Journal of Machine Learning Research 15 (2014) 1-58 Submitted 7/10; Revised 7/12; Published 1/14

Bridging Viterbi and Posterior Decoding: A Generalized Risk Approach to Hidden Path Inference Based on Hidden Markov Models

Jüri Lember JURI.LEMBER@UT.EE
Institute of Mathematical Statistics
Tartu University
J. Liivi 2-507, Tartu, 50409, Estonia

Alexey A. Koloydenko ALEXEY.KOLOYDENKO@RHUL.AC.UK
Department of Mathematics
Royal Holloway University of London
Egham, TW20 0EX, UK

Editor: Richard Maclin

Abstract

Motivated by the unceasing interest in hidden Markov models (HMMs), this paper re-examines hidden path inference in these models, using primarily a risk-based framework. While the most common *maximum a posteriori* (MAP), or Viterbi, path estimator and the *minimum error*, or *Posterior Decoder* (PD) have long been around, other path estimators, or decoders, have been either only hinted at or applied more recently and in dedicated applications generally unfamiliar to the statistical learning community. Over a decade ago, however, a family of algorithmically defined decoders aiming to hybridize the two standard ones was proposed elsewhere. The present paper gives a careful analysis of this hybridization approach, identifies several problems and issues with it and other previously proposed approaches, and proposes practical resolutions of those. Furthermore, simple modifications of the classical criteria for hidden path recognition are shown to lead to a new class of decoders. Dynamic programming algorithms to compute these decoders in the usual forward-backward manner are presented. A particularly interesting subclass of such estimators can be also viewed as hybrids of the MAP and PD estimators. Similar to previously proposed MAP-PD hybrids, the new class is parameterized by a small number of tunable parameters. Unlike their algorithmic predecessors, the new risk-based decoders are more clearly interpretable, and, most importantly, work "out-of-the box" in practice, which is demonstrated on some real bioinformatics tasks and data. Some further generalizations and applications are discussed in the conclusion.

Keywords: admissible path, decoder, HMM, hybrid, interpolation, MAP sequence, minimum error, optimal accuracy, power transform, risk, segmental classification, symbol-by-symbol, posterior decoding, Viterbi algorithm

1. Introduction

Besides their classical and traditional applications in signal processing and communications (Viterbi, 1967; Bahl et al., 1974; Hayes et al., 1982; Brushe et al., 1998) (see also further references in Cappé et al., 2005) and speech recognition (Huang et al., 1990; Jelinek, 1976,

2001; McDermott and Hazen, 2004; Ney et al., 1994; Padmanabhan and Picheny, 2002; Rabiner and Juang, 1993; Rabiner et al., 1986; Shu et al., 2003; Steinbiss et al., 1995; Ström et al., 1999), hidden Markov models have recently become indispensable in computational biology and bioinformatics (Burge and Karlin, 1997; Durbin et al., 1998; Eddy, 2004; Krogh, 1998; Brejová et al., 2007b; Majoros and Ohler, 2007) as well as in natural language modeling (Manning and Schütze, 1999; Vogel et al., 1996) and information security (Mason et al., 2006).

At the same time, their spatial extensions, known as hidden Markov random field models (HMRFM), have been immensely influential in spatial statistics (Besag and Green, 1993; Green and Richardson, 2002; Künsch et al., 1995; McGrory et al., 2009), and particularly in image analysis, restoration, and segmentation (Besag, 1986; Geman and Geman, 1984; Li et al., 2000; Marroquin et al., 2003; Winkler, 2003). Indeed, hidden Markov models have been called 'one of the most successful statistical modeling ideas that have [emerged] in the last forty years' (Cappé et al., 2005).

HM(RF)Ms owe much of their success to the following: The posterior distribution of the hidden layer inherits the Markov property from the prior distribution (although the posterior distribution is generally inhomogeneous even if the prior distribution is homogeneous). At the same time, the marginal law of the observed layer can still include global, that is non-Markovian, dependence, hence the richness of the observed system (Künsch et al., 1995).

The Markov property of the posterior distribution and the conditional independence of the observed variables given the hidden ones, have naturally led to a number of computationally feasible methods for inference about the hidden realizations as well as model parameters. HMMs are also naturally a special case of *graphical models* (Lauritzen, 1996; Bishop, 2006, Chap. 8).

HMMs, or one dimensional HMRFMs, have been particularly popular not least due to the fact that the linear order of the indexing set (usually associated with time) makes exploration of hidden realizations relatively straightforward from the computational viewpoint. In contrast, higher dimensional HMRFMs generally require approximate, possibly stochastic, techniques in order to compute optimal configurations of the hidden field (Cocozza-Thivent and Bekkhoucha, 1993; Joshi et al., 2006; Winkler, 2003; McGrory et al., 2009). In particular, a *maximum a posteriori* (MAP) estimator of the hidden layer of an HMM is efficiently and exactly computed by a dynamic programming algorithm bearing the name of Viterbi, whereas a general higher dimensional HMRFM would employ, for example, a simulated annealing type method (Geman and Geman, 1984; Winkler, 2003) to produce approximate solutions to the same task.

There are also various useful extensions of the ordinary HMM, such as variable duration semi-Markov models, coupled HMMs (Brand et al., 1997), and factorial HMMs (Bishop, 2006, Chap. 13), etc. All of the material in this paper is applicable to those extensions in a straightforward way. However, to simplify the exposition we focus below on the ordinary HMM.

1.1 Notation and Main Ingredients

We adopt the machine and statistical learning convention, referring to the hidden and observed processes as Y and X, respectively, in effect reversing the convention that is more

commonly used in the HMM context. Thus, let $Y = \{Y_t\}_{t\geq 1}$ be a Markov chain with state space $S = \{1, \ldots, K\}$, $K > 1$, and initial probabilities $\pi_s = P(Y_1 = s)$, $s \in S$. Although we include inhomogeneous chains in most of what follows, for brevity we will still be suppressing the time index wherever this does not cause ambiguity. Hence, we write $\mathbb{P} = (p_{ij})_{i,j\in S}$ for all transition matrices. Let $X = \{X_t\}_{t\geq 1}$ be a process with the following properties. First, given $\{Y_t\}_{t\geq 1}$, the random variables $\{X_t\}_{t\geq 1}$ are conditionally independent. Second, for each $t = 1, 2, \ldots$, the distribution of X_t depends on $\{Y_t\}_{t\geq 1}$ (and t) only through Y_t. The process X is sometimes called the *hidden Markov process* (HMP) and the pair (Y, X) is referred to as a *hidden Markov model* (HMM). The name is motivated by the assumption that the process Y (sometimes called a *regime*) is generally non-observable. The conditional distribution of X_1 given $Y_1 = s$ is called an *emission distribution*, written as P_s, $s \in S$. We shall assume that the emission distributions are defined on a measurable space $(\mathcal{X}, \mathcal{B})$, where $\mathcal{X}$ is usually $\mathbb{R}^d$ and $\mathcal{B}$ is the corresponding Borel σ-algebra. Without loss of generality, we assume that the measures P_s have densities f_s with respect to some reference measure λ, such as the counting or Lebesgue measure.

Given a set $\mathcal{A}$, integers m and n, $m < n$, and a sequence $a_1, a_2, \ldots \in \mathcal{A}^\infty$, we write a_m^n for the subsequence $(a_m, \ldots, a_n)$. When $m = 1$, it will be often suppressed. Thus, $x^T := (x_1, \ldots, x_T)$ and $y^T := (y_1, \ldots, y_T)$ stand for the fixed observed and unobserved realizations, respectively, of the HMM $(X_t, Y_t)_{t\geq 1}$ up to time $T \geq 1$. Any sequence $s^T \in S^T$ is called a *path*. This parallel notation (that is, s^T in addition to y^T) is necessitated largely by our forthcoming discussion of various loss functions, which do require two arguments. We shall denote the joint probability density of (x^T, y^T) by $p(x^T, y^T)$, that is,

$$p(x^T, y^T) := \mathbf{P}(Y^T = y^T) \prod_{t=1}^{T} f_{y_t}(x_t).$$

To make mathematical expressions more compact, we overload the notation when this causes no ambiguity. Thus, $p(s^T)$ stands for the probability mass function $\mathbf{P}(Y^T = s^T)$ of path s^T, and $p(x^T)$ stands for the (unconditional) probability density function $\sum_{s^T\in S^T} p(x^T, s^T)$ of the observed data x^T. Furthermore, we write $p_t(s)$ and $p_t(s \mid x^T)$ for $\mathbf{P}(Y_t = s)$ and $\mathbf{P}\left(Y_t = s \mid X^T = x^T\right)$, respectively. It is standard (see Bishop, 2006, Chap. 13; Ephraim and Merhav, 2002; Cappé et al., 2005) in this context to define the so-called *forward* and *backward* variables

$$\alpha_t(s) := p(x^t \mid Y_t = s)P(Y_t = s), \quad \beta_t(s) := \begin{cases} 1, & \text{if } t = T \\ p(x_{t+1}^T \mid Y_t = s), & \text{if } t < T \end{cases}, \tag{1}$$

where $p(x^t \mid Y_t = s)$ and $p(x_{t+1}^T \mid Y_t = s)$ are the conditional densities of the data segments x^t and x_{t+1}^T, respectively, given $Y_t = s$.

1.2 Path Estimation

Our focus here is estimation of the hidden path y^T. This task can also be viewed as *segmentation* of the data sequence into regions with distinct class labels (Lember et al., 2011). Treating y^T as missing data (Rabiner, 1989), or parameters, a classical and by far the most popular solution to this task is to maximize $p(x^T, s^T)$ in $s^T \in S^T$. Often, especially

in the digital communication literature (Lin and Costello Jr., 1983; Brushe et al., 1998), $p(x^T, s^T)$ is called the *likelihood function* which might become potentially problematic in the presence of any genuine model parameters. Such "maximum likelihood" paths are also called *Viterbi paths* or *Viterbi alignments* after the Viterbi algorithm (Viterbi, 1967; Rabiner, 1989) commonly used for their computation. If $p(s^T)$ is thought of as the prior distribution of Y^T, then the Viterbi path also maximizes $p(s^T \mid x^T) := \mathbf{P}(Y^T = s^T \mid X^T = x^T)$, the probability mass function of the posterior distribution of Y^T, hence the term '*maximum a posteriori (MAP) path*'.

In spite of its computational attractiveness, inference based on the Viterbi paths may be unsatisfactory for a number of reasons, including its sub-optimality with regard to the number of correctly estimated states y_t. Also, using the language of information theory, there is no reason to expect a Viterbi path to be typical (Lember and Koloydenko, 2010). Indeed, "there might be many similar paths through the model with probabilities that add up to a higher probability than the single most probable path" (Käll et al., 2005). The fact that a MAP estimate need not be representative of the posterior distribution has also been recently discussed in a more general context by Carvalho and Lawrence (2008). Atypicality of Viterbi paths particularly concerns situations when estimation of y^T is combined with inference about model parameters, such as the transition probabilities p_{ij} (Lember and Koloydenko, 2010). Even when estimating, say, the probability of heads from independent tosses of a biased coin, we naturally hope to observe a typical realization and not the constant one of maximum probability.

An alternative and very natural way to estimate y^T is by maximizing the posterior probability $p_t(s \mid x^T)$ of each individual hidden state Y_t, $1 \le t \le T$ (Bahl et al., 1974). We refer to the corresponding estimator as *pointwise maximum a posteriori (PMAP)*. PMAP is well-known to maximize the expected number of correctly estimated states (Section 2), hence the characterization '*optimal accuracy*' (Holmes and Durbin, 1998). In statistics, especially spatial statistics and image analysis, this type of estimation is known as *Marginal Posterior Mode* (Winkler, 2003) or *Maximum Posterior Marginals* (Rue, 1995) (MPM) estimation. This is also known as the *posterior decoding* (PD) in computational biology (Brejová et al., 2007b) and machine translation (Ganchev et al., 2008), and has been reported to be particularly successful in pairwise sequence alignment (Holmes and Durbin, 1998) and when more than one path has its posterior probability as "high" or nearly as "high" as that of the Viterbi path (Eddy, 2004). In the wider context of biological applications of discrete high-dimensional probability models, this has also been called *consensus* estimation, and in the absence of constraints, *centroid* estimation (Carvalho and Lawrence, 2008). In communications applications of HMMs, largely influenced by the BCJR algorithm (Bahl et al., 1974), the terms '*optimal symbol-by-symbol detection*' (Hayes et al., 1982), '*symbol-by-symbol MAP estimation*' (Robertson et al., 1995), and '*MAP state estimation*' (Brushe et al., 1998) have been used for this. Remarkably, even before observing the data, optimal accuracy (that is, based on the prior instead of the posterior distribution) decoding can still be more accurate than the Viterbi decoding (Subsection 5.4).

1.2.1 How Different are PMAP and MAP Inferences and How Much Room is in between the Two?

This is a natural question in both practice and theory, especially for anyone interested in improving performance of applications based on these methods while maintaining their computational attractiveness.

A not so uncommon misconception that the difference between PMAP and Viterbi inferences is negligible may in part be explained by the concluding remark made by Bahl et al. (1974) in the special context of linear codes: "Even though Viterbi decoding is not optimal in the sense of bit error rate, in most applications of interest the performance of both [PMAP and Viterbi] algorithms would be effectively identical." This conclusion may in turn be explained by the dominance of binary chains in the telecommunication applications, and the binary state space indeed leaves too little room for the two inferences to differ. However, as HMMs with larger state spaces gained more prominence, it became clear that appreciable differences between the PMAP and Viterbi inferences do occur (see, for example, Ganchev et al., 2008). In fact, already two decades after Bahl et al. (1974), Brushe et al. (1998) contemplated hybridization of the PMAP and Viterbi decoders, writing "Indeed, there may be applications where a delicate performance dependence exists between [the Viterbi and PMAP] estimates. In such cases, the use of a hybrid scheme ... may result in performance gains." We return to their idea later in this paper.

Although interesting comparisons of the PMAP and Viterbi decoders on special tasks (e.g., Ganchev et al., 2008), have been recently reported, we are not aware of any systematic general studies of the two decoders that would exploit such comparisons in order to design new interesting hybrid schemes. Soon after the first version of this article was posted on `arXiv`, however, Yau and Holmes (2010) reported similar interests in this subject, supported by real and simulated examples. Of course, it has long been well-known (Rabiner, 1989) that despite being optimal in the sense of maximizing the expected number of correctly estimated states, a PMAP path can at the same time have very low, possibly zero, probability. Thus, on the logarithmic scale, the difference in path probabilities between the PMAP and Viterbi decoders can easily be *infinite*. In Section 5, we give a real data example with only six hidden states to show that besides the infinite difference in the log-probabilities, the two decoders can differ significantly (by more than 13%) in accuracy. This could have been expected if the data were indeed generated by an HMM and if that same HMM were used for decoding. However, when the model is misspecified, which is very common in practice, empirical performance measures, such as the symbol-by-symbol error rate, are generally biased as estimators of corresponding model based expected performance measures. In particular, in such situations there is no guarantee that the PMAP decoding is empirically more accurate than MAP. Although these points are fairly straightforward, we felt, especially during the reviewing process, that some readers might still appreciate a concrete illustration, which we give in Section 5. Other readers can simply glance over Section 5 without interrupting the overall flow of the manuscript.

It is actually not difficult to constrain the PMAP decoder to *admissible* paths (Subsection 2.2.1), where admissibility is defined relative to the posterior distribution. Specifically, given x^T, a path y^T is called *admissible if its posterior probability* $p(y^T \mid x^T)$ *is defined and positive*, that is, if $p(x^T, y^T) > 0$. We then point out that constraining the PMAP decoder

to the paths of positive prior probability, as already done by others (see more below), is not sufficient (albeit necessary) for admissibility of the PMAP paths. Note that in a slightly more general form allowing for state aggregation, Käll et al. (2005) do exactly this, that is, force PMAP paths to have positive prior probability, referring to the result as "a possible path through the model". Thus, Käll et al. (2005) appear to ignore that having a positive prior probability is *not sufficient in general for a PMAP path to be "a possible path through the model"*, unless, of course, "the model" is to be understood as the hidden Markov chain only and not the whole HMM. We will refer to the PMAP decoder constrained to the admissible paths as the *admissibly constrained PMAP*, or, simply *constrained PMAP*. This also details and clarifies our earlier discussion of admissibility (Lember et al., 2011, Section 2), which, like Rabiner (1989); Käll et al. (2005), also ignored the distinction between *a priori* and *a posteriori* modes of admissibility.

A variation on the same idea of making PMAP paths admissible has been applied for prediction of membrane proteins, giving rise to the *posterior Viterbi decoding (PVD)* (Fariselli et al., 2005). PVD, however, maximizes the product $\prod_{t=1}^{T} p_t(s_t \mid x^T)$ (Fariselli et al., 2005) (and also Equation 9 below) and not the sum $\sum_{t=1}^{T} p_t(s_t \mid x^T)$, whereas the two criteria are *no longer equivalent in the presence of path constraints* (Subsection 2.2.1). While acknowledging this latter distinction between their decoder and PVD and not distinguishing between the prior and posterior modes of admissibility, Käll et al. (2005) appear to be unaware of the other distinction between their decoder and PVD: PVD paths are guaranteed to be of not only positive prior probability but also of positive posterior probability, that is, admissible (in our sense of the term). Holmes and Durbin (1998) proposed a PMAP decoder to compute optimal pairwise sequence alignments. Holmes and Durbin (1998) used the term "legitimate alignment", which suggests admissibility, but the description of their algorithm (Holmes and Durbin, 1998, Section 3.8) appears to be insufficiently detailed to verify if the output is guaranteed to be admissible, or only of positive prior probability, or, if inadmissible solutions are altogether an issue in that context.

Our own experiments (Section 5) show that both PVD and constrained PMAP decoder can return paths of very low (posterior) probabilities. Moreover, in many applications, for example, gene identification and protein secondary structure prediction, the pointwise (e.g., nucleotide level) error rate is not necessarily the main measure of accuracy (see also Subsection 1.2.2 below), hence the constrained PMAP need not be an ultimate answer in that respect either. Together with the above problem of atypicality of MAP paths, this has been addressed by moving from single path inference towards *envelopes* (Holmes and Durbin, 1998). Thus, for example, in computational biology a common approach would be to aggregate individual states into a smaller number of semantic labels (e.g., codon, intron, intergenic). In effect, this would realize the notion of path similarity by mapping many "similar" state paths to a single label path, or *annotation* (Krogh, 1997; Käll et al., 2005; Fariselli et al., 2005; Brejová et al., 2007b). However, since this mapping would usually be many-to-one (what Brejová et al., 2007a refer to as the "multiple path problem"), the annotation of the Viterbi path would generally be inferior to the optimal (in the MAP sense) annotation. On the other hand, to compute the MAP annotation in many practically important HMMs can be NP-hard (Brejová et al., 2007a) (which is not surprizing given that the coarsened hidden chain on the set of labels is generally no longer Markov). Unlike the Viterbi/MAP decoder, the PMAP decoder, owing it to its symbol-by-symbol

nature, handles annotations as easily as it does state paths, including the enforcement of admissibility. Interpreting admissibility relative to the prior distribution, this was shown by Käll et al. (2005), and this paper extends their result to admissible (that is, of positive posterior probability) paths and indicates further extensions (Section 8).

A number of alternative heuristic approaches are also known in computational biology, but none appears to be fully satisfactory (Brejová et al., 2007b). Overall, although the original Viterbi decoder has still been the most popular paradigm in many applications, and in computational biology in particular, alternative approaches have often demonstrated significantly better performance, for example, in predicting various biological features. For example, Krogh (1997) suggested the *1-best* algorithm for optimal labeling. More recently, Fariselli et al. (2005) have demonstrated PVD to be superior to the 1-best algorithm, and, not surprisingly, to the Viterbi and PMAP decoders, on tasks of predicting membrane proteins.

Thus, a starting point of this contribution was that restricting the PMAP decoder to admissible paths is but one of *numerous ways to combine the strong points of the MAP and PMAP path estimators*. Indeed, the popular seminal tutorial (Rabiner, 1989) briefly mentions maximization of the expected number of correctly decoded (overlapping) blocks of length two or three, rather than single states as a sensible remedy against vanishing probabilities (albeit leaving it unclear if prior or posterior probability was meant). With $k \geq 1$ and $\widehat{y^T}(k)$ being the block length and corresponding path estimate, respectively, this approach yields Viterbi inference as k increases to T (with $\widehat{y^T}(1)$ corresponding to PMAP). Therefore, this could be interpreted as discrete interpolation between the PMAP and Viterbi inferences. Intuitively, following Rabiner's logic, one might also expect $p(x^T, \widehat{y^T}(k))$ to increase with k. However, *this is not true* and it is possible for the decoder with $k = 2$ to produce an inadmissible (with the prior probability being also zero) path $\widehat{y^T}(2)$ while the PMAP path is admissible: $p(x^T, \widehat{y^T}(2)) = 0 = p(\widehat{y^T}(2)) < p(x^T, \widehat{y^T}(1))$. *We are not aware of this observation being previously made in the literature.* Moreover, our experiments in Section 5 show that this situation is far from being uncommon.

