between different strains or species and, as discussed in [8], corresponds to a change of the tree topology in the affected region. We here follow [8] and model the process with an HMM, where each state s_t represents a different topology,

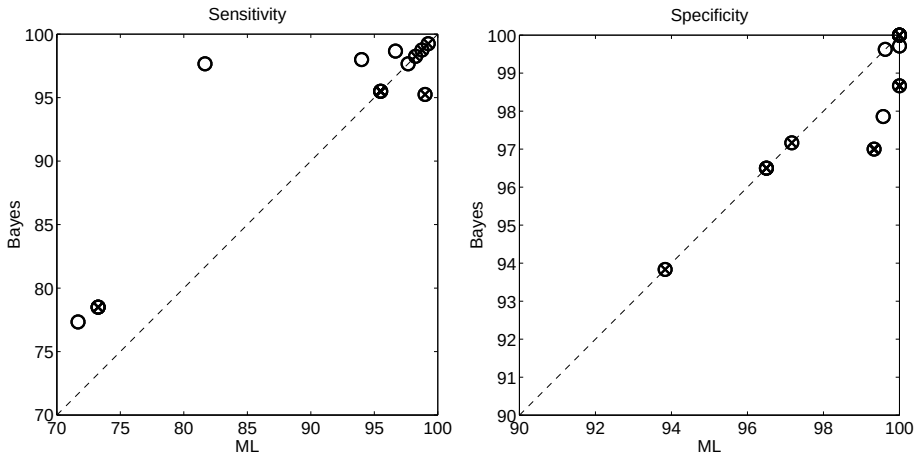

Figure 3: Comparison of the sensitivities *(left)* and specificities *(right)* obtained with the ML *(horizontal axis)* and the Bayesian *(vertical axis)* training schemes. Each cross or circle represents a different combination of training and test sets, where the difference in the symbols is related to the different lengths of the recombinant regions in the respective test set (crosses: 200, 200; circles: 50, 250). The diagonal dashed line indicates an equal performance of the ML and the Bayesian scheme, symbols above this line indicate a performance improvement as a result of applying the Bayesian scheme (and vice versa).

while the emission parameters are given by the branch lengths of the tree.

In our experiment, we simulated the evolution and recombination processes in a 4-species tree according to the way described in [3] and [8]. All synthetic DNA sequence alignments contained 1000 sites and two recombinant zones, but varied in the edge lengths of the true tree and the locations and lengths of the recombinant regions. Each of the data sets in turn was used for training an HMM with either ML or according to the Bayesian scheme described above [using two different settings of the hyperparameters: $(\alpha, \beta) = (10, 10), (150, 50)$]. The prediction performance was subsequently assessed on the remaining data sets not used for training, where we chose two performance measures. From the Viterbi path (that is, the mode of $P(\mathbf{s}|\mathbf{Y}) = P(s_1, \ldots, s_N|\mathbf{Y})$) we determined the true prediction rate for detecting recombinant sites (the *sensitivity*) as well as the true prediction rate for non-recombinant sites (the *specificity*). A comparison of the performance between training with ML and the Bayesian scheme is shown in Figure 3.[3]

[3]This comparison is based on the more informative prior, $(\alpha, \beta) = (150, 50)$. For the vaguer prior, $(\alpha, \beta) = (10, 10)$, the pattern in the plots is similar, but less pronounced.

5 Discussion

While the Bayesian approach clearly outperformed ML on the synthetic toy problem, the results on the phylogenetic data sets are less clear-cut. The simulations suggest that the Bayesian approach improves the *sensitivity* for detecting recombinations at the cost of a slight degradation in the *specificity*. Since the values for the latter are usually quite high (over 95%), the improvement in the former seems to be more relevant for practical applications. Further simulation studies are required to clarify this issue.

References

[1] P. Baldi and P. Brunak. *Bioinformatics - The Machine Learning Approach*. MIT Press, 1988.

[2] Y. Bengio and P. Frasconi. Input-Output HMMs for Sequence Processing. *IEEE Transactions on Neural Networks*, 7(5), 1996.

[3] R. Durbin, S. R. Eddy, A. Krogh, and Mitchison G. *Biological sequence analysis. Probabilistic models of proteins and nucleic acids*. Cambridge University Press, Combridge, UK, 1998.

[4] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter. *Markov Chain Monte Carlo in Practice*. Chapman and Hall, Suffolk, 1996. ISBN: 0-412-05551-1.

[5] M. I. Jordan, Z. Ghahramani, T. S. Jaakola, and L. K. Saul. An Introduction to Variational Methods for Graphical Models. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 105–161, Cambridge, MA, 1999. MIT Press.

[6] D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

[7] D.J.C. MacKay. Ensemble Learning for Hidden Markov Models. Technical report, Cavendish Laboratory, Cambridge CB3 0HE, UK, 1998.

[8] G. McGuire. *Statistical Methods for DNA Sequences: Detection of Recombination and Distance Estimation*. PhD thesis, University of Edinburgh, 1998.

[9] R. M. Neal. *Bayesian Learning for Neural Networks*, volume 118 of *Lecture Notes in Statistics*. Springer, New York, 1996. ISBN 0-387-94724-8.

[10] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.