On a related note, concerned with the same deficiencies of the MAP and PMAP inferences, Yau and Holmes (2010) have most recently also used the decision-theoretic framework to allow for full asymmetry in the otherwise symmetric pairwise loss (Equation 30 below with $k = 2$) that underpins the $\widehat{y^T}(2)$ inference. This is no doubt a very natural extension to provide to the end user, and (partially) asymmetric pairwise losses had indeed been incorporated in a prominent web-server in the context of RNA secondary structure prediction (Sato et al., 2009).

Despite the possibility of $\widehat{y^T}(2)$ or its asymmetric siblings to be inadmissible, we find the idea of interpolation between the PMAP and Viterbi inferences very interesting. Besides Yau and Holmes (2010) acknowledging the need for intermediate modes of inference, to the best of our knowledge, the only published work that explicitly proposed such an interpolation is that of Brushe et al. (1998). However, the approach of Brushe et al. (1998) is algorithmic, which makes it difficult to interpret its paths in general and analyze their properties (e.g., asymptotic behavior in particular). More importantly, Brushe et al. (1998) claim that the family of their interpolating decoders will work in practice, which, as we explain in detail in Section 6, need not be true apart from trivial situations. Despite these

and other deficiencies of their approach, it raises some interesting questions and inspires interesting modifications, which we also discuss in Section 6. It had not been our original intention to dwell on the algorithmic approach in this manuscript as this approach is peripheral to the present theme of the risk-based approach. However, encouraged by some of the reviewers and taking into account their queries on and interest in that particular discussion, we have now made that discussion into a full section (Section 6), which might, however, appear somewhat hypertrophied to some readers.

1.2.2 Further Motivation

One other motivation for considering new decoders is that unlike the error rate or path probability, analytic optimization of other performance measures (e.g., Matthew's correlation Aydin et al., 2006, Q_2, Q_{ok}, SOV Fariselli et al., 2005, etc.) used in practice is difficult if at all possible. Having a large family of computationally efficient decoders, such as the new generalized hybrid decoders, and using some training data, one can select empirically a member from the family that optimizes the performance measure of interest. More generally, it seems advantageous for applications to be aware of the new choices of decoders and their properties.

Also, depending on the application, the emphasis sometimes shifts from purely automatic decoding with hard decisions to data exploration. Indeed, some performance measures may be hard to formalize and subsequently hard to compute. For example, an estimated path can be deemed correct if it is only structurally identical to the true path, say, conforming to the description "a long run of 1's followed by a short run of 2's followed by a long run of alternating 2's and 3's". It is then particularly valuable to gain insights into the topology of the state space in the sense of identifying compartments of high concentration of the posterior distribution. The significance of identifying clusters (of similar sequences) of high (total) posterior probability in high-dimensional discrete spaces has been recently discussed by Carvalho and Lawrence (2008), and a thorough discussion of the advantages of topological and geometric approaches to analysis of complex data in general has more recently been given by Carlsson (2009). Thus, it may be beneficial to output a family of related decodings instead of one or several ("N best") decodings that are optimal relative to a single criterion such as MAP. For instance, by slowly varying the optimization criterion (e.g., decreasing the penalty for false discovery of rare states or transitions), saliency of detections of interesting features can be assessed and a better understanding of a neighborhood of solutions can be gained (e.g., discerning between an "archipelago" and a "continent"), all without having to compute, or even define explicitly, a path similarity measure (such as those based on, for example, BLAST scores Altschul et al., 1990). At the same time, by varying the optimization criteria more aggressively, alternative structures might be encountered coming from neighborhoods of remote (say, in the Hamming distance sense) local maxima of the posterior distribution. Viewed within this context, this relatively inexpensive type of "neighborhood" inference might become alternative or complementary to the direct sampling (from the posterior distribution); see also Section 5 and Section 8.

1.3 Further Notation and Organization of the Rest of the Paper

In this paper, we consider the path inference problem in the more general framework of statistical learning. Namely, we consider sequence *classifier* mappings

$$g : \mathcal{X}^T \to S^T, \quad T = 1, 2, \ldots,$$

and optimality criteria for their selection. When all g's are obtained using the same decoding principle, or optimality criterion, regardless of T, we refer to them collectively as a *classification method*, or simply, *decoder*. This will be the case in this paper, and therefore we simplify the notation by writing $g(x^T)$ instead of $g(x^T; T)$ or the like. In Section 2, criteria for optimality of g are naturally formulated in terms of risk minimization whereby $R(s^T \mid x^T)$, the *risk of* of outputting path s^T, derives from a suitable *loss function*. A Bayes decoder, that is one that minimizes $R(g(x^T) \mid x^T)$ over all possible g, will be denoted by v with a suitable reference to the risk R. In Section 3, we consider families of risk functions which naturally generalize those corresponding to the Viterbi and PMAP solutions (Subsection 2.1). There we will need the full two argument notation $v(x^T; \cdot)$ using the second argument to single out an individual member of such a family. Furthermore, as shown in Section 4, these risk functions define a family of path decoders $v(x^T; k)$ parameterized by an integer k with $k = 1$ and $k \to \infty$ corresponding to the PMAP and Viterbi cases, respectively (Theorem 6). A continuous mapping via $k = 1/(1-\alpha)$, $0 \leq \alpha \leq 1$ compactifies this parameterization and further enriches the solution space by including fractional k. It is then discussed how the new family of decoders can be embedded into yet a wider class with a principled criterion of optimality. We also compare the new family of decoders with the Rabiner k-block approach. Any decoder would only be of theoretical interest if it could not be efficiently computed. In Section 3, we show that all of the newly defined decoders can be implemented efficiently as a dynamic programming algorithm in the usual forward-backward manner with essentially the same (computational as well as memory) complexity as the PMAP or Viterbi decoders (Theorem 4). Recent advances in the asymptotic theory of some of the main decoders and risks presented in this paper are reviewed in Section 7 together with sketches of how these may be relevant in practice. Various further extensions are discussed in the concluding Section 8.

1.4 Contributions of the Paper

We review HMM-based decoding within the sound framework of statistical decision theory, and do so notably more broadly than has been done before, for example, in the prominent work of Carvalho and Lawrence (2008). We also investigate thoroughly previous work on combining the desirable properties of the two most common decoders, that is the Viterbi and optimal accuracy decoders. In doing so, we discover several relevant claims and suggestions to be unjustified, misleading, or plainly incorrect. We explain in detail those deficiencies, giving relevant counterexamples, and show how they can be resolved. Some such resolutions are naturally left within the native frameworks of the originals, whereas others are more naturally given within the general risk-based framework. All of the resulting decoders are shown to be easily implementable within the usual forward-backward computational frameworks of the optimal accuracy and Viterbi decoders. We argue that the richness, flexibility,

and analytic interpretation of the resulting families of decoders offer new possibilities for applications and invite further theoretical analysis. Specifically, this paper

1) clarifies the definition of admissibility of hidden paths and shows that, when constrained to the paths of positive prior probability, the optimal accuracy decoding can still return inadmissible paths;

2) shows that the suggestion of Rabiner (1989) to maximize the expected rate of correctly recognized blocks can lead to inadmissible paths for blocks of size two, and therefore can be misleading;

3) proposes suitable risk functions to "repair" the above suggestion, and subsequently designs new families of computationally efficient decoders, providing an experimental illustration;

4) unifies virtually all of the key decoders within the same risk-based framework;

5) analyzes the relationships between the risks achieved by the different decoders, yielding a general result on convex decomposition of the key risk functionals for Markov chains;

6) analyzes the related earlier work of Brushe et al. (1998), and in particular:

(a) explains how the idea of hybridization of the Viterbi and optimal accuracy decoders proposed in the above work can fail when the Viterbi path is not unique;

(b) establishes that the claims made in the same work regarding the implementation of their algorithm to hybridize the Viterbi and optimal accuracy decoders are incorrect;

(c) shows how the corresponding forward and backward variables given in the same work can be scaled to produce an operational decoding algorithm;

(d) shows that the resulting decoders are different from the original hybrid decoders of Brushe et al. (1998);

(e) proposes an immediately operational algorithm to hybridize the Viterbi and optimal accuracy decoders (at least when the Viterbi path is unique), which is based on the more common power-transform, and which also allows for extrapolations "beyond" the optimal accuracy decoder;

7) indicates a number of further extensions of the new families of decoders.

At the same time, a thorough performance evaluation, including asymmetric variants of the main loss functions, and using several applications with their own performance measures, is outside the scope of this paper (Section 8).

2. Risk-Based Path Inference

Given a sequence of observations x^T with $p(x^T) > 0$, we view the (posterior) *risk* as a function

$$R(\cdot \mid x^T): \quad S^T \mapsto [0, \infty].$$

Naturally, we seek a state sequence with minimum risk: $v(x^T) := \arg\min_{s^T \in S^T} R\big(s^T \mid x^T\big)$. In the *statistical decision and pattern recognition theories*, the classifier v is known as the *Bayes classifier* (relative to risk R). Within the same framework, the risk is often specified via a *loss-function*

$$L : S^T \times S^T \to [0, \infty],$$

interpreting $L(s^T, y^T)$ as the loss incurred by the decision to predict s^T when the actual state sequence was y^T. Therefore, for any state sequence $s^T \in S^T$, the risk is given by

$$R(s^T \mid x^T) := E[L(s^T, Y^T) \mid X^T = x^T] = \sum_{y^T \in S^T} L(s^T, y^T) p(y^T \mid x^T).$$

2.1 Standard Path Inferences Re-Examined

The most popular loss function is the so-called *symmetrical* or *zero-one* loss L_∞ defined as follows:

$$L_\infty(s^T, y^T) = \begin{cases} 1, & \text{if } s^T \neq y^T; \\ 0, & \text{if } s^T = y^T. \end{cases}$$

We shall denote the corresponding risk by R_∞. With this loss, clearly

$$R_\infty(s^T \mid x^T) = \mathbf{P}(Y^T \neq s^T \mid X^T = x^T) = 1 - p(s^T \mid x^T), \tag{2}$$

thus $R_\infty(\cdot \mid x^T)$ is minimized by a Viterbi path, that is, a sequence of maximum posterior probability. Let $v(\cdot; \infty)$ stand for the corresponding classifier, that is

$$v(x^T; \infty) := \arg \max_{s^T \in S^T} p(s^T \mid x^T),$$

with a suitable tie-breaking rule.

Note that Viterbi paths also minimize the following risk

$$\bar{R}_\infty(s^T \mid x^T) := -\frac{1}{T} \log p(s^T \mid x^T). \tag{3}$$

It can actually be advantageous to use the logarithmic risk (3) since, as we shall see later, this leads to various natural generalizations (Sections 3 and 4).

When sequences are compared pointwise, it is common to use additive loss functions of the form

$$L_1(s^T, y^T) = \frac{1}{T} \sum_{t=1}^{T} l(s_t, y_t), \tag{4}$$

where $l(s, y) \geq 0$ is the loss associated with classifying y as s. Typically, for every state s, $l(s, s) = 0$. It is not hard to see that, with L_1 as in (4), the corresponding risk can be represented as follows

$$R_1(s^T \mid x^T) = \frac{1}{T} \sum_{t=1}^{T} \rho_t(s_t \mid x^T),$$

where $\rho_t(s \mid x^T) = \sum_{y \in S} l(s, y) p_t(y \mid x^T)$. Most commonly, l is again symmetrical, or zero-one, that is $l(s, y) = \mathbb{I}_{\{s \neq y\}}$, where $\mathbb{I}_A$ stands for the indicator function of set A. In

this case, L_1 is naturally related to the *Hamming distance* (Carvalho and Lawrence, 2008). Then also $\rho_t(s_t \mid x^T) = 1 - p_t(s_t \mid x^T)$ so that the corresponding risk is

$$R_1(s^T \mid x^T) = 1 - \frac{1}{T}\sum_{t=1}^{T} p_t(s_t \mid x^T). \tag{5}$$

Let $v(\cdot;1)$ stand for the Bayes classifier relative to this R_1-risk. It is easy to see from the above definition of R_1, that $v(\cdot;1)$ delivers PMAP paths, which minimize the expected number of misclassification errors. In addition to maximizing $\sum_{t=1}^{T} p_t(s_t \mid x^T)$, $v(\cdot;1)$ also maximizes $\prod_{t=1}^{T} p_t(s_t \mid x^T)$, and therefore minimizes the following risk

$$\bar{R}_1(s^T \mid x^T) := -\frac{1}{T}\sum_{t=1}^{T} \log p_t(s_t \mid x^T). \tag{6}$$

2.2 Generalizations

Next, we begin to consider various generalizations of the the standard path inferences.

2.2.1 Admissible PMAP and Posterior Viterbi Decoders

Recall (Subsection 1.2.1) that PMAP paths can be *inadmissible*. According to our definition of admissibility (Subsection 1.2.1), a path is inadmissible if it is of zero posterior probability. Although Rabiner (1989) gives no explicit definition of admissibility, or *validity*, he refers to forbidden transitions, that is, of zero prior probability (which, of course, also implies zero posterior probability) as an example of how a path can be "not valid"; the possibility of a path to have a positive prior probability but zero posterior probability is not discussed there. As far as we are aware, Käll et al. (2005) were the first to formally write down an amended PMAP optimization problem to guarantee path validity, or admissibility. However, they too do not state explicitly if "a possible path through the model" means for them positivity only of the prior probability or also of the posterior probability. If "the model" is to be understood as the HMM in its entirety, then this would require positivity of the posterior probability. However, the optimization presented by Käll et al. (2005) does not guarantee positivity of the posterior probability, that is, it only guarantees positivity of the prior probability. Perhaps, it does not happen very often in practice that the PMAP decoder constrained to return *a priori* possible paths returns an inadmissible path (it does not happen in our own experiments in Section 5 as all of our emission probabilities are non-zero on the entire emission alphabet). However, as the example in Appendix A shows, this is indeed possible.

Thus, to enforce admissibility properly, R_1-risk needs to be minimized over the admissible paths (R_1 minimization over the paths of positive prior probability is revisited in Subsection 2.2.2 below):

$$\min_{s^T : p(s^T \mid x^T) > 0} R_1(s^T \mid x^T) \quad \Leftrightarrow \quad \max_{s^T : p(s^T \mid x^T) > 0} \sum_{t=1}^{T} p_t(s_t \mid x^T). \tag{7}$$

Assuming that $p_t(s \mid x^T)$, $1 \le t \le T$, $s \in S$, have been precomputed (e.g., by the classical forward-backward recursion Rabiner, 1989), a solution to (7) can be easily found by a

Viterbi-like recursion (8)

$$\begin{aligned} \delta_1(j) &:= p_1(j \mid x^T), \ \ \forall\, j \in S, \\ \delta_{t+1}(j) &:= \max_i \left(\delta_t(i) + \log r_t(i,j)\right) + p_{t+1}(j \mid x^T) \text{ for } t = 1, 2, \ldots, T-1, \text{ and } \forall j \in S, \end{aligned} \tag{8}$$

where $r_t(i,j) := \mathbb{I}_{\{p_{ij} f_j(x_{t+1}) > 0\}}$ (recall that $p_{ij} = \mathbf{P}(Y_{t+1} = j \mid Y_t = i)$ and f_j is the density of the conditional probability distribution of X_{t+1} conditioned on $Y_{t+1} = j$). *To the best of our knowledge this has not been stated in the literature before.* We will refer to this decoder as the *Constrained PMAP* decoder.

Next note that in the presence of path constraints, minimization of the R_1-risk (5) is *no longer equivalent* to minimization of the $\bar{R}_1$-risk (6). In particular, the problem (7) is not equivalent to the following problem

$$\min_{s^T : p(s^T \mid x^T) > 0} \bar{R}_1(s^T \mid x^T) \quad \Leftrightarrow \quad \max_{s^T : p(s^T \mid x^T) > 0} \sum_{t=1}^{T} \log p_t(s_t \mid x^T). \tag{9}$$

It is also important to note that the problem (9) above *is equivalent* to what has been termed the *posterior-Viterbi decoding*, or *PVD* (Fariselli et al., 2005):

$$\min_{s^T : p(s^T) > 0} \bar{R}_1(s^T \mid x^T) \quad \Leftrightarrow \quad \max_{s^T : p(s^T) > 0} \sum_{t=1}^{T} \log p_t(s_t \mid x^T),$$

that is, unlike in the case of $R_1(s^T \mid x^T)$ minimization, minimization of $\bar{R}_1(s^T \mid x^T)$ over the paths of positive prior probability is indeed sufficient to produce admissible paths.

A solution to (9) can be computed by a related recursion given in (10) below

$$\begin{aligned} \delta_1(j) &:= \log p_1(j \mid x^T), \ \forall j \in S, \\ \delta_{t+1}(j) &:= \max_i \left(\delta_t(i) + \log r_{ij}\right) + \log p_{t+1}(j \mid x^T), \text{ for } t = 1, 2, \ldots, T-1, \ \forall j \in S, \end{aligned} \tag{10}$$

where $r_{ij} := \mathbb{I}_{\{p_{ij} > 0\}}$ (which for inhomogeneous chains will depend on t).

2.2.2 Beyond PVD and *A priori* Admissible PMAP

Although admissible minimizers of R_1 and $\bar{R}_1$ risk are by definition of positive probability, this probability can still be very small. Indeed, in the above recursions, the weight r_{ij} is 1 even when p_{ij} is very small. We next replace r_{ij} by the true transition probability p_{ij} in minimizing the $\bar{R}_1$-risk (that is maximization of $\prod_{t=1}^{T} p_t(s_t \mid x^T)$). Then the solutions remain admissible and also tend to maximize the prior path probability. To bring the newly obtained optimization problem to a more elegant form (11), we pretend that $\delta_1(j)$ in (10) above was defined as $\delta_1(j) := \log p_1(j \mid x^T) + \log \mathbb{I}_{\{\pi_j > 0\}}$ (which indeed does not change the results of the recursion (10)) and replace the last term by $\log \pi_j$.

Thus, with the above replacements, the recursion (10) now solves the following *seemingly unconstrained* optimization problem (see Theorem 4)

$$\max_{s^T} \Big[\sum_{t=1}^{T} \log p_t(s_t \mid x^T) + \log p(s^T) \Big] \quad \Leftrightarrow \quad \min_{s^T} \Big[\bar{R}_1(s^T \mid x^T) + h(s^T) \Big], \tag{11}$$

where the penalty term

$$h(s^T) \quad = -\tfrac{1}{T}\log p(s^T) \quad =: \bar{R}_\infty(s^T) \tag{12}$$

is the logarithmic risk based on the prior distribution,[1] which does not involve the observed data.

The thereby modified recursions immediately generalize as follows:

$$\begin{aligned}\delta_1(j) &:= \log p_1(j \mid x^T) + C\log\pi_j,\ \forall j \in S,\\ \delta_{t+1}(j) &:= \max_i \big(\delta_t(i) + C\log p_{ij}\big) + \log p_{t+1}(j \mid x^T) \text{ for } t = 1,2,\ldots,T-1,\ \forall j \in S,\end{aligned}$$

solving

$$\min_{s^T}\Big[\bar{R}_1(s^T \mid x^T) + Ch(s^T)\Big], \tag{13}$$

where $C > 0$ is a trade-off constant, which can also be viewed as a regularization parameter. Indeed, Proposition 2 below states that $C > 0$ implies admissibility of solutions to (13). In particular, PVD, that is the problem solved by the original recursion (10), can now be recovered by taking C sufficiently small. (Alternatively, the PVD problem can also be formally written in the form (13) with $C = \infty$ and $h(s^T)$ given, for example, by $\mathbb{I}_{\{p(s^T)=0\}}$.)

What if the actual probabilities p_{ij} (π_j) were also used in the optimal accuracy/PMAP decoding? To motivate this, we re-consider the optimal accuracy/PMAP decoding imposing the positivity constraint not on the posterior but on the prior path probability:

$$\min_{s^T:p(s^T)>0} R_1(s^T \mid x^T) \quad \Leftrightarrow \quad \max_{s^T:p(s^T)>0}\sum_{t=1}^T p_t(s_t \mid x^T). \tag{14}$$

Solution to (14) can be easily found by yet another Viterbi-like recursion given in (15) below

$$\begin{aligned}\delta_1(j) &:= p_1(j \mid x^T),\ \ \forall\, j \in S, \qquad (15)\\ \delta_{t+1}(j) &:= \max_i\,(\delta_t(i) + \log r_{ij}) + p_{t+1}(j \mid x^T) \text{ for } t = 1,2,\ldots,T-1, \text{ and } \forall j \in S,\end{aligned}$$

which is the same as (8) apart from the r_{ij} in place of the $r_t(i,j)$.

We again replace the indicators r_{ij} by the actual probabilities p_{ij}. We once more pretend that $\delta_1(j)$ in (15) above was defined, this time, as $\delta_1(j) := p_1(j \mid x^T) + \log\mathbb{I}_{\{\pi_j>0\}}$. Replacing the last term by $\log\pi_j$ yields the following problem:

$$\max_{s^T}\Big[\sum_{t=1}^T p_t(s_t \mid x_t) + \log p(s^T)\Big] \quad \Leftrightarrow \quad \min_{s^T}\Big[R_1(s^T \mid x^T) + \bar{R}_\infty(s^T)\Big]. \tag{16}$$

A more general problem can be written in the form

$$\min_{s^T}\Big[R_1(s^T \mid x^T) + Ch(s^T)\Big], \tag{17}$$

1. More generally, the same type of risk (e.g., $\bar{R}_\infty$) can be based on the posterior ($p(s^T \mid x^T)$), joint ($p(s^T, x^T)$) or prior ($p(s^T)$) distribution. Compromising between notational accuracy on the one hand and notational simplicity and consistency on the other hand, throughout the paper we disambiguate these cases solely by the argument.

where h is some penalty function (independent of the data x^T). Thus, the problem (14) of optimal accuracy/PMAP decoding over the paths of positive prior probability is obtained by taking C sufficiently small and $h(s^T) = \bar{R}_\infty(s^T)$. (Setting $C \times h(s^T) = \infty \times \mathbb{I}_{\{p(s^T)=0\}}$ also reduces the problem (17) back to (7).)

Clearly, if instead of (14) we started off with (7) ($R_1(s^T \mid x^T)$ minimization over the admissible paths), we would arrive at $\bar{R}_\infty(s^T \mid x^T)$ in place of $\bar{R}_\infty(s^T)$ in (16) above. Inclusion of $\bar{R}_\infty(s^T \mid x^T)$ more generally is treated next in Section 3.

3. Combined Risks

Motivated by the previous section, we consider the following general problem

$$\min_{s^T} \Big[C_1 \bar{R}_1(s^T \mid x^T) + C_2 \bar{R}_\infty(s^T \mid x^T) + C_3 \bar{R}_1(s^T) + C_4 \bar{R}_\infty(s^T) \Big], \tag{18}$$

where $C_i \geq 0$, $i = 1, 2, 3, 4$, $\sum_{i=1}^4 C_i > 0$.[2] This is also equivalent to

$$\min_{s^T} \Big[C_1 \bar{R}_1(s^T \mid x^T) + C_2 \bar{R}_\infty(s^T, x^T) + C_3 \bar{R}_1(s^T) + C_4 \bar{R}_\infty(s^T) \Big], \tag{19}$$

where, recalling (6), $\quad \bar{R}_1(s^T \mid x^T) = -\frac{1}{T} \sum_{t=1}^T \log p_t(s_t \mid x^T)$,

$$\begin{aligned}
\bar{R}_\infty(s^T, x^T) &:= -\frac{1}{T} \log p(x^T, s^T), \\
&= -\frac{1}{T} [\log p(s^T) + \sum_{t=1}^T \log f_{s_t}(x_t)], \\
&= -\frac{1}{T} [\log \pi_{s_1} + \sum_{t=1}^{T-1} \log p_{s_t s_{t+1}} + \sum_{t=1}^T \log f_{s_t}(x_t)],
\end{aligned}$$

recalling (3), $\quad \bar{R}_\infty(s^T \mid x^T) = -\frac{1}{T} \log p(s^T \mid x^T)$,

$$\begin{aligned}
&= \bar{R}_\infty(s^T, x^T) + \frac{1}{T} \log p(x^T), \\
\bar{R}_1(s^T) &:= -\frac{1}{T} \sum_{t=1}^T \log p_t(s_t), \qquad (20) \\
\bar{R}_\infty(s^T) &= -\frac{1}{T} \log p(s^T), \quad \text{recalling (12)}, \\
&= -\frac{1}{T} [\log \pi_{s_1} + \sum_{t=1}^{T-1} \log p_{s_t s_{t+1}}]. \qquad (21)
\end{aligned}$$

The newly introduced risk $\bar{R}_1(s^T)$ involves only the prior marginals. Note that the combination $C_1 = C_3 = C_4 = 0$ corresponds to the MAP/Viterbi decoding; the combination

2. For uniqueness of representation, one may want to additionally require $\sum_{i=1}^4 C_i = 1$.

$C_2 = C_3 = C_4 = 0$ yields the PMAP case, whereas the combinations $C_1 = C_2 = C_3 = 0$ and $C_1 = C_2 = C_4 = 0$ give the *maximum a priori* decoding and *marginal prior mode* decoding, respectively. The case $C_2 = C_3 = 0$ subsumes (13) and the case $C_1 = C_3 = 0$ is the problem

$$\min_{s^T} \left[\bar{R}_\infty(s^T \mid x^T) + C\bar{R}_\infty(s^T) \right]. \tag{22}$$

Thus, a solution to (22) is a generalization of the Viterbi decoding that allows one to suppress ($C > 0$) contribution of the data.

Remark 1 *If $C_2 > 0$, then every solution of* (18) *is admissible and the minimized risk is finite.*

No less important and perhaps a little less obvious is that $C_1, C_4 > 0$ also guarantees admissibility of the solutions, as stated in Proposition 2 below.

Proposition 2 *Let $C_1, C_4 > 0$. Then, the minimized risk* (18) *is finite and any minimizer s^T is admissible.*

Proof Without loss of generality, assume $C_2 = C_3 = 0$. Since $p(x^T) > 0$ (assumed in the beginning of Section 2), there exists some admissible path s^T. Clearly, the combined risk of this path is finite, hence so is the minimum risk. Now, suppose s^T is a minimizer of the combined risk and suppose further that s^T is inadmissible, that is $p(s^T \mid x^T) = 0$. Since the minimized risk (18) is finite, we must have $p(s^T) > 0$. Therefore, it must be that $p(x^T \mid s^T) = 0$, and therefore we must have some t, $1 \le t \le T$, such $f_{s_t}(x_t) = 0$. This would imply that any path through (t, s_t) is inadmissible, hence $p_t(s_t \mid x^T)$, the sum of the posterior probabilities of all such paths, is zero. This implies $\bar{R}_1(s^T \mid x^T) = \infty$, contradicting optimality of s^T. ■

Remark 3 *Note that for any x^T, the Posterior-Viterbi decoding (Fariselli et al., 2005) (Problem 9 above) can be obtained by setting $C_3 = C_4 = 0$ and taking C_2 sufficiently small, that is, $0 < C_2 \ll C_1$. Also, PVD can be obtained almost surely by setting $C_2 = C_3 = 0$ and taking C_4 sufficiently small, that is, $0 < C_4 \ll C_1$.*

It is fairly intuitive that PVD can be realized as solutions to (18), but we nonetheless prove this formally in Appendix B.

If the smoothing probabilities $p_t(s \mid x^T)$, $t = 1, \ldots, T$ and $s \in S$, have been already computed, a solution to (18) can be found also by a standard dynamic programming algorithm. Let us first introduce more notation. For every $t \in 1, \ldots, T$ and $j \in S$, let

$$\gamma_t(j) := C_1 \log p_t(j \mid x^T) + C_2 \log f_j(x_t) + C_3 \log p_t(j).$$

Note that the function γ_t depends on the entire data x^T. Next, let us also define the following scores

$$\begin{aligned} \delta_1(j) &:= (C_2 + C_4) \log \pi_j + \gamma_1(j), \ \forall j \in S, \\ \delta_t(j) &:= \max_i \left(\delta_{t-1}(i) + (C_2 + C_4) \log p_{ij} \right) + \gamma_t(j), \\ &\qquad \text{for } t = 2, 3, \ldots, T, \text{ and } \forall j \in S. \end{aligned} \tag{23}$$

Using the above scores $\delta_t(j)$ and a suitable tie-breaking rule, below we define the back-pointers $i_t(j)$, terminal state i_T, and the optimal path $\widehat{y^T}(i_T)$.

$$\begin{aligned} i_t(j) &:= \arg\max_{i\in S}[\delta_t(i) + (C_2 + C_4)\log p_{ij}], \quad \text{when } t = 1,\dots,T-1; \\ i_T &:= \arg\max_{i\in S}\delta_T(i); \end{aligned} \tag{24}$$

$$\widehat{y^t}(j) := \begin{cases} i_1(j), & \text{when } t = 1; \\ \big(\widehat{y^{t-1}}(i_{t-1}(j)), j\big), & \text{when } t = 2,\dots,T. \end{cases} \tag{25}$$

Thus, given x^{t+1} and the best path that ends in state j (at time $t+1$), $i_t(j)$ represents the t-th state in this path.

The following theorem formalizes the dynamic programming argument; its proof is standard and we state it below for completeness only.

Theorem 4 *Any solution to* (18) *can be represented in the form* $\widehat{y^T}(i_T)$ *provided the ties in* (24) *are broken accordingly.*

Proof With a slight abuse of notation, for every $s^t \in S^t$, let

$$U(s^t) = \sum_{u=1}^{t} \left[\gamma_u(s_u) + (C_2 + C_4)\log p_{s_{u-1}s_u}\right],$$

where $s_0 := 0$ and $p_{0s} := \pi_s$. Hence,

$$-T[C_1\bar{R}_1(s^T \mid x^T) + C_2\bar{R}_\infty(s^T, x^T) + C_3\bar{R}_1(s^T) + C_4\bar{R}_\infty(s^T)] = U(s^T)$$

and any maximizer of $U(s^T)$ is clearly a solution to (18) and (19).

Next, let $U(j) := \delta_1(j)$ for all $j \in S$, and let

$$U(s^{t+1}) = U(s^t) + (C_2 + C_4)\log p_{s_t s_{t+1}} + \gamma_{t+1}(s_{t+1}),$$

for $t = 1, 2, \dots, T-1$ and also $s^t \in S^t$. By induction on t, these yield

$$\delta_t(j) = \max_{s^t : s_t = j} U(s^t)$$

for every $t = 1, 2, \dots, T$ and for all $j \in S$. Clearly, every maximizer $\widehat{y^T}$ of $U(s^T)$ over the set S^T must end up in i_T, or, more precisely, in the set $\arg\max_{j\in S}\delta_T(j)$, allowing for non-uniqueness. Continuing to interpret arg max as a set, recursion (23) implies recursions (24) and (25), hence any maximizer $\widehat{y^T}$ can indeed be computed in the form $\widehat{y^T}(i_T)$ via the *forward* (recursion (24))-*backward* (recursion (25)) procedure. ■

Similarly to the generalized risk minimization of (18), the generalized problem of accuracy optimization (17) can also be further generalized as follows:

$$\min_{s^T}\Big[C_1 R_1(s^T \mid x^T) + C_2\bar{R}_\infty(s^T \mid x^T) + C_3 R_1(s^T) + C_4\bar{R}_\infty(s^T)\Big], \tag{26}$$

where risk

$$R_1(s^T) := \frac{1}{T}\sum_{t=1}^{T}\mathbf{P}(Y_t \neq s_t) = 1 - \frac{1}{T}\sum_{t=1}^{T} p_t(s_t) \tag{27}$$

is the error rate relative to the prior distribution. This problem can also be solved by a recursion formally identical to that in (23) except for the removed logarithms in the marginal probabilities:

$$\gamma_t(j) \quad = \quad C_1 p_t(j \mid x^T) + C_2 \log f_j(x_t) + C_3 p_t(j). \tag{28}$$

The following remarks compare this generalized Problem with the generalized Problem (18) (Remarks 1 and 3, Proposition 2).

Remark 5 *1. As in the generalized posterior-Viterbi decoding* (18)*, here $C_2 > 0$ also implies admissibility of the optimal paths.*

2. Now, $C_4 > 0$ implies that the minimized risk is finite for any x^T, but unlike in (18)*, $C_1, C_4 > 0$ is not sufficient to guarantee admissibility almost surely of the solutions to the problem* (26)*.*

3. Taking $C_3 = C_4 = 0$, the constrained PMAP problem (Käll et al., 2005) (Problem 7 above) is obtained for some C_1, C_2 such that $0 < C_2 \ll C_1$.

We refer to a decoder solving the generalized risk minimization Problem (18) as a *generalized posterior-Viterbi hybrid decoder*. Similarly, a decoder solving the generalized optimal accuracy Problem (26) is referred to as a *generalized PMAP hybrid decoder* to distinguish the product-based risk $\bar{R}_1(s^T \mid x^T)$ in the former case from the sum-based risk $R_1(s^T \mid x^T)$ in the latter case. Both the generalized families, however, naturally extend the PMAP/optimal accuracy/posterior decoder (Section 2.1).

Corollary 15 of Apendix C establishes the usual trade-off type of resuls for the solutions to Problems (18) and (26). The results on the trade-off between $\bar{R}_1$ and $\bar{R}_\infty$ risks will in particular be useful in Corollary 8 (see further below) for establishing monotonicity of the solution to Problem (18).

4. The k-Block Posterior-Viterbi Decoding

The next approach provides a surprisingly different insight into what otherwise has already been formulated as the generalized Problem (18). This, first of all, helps better understand how the generalized Problem (18) resolves the drawback of Rabiner's suggestion (introduced in the last paragraph of Subsection 1.2.1 above). Secondly, the same approach gives an elegant relationship (Theorem 6, Corollary 7) between the main types of risk, which surprisingly amounts to, as far as we know, a novel property of ordinary Markov chains (Equation 34, and Proposition 14 of the concluding Section 8).

Recall (Subsection 1.2) that Rabiner's compromise between MAP and PMAP is to maximize the expected number of correctly decoded pairs or triples of (adjacent) states. With k being the length of the overlapping block ($k = 2, 3, \ldots$) this means to minimize the

conditional risk

$$R_k(s^T \mid x^T) := 1 - \frac{1}{T-k+1} \sum_{t=1}^{T-k+1} p(s_t^{t+k-1} \mid x^T), \tag{29}$$

which derives from the following loss function:

$$L_k(s^T, y^T) := \frac{1}{T-k+1} \sum_{t=1}^{T-k+1} \mathbb{I}_{\{s_t^{t+k-1} \neq y_t^{t+k-1}\}}. \tag{30}$$

When $k = 1$ this gives the usual R_1 maximization, that is, the PMAP decoding, which is known to fault by allowing inadmissible paths. Just as in (4) with $k = 1$, we could also consider a general (possibly asymmetric) loss function $l_k(s_t^{t+k-1}, y_t^{t+k-1})$ for larger k in (30) above. Thus, for $k = 2$ this is the Markov loss function studied by Yau and Holmes (2010).

It is natural to think that minimizers of $R_k(s^T \mid x^T)$ "move" towards Viterbi paths "monotonically" as k increases to T. Indeed, when $k = T$, minimization of $R_k(s^T \mid x^T)$ (29) is equivalent to minimization of $\bar{R}_\infty(s^T \mid x^T)$ achieved by the Viterbi decoding. However, as the experiments in Section 5 below show, minimizers of (29) are not guaranteed to be admissible (even if admissibility were defined relative to the prior distribution) for $k > 1$. Also, as we already pointed out in Subsection 1.2.1, this approach does not give monotonicity, that is, allows the optimal path for $k = 2$ to have lower (prior and posterior) probabilities than those of the PMAP path (that is, $k = 1$). Another drawback of using the loss L_k (30) and its more general variants is that, unlike in the generalized PVD and PMAP hybrid decoders, *the computational complexity of Rabiner's approach grows with the block length k*. We now show how these drawbacks go away when the sum in (29) is replaced by a product, eventually arriving at a subfamily of the generalized posterior Viterbi decoders. Certainly, replacing the sum by the product alters the problem, and it does so in a way that makes the block-wise coding idea work well. Namely, the longer the block, the larger the resulting path probability, which is also now guaranteed to be positive already for $k = 2$. Moreover, this gives another interpretation of the risks $\bar{R}_1(s^T \mid x^T) + C\bar{R}_\infty(s^T \mid x^T)$ (see also Remark 3 above), the prior risks $\bar{R}_1(s^T) + C\bar{R}_\infty(s^T)$, and consequently the generalized Problem (18).

Let k be a positive integer. For the time being, let p represent any first order Markov chain on S^T, and let us define

$$\bar{U}_k(s^T) := \prod_{j=1-k}^{T-1} p\big(s_{\max(j+1,1)}^{\min(j+k,T)}\big), \quad \bar{R}_k(s^T) := -\frac{1}{T} \ln \bar{U}_k(s^T).$$

Thus

$$\bar{U}_k(s^T) = U_1^k \cdot U_2^k \cdot U_3^k,$$

where

$$\begin{aligned} U_1^k &:= p(s_1) \cdots p(s_1^{k-2}) p(s_1^{k-1}), \\ U_2^k &:= p(s_1^k) p(s_2^{k+1}) \cdots p(s_{T-k}^{T-1}) p(s_{T-k+1}^T), \\ U_3^k &:= p(s_{T-k+2}^T) p(s_{T-k+3}^T) \cdots p(s_T). \end{aligned}$$

Thus, $\bar{R}_k$ is a natural generalization of $\bar{R}_1$ (introduced first for the posterior distribution in (6)) since when $k=1$, $\bar{R}_k = \bar{R}_1$.

Theorem 6 *Let k be such that $T \geq k > 1$. Then the following recursion holds*

$$\bar{R}_k(s^T) = \bar{R}_\infty(s^T) + \bar{R}_{k-1}(s^T), \quad \forall s^T \in S^T.$$

Proof Note that

$$U_1^k = U_1^{k-1} p(s_1^{k-1}), \quad U_3^k = p(s_{T-k+2}^T) U_3^{k-1}.$$

Next, for all j such that $j+k \leq T$, the Markov property gives

$$p(s_{j+1}^{j+k}) = p(s_{j+k} \mid s_{j+k-1}) p(s_{j+1}^{j+k-1})$$

and

$$\begin{aligned}
&U_2^k p(s_{T-k+2}^T) = p(s_1^k) p(s_2^{k+1}) \cdots p(s_{T-k+1}^T) p(s_{T-k+2}^T) = \\
&p(s_k \mid s_{k-1}) p(s_1^{k-1}) p(s_{k+1} \mid s_k) p(s_2^k) \cdots p(s_T \mid s_{T-1}) p(s_{T-k+1}^{T-1}) p(s_{T-k+2}^T) = \\
&p(s_k \mid s_{k-1}) p(s_{k+1} \mid s_k) \cdots p(s_T \mid s_{T-1}) p(s_1^{k-1}) \cdots p(s_{T-k+1}^{T-1}) p(s_{T-k+2}^T) = \\
&p(s_k \mid s_{k-1}) \cdots p(s_T \mid s_{T-1}) U_2^{k-1}.
\end{aligned}$$

Hence,

$$\begin{aligned}
\bar{U}_k(s^T) &= U_1^{k-1} p(s_1^{k-1}) p(s_k \mid s_{k-1}) \cdots p(s_T \mid s_{T-1}) U_2^{k-1} U_3^{k-1}, \\
&= p(s_1^T) U_1^{k-1} U_2^{k-1} U_3^{k-1} = p(s^T) \bar{U}_{k-1}(s^T).
\end{aligned}$$

The second equality above also follows from the Markov property. Taking logarithms on both sides and dividing by $-T$ completes the proof. ■

Now, we specialize this result to our HMM context, and, thus, $p(s^T)$ and $p(s^T \mid x^T)$ are again the prior and posterior hidden path distributions.

Corollary 7 *Let k be such that $T \geq k > 1$. For all paths $s^T \in S^T$ the prior risks $\bar{R}_k$ and $\bar{R}_\infty$ satisfy* (31)*. For every $x^T \in \mathcal{X}^T$ and for all paths $s^T \in S^T$, the posterior risks $\bar{R}_k$ and $\bar{R}_\infty$ satisfy* (32)*.*

$$\bar{R}_k(s^T) = \bar{R}_\infty(s^T) + \bar{R}_{k-1}(s^T), \tag{31}$$

$$\bar{R}_k(s^T \mid x^T) = \bar{R}_\infty(s^T \mid x^T) + \bar{R}_{k-1}(s^T \mid x^T). \tag{32}$$

Proof Clearly, conditioned on the data x^T, Y^T remains a first order Markov chain (generally inhomogeneous even if it was homogeneous *a priori*). Hence, Theorem 6 applies. ■

Below, *we focus on the posterior distribution and risks, but the discussion readily extends to any first order Markov chain.*

Let $v(x^T; k)$ be a decoder that minimizes $\bar{R}_k(s^T \mid x^T)$, returning a path $\hat{y}(k)$, that is,

$$\hat{y}(k) = \arg \max_{s^T \in S^T} \bar{U}_k(s^T \mid x^T) = \arg \min_{s^T \in S^T} \bar{R}_k(s^T \mid x^T). \tag{33}$$

Corollary (8) below states how $\bar{R}_k(s^T \mid x^T)$ minimization is a special case of the generalized Problem (18). We refer to the generalized posterior-Viterbi hybrid decoders $v(x^T; k)$ as *k-block* PVD and summarize their properties in Corollary (8).

Corollary 8 *For every $x^T \in \mathcal{X}^T$, and for every $s^T \in S^T$, we have*

$$\bar{R}_k(s^T \mid x^T) = (k-1)\bar{R}_\infty(s^T \mid x^T) + \bar{R}_1(s^T \mid x^T), \qquad \forall k \text{ such that } 1 \le k \le T. \tag{34}$$

$$\hat{y}(k) \text{ is admissible}, \qquad \forall k \text{ such that } k > 1. \tag{35}$$

$$\bar{R}_\infty(\hat{y}(k) \mid x^T) \le \bar{R}_\infty(\hat{y}(k-1) \mid x^T), \qquad \forall k \text{ such that } 1 < k \le T. \tag{36}$$

$$\bar{R}_1(\hat{y}(k) \mid x^T) \ge \bar{R}_1(\hat{y}(k-1) \mid x^T), \qquad \forall k \text{ such that } 1 < k \le T. \tag{37}$$

Proof Equation (34) follows immediately from Equation (32) of Corollary 7. Admissibility of $\hat{y}(k)$ for $k > 1$ in (35) becomes obvious recalling Remark 1. Inequalities (36) and (37) are established by Corollary 15. ■

Equation (34) is also of practical significance showing that $\hat{y}(k)$ is a solution to (18) with $C_1 = 1$, $C_2 = k-1$, $C_3 = C_4 = 0$, and as such can be computed in the same fashion for all k, $1 \le k \le T$ (see Theorem 4 above).

Inequality (36) means that the posterior path probability $p(\hat{y}(k) \mid x^T)$ increases with k. At the same time, increasing k also increases $\bar{R}_1$-risk, that is, decreases the product of the (posterior) marginal probabilities of states along the path $\hat{y}(k)$. Inequalities (36) and (37) clearly show that as k increases, $v(\cdot; k)$ monotonically moves from $v(\cdot; 1)$ (PMAP) towards the Viterbi decoder, that is $v(\cdot; \infty)$. However, the maximum block length is $k = T$.

A natural way to complete this bridging of PMAP with MAP is by embedding the $\bar{R}_k$ risks into the family $\bar{R}_\alpha$ via $\alpha = \frac{k-1}{k} \in [0, 1]$. Thus, (34) extends to

$$\bar{R}_\alpha(s^T \mid x^T) := \alpha\bar{R}_\infty(s^T \mid x^T) + (1-\alpha)\bar{R}_1(s^T \mid x^T) \tag{38}$$

with $\alpha = 0$ and $\alpha = 1$ corresponding to the PMAP and Viterbi cases, respectively. This embedding is clearly still within the generalized Problem (18) via $C_1 = 1 - \alpha$, $C_2 = \alpha$, $C_3 = C_4 = 0$. In particular, $v(x^T; k(\alpha))$ can be computed by using the same dynamic programming algorithm of Theorem 4 for all $k \in [1, \infty]$ (that is, all $\alpha \in [0, 1]$), and inequalities (36) and (37) are special cases of Corollary 15 (part 1) to Lemma 16.

Recalling Remark 3, we note that on the lower end of $0 \le \alpha \le 1$, before reaching PMAP ($\alpha = 0$) we encounter PVD for some sufficiently small $\alpha \approx 0$. Note also that in (35) k need not be integer either, that is, Remark 1 establishes admissibility of $\hat{y}(k(\alpha))$, $k(\alpha) = 1/(1-\alpha)$, for all $\alpha \in (0, 1]$ (that is, all $k \in (1, \infty]$).

Given x^T and a sufficiently large k (equivalently, $\alpha \approx 1$), $\hat{y}(k)$, the minimizer of $\bar{R}_\alpha(s^T \mid x^T)$ (38) (and (34)) would become a Viterbi path $\hat{y}(\infty)$ (since S^T is finite). However, such α (and k) would generally depend on x^T, and in particular k may need to be larger than T, that is, $\hat{y}(T)$ may be different from $\hat{y}(\infty)$.

At the same time, for $k > 1$ we have

$$\bar{R}_\infty(\hat{y}(\infty) \mid x^T) \le \bar{R}_\infty(\hat{y}(k) \mid x^T) \le \bar{R}_\infty(\hat{y}(\infty) \mid x^T) + \frac{\bar{R}_1(\hat{y}(\infty) \mid x^T)}{k-1}, \tag{39}$$

on which we comment more in Section 7 below. The first inequality of (39) above follows immediately from the definition of the Viterbi decoder. To obtain the second inequality, apply (34) to both $\hat{y}(k)$ and $\hat{y}(\infty)$ and subtract one equation from the other. Dividing the

resulting terms by $k-1$, noticing that $\bar{R}_k(\hat{y}(\infty) \mid x^T) \geq \bar{R}_k(\hat{y}(k) \mid x^T)$ and $\bar{R}_1(\hat{y}(k) \mid x^T) \geq 0$, and rearranging the other terms yields the result.

Considering the prior chain Y^T and risks in (31), we immediately obtain statements analogous to (34)-(38) extending these new interpretations to the entire generalized Problem (18). In particular, it might be of general interest to note that for any first order Markov chain (that is, not necessarily representing the posterior distribution of an HMM) the following convexly combined risk

$$\bar{R}_\alpha(s^T) := \alpha \bar{R}_\infty(s^T) + (1-\alpha)\bar{R}_1(s^T)$$

can be efficiently minimized in the usual forward-backward manner (Theorem 4).

5. Experiments

We *illustrate* the performance of the Viterbi, PMAP, and some of the other known and new decoders on the task of predicting protein secondary structure in single amino-acid sequences. We show that the differences in performance between the various decoders can be significant. For this illustration purpose, our decoders are based entirely on the ordinary first order HMM. In particular, when decoding an amino-acid sequence, they do not use cues from decoded homologous sequences (other than by allowing homologous sequences to be part of the training set for estimation of the model parameters). Certainly, successful predictors in practice are significantly more elaborate. In particular, they do exploit intensively information from decoded homologs, and also include interactions at ranges considerably longer than that of the first order HMM (Aydin et al., 2006). However, our current goal is not to compete for the absolute record on the task (which, not so long ago, was reported to be about 70% (Aydin et al., 2006)), but to merely emphasize the following two points. First, the difference in performance between the Viterbi and PMAP decoders can be appreciable in practice already with the ordinary first order HMMs having as few as six hidden states. Secondly, using the new family of decoders (that is, solutions to the generalized risk minimization 18 and 26) gives a potentially useful additional flexibility by exercising trade-offs between principled performance measures (Subsection 1.2.2).

Our data are a non-redundant subset of the *Protein Data Bank* (Berman et al., 2000). Specifically, the secondary structural elements have been found from their atomic coordinates using `SSENVID` (Softberry, Inc., 2001) and the resulting data can be freely downloaded from `http://personal.rhul.ac.uk/utah/113/VA/env_seqssnr.txt`. The data contain $N = 25713$ realizations $(x^{T_n}(n), y^{T_n}(n))$, $n = 1, 2, \ldots, N$, with three original hidden states $\{a, b, c\}$, representing $\alpha-$helix, $\beta-$strand, and coil, respectively. The average length $\bar{T}$ of a realization is 167 positions. The observations $x^{T_n}(n)$ come from a 20 symbol emission alphabet of amino-acids

$$\mathcal{X} = \{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}.$$

We further distinguish four subclasses of the α-helix class a. The definition and enumeration of the final six classes are as follows: Class one consists of the short, up to seven a long, α-helices. Classes two and three consist of the β-strands (any number of b's) and coil sequences (any number of c's), respectively. Classes four, five, and six derive from the a's

that comprise an α-helix of length at least eight, thereafter referred to as long. Specifically, class four is the so-called N-end, which is the first four a's of a long α-helix. Similarly, class six is the so called C-end, which is the last four a's of a long α-helix. Any a's in the middle of a long α-helix are class five. Refining the original classification has been known to improve prediction of protein secondary structure (Salamov and Solovyev, 1995). For simplicity, here we only sub-divide the α-helix class (whereas Salamov and Solovyev, 1995 go further) given the limited goals of these experiments.

The (maximum likelihood estimates of the) transition and emission distribution matrices as well as the vector of the initial probabilities computed from *all of the realizations* are given in Appendix E.

The following experiments emulate a typical practical situation by re-estimating these parameters from $N-1$ sequences and using the re-estimated values to decode a remaining sequence. We repeat the process N times in the leave-one(sequence)-out fashion. We do not impose stationarity in these experiments as we did not have any prior evidence of stationarity. Indeed, the (estimated) initial distribution $\hat{\pi}$ appears to be very different from the stationary one ($\hat{\pi}_{inv}$, see Appendix E) and many sequences in the data set are quite short.

Figure 1 displays case 877, which is 149 positions long and is split into two pieces at position $t = 72$ (shown in both images). The top (0) row is the ground truth. This case is typical in several senses. First, in this case the PMAP decoder (row 2) shows the median gain in accuracy (of about 11%) over the Viterbi decoder (row 1); see subsequent subsections for a discussion of performance measures. Secondly, the PMAP, or optimal accuracy output, is inadmissible in this case, which is evident from, for example, the isolated state five (yellow) island (transitions between states three and five are forbidden). Rows 3 through 5 are outputs from the PVD, Constrained PMAP, and Rabiner $k = 2$ decoders, respectively. It is typical of the PVD and Constrained PMAP decoders to tie. Outputs from other members of the generalized posterior Viterbi (18) and PMAP (26) hybrid decoders are given in rows 6-18, and 19-31, respectively. Table 1 gives a detailed legend for interpreting the outputs. The monotonicity of the generalized PVD hybrid inference (Corollary 15, part 1, and Corollary 8, inequalities 36 and 37) is illustrated by following the posterior risk columns $\bar{R}_\infty$ and $\bar{R}_1$ across rows 2 (PMAP), then 6 through 17, and finally 1 (Viterbi); PVD (row 3) is attained when $\alpha \approx 0$ (rows 6-9) and here is also indistinguishable from Constrained PMAP (row 4). The monotonicity of the generalized PMAP hybrid inference (Corollary 15, part 3) is illustrated by following the $\bar{R}_\infty$ and R_1 columns across rows 2 (PMAP), then 19 through 30, and finally 1 (Viterbi); Constrained PMAP (row 4) is attained when $\alpha \approx 0$ (rows 19-20) and here is also indistinguishable from PVD (row 3).

Note how the decoder in row 16 (Figure 1) differs from its neighbors, specifically, how it completely misses the terminal activity, which is to a variable extent captured by both its "more accurate" (row 15) and "more probable" (row 17) neighbors.

Rows 18 and 31 are the "data blind" *maximum a priori* and *pointwise maximum a priori* decodings, which are members of both the generalized hybrid families. These decoders tie not only in this but in all the other cases as well; see the structure of the (overall) transition matrix $\mathbb{P}$ in Appendix E also to understand the overwhelming dominance of class 3 ("coil") in the absence of the amino-acid information. By adjusting the R_1 and $\bar{R}_1$ risk terms in the generalized decoders, we can easily accommodate unequal classification penalties to begin

exploring the topology of the posterior distribution (see also Section 8). Thus, for example, we suppress the dominating class 3 to better reveal activity of the remaining classes as shown in Figure 2. Specifically, the marginal posterior probabilities $p_t(s \mid x^T)$ are replaced by $p_t(s \mid x^T)/21$ and $4p_t(s \mid x^T)/21$ for $s = 3$ and $s \neq 3$, respectively; the same re-weighting is also applied to the prior marginal distributions; the Viterbi, Rabiner $k = 2$, as well as the MAPriori decoder (rows 2, 5, 18, respectively) are not affected by this adjustment.

Application specific performance measures will usually be of more interest than the simple measures used here for illustration of the ideas (Section 8). Thus, for example, regarded as β-strand (state 2) detectors, the original decoders (Figure 1) miss four of the seven 2-islands. On the other hand, a more dynamic class 2 activity revealed in Figure 2 correlates very well with the seven objects of class 2. The presence of the adjusted PMAPriori decoder (row 31) also helps to better assess the value of the observed data.

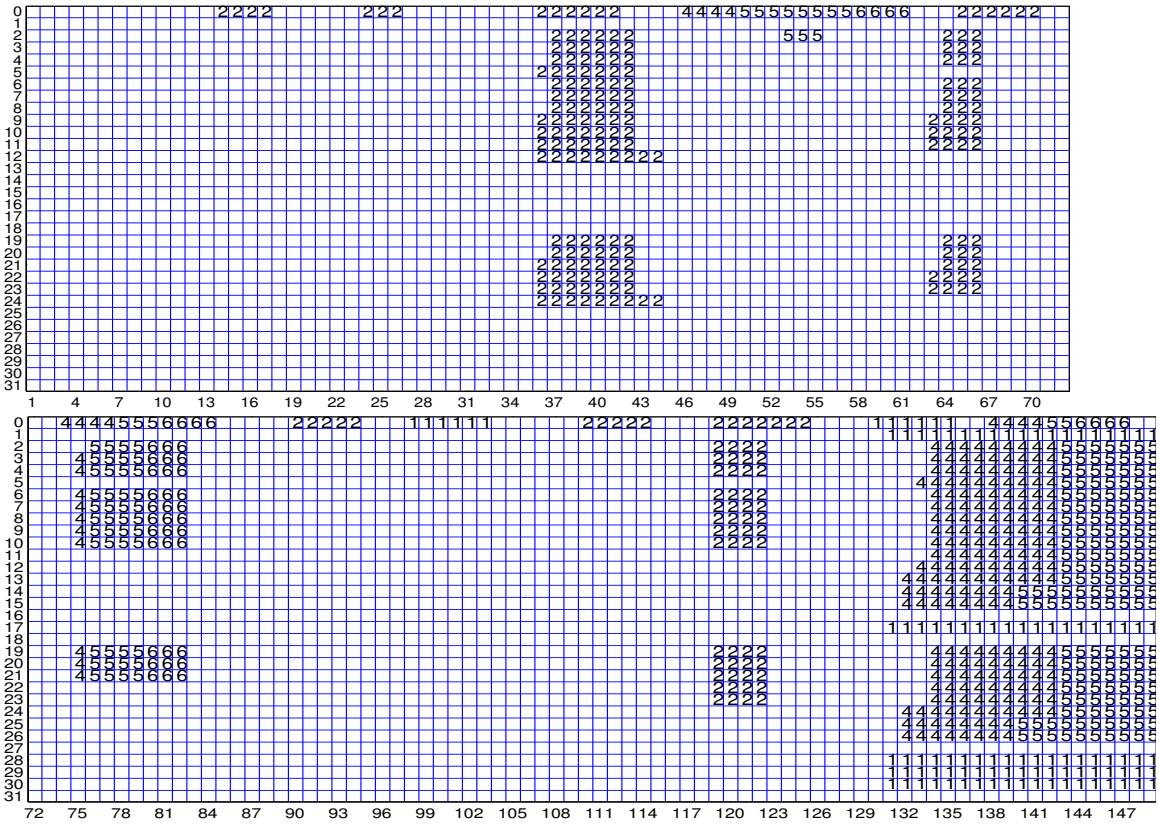

Figure 1: Performance of the well-known and some of the new decoders on Case 877. The dominant class 3 is represented by blank entries. For further legend, see Table 1.

In addition to using the real data, we simulate synthetic data sets each of which having the same number $N = 25713$ of sequences, in the following way. Let $\{\hat{\pi}_{sn}\}_{s\in S}$, $\widehat{\mathbb{P}}_n$, $\{\widehat{P}_{sn}, s \in S\}$ be the estimates of the HMM parameters (initial, transition, and emission distributions, respectively) obtained from $(x^{T_n}(n), y^{T_n}(n))$, the n-th actual realization. Then the n-th

Row	Output $\widehat{y^{149}}$							Empir. error rate(%)	posterior risks		
	Generalized		Alias	C_1	C_2	C_3	C_4				
	PVD	PMAP							$\overline{R}_\infty$	$\overline{R}_1$	$R_1(\%)$
0	Truth							0	0.4907	1.1311	59.2173
1	+	+	Viterbi	0	1	0	0	56.3758	0.1604	0.8296	50.3368
2	+	+	PMAP	1	0	0	0	45.6376	∞	0.6905	46.7752
3a	+		PVD	≈ 1	≈ 0	0	0	46.9799	0.2486	0.6961	46.9188
3b	+			≈ 1	0	0	≈ 0				
4		+	Constr. PMAP	≈ 1	≈ 0	0	0	46.9799	0.2468	0.6961	46.9188
5			Rabiner $k=2$	n/a	n/a	n/a	n/a	53.0201	0.1823	0.7118	47.4429
6	+			0.999	0.001	0	0	46.9799	0.2486	0.6961	46.9188
7	+			0.995	0.005	0	0	46.9799	0.2486	0.6961	46.9188
8	+			0.990	0.010	0	0	46.9799	0.2486	0.6961	46.9188
9	+			0.950	0.050	0	0	46.9799	0.2352	0.6964	46.9322
10	+			0.900	0.100	0	0	46.9799	0.2352	0.6964	46.9322
11	+			2/3	1/3	0	0	53.0201	0.1897	0.7065	47.2499
12	+			0.500	0.500	0	0	54.3624	0.1791	0.7142	47.5372
13	+			1/3	2/3	0	0	56.3758	0.1700	0.7277	48.0356
14	+			0.250	0.750	0	0	57.0470	0.1680	0.7331	48.1738
15	+			0.200	0.800	0	0	57.0470	0.1680	0.7331	48.1738
16	+			0.100	0.900	0	0	57.0470	0.1645	0.7637	48.9620
17	+			0.010	0.990	0	0	56.3758	0.1604	0.8296	50.3368
18	+	+	MA-Prior	0	0	0	1	57.0470	0.1645	0.7637	48.9620
19		+		0.999	0.001	0	0	46.9799	0.2486	0.6961	46.9188
20		+		0.995	0.005	0	0	46.9799	0.2486	0.6961	46.9188
21		+		0.990	0.010	0	0	46.3087	0.2417	0.6962	46.9245
22		+		0.950	0.050	0	0	50.3356	0.2009	0.7021	47.0773
23		+		0.900	0.100	0	0	50.3356	0.2009	0.7021	47.0773
24		+		2/3	1/3	0	0	54.3624	0.1776	0.7165	47.6139
25		+		0.500	0.500	0	0	57.0470	0.1680	0.7331	48.1738
26		+		1/3	2/3	0	0	57.0470	0.1680	0.7331	48.1738
27		+		0.250	0.750	0	0	57.0470	0.1645	0.7637	48.9620
28		+		0.200	0.800	0	0	56.3758	0.1604	0.8296	50.3368
29		+		0.100	0.900	0	0	56.3758	0.1604	0.8296	50.3368
30		+		0.010	0.990	0	0	56.3758	0.1604	0.8296	50.3368
31	+	+	PMA-Prior	0	0	1	0	57.0470	0.1645	0.7637	48.9620

Table 1: Case 877. Performance of the well-known and some of the new decoders. Worst, second worst, best and second best entries in each category are highlighted in red, magenta, blue and cyan respectively. In rows 1, 2, 3a, 6-17, $C_1 = 1 - \alpha = \frac{1}{k}$ and $C_2 = \alpha = 1 - \frac{1}{k}$.

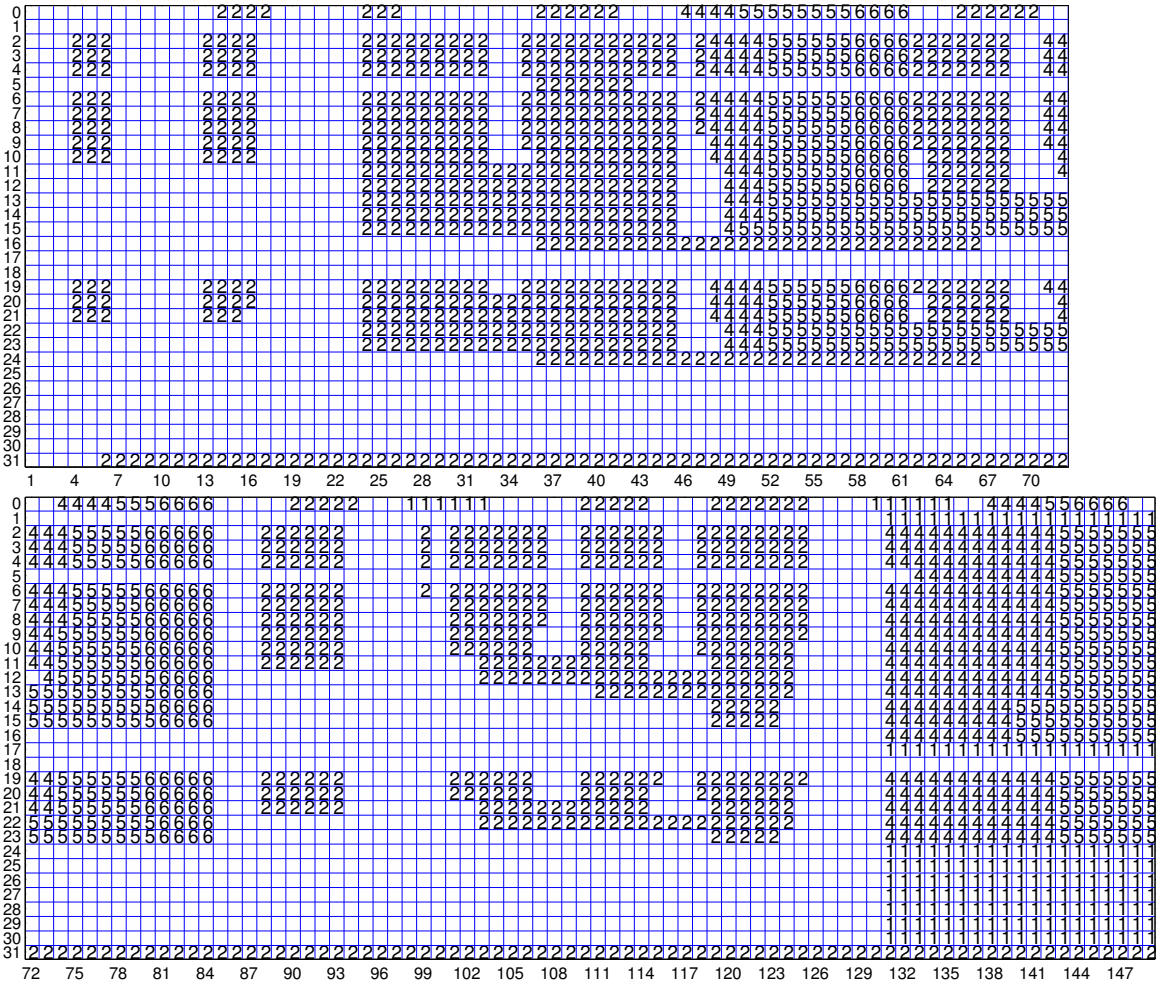

Figure 2: Performance of the selected decoders on Case 877. The dominant class 3 (blank entries) is suppressed by an asymmetric loss incorporated into the R_1 and $\bar{R}_1$ risks of the generalized hybrid decoders. Subsequently, the remaining classes reveal more activity, and in particular all of the seven instances of class 2 can be recognized with essentially only two false alarms.

simulated realization is a sample of length T_n from the (first order homogeneous) HMM with these parameters (note that the initial distributions $\{\hat{\pi}_{sn}\}_{s\in S}$ are necessarily degenerate). The simulations, first of all, help us obtain interval estimates of the performance measures (see more below). Also, they are valuable theoretically. Indeed, the analysis based on the real data tells us what happens in a typical practical scenario in which the (HMM) model is known to be too crude and yet has to be used for its simplicity. The simulations on the contrary tell us what happens when the model is correct. By default, the analysis below refers to the real data, whereas the use of the synthetic data will be acknowledged explicitly.

5.1 Performance Measures and Their Estimation

The performance measures discussed in this subsection will be used in the following two subsections to more completely assess and compare the performance of all the known decoders (including PMAP and Viterbi), and several new members of the generalized families.

Given a decoder v, our principal performance measures are the $R_1(v)$ risk $E[R_1(v(X^T) \mid X^T)]$ (see Equation 5) and the $\bar{R}_\infty$ risk $E[\bar{R}_\infty(v(X^T) \mid X^T)]$ (3); it is not practical to operate with R_∞ (2) since it is virtually 1 for reasonably long realizations. For the $\bar{R}_\infty$ results, see Subsection 5.3.

The R_1 risk is simply the point-wise error rate $\frac{1}{T}\sum_{t=1}^{T} P(\hat{Y}_t \neq Y_t)$, where $\widehat{Y^T}$ is the output of $v(X^T)$. This assumes T to be non-random; more generally, T is random and the R_1 risk is then given by $E_T\left[\frac{1}{T}\sum_{t=1}^{T} P\left(\hat{Y}_t \neq Y_t \mid T\right)\right]$. We refer to $1-R_1$ as *accuracy* when comparing our decoders (e.g., Section 5.2 below). Note that given a decoder v, $R_1(v)$, is simply a parameter of the underlying population of all (T, x^T, y^T) that could potentially be observed. If the current hidden Markov model were not too crude for this population, we would compute such risks if not analytically, then at least by using Monte-Carlo simulations, for any g of interest. In reality, however, we need to estimate them from the given data. The situation is further complicated by the fact that the classification method v is specified only up to the model parameters, which are unknown and also need to be estimated from the data.

All in all, we use the usual cross-validation (CV) estimation. Specifically, to decode $x^{T_n}(n)$, we make g use the estimates of the parameters obtained from the remaining $N-1$ sequences. Thus, if v outputs $\widehat{y^{T_n}}$, then we take the empirical point-wise error rate

$$\hat{e}_n = \frac{1}{T_n}\sum_{t=1}^{T_n} \mathbb{I}_{\{\hat{y}_t \neq y_t(n)\}} \tag{40}$$

to be an estimate of $R_1(v)$. Clearly, if v used the same fixed parameters as used in the definition of $R_1(v)$, then $E[\hat{e}_n] = R_1(v)$, that is, $\hat{e}_n$ would be unbiased for $R_1(v)$, and so would be the average

$$\hat{e}_{CV} = \frac{1}{N}\sum_{n=1}^{N} \hat{e}_n. \tag{41}$$

Obviously, in reality $\hat{e}_{CV}$ is likely to be biased. For this reason we also look at the model-based CV estimate of R_1 given by

$$\hat{R}_1 = \frac{1}{N}\sum_{n=1}^{N} R_1(\widehat{y^{T_n}} \mid x^{T_n}(n)). \tag{42}$$

Computation of $R_1(\cdot \mid x^T)$ indeed relies on the model being correct, hence $\hat{R}_1$ is also likely to be biased. We also report approximate 95% confidence intervals which are based on the usual normal approximation disregarding, among others, any effects of the variability in the realization length T.

If the variation in T were merely an observational artifact, then instead of the above cross-validation averages (42), we would focus on the total error rate for the entire data set given by (43) below.

$$\hat{e} \;=\; \frac{\sum_{n=1}^{N}\sum_{t=1}^{T_n}\mathbb{I}_{\{\hat{y}_t(n)\neq y_t(n)\}}}{\sum_{n=1}^{N}T_n} = \sum_{n=1}^{N} w(n)\hat{e}_n, \quad \text{where } w(n) = \frac{T_n}{\sum_{n=1}^{N}T_n}. \tag{43}$$

However, to obtain sensible confidence intervals in this setting, we need to estimate the variance of $\hat{e}$. Bootstrapping is a possibility, but we instead simulate several (specifically, 15) synthetic data sets as described above in the introduction to this Section, that is, re-sampling individual realizations $(x^{T_n}(n), y^{T_n}(n))$ from the HMM with parameters $\{\hat{\pi}_{sn}\}_{s\in S}$, $\widehat{\mathbb{P}}_n$, $\{\widehat{P}_{sn}, s \in S\}$, $n = 1, 2, \ldots, N$. We then use the t-distribution (on 14 degrees of freedom) to obtain the 95% margins of error.

5.2 Comparison of the Accuracy of the Viterbi and PMAP Decoders

A histogram of the difference $\hat{e}(\text{Viterbi}, n) - \hat{e}(\text{PMAP}, n)$ between the empirical errors (40) of the Viterbi and PMAP decoders is plotted in Figure 3 (black narrow bins). We also observe that in 85.35% of the CV rounds the PMAP decoder is more accurate, and in 10.67%—less accurate, than the Viterbi decoder (in 3.98% of the cases the two methods show the same accuracy). To examine sensitivity of these results to the variation in the realization length, we superimpose in the same Figure 3 a histogram of the subsample consisting of the 1000 longest realizations (blue wide bins). Although the subsample spans a less extreme range $(-16.75\%, 52.62\%)$ than that of the entire sample, the locations of the two histograms are very similar, suggesting the average *gain of accuracy of about 12% when replacing the Viterbi decoder by the PMAP one.*

We also compare the performance of the Viterbi and PMAP decoders by examining their $R_1(\cdot \mid x^{T(n)}(n))$ risks (5), see Figure 4. Note that the difference $\hat{R}_1(\text{Viterbi}) - \hat{R}_1(\text{PMAP})$ is 9% on average, and is largely unchanged (apart from a minor increase) when recomputed on the subsample of the 1000 longest realizations (450-2060 positions).

Finally, $\hat{e}$ (43) is 59.68% ($\pm$0.068%) and 46.10% ($\pm$0.047%) for the Viterbi and PMAP decoders, respectively, and the PMAP comes out $13.58\% \pm 0.0463\%$ more accurate than the Viterbi decoder. The above confidence intervals are, however, likely to be deflated since the model-based simulations show little variation of $\hat{e}$(Viterbi), $\hat{e}$(PMAP), or the differences $\hat{e}(\text{Viterbi}) - \hat{e}(\text{PMAP})$. In fact, based on the 15 model-based simulations, the PMAP is only $7.46\% \pm 0.0463\%$ more accurate than the Viterbi decoder, with the individual error rates of $47.49\% \pm 0.047\%$ and $54.95\% \pm 0.068\%$ for the former and the latter, respectively. Finally, replacing the empirical error rates by the $R_1(\cdot \mid x^T)$ risks (which are now computed exactly since the simulations are model-based), we obtain the difference of $8.55\% \pm 0.0213\%$.

In summary, the PMAP decoder can be notably more accurate than the Viterbi decoder in scenarios with as few as six hidden states.

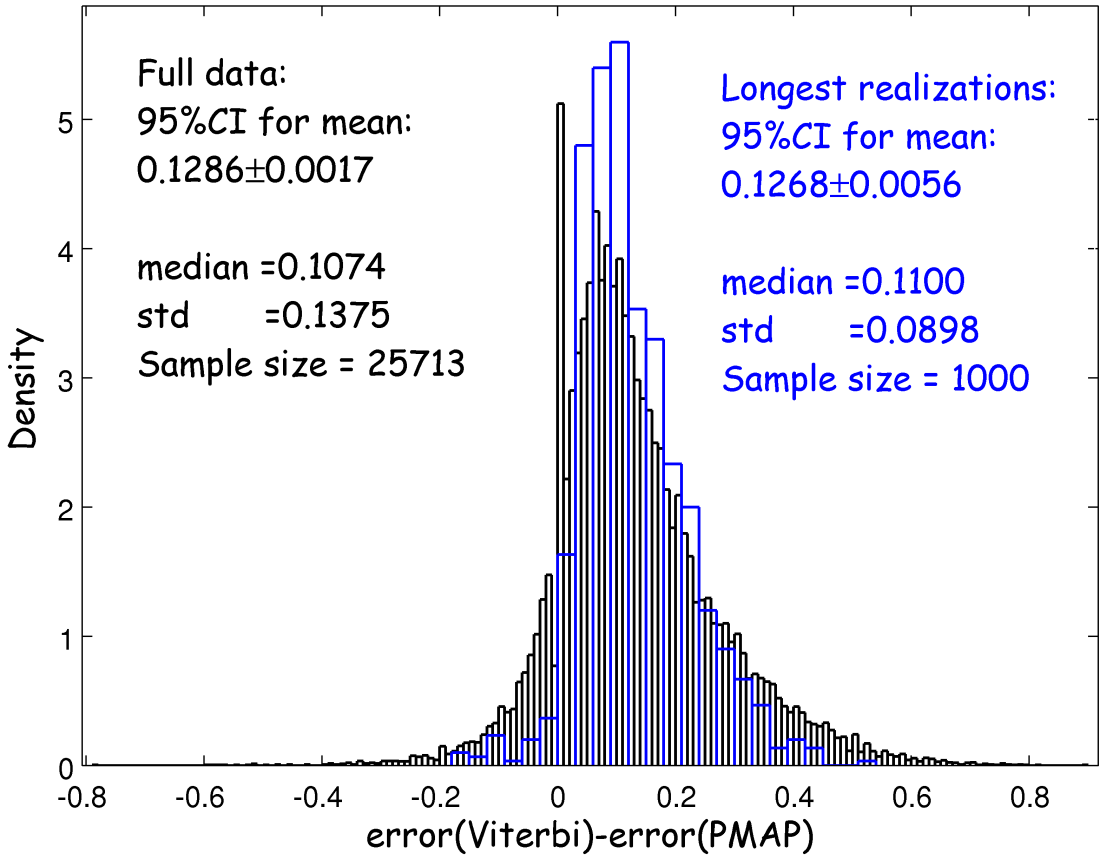

Figure 3: A histogram of the difference between the empirical error rates $\hat{e}(\text{Viterbi}, n) - \hat{e}(\text{PMAP}, n)$ obtained from the full data (black narrow bins) and the subsample consisting of 1000 longest realizations (blue wide bins). Although in 3.98% of the entire data set the two methods show the same accuracy (spike at 0), overall their performance appears to be notably different. The Viterbi decoder is more accurate in 10.67% of all the cases, and the PMAP decoder is more accurate in 85.35% of all the cases. The extreme differences (min $= -78.69\%$,max $=$ 89.74%) tend to be observed on short sequences (136 positions and shorter), but the subsample of the 1000 longest realizations (450-2060 positions) confirms the effect of the PMAP decoder being more accurate. In particular, on the longest sequences, the PMAP decoder can be 52.62% more accurate than the Viterbi decoder, whereas the latter can be at most 16.75% more accurate than the former.

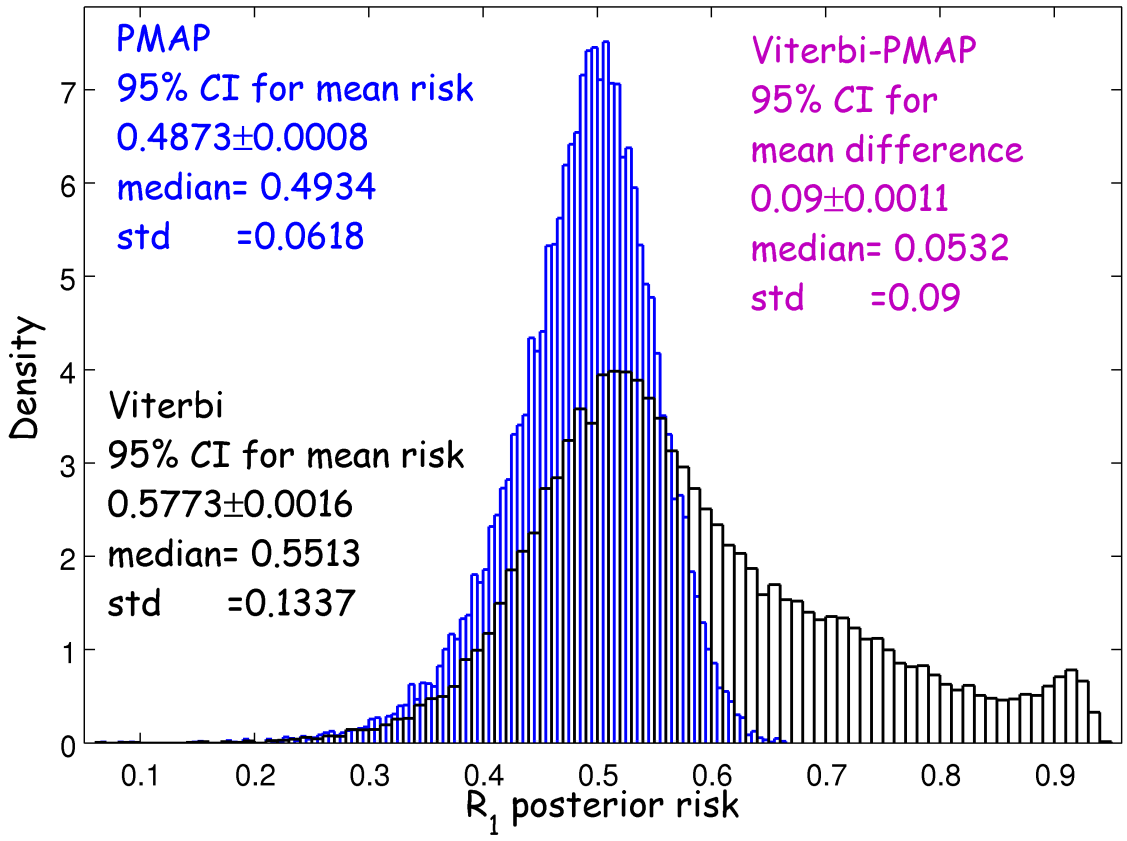

Figure 4: Histograms of the $R_1(\widehat{y^{T(n)}} \mid x^{T(n)}(n))$ risk of the Viterbi (black, more spread) and PMAP (blue, more peaked) decoders. Since the first order homogeneous HMM is only an approximation to the data source, the cross-validation averages of 48.73% (PMAP), 57.73% (Viterbi), and 9% (PMAP's gain over Viterbi) are likely to be biased as estimates of the respective pointwise error rates; see also Figure 3 for a model independent analysis.

5.3 The $\bar{R}_\infty$ Risk of the Viterbi, PMAP and Other Decoders

Next we look at the log-posterior probability rates $\log(P(\widehat{y^T} \mid x^T))/T = -\bar{R}_\infty(\widehat{y^T} \mid x^T)$ of the PMAP, Viterbi and other decoders. In 74.14% of the cases, the PMAP decoder returns an inadmissible path, that is, $\log(P(\widehat{y^T} \mid x^T))/T = -\infty$. To avoid dealing with an infinite range, we switch to the exponential scale. Thus, Figure 5 below displays histograms of the geometric rates $\sqrt[T]{P(\widehat{y^T} \mid x^T)}$.

The Rabiner 2-block decoder $\hat{y}(2)$ returns inadmissible paths in 70.94% of the cases. In 7.32% of the cases this decoder gives an inadmissible path even when the PMAP path (for the same realization) is admissible. This illustrates the violation of monotonicity (see

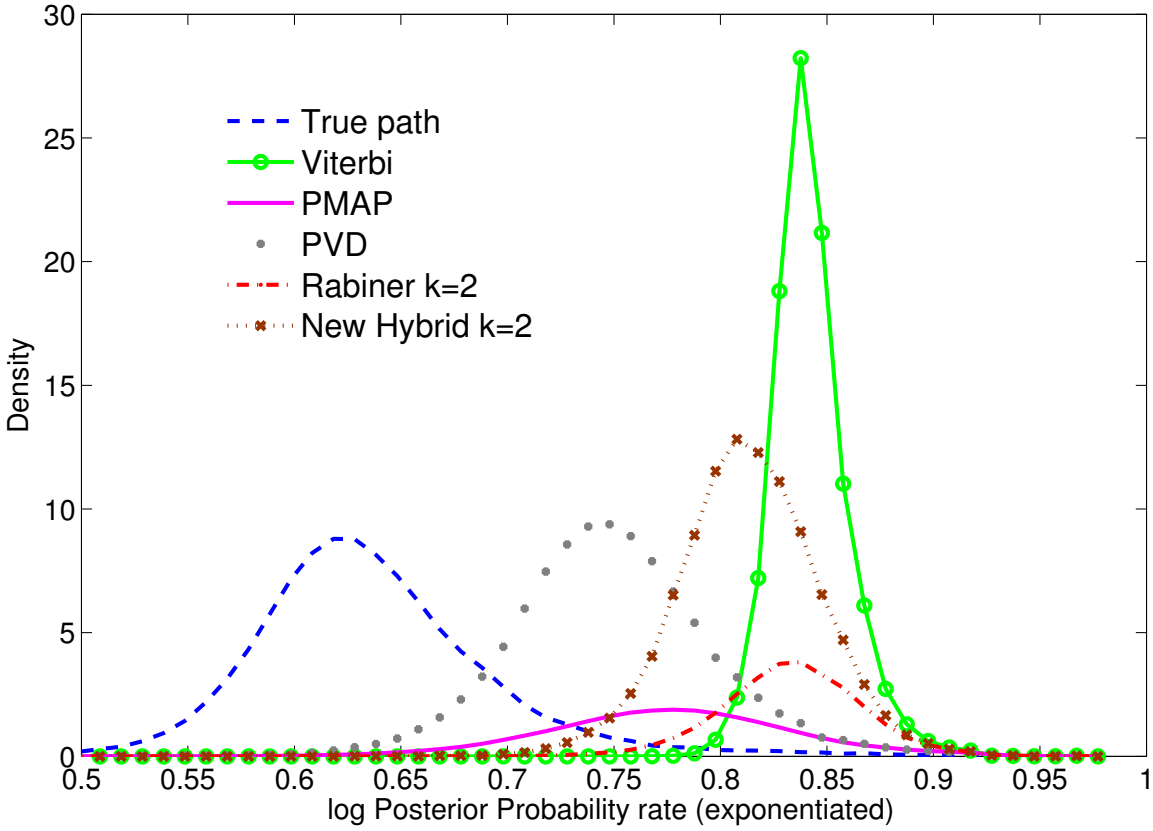

Figure 5: Distributions of the (geometric rates of the) posterior probabilities of selected decoders. The Constrained PMAP decoder is virtually indistinguishable from PVD, hence omitted. The PMAP and Rabiner 2-block (see Subsection 1.2.1) decoders return inadmissible paths in 74.14% and 70.94% of the cases (not shown), respectively (hence only 25.86% and 29.06% of the respective distributions are shown). Just like PVD and the Constrained PMAP decoder, the new hybrid 2-block posterior-Viterbi decoder (33) is guaranteed to produce admissible paths. Moreover, those paths would generally have a higher probability than the probabilities of the PVD and Constrained PMAP paths.

Subsection 1.2.1) in the path (posterior) probability when using Rabiner's suggestion to base decoding on the loss (30).

We also note that the posterior probabilities of the actual hidden paths (blue histogram) are notably lower than those of the admissible decodings, especially the Viterbi outputs. However, these effects are not out of line with the model-based simulations.

5.4 Summary of the Experiments

Figure 6 compares performance of these and other decoders as measured by the averaged error rate and the averaged (exponentiated) path log-posterior rate

$$\widehat{\sqrt[T]{P(\widehat{y^T} \mid x^T)}}_{CV} = \frac{1}{N}\sum_{n=1}^{N} \sqrt[T_n]{P(\widehat{y^{T_n}} \mid x^{T_n}(n))}. \tag{44}$$

Recall that the family of k-block posterior-Viterbi decoders is naturally parameterized by the block length k ($k = 1$ and $k \to \infty$ giving the PMAP and Viterbi decoders, respectively). We have also included the continuous re-parameterization (38) via $k = \frac{1}{1-\alpha}$ (and $\alpha = \frac{k-1}{k}$) which embeds these special cases into the generalized PVD Problem (18) via $C_1 = \alpha$, $C_2 = 1-\alpha$, $C_3 = C_4 = 0$.

Figure 6 displays performance of members of the generalized PVD and generalized PMAP (Problem 26) families with $C_1 = \alpha$, $C_2 = 1-\alpha$, $C_3 = C_4 = 0$ for a subset of values of α used in Figure 1 and Table 1. The point-wise maximum a priori ($C_1 = C_2 = C_4 = 0$, $C_3 = 1$) and the prior-based Viterbi ($C_1 = C_2 = C_3 = 0$, $C_4 = 1$) decoders are also included, showing identical performance on these data. Remarkably (but not very surprisingly given the crudeness of the hidden Markov model for these data), the accuracy of these "data-blind" decoders on average is still higher than that of the Viterbi (MAP) decoder. We reiterate that the hidden Markov model is rather crude as a model for the given data. Furthermore, the estimates of the model parameters used for decoding any given sequence are obtained from sequences that can generally have very different characteristics from the sequence being decoded. Therefore, the risks optimized under these conditions may be misleading, for example, a PMAP path need not have the lowest empirical error rate. Nonetheless, the empirical error rates of the generalized decoders are still found to follow the theoretical order of the posterior R_1 and $\bar{R}_1$ risks.

6. Algorithmic Approaches

It is also possible (at least when the Viterbi path is unique) to hybridize MAP and PMAP inferences without introduction of risk/loss functions. We discuss such approaches mainly because one such approach was taken by Brushe et al. (1998) in what appears to be the only publication dedicated to hybridization of the MAP and PMAP inferences in HMMs.

First note that the hybridization can be achieved by a suitable transformation of the forward and backward variables $\alpha_t(i)$ and $\beta_t(i)$ defined in (1). To make this concrete, consider the recursively applied power transformations with $\mu > 0$ given in (45) below

$$\begin{aligned}
\alpha_1(i;\mu) &:= \alpha_1(i); \qquad (45)\\
\alpha_t(i;\mu) &:= \left[\sum_{j=1}^{K} (\alpha_{t-1}(j;\mu)p_{ji})^{\mu}\right]^{\frac{1}{\mu}} f_i(x_t), \quad t = 2,3\ldots,T;\\
\beta_T(i;\mu) &:= \beta_T(i) = 1;\\
\beta_t(i;\mu) &:= \left[\sum_{j=1}^{K} (p_{ij}f_j(x_{t+1})\beta_{t+1}(j;\mu))^{\mu}\right]^{\frac{1}{\mu}}, \quad t = T-1, T-2, \ldots, 1,
\end{aligned}$$

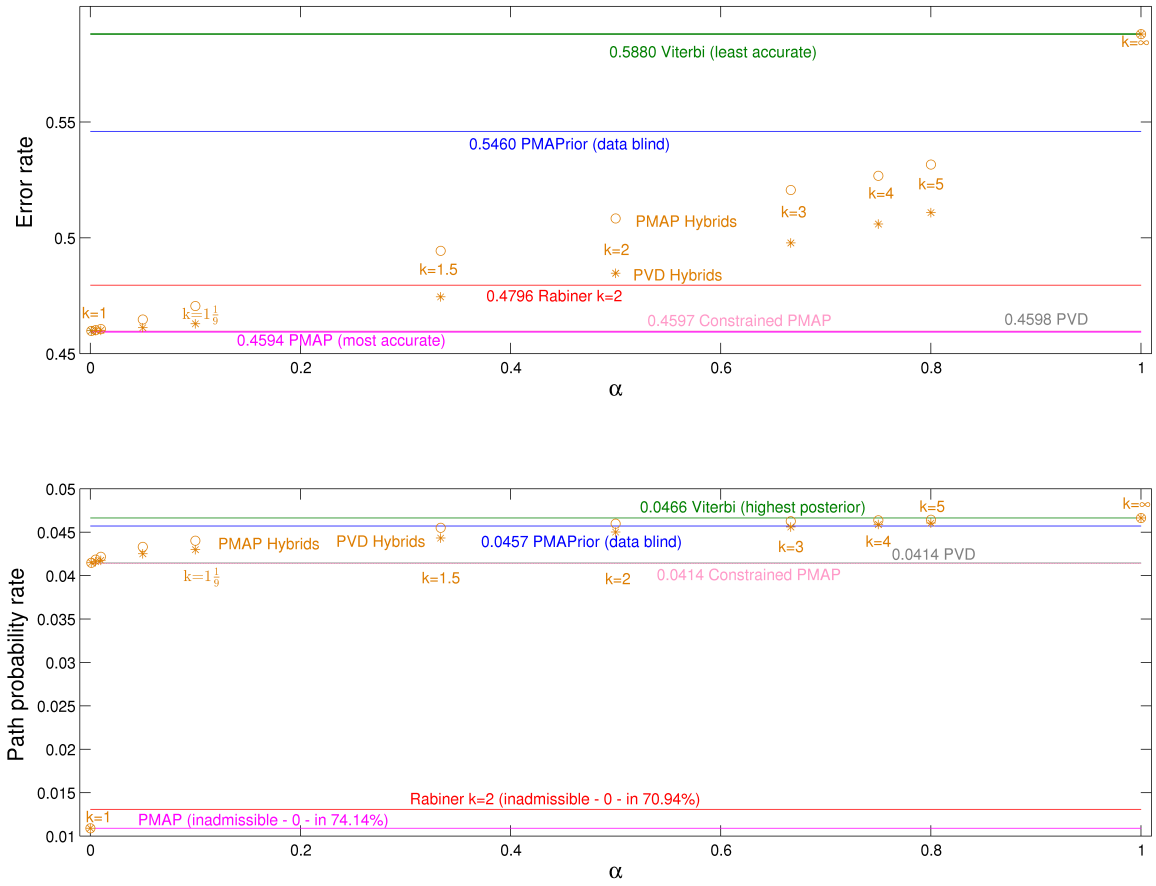

Figure 6: Empirical error (41) (top) and probability rates (44) (bottom) of the popular and some new members of the generalized PVD (asterisk) and PMAP (circle) families.

for all $i \in S$. Clearly, $\alpha_t(i;1) = \alpha_t(i)$ and $\beta_t(i;1) = \beta_t(i)$, for all $i \in S$ and all $t = 1, 2, \ldots, T$. Thus, $\mu = 1$ leads to the PMAP decoding, that is, at time t returning

$$\hat{y}_t(1) = \arg\max_{i \in S} \{\alpha_t(i;1)\beta_t(i;1)\}, \tag{46}$$

provided some tie-breaking rule.

Using induction on t and continuity of the power transform, it can also be seen that the following limits exist and are finite for all $i \in S$ and all $t = 1, 2, \ldots, T$: $\lim_{\mu\to\infty} \alpha_t(i;\mu) =: \alpha_t(i,\infty)$ and $\lim_{\mu\to\infty} \beta_t(i;\mu) =: \beta_t(i;\infty)$, where

$$\begin{aligned} \alpha_t(i;\infty) &= \max_{s^t : s_t = i} p(x^t, s^t), \quad t = 1, 2, \ldots, T, \\ &= \max_{j \in S} \left(\alpha_{t-1}(j;\infty) p_{ji}\right) f_i(x_t), \quad t = 2, 3, \ldots, T, \end{aligned} \tag{47}$$

$$\beta_t(i;\infty) = \max_{s_{t+1}^T \in S^{T-t}} p(x_{t+1}^T, s_{t+1}^T \mid Y_t = i), \quad t = T-1, T-2, \ldots, 1, \quad \text{and } \beta_T(i;\infty) = 1,$$
$$\max_{j \in S} \left(p_{ij} f_j(x_{t+1}) \beta_{t+1}(j;\infty)\right).$$

The above convergence follows from the following trivial observation, which we nonetheless prove below for reasons to become clear later on in the context of Equation (50).

Proposition 9 *Let $a_j(\mu)$, $j = 1, 2, \ldots, K$, be non-negative as functions of $\mu \in (0, \infty)$. Assume that $a_j(\mu)$ converges to some (finite) limit a_j as $\mu \to \infty$. Assume further that for any μ, at least some of the $a_j(\mu)$ are positive. Then we have*

$$\lim_{\mu \to \infty} \left(\sum_{j=1}^{K} a_j(\mu)^\mu \right)^{\frac{1}{\mu}} = \max_{1 \le j \le K} \{a_j\}.$$

Proof Let $M(\mu) = \max_{1 \le j \le K} \{a_j(\mu)\}$, and let $M = \max_{1 \le j \le K} \{a_j\}$. Write $\left(\sum_{j=1}^{K} a_j(\mu)^\mu \right)^{\frac{1}{\mu}} = M(\mu) \left(\sum_{j=1}^{K} \left(\frac{a_j(\mu)}{M(\mu)} \right)^\mu \right)^{\frac{1}{\mu}}$ and note that as $\mu \to \infty$, $M(\mu)$ converges to M. Also, we have

$$1 \le \left(\sum_{j=1}^{K} \left(\frac{a_j(\mu)}{M(\mu)} \right)^\mu \right)^{\frac{1}{\mu}} \le K^{\frac{1}{\mu}}.$$

Since $K^{\frac{1}{\mu}} \to 1$, by the Sandwich Theorem the middle term also converges to 1, yielding the proposed result. ■

Returning to (47), we note that any Viterbi path $\widehat{y^T}(\infty)$ satisfies the following property:

$$\hat{y}_t(\infty) = \arg\max_{i \in S} \{\alpha_t(i;\infty)\beta_t(i;\infty)\}. \tag{48}$$

The above property (48) has already been pointed out by Brushe et al. (1998). The main motivation of Brushe et al. (1998), however, seems to be the case of continuous emission distributions P_s, which might explain why the authors do not consider the fact that *not every path that satisfies* (48) *is necessarily Viterbi, or MAP*. Thus, ignoring potential non-uniqueness of the Viterbi paths, Brushe et al. (1998) state, based on (48), that the Viterbi path can be found *symbol-by-symbol*. As the following simple example shows, *when the Viterbi path is not unique, the attempt to implement the Viterbi decoding in the symbol-by-symbol fashion (based on Equation 48) can produce suboptimal (in the MAP sense), or even inadmissible, paths.*

Example 1 *Let $S = \{1, 2, 3\}$ and let $\{A, B, C, D\}$ be the emission alphabet. Let the initial distribution π, transition probability matrix $\mathbb{P}$, and the emission distributions f_s, $s \in S$, be defined as follows:*

$$\pi = \begin{pmatrix} 0.4 \\ 0.54 \\ 0.06 \end{pmatrix} \quad \mathbb{P} = \begin{pmatrix} 0.6 & 0.4 & 0 \\ 0.1 & 0.1 & 0.8 \\ 0 & 0.02 & 0.98 \end{pmatrix} \quad \begin{array}{lcccc} & A & B & C & D \\ f_1(\cdot) & 0.3 & 0.15 & 0.25 & 0.3 \\ f_2(\cdot) & 0.2 & 0.3 & 0.3 & 0.2 \\ f_3(\cdot) & 1/6 & 1/6 & 1/6 & 1/2 \end{array}.$$

Suppose the sequence $x^2 = (A, B)$ *has been observed. The (posterior) probabilities of all the nine paths* (i, j) *are then summarized in the matrix* $PP = (P(Y^2 = (i, j) \mid AB))$ *below:*

$$PP = \begin{pmatrix} 0.0108 & 0.0144 & 0 \\ 0.0016 & 0.0032 & 0.0144 \\ 0 & 0.0001 & 0.0016 \end{pmatrix},$$

hence there are two Viterbi paths in this case, namely $(1, 2)$ *and* $(2, 3)$*. Now,* $\alpha_1(i; \infty) = \pi_i f_i(A)$*,* $i \in S$*, and* $\beta_1(i; \infty) = \max_{j \in S} P(X_2 = B, Y_2 = j \mid Y_1 = i) = \max_{j \in S} f_j(B) p_{ij}$*, or, in the vector form:*

$$\begin{pmatrix} \alpha_1(1;\infty) \\ \alpha_1(2;\infty) \\ \alpha_1(3;\infty) \end{pmatrix} = \begin{pmatrix} 0.12 \\ 0.108 \\ 0.01 \end{pmatrix}, \quad \begin{pmatrix} \beta_1(1;\infty) \\ \beta_1(2;\infty) \\ \beta_1(3;\infty) \end{pmatrix} = \begin{pmatrix} 0.12 \\ 2/15 \\ 49/300 \end{pmatrix}, \quad \begin{pmatrix} \alpha_1(1;\infty)\beta_1(1;\infty) \\ \alpha_1(2;\infty)\beta_1(2;\infty) \\ \alpha_1(3;\infty)\beta_1(3;\infty) \end{pmatrix} = \begin{pmatrix} 0.0144 \\ 0.0144 \\ 49/30000 \end{pmatrix},$$

so we have $\hat{y}_1(\infty) = 1$ *or* $\hat{y}_1(\infty) = 2$*. On the other hand,* $\alpha_2(i; \infty) = \max_{j \in S} P(X^2 = (A, B), Y^2 = (j, i))$*, and* $\beta_2(i, \infty) = 1$ *for all* $i \in S$*. Therefore,*

$$\begin{pmatrix} \alpha_2(1;\infty) \\ \alpha_2(2;\infty) \\ \alpha_2(3;\infty) \end{pmatrix} = \begin{pmatrix} \alpha_2(1;\infty)\beta_2(1;\infty) \\ \alpha_2(2;\infty)\beta_2(2;\infty) \\ \alpha_2(3;\infty)\beta_2(3;\infty) \end{pmatrix} = \begin{pmatrix} \max\{0.0108, 0.0016, 0\} \\ \max\{0.0144, 0.0032, 0.0001\} \\ \max\{0, 0.0144, 0.0016\} \end{pmatrix} = \begin{pmatrix} 0.0108 \\ 0.0144 \\ 0.0144 \end{pmatrix}.$$

Therefore, $\hat{y}_2(\infty) = 2$ *or* $\hat{y}_2(\infty) = 3$*. However, the symbol-by-symbol decoding is not aware that gluing* $\hat{y}_1(\infty) = 1$ *and* $\hat{y}_2(\infty) = 3$ *is not only suboptimal, but is actually forbidden, that is, results in the inadmissible path* $(1, 3)$*.*

In contrast to Viterbi, the PMAP inference (in the absence of constraints) is by definition *point-wise*, or symbol-by-symbol, hence violation of admissibility is not surprising there regardless of the non-uniqueness issue.

All in all, the main idea of Brushe et al. (1998) is to consider "hybrid" decoders that use intermediate values of the interpolation parameter μ. That is, the hybrid decoder with parameter μ is defined as a decoder that at time t returns

$$\hat{y}_t(\mu) = \arg\max_{i \in S} \{\alpha_t(i; \mu)\beta_t(i; \mu)\}, \tag{49}$$

provided some tie-breaking rule.

Note also that in their attempt to hybridize PMAP with Viterbi in this manner, Brushe et al. (1998) instead of (45) use different transformations that are based on the following $(0, \infty) \to \mathbb{R}$ composite mapping

$$F(\mu, d_1(\mu), d_2(\mu), \ldots, d_N(\mu)) := \frac{1 + (N-1)\exp(-\mu)}{\mu} \log\left(\frac{1}{N}\sum_{j=1}^{N} \exp\left(\mu d_j\left(\mu\right)\right)\right), \tag{50}$$

where $N = K$ (in our notation) and functions $d_j\left(\mu\right)$ are continuous on $[0, \infty)$ with finite limits $d_j(\infty)$ as $\mu \to \infty$. It is then not hard to verify that as $\mu \to 0$, the function (50) converges to $\sum_{j=1}^{N} d_j(0)$ (based on Brushe et al., 1998, Proposition 1a). At the same time, as $\mu \to \infty$ the same function converges to $\max_{1 \le j \le N}\{d_j(\infty)\}$ (based on Brushe et al.,

1998, Proposition 1b). To establish the latter convergence, Brushe et al. (1998) refer to the Varadhan-Laplace Lemma, although the result can also be obtained with basic calculus, for example, by using continuity of the logarithmic function, taking the logarithm inside the limit in Proposition 9, and identifying $a_j(\mu)$ with $e^{d_j(\mu)}$.

This mapping is then applied recursively to $\alpha_t(i;\mu)$ and $\beta_t(i;\mu)$, the analogs of the forward and backward variables ($\kappa_t^\mu(i)$ and $\tau_t^\mu(i)$, respectively, in the notation of Brushe et al., 1998), to produce the correct end points/limits, that is, PMAP and Viterbi/MAP (when the latter is unique). Specifically, the transformed forward and backward variables would be re-defined as follows:

$$\begin{aligned}
\alpha_1(i;\mu) &:= \alpha_1(i); \qquad (51)\\
\alpha_t(i;\mu) &:= \frac{1+(N-1)\,e^{-\mu}}{\mu}\log\left(\frac{1}{N}\sum_{j=1}^{N} e^{\mu\alpha_{t-1}(j;\mu)p_{ji}}\right) f_i\left(x_t\right),\ t=2,3,\ldots,T;\\
\beta_T(i;\mu) &:= \beta_T(i)=1;\\
\beta_t(i;\mu) &:= \frac{1+(N-1)\,e^{-\mu}}{\mu}\log\left(\frac{1}{N}\sum_{j=1}^{N} e^{\mu\beta_{t+1}(j;\mu)p_{ij}f_j(x_{t+1})}\right),\ t=T-1,T-2,\ldots,1.
\end{aligned}$$

Above, we took the liberty to correct $\kappa_1^\mu(i)=\pi(i)$ ($\alpha_1(i;\mu)=\pi_i$ in our notation), which appears in the paper of Brushe et al. (1998) as Equation (22) and also in the proofs of parts (a) and (b) of their Lemma 1. Clearly, in order for $\kappa_1^\mu(i)$ ($\alpha_1(i;\mu)$ in our notation) to match $\alpha_1(i)=P(Y_1=i,X_1=x_1)$ (as claimed in their Lemma 1), $\kappa_1^\mu(i)$ has to equal $\pi(i)b_i(O_1)$ (which is $\pi_i f_i(x_1)$ in our notation). Note that Equation (15) of Brushe et al. (1998) leaves $\alpha_1(i)$ undefined, but instead introduces $\alpha_0(i)$, which is defined to be $\pi(i)$. If that was an implicit intention to introduce a "silent" state at $t=0$, then their Equation (22) and the relevant parts of the proof of Lemma 1 would also have to start with $t=0$ and not with $t=1$. If, on the other hand, $t=0$ in Equation (15) was simply a typing error and the intention was to have $t=1$, then the would-be definition of $\alpha_1(i)=\pi(i)$ contradicts an earlier equation just below their Equation (14), which gives $\alpha_1(i)=P(O_1,q_1=S_i)=\pi(i)b_1(O_1)$ (that is, $P(Y_1=i,X_1=x_1)=\pi_1 f_1(x_1)$ in our notation).

Returning to the essence of the approach, note that the only reason stated by Brushe et al. (1998) for choosing (51) as the family of interpolating transformations is the attainment of the required limits (that is, PMAP when $\mu\to 0$, and Viterbi when $\mu\to\infty$). It is therefore not clear if Brushe et al. (1998) realized that besides (51), there are other (single parameter) families of transformations, such as (45), with the same limiting behavior. Naturally, the resulting interpolation generally depends on the choice of the transformations used. In the absence of any special reason for using (51), (45) may have an appeal for its simplicity, should one really wish to pursue the idea of algorithmic hybridization. Moreover, we explain next (Subsection 6.1) *why the hybrid decoder defined by* (49) *and the transformations* (51) *does not work in practice except with trivial examples*, and we also show (Subsection 6.3) *how this decoder can be modified to become operational*. In contrast to this, we will show (Subsection 6.2) that *the hybrid decoder based on the transformations* (45) *becomes operational by modifying just the algorithm used for its computation, and not the decoder*. This makes the transformations (45) even more attractive as an alternative to (51).

6.1 The Hybrid Decoder Based on the Transformations (51) Does Not Work in Practice Except with Trivial Examples

The key point is that the transform-based algorithmic hybridization attempts to compute quantities which, at least for $\mu \approx 0$, are the same order of magnitude as the forward and backward probabilities $\alpha_t(i) = P(X^t = x^t, Y_t = i)$ and $\beta_t(i) = P(x_{t+1}^T \mid Y_t = i)$. These are well-known to vanish exponentially fast with T, see, for example, Bishop (2006, 13.2.4) who also note that "[f]or moderate lengths of chain (say 100 or so), the calculation of the $[\alpha_t(j)]$ will soon exceed the dynamic range of the computer, even if double precision floating point is used." The situation clearly gets worse as μ increases. Indeed, recall (47), and note that $\max_{s^t:s_t=i} p(x^t, s^t) = \alpha_t(i;\infty) \leq \sum_{s^t:s_t=i} p(x^t, s^t) = \alpha_t(i)$ (which is also $\alpha_t(j;1)$ in Equation 45 and $\alpha_t(j;0)$ in Equation 51). This easily leads to a collapse of computations already with chains as short as $T = 10$ (which indeed happens using the data and model from our experiments of Section 5 above).

We disagree with Brushe et al. (1998) in interpreting the nature of the above numerical problems when they divert the reader's attention to the computation of the `logsumexp` function used in their transforms (50), (51). We find this is misleading as the $\log(e^a + e^b) = \max\{a, b\} + \log\left(1 + e^{-|a-b|}\right)$ trick (alluded to by Brushe et al., 1998 in their Remark below Equation 25) is relevant to the problem of underflow only of the *intermediate values* (that is, $e^a + e^b$ when a or b is negative of a large magnitude, such as the logarithm of a very small probability). In the case of the transform (50), however, computations of the transformed, say, forward variable $\alpha_t(i;\mu)$ (51), do require $\mu d_j(\mu) = \mu\alpha_{t-1}(j;\mu)p_{ji}$ and not their logarithm. Thus, at some t underflow in $\alpha_t(i;\mu)$ occurs for some i, and then eventually for all i. In terms of the `logsumexp` function, this means that both e^a and e^b become 1 (and not zero!) but the logarithm of their average (the core of the transform 50) becomes 0, transferring the underflow to the next generation, that is, $\alpha_{t+1}(i;\mu)$. Thus, storing $\alpha_t(i;\mu)$ in the log-domain is irrelevant here since the transforms (50), (51) with or without the `logsumexp` trick, do require the actual value of $\alpha_t(i;\mu)$. One could conceivably introduce the `loglogsumexpexp` function to operate on $\log(\alpha_t(i;\mu))$ and resolve this problem in that way, but it is not clear if the goal is worth the effort.

Furthermore, insisting that "[t]he computational complexity and numerical implementation issues associated with the hybrid algorithm can be overcome using the Jacobian logarithm", Brushe et al. (1998, p. 3133) repeatedly refer to another paper, which proposes to compute the `logsumexp` function $\log(\sum_k \exp(a_k))$ via recursive application of $\log(e^a + e^b) = \max\{a, b\} + \log\left(1 + e^{-|a-b|}\right)$. Although this recursive implementation should indeed be generally more accurate (albeit also computationally more expensive) than the commonly used single-shift implementation $\log(\sum_k \exp(a_k)) = M + \log(\exp(a_k - M))$ ($M = \max_k\{a_k\}$), as we just explained above, it is irrelevant to the real problem of computing the transformed forward and backward variables $\alpha_t(i;\mu)$, $\beta_t(i;\mu)$ ($\kappa_t^\mu(i)$, $\tau_t^\mu(i)$, respectively, of Brushe et al., 1998). Thus, the approach of Brushe et al. (1998) *does not immediately provide an operational decoding algorithm* except for trivially short chains. For example, using the two-state HMM from the Example 2 and the 64-bit MATLAB (MATLAB, 2011) (but without The Symbolic Math Toolbox) installation on a (64-bit) Linux machine, the hybrid decoder based on (51) with $\mu = 1$ already fails for $T = 40$ (with or

without the `logsumexp` trick). For comparison, the hybrid decoder based on the power transform (45) ($\mu = 1$) survives an order of magnitude longer.

A natural question is then whether the transform-based algorithmic hybridization approach (using (51) or (45), or the like) can at all work in practice. The fact that no such example has been given by Brushe et al. (1998), or anyone else up to date, casts some doubt. Below we give reassuring answers, which have been verified to work on several realistic examples.

Indeed, it is well-known that in practice, to decode the t-th symbol the PMAP decoder uses the posterior probabilities $p_t(i \mid x^T)$ and not the vanishing joint probabilities $p_t(i \mid x^T)p(X^T = x^T) = P(x^T, Y_t = i) = \alpha_t(i)\beta_t(i)$. The posterior probabilities $p_t(i \mid x^T)$ are computed as $\tilde{\alpha}_t(i)\tilde{\beta}_t(i)$, where $\tilde{\alpha}_t(i) = P(Y_t = i \mid x^t)$ and $\tilde{\beta}_t(i) = P(x_{t+1}^T \mid Y_t = i)/p(x_{t+1}^T \mid x^t)$ are the scaled analogs of the forward and backward probabilities $\alpha_t(i)$ and $\beta_t(i)$ (Bishop, 2006, 13.2.4). This allows PMAP to bypass the aforementioned problem of numerical underflow.

6.2 The Hybrid Decoder (49) is Invariant to Rescaling of the Power-Transformed (45) Forward and Backward Variables $\alpha(\cdot;\mu)$, $\beta(\cdot;\mu)$.

Let us apply the same normalization approach to the transformed forward and backward variables, first, using the power transform (45) and then (51). First, recall (e.g., Bishop, 2006, 13.2.4) that $\tilde{\alpha}_t(i)$ are obtained by replacing the recursive definition

$$\alpha_t(i) = f_i(x_t) \sum_{j=1}^{K} \alpha_{t-1}(j) p_{ji}, \quad i = 1, 2, \ldots, K,$$

by the two-step self-normalized definition

$$\begin{aligned} p(x_t \mid x^{t-1})\tilde{\alpha}_t(i) &= f_i(x_t) \sum_{j=1}^{K} \tilde{\alpha}_{t-1}(j) p_{ji}, \quad i = 1, 2, \ldots, K, \\ \tilde{\alpha}_t(i) &= \frac{p(x_t \mid x^{t-1})\tilde{\alpha}_t(i)}{\sum_{s=1}^{K} p(x_t \mid x^{t-1})\tilde{\alpha}_t(s)}, \quad \text{for } t = 2, \ldots, T, \\ \text{where} \quad \tilde{\alpha}_1(i) &= \alpha_1(i)/c_1, \text{ and } c_1 := p(x_1) = \sum_{s=1}^{K} \alpha_1(s). \end{aligned}$$

Thus, for all $t = 2, 3 \ldots T$, and for all $i = 1, 2, \ldots, K$,

$$\tilde{\alpha}_t(i) = \frac{f_i(x_t) \sum_{j=1}^{K} \tilde{\alpha}_{t-1}(j) p_{ji}}{c_t}, \quad \text{where, also according to Bishop (2006, Equation 13.56),}$$

$$c_t := p(x_t \mid x^{t-1}) = \sum_{s=1}^{K} f_s(x_t) \sum_{j=1}^{K} \tilde{\alpha}_{t-1}(j) p_{js}.$$

Similarly, the rescaled backward variables are given by

$$\begin{aligned} \tilde{\beta}_T(i) &:= 1; \\ \tilde{\beta}_t(i) &:= \frac{\sum_{j=1}^{K} p_{ij} f_j(x_{t+1}) \tilde{\beta}_{t+1}(j)}{c_{t+1}}, \quad t = T-1, T-2, \ldots, 1. \end{aligned}$$

In the same manner, we normalize the $\alpha_t(i;\mu)$ and $\beta_t(i;\mu)$ (defined by equations 45) for any $\mu > 0$ as follows:

$$\begin{aligned}
\tilde{\alpha}_1(i;\mu) &:= \alpha_1(i)/c_1(\mu) = \tilde{\alpha}_1(i), \quad \text{where } c_1(\mu) := c_1 \text{ for all } \mu; \\
\tilde{\alpha}_t(i;\mu) &:= \frac{\left[\sum_{j=1}^K \left(\tilde{\alpha}_{t-1}(j;\mu)p_{ji}\right)^\mu\right]^{\frac{1}{\mu}} f_i(x_t)}{c_t(\mu)}, \quad t = 2,3,\ldots,T; \\
\tilde{\beta}_T(i;\mu) &:= \beta_T(i) = 1; \\
\tilde{\beta}_t(i;\mu) &:= \frac{\left[\sum_{j=1}^K \left(p_{ij}f_j(x_{t+1})\tilde{\beta}_{t+1}(j;\mu)\right)^\mu\right]^{\frac{1}{\mu}}}{c_{t+1}(\mu)}, \quad t = T-1, T-2, \ldots, 1,
\end{aligned} \tag{52}$$

where

$$c_t(\mu) := \sum_{s=1}^K \left[\sum_{j=1}^K \left(\tilde{\alpha}_{t-1}(j;\mu)p_{js}\right)^\mu\right]^{\frac{1}{\mu}} f_s(x_t), \quad t = 2,3,\ldots,T.$$

Thus, $c_t(1) = c_t$ for all $t = 1,2,\ldots,T$. Also note that, using induction on t and (47), $\lim_{\mu\to 1} c_t(\mu) = c_t(1)$, and the limits $c_t(\infty) := \lim_{\mu\to\infty} c_t(\mu)$ exist and are finite for all $t = 1,2,\ldots,T$.

Proposition 10 *For any $i \in S$, we have*

1) $\tilde{\alpha}_t(i;\mu) = \frac{\alpha_t(i;\mu)}{\sum_{s=1}^K \alpha_t(s;\mu)} = \frac{\alpha_t(i;\mu)}{\prod_{m=1}^t c_m(\mu)}$ *for all* $t = 1,2,\ldots,T$, *and* $\tilde{\beta}_t(i;\mu) = \frac{\beta_t(i;\mu)}{\prod_{m=t+1}^T c_m(\mu)}$ *for all* $t = 1,2,\ldots,T-1$ *and for all* $\mu > 0$*;*

2) $\lim_{\mu\to 1}\tilde{\alpha}_t(i;\mu) = \tilde{\alpha}_t(i)$, $\lim_{\mu\to 1}\tilde{\beta}_t(i;\mu) = \tilde{\beta}_t(i)$ *for all* $t = 1,2,\ldots,T$*;*

3) $\lim_{\mu\to\infty}\tilde{\alpha}_t(i;\mu) = \tilde{\alpha}_t(i;\infty) := \frac{\alpha_t(i;\infty)}{\sum_{s=1}^K \alpha_t(s,\infty)}$, *for all* $t = 1,2,\ldots,T$, *and* $\lim_{\mu\to\infty}\tilde{\beta}_t(i;\mu) =: \tilde{\beta}_t(i;\infty) = \frac{\beta_t(i;\infty)}{\prod_{m=t+1}^T c_m(\infty)}$, *for all* $t = 1,2,\ldots,T-1$, *and, finally,* $\lim_{\mu\to\infty}\tilde{\beta}_T(i;\mu) =: \tilde{\beta}_T(i;\infty) = 1$ *trivially;*

4) *The hybrid decoder* (49) *based on the transformations* (45) *and the hybrid decoder* (49) *based on the transformations* (52) *are one and the same decoder, provided that both use the same tie-breaking rule.*

Proof The first claim concerning the $\tilde{\alpha}_t$ is trivially true for $t = 1$ by definition of $\alpha_1(i;\mu)$, that is (45). Now, using induction on t, assume that the claim is true for $t-1$. Write $a_{t-1}(\mu)$ for $(\sum_{s=1}^K \alpha_{t-1}(s;\mu))^{-1}$ so that $a_{t-1}(\mu)\alpha_{t-1}(j;\mu) = \tilde{\alpha}_{t-1}(j;\mu)$ and $a_{t-1}(\mu) = (\prod_{m=1}^{t-1} c_m(\mu))^{-1}$. Then, using (52), we get

$$\tilde{\alpha}_t(i;\mu) = \frac{\left(\sum_{j=1}^K \left(a_{t-1}(\mu)\alpha_{t-1}(j;\mu)p_{ji}\right)^\mu\right)^{\frac{1}{\mu}} f_i(x_t)}{\sum_{s=1}^K \left(\sum_{j=1}^K \left(a_{t-1}(\mu)\alpha_{t-1}(j;\mu)p_{js}\right)^\mu\right)^{\frac{1}{\mu}} f_s(x_t)},$$

which, upon cancellation of the $a_{t-1}(\mu)$, yields the required result

$$\frac{\left(\sum_{j=1}^K (\alpha_{t-1}(j;\mu)p_{ji})^\mu\right)^{\frac{1}{\mu}} f_i(x_t)}{\sum_{s=1}^K \left(\sum_{j=1}^K (\alpha_{t-1}(j;\mu)p_{js})^\mu\right)^{\frac{1}{\mu}} f_s(x_t)} = \frac{\alpha_t(i;\mu)}{\sum_{s=1}^K \alpha_t(s;\mu)}.$$

To see that $\tilde{\alpha}_t(i;\mu)$ also equals $\frac{\alpha_t(i;\mu)}{\prod_{m=1}^t c_m(\mu)}$, write

$$\tilde{\alpha}_t(i;\mu) = \frac{\left(\sum_{j=1}^K (a_{t-1}(\mu)\alpha_{t-1}(j;\mu)p_{ji})^\mu\right)^{\frac{1}{\mu}} f_i(x_t)}{c_t(\mu)} = \frac{\left(\sum_{j=1}^K (\alpha_{t-1}(j;\mu)p_{ji})^\mu\right)^{\frac{1}{\mu}} f_i(x_t)}{(\prod_{m=1}^{t-1} c_m(\mu))c_t(\mu)},$$

which, recalling the original (unscaled) $\alpha_t(i;\mu)$ recursion, yields the result.

The β variables are handled analogously.

The second claim is then a straightforward consequence of the first claim and the continuity (with respect to μ, and in particular at $\mu = 1$) of the power transform; for example, to establish the result for the $\tilde{\beta}_t(i;\mu)$, observe that $\prod_{m=t+1}^T c_m(\mu) \to \prod_{m=t+1}^T c_m(1)$ when $\mu \to 1$. The third claim also immediately follows from the first one and Proposition 9, also noticing that $\prod_{m=t+1}^T c_m(\mu) \to \prod_{m=t+1}^T c_m(\infty)$ as $\mu \to \infty$. The fourth claim also immediately follows from the first claim as v_t maximizes $\alpha_t(i;\mu)\beta_t(i;\mu)$ if and only if it maximizes $\tilde{\alpha}_t(i;\mu)\tilde{\beta}_t(i;\mu)$. ■

In particular, we arrive at the following characterization of the Viterbi paths $\widehat{y^T}(\infty)$, which is now possible to compute in practice for a wide range of models and parameters in contrast to the condition (48):

Corollary 11 *For any $t = 1, 2, \ldots, T$, $\hat{y}_t(\infty) = \arg\max_{i\in S}\{\tilde{\alpha}_t(i;\infty)\tilde{\beta}_t(i;\infty)\}$.*

Recall (46), and thus note that the PMAP decoder also maximizes $\tilde{\alpha}_t(i;1)\tilde{\beta}_t(i;1)$. As a side note, consider also the following decoder $v(x^T;0)$ that extrapolates the normalized power-transformed decoder to $\mu \to 0$, that is "beyond" the PMAP decoding. Namely, for any $t = 1, 2, \ldots, T$, let $v_t = \arg\max_{i\in S}\{\tilde{\alpha}_t(i;0)\tilde{\beta}_t(i;0)\}$, where for any $i \in S$,

$$\begin{aligned}
\tilde{\alpha}_1(i;0) &:= \alpha_1(i)/c_1 = \tilde{\alpha}_1(i); \qquad (53)\\
\tilde{\alpha}_t(i;0) &:= \frac{\left[\prod_{j\in S_t(i)} \tilde{\alpha}_{t-1}(j;0)p_{ji}\right]^{\frac{1}{K_t(i)}} f_i(x_t)}{\sum_{s=1}^K \left[\prod_{j\in S_t(s)} \tilde{\alpha}_{t-1}(j;0)p_{js}\right]^{\frac{1}{K_t(s)}} f_s(x_t)}, \quad t = 2, 3, \ldots, T,\\
\text{where } S_t(i) &:= \{j \in S :\ \tilde{\alpha}_{t-1}(j;0)p_{ji} > 0\} \text{ and } K_t(i) := |S_t(i)|, \text{ that is size of } S_t(i);\\
\tilde{\beta}_T(i;0) &:= \beta_T(i) = 1;\\
\tilde{\beta}_t(i;0) &:= \frac{\left[\prod_{j\in S_t^*(i)} p_{ij} f_j(x_{t+1})\tilde{\beta}_{t+1}(j;0)\right]^{\frac{1}{K_t^*(i)}}}{\sum_{s=1}^K \left[\prod_{j\in S_{t+1}(s)} \tilde{\alpha}_t(j;0)p_{js}\right]^{\frac{1}{K_{t+1}(s)}} f_s(x_{t+1})}, \quad t = T-1, T-2, \ldots, 1,\\
\text{where } S_t^*(i) &:= \{j \in S :\ p_{ij} f_j(x_{t+1})\tilde{\beta}_{t+1}(j;0) > 0\} \text{ and } K_t^*(i) := |S_t^*(i)|.
\end{aligned}$$

Corollary 12 *Assume that* $\lim_{\mu\to 0}\tilde{\alpha}_t(i;\mu) > 0$ *and* $\lim_{\mu\to 0}\tilde{\beta}_t(i;\mu) > 0$ *for all* $i \in S$ *and all* $t = 1, 2, \ldots, T$. *Then* $\tilde{\alpha}_t(i;0) = \lim_{\mu\to 0}\tilde{\alpha}_t(i;\mu)$ *and* $\lim_{\mu\to 0}\tilde{\beta}_t(i;\mu) = \tilde{\beta}_t(i;0)$ *for all* $i \in S$ *and all* $t = 1, 2, \ldots, T$, *that is the decoder* (49) *based on the transformations* (52) *converges (upto the tie-breaking rule) to the decoder defined by* (53) *above.*

Proof This is a straightforward exercise in calculus, that is, using continuity of the exponential function and invoking Proposition 1a of Brushe et al. (1998), with the positivity assumption making all $K_t(i)$ and $K_t^*(i)$ equal to K. ■

Note also that the hybrid decoder (49) based on the original, that is, unnormalized variables (45), generally does not have a limit as $\mu \to 0$.

6.3 Rescaling of the Forward and Backward Variables $\alpha(\cdot;\mu)$ and $\beta(\cdot;\mu)$ Defined by (51) Alters the Hybrid Decoder (49).

In the same manner as in (52) above, we now normalize the $\alpha(\cdot;\mu)$ and $\beta(\cdot;\mu)$ variables transformed according to (51). Thus, for any $\mu > 0$ and for any $i \in S$, let

$$
\begin{aligned}
\check{\alpha}_1(i;\mu) &:= \alpha_1(i)/\sum_{s=1}^{K}\alpha_1(s) = \tilde{\alpha}_1(i); \qquad (54)\\
\check{\alpha}_t(i;\mu) &:= \frac{\log\left[\frac{1}{K}\sum_{j=1}^{K} e^{\mu\check{\alpha}_{t-1}(j;\mu)p_{ji}}\right] f_i(x_t)}{\sum_{s=1}^{K}\log\left[\frac{1}{K}\sum_{j=1}^{K} e^{\mu\check{\alpha}_{t-1}(j;\mu)p_{js}}\right] f_s(x_t)}, \quad t = 2, 3, \ldots, T;\\
\check{\beta}_T(i;\mu) &:= \beta_T(i) = 1, \quad t = T-1, T-2, \ldots, 1;\\
\check{\beta}_t(i;\mu) &:= \frac{\log\left[\frac{1}{K}\sum_{j=1}^{K} e^{\mu p_{ij} f_j(x_{t+1})\check{\beta}_{t+1}(j;\mu)}\right]}{\sum_{s=1}^{K}\log\left[\frac{1}{K}\sum_{j=1}^{K} e^{\mu\check{\alpha}_t(j;\mu)p_{js}}\right] f_s(x_{t+1})}, \quad t = T-1, T-2, \ldots, 1.
\end{aligned}
$$

Proposition 13 *For any* $i \in S$, *we have*

1) $\lim_{\mu\to 0}\check{\alpha}_t(i;\mu) = \tilde{\alpha}_t(i)$, $\lim_{\mu\to 0}\check{\beta}_t(i;\mu) = \tilde{\beta}_t(i)$ *for all* $t = 1, 2, \ldots, T$;

2) $\lim_{\mu\to\infty}\check{\alpha}_t(i;\mu) = \tilde{\alpha}_t(i;\infty)$ *and* $\lim_{\mu\to\infty}\check{\beta}_t(i;\mu) = \tilde{\beta}_t(i;\infty)$, *for all* $t = 1, 2, \ldots, T$.

3) *The hybrid decoder* (49) *based on the transformations* (51) *and the hybrid decoder* (49) *based on the transformations* (54) *are generally different, even if both use the same tie-breaking rule.*

Proof The first two claims are straightforward extensions of Lemmas 1 and 2 of Brushe et al. (1998). To see this, first restore the previously reduced factor $\frac{1+(K-1)e^{-\mu}}{\mu}$ in both the numerator and denominator of the expressions for $\check{\alpha}_t(i;\mu)$ and $\check{\beta}_t(i;\mu)$. Then apply induction on t (first in the forward manner for the α variables and then backward for the β variables). For example, assume that $\lim_{\mu\to\infty}\check{\beta}_{t+1}(i;\mu) = \tilde{\beta}_{t+1}(i;\infty)$. Then, as $\mu \to \infty$,

$$
\frac{1+(K-1)e^{-\mu}}{\mu}\log\left[\frac{1}{K}\sum_{j=1}^{K} e^{\mu p_{ij} f_j(x_{t+1})\check{\beta}_{t+1}(j;\mu)}\right] \;\to\; \max_{j\in S}\left(p_{ij} f_j(x_{t+1})\tilde{\beta}_{t+1}(j;\infty)\right),
$$

which is, according to claim 3 of Proposition 10,

$$\max_{j\in S}\left(p_{ij}f_j(x_{t+1})\beta_{t+1}(j;\infty)/\prod_{m=t+2}^{T}c_m(\infty)\right) \;=\; \max_{j\in S}\left(p_{ij}f_j(x_{t+1})\beta_{t+1}(j;\infty)\right)/\prod_{m=t+2}^{T}c_m(\infty).$$

Next, recalling (47), we get that the numerator in the expression for $\lim_{\mu\to\infty}\check{\beta}_t(i;\mu)$ is given by $\beta_t(i;\infty)/\prod_{m=t+2}^{T}c_m(\infty)$. Observing that the denominator is given by

$$\lim_{\mu\to\infty}\frac{1+(K-1)e^{-\mu}}{\mu}\sum_{s=1}^{K}\log\left[\frac{1}{K}\sum_{j=1}^{K}e^{\mu\check{\alpha}_t(j;\mu)p_{js}}\right]f_s(x_{t+1}) \;=\; \sum_{s=1}^{K}\max_{j\in S}\left(\tilde{\alpha}_t(j;\infty)p_{js}\right)f_s(x_{t+1}),$$

which is just $c_{t+1}(\infty)$, finally gives $\lim_{\mu\to\infty}\check{\beta}_t(i;\mu)=\beta_t(i;\infty)/\prod_{m=t+1}^{T}c_m(\infty)=\tilde{\beta}_t(i;\infty)$, as required.

As a counter-example proving the last claim, consider the simple HMM from The MathWorks, Inc. (2012, p. 1840).

Example 2 *Let $S=\{1,2\}$ and let $\{1,2,\ldots,6\}$ be the emission alphabet. Let the initial distribution π, transition probability matrix $\mathbb{P}$, and the emission distributions f_s, $s\in S$, be defined as follows:*

$$\pi=\begin{pmatrix}2/3\\1/3\end{pmatrix},\quad \mathbb{P}=\begin{pmatrix}0.95 & 0.05\\0.1 & 0.9\end{pmatrix},\ \pi^t\mathbb{P}=\pi^t,\qquad \begin{array}{lcccccc} & 1 & 2 & 3 & 4 & 5 & 6\\ f_1(\cdot) & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6\\ f_2(\cdot) & 0.1 & 0.1 & 0.1 & 0.1 & 0.1 & 0.2\end{array}.$$

Suppose $x^5=(2,6,6,4,1)$ has been observed. Take $\mu=7$. Table 2 shows outputs of the original (top) and normalized (bottom) transformed decoders, respectively. Clearly, the decoders return different paths.

■

Note that unlike the normalized hybrid decoder based on the power-transform, this normalized hybrid decoder generally does not satisfy the first claim of Proposition 10. (Indeed, satisfying these conditions would contradict the third claim of the latter Proposition 13.)

We have also experimented with these normalized hybrid decoders using a subset of real data (and a realistic HMM with $K=6$ states) from our experimental Section 5 and can indeed confirm convergence of the hybrid decoder based (54) to the PMAP decoder with $\mu=0.001$ and to the Viterbi decoder with $\mu=10000$ for sequences of length $T=100$. Naturally, the above range of μ values would generally need to increase significantly with T.

Below, we summarize our views on the idea of purely algorithmic hybridization of MAP and PMAP.

1. The method presented by Brushe et al. (1998) need not work, that is, can fail to converge to the Viterbi path, when the Viterbi path is not unique, see Example 1 above.

2. Since the method depends on the transformation used, more work may be needed to understand which (if any) particular transformation/interpolation could be suitable for a specific application; the choice of (51) made by Brushe et al. (1998) seems to be rather arbitrary.

t	$\alpha_t(1;\mu)$	$\beta_t(1;\mu)$	$\alpha_t(2;\mu)$	$\beta_t(2;\mu)$	$\alpha_t(1;\mu)\beta_t(1;\mu)$	$\alpha_t(2;\mu)\beta_t(2;\mu)$
					10^{-6}	10^{-6}
1	0.11111	6.6968e-05	0.033333	0.00019826	**7.4409**	6.6088
2	0.010576	0.00071029	0.0091583	0.00085352	7.5121	**7.8168**
3	0.0009266	0.0083987	0.0022209	0.003471	**7.7823**	7.7088
4	9.201e-05	0.10141	0.00010268	0.058041	**9.3311**	5.9598
5	8.1481e-06	1	4.8559e-06	1	**8.1481**	4.8559

t	$\check{\alpha}_t(1;\mu)$	$\check{\beta}_t(1;\mu)$	$\check{\alpha}_t(2;\mu)$	$\check{\beta}_t(2;\mu)$	$\check{\alpha}_t(1;\mu)\check{\beta}_t(1;\mu)$	$\check{\alpha}_t(2;\mu)\check{\beta}_t(2;\mu)$
1	0.76923	0.30879	0.23077	0.97296	**0.23753**	0.22453
2	0.58963	0.55137	0.41037	0.55227	**0.32510**	0.22664
3	0.35383	1.15172	0.64617	0.39942	**0.40751**	0.25809
4	0.46886	1.03712	0.53114	0.59356	**0.48626**	0.31526
5	0.60611	1	0.39389	1	**0.60611**	0.39389

Table 2: $\mu = 7$. Top: Output from the original (unnormalized) transformed decoder based on the transformations (51); the optimal path is $(1, 2, 1, 1, 1)$. Bottom: Output from the normalized transformed decoder based on the transformations (54); the optimal path is $(1, 1, 1, 1, 1)$.

3. Also, the choice of (51) does not work in practice except with trivially short sequences; the underlying transformations can be normalized but this alters the decoder (Proposition 13). The choice of (45) is better in several aspects, mainly for its rescaling property (subsection 6.2), that is, the decoder is indeed ready to work in practice.

4. Algorithmically defined estimators are notoriously hard to analyze analytically (Winkler, 2003, pp. 25, 129-131). Indeed, it is not clear if the general members of the above interpolating families (regardless of the transformation used) satisfy any explicit optimality criteria; this makes it difficult to interpret such decoders. This may also discourage the use of such decoders in more complex inference cycles (that is, when any genuine model parameters are to be estimated as well, for example, as in Viterbi Training Koski, 2001; Lember and Koloydenko, 2008, 2010).

5. The point-wise hybridization scheme (49) can itself be altered. Indeed, other recursion schemes (see, for example, Koski, 2001, pp. 272-273 for Derin's formula) can also be applied for this purpose. However, now more than a decade after Brushe et al. (1998), we are not aware of any practical application of the idea of algorithmic hybridization of the MAP-PMAP inferences. Besides the plausible reasons already discussed in Subsection 1.2.1 (that actually extend to any type of MAP-PMAP hybridization), it is plausible that this particular type of hybridization has not yet seen application because of the lack of interpretation of its solutions, and possibly also because of the aforementioned difficulties with implementation of the original idea of Brushe et al. (1998).[3]

3. We recently attempted to contact the authors of that paper, but have not received any response by the time of sending this manuscript to the production editor.

Appendix D gives a pseudo-code to compute a decoded sequence $\widehat{y^T}(\mu)$ for any $\mu > 0$ using the power-transform approach (49) with scaling. Naturally, the decoding process can be parallelized over a range of μ values.

7. Asymptotic Risks

Given an arbitrary decoder g and a risk function R, the quantity $R(g(x^T) \mid x^T)$ evaluates the risk when g is applied to a given sequence x^T. Below we will write $R(x^T)$ for the minimum risk $\min_{s^T} R(s^T \mid x^T)$ which is achieved by the Bayes decoder v: $R(v(x^T) \mid x^T) = R(x^T)$. Besides $R(X^T)$, we are also interested in the random variables $R(g(X^T) \mid X^T)$ (depending on R and g). Thus, Kuljus and Lember (2012) have considered convergence of various risks of the Viterbi decoder $v(\cdot;\infty)$. Since Viterbi paths $v(x^T;\infty)$ and $v(x^{T+1};\infty)$ may differ significantly, asymptotic analysis of the Viterbi decoding is far from being trivial. Koloydenko and Lember (2008); Lember and Koloydenko (2008, 2010) constructed a well-defined process $v(X^\infty;\infty)$, named also after Viterbi, that for a wide class of HMMs extends *ad infinitum* finite Viterbi paths $v(x^T;\infty)$ and possesses useful ergodic properties. Based on the asymptotic theory of Viterbi processes $v(X^\infty;\infty)$, Kuljus and Lember (2012) have shown that under fairly general assumptions on the HMM, the random variables $R_k(v(X^T;\infty) \mid X^T)$, $\bar{R}_k(v(X^T;\infty) \mid X^T)$, where $k = 1, 2, \ldots$, and $\bar{R}_\infty(v(X^T;\infty) \mid X^T)$, as well as $\bar{R}_\infty(v(X^T;\infty))$ (see Equation 12), $\bar{R}_1(v(X^T;\infty))$ (see Equation 20), and $R_1(v(X^T;\infty))$ (see Equation 27) all converge (as $T \to \infty$) *a.s.* to constant (that is non-random) limits. Convergence of these risks implies *a.s.* convergence of

$$C_1\bar{R}_1(v(X^T;\infty) \mid X^T) + C_2\bar{R}_\infty(v(X^T;\infty) \mid X^T) + C_3\bar{R}_1(v(X^T;\infty)) + C_4\bar{R}_\infty(v(X^T;\infty)),$$

and

$$C_1R_1(v(X^T;\infty) \mid X^T) + C_2\bar{R}_\infty(v(X^T;\infty) \mid X^T) + C_3R_1(v(X^T;\infty)) + C_4\bar{R}_\infty(v(X^T;\infty)),$$

the risks appearing in the generalized problems (18) and (26), respectively. Actually, convergence of $\bar{R}_\infty(v(X^T;\infty), X^T)$ is also proved (and used in the proof of convergence of $\bar{R}_\infty(v(X^T;\infty) \mid X^T)$). Hence, the minimized risk in (19), evaluated at the Viterbi paths, converges as well.

The limits—*asymptotic risks*—are (deterministic) constants that depend only on the model, and help us assess the Viterbi inference in the following principled way. For example, let $R_1(k = \infty)$ be the limit (as $T \to \infty$) of $R_1(v(X^T;\infty) \mid X^T)$, which is the asymptotic misclassification rate of the Viterbi decoding. Thus, for large T, the Viterbi decoding makes about $TR_1(k = \infty)$ misclassification errors. The asymptotic risks might be, in principle, found theoretically, but in reality this can be rather difficult. However, since all these asymptotic results also hold in the L_1 sense, which implies convergences of expectations, the limiting risks can be estimated by simulations.

Lember (2011a,b) has also shown that under the same assumptions $R_1(X^T) = R_1(v(X^T;1) \mid X^T)$ converges to a constant limit, say R_1. Kuljus and Lember (2012) have at the same time also shown $\bar{R}_1(X^T) = \bar{R}_1(v(X^T;1) \mid X^T)$ to converge. Clearly $R_1(k = \infty) \geq R_1(1)$, and even if their difference is small, the total number of errors made by the Viterbi decoder in excess of PMAP in the long run can still be significant.

Presently, we are not aware of a universal method for proving (or improving upon) the limit theorems for these risks. Recall that convergence of the risks of the Viterbi decoding is possible due to the existence of the Viterbi process which has nice ergodic properties. The question whether infinite PMAP processes have similar properties, is still open. Therefore, convergence of $R_1(X^T)$ was proven with a completely different method based on the smoothing probabilities. In fact, all of the limit theorems obtained thus far have been proven with different methods. We conjecture that these different methods can be combined so that convergence of the minimized combined risk (18) or (26) could be proven as well. In summary, as mentioned before, convergence of the minimized combined risks has thus far been obtained for trivial combinations only, that is with three of the four constants being zero. Note that while convergence of the intermediate case (38) with its minimizer $v(x^T; k(\alpha))$ is an open question, (39) gives

$$0 \leq \bar{R}_\infty(v(x^T; k(\alpha)) \mid x^T) - \bar{R}_\infty(v(x^T; \infty) \mid x^T) \leq \frac{\bar{R}_1(v(x^T; \infty) \mid x^T)}{k-1}.$$

This, together with the *a.s.* convergence of $\bar{R}_1(v(X^T; \infty) \mid X^T)$, implies that in the long run, for most sequences x^T, $\bar{R}_\infty(v(x^T; k) \mid x^T)$ will not exceed $\bar{R}_\infty(v(x^T; \infty) \mid x^T)$ by more than $\frac{1}{k-1} \lim_{T\to\infty} \bar{R}_1(v(X^T; \infty) \mid X^T)$. Since this limit is finite, letting k increase with T, we get that $\bar{R}_\infty(v(X^T; k_T))$ approach $\lim_{T\to\infty} \bar{R}_\infty(v(X^T; \infty))$ *a.s.*, that is, as the intuition predicts, the likelihood of $v(X^T; k_T)$ approaches that of $v(X^T; \infty)$.

Finally, Lember and Koloydenko (2010); Lember et al. (2011) also outline possible applications of the above asymptotic risk theory. For example, if a certain number of the true labels $y_1, y_2, \ldots, y_T$ can be revealed (say, at some cost), the remaining labels would be computed by a constrained decoder, for example, the constrained Viterbi decoder. Having observed x^T, the user then needs to decide which positions are "most informative" and then acquires their labels. Assuming further that the HMM is stationary, the R_1-like risks $P(v(X^\infty; \infty)_t \neq Y_t \mid X_{t-m}^{t+m} \in A)$ (for any $m \geq 1$ and any measurable set $A \in \mathcal{X}^{2m+1}$), are independent of t (for $t = m+1, m+2, \ldots$), and could therefore be used in the above active learning protocol for the selection of the most informative positions. Specifically, if A is such that $P(v(X^\infty; \infty)_t \neq Y_t \mid X_{t-m}^{t+m} \in A)$ is high, then acquire labels at positions t of occurrence of A. Naturally, there are different ways to make this concrete. For one simple example, suppose only a batch of L labels can be acquired. Assuming $\mathcal{X}$ to be discrete, order all the $\mathcal{X}$ words A of length q (that is, $A \in \mathcal{X}^q$) by $P(v(X^\infty; \infty)_t \neq Y_t \mid X_{t-m}^{t+m} \in A)$. Finally, from the $\mathcal{X}$ of length q that occur in x^T, choose L with the highest $P(v(X^\infty; \infty)_t \neq Y_t \mid X_{t-m}^{t+m} \in A)$. The above asymptotic theory is crucial also for establishing $P(v(X^\infty; \infty)_t \neq Y_t \mid X_{t-m}^{t+m} \in A)$ as the *a.s.* limit of easily computable (e.g., via off-line simulations) empirical measures. In practice, these latter measures would be used as estimates of $P(v(X^\infty; \infty)_t \neq Y_t \mid X_{t-m}^{t+m} \in A)$ and first experiments along these lines are given by Lember et al. (2011, Section 4.4). It may also be of interest to test these ideas with other risks and decoders, such as members of the generalized hybrid families presented here.

8. Discussion

The point-wise symmetric zero-one loss $l(s,y) = \mathbb{I}_{\{s\neq y\}}$ in (4), (5), and consequently in the generalized PMAP hybrid decoding (26), can be easily replaced by a general loss $l(s,y) \geq 0$, $s, y \in S$. In computational terms, this would require multiplying the loss matrix $(l(s,y))_{s,y\in S}$ by the (prior or) posterior probability vectors $(p_t(1 \mid x^T), p_t(2 \mid x^T), \ldots, p_t(K \mid x^T))'$ to obtain the (prior or) posterior risk $(\rho_t(1 \mid x^T), \rho_t(2 \mid x^T), \ldots, \rho_t(K \mid x^T))'$ vectors (we use the apostrophe to denote vector transpose). The dynamic programming algorithm defined by (23) with (28) still stands provided $p_t(j \mid x^T)$ (or $p_t(j)$, or both) is replaced by $1 - \rho_t(j \mid x^T)$ (or $1 - \rho_t(j)$, or both respectively) in the definition of $\gamma_t(j)$. If all confusions of state y are equally undesirable, that is, $l(s,y)$ is of the form $l(y) \times \mathbb{I}_{\{s\neq y\}}$, then the above adjustment reduces to replacing $p_t(j \mid x^T)$ by $l(j)p_t(j \mid x^T)$ (for all $j \in S$), which we illustrated in Figure 2 when suppressing state 3. Similar adjustments can be made to the $\bar{R}_1$ risks of the generalized PVD family, which was also illustrated in Figure 2.

Using an asymmetric loss could be particularly valuable in practice when, for example, detection of a rare state or transition needs to be encouraged. Similar views have been most recently expressed also by Yau and Holmes (2010), who, staying within the additive risk framework, have proposed a general asymmetric form of the loss (30) with $k = 2$. Hybridizing this general asymmetric pairwise loss with the other losses considered in this work should provide additional flexibility to path inference. A way to incorporate this loss into our generalized framework is by vectorizing the chain $\{Y_t\}_{t\geq 1}$ as $\{(Y_t, Y_{t+1})\}_{t\geq 1}$ and then following the opening lines of this Section.

Also, using a range of perturbed versions of a loss function can help assess saliency of particular detections ("islands"). In fact, at the stage of data exploration one may more generally want to use a collection of outputs produced by using a range of different loss functions instead of a single one.

The logarithmic risks (3), (6), (12), (20) on the one hand, and the ordinary risks (2), (5), $R_\infty(s^T) = 1 - p(s^T)$, (27), on the other hand, can be respectively combined into a single parameter family of risks by using, for example, the power transformation as shown below with p for the moment standing for any probability distribution on S^T:

$$\begin{aligned} R_1(s^T;\beta) &= \begin{cases} -\frac{1}{T}\sum_{t=1}^T \frac{p_t(s_t)^\beta - 1}{\beta}, & \text{if } \beta \neq 0; \\ -\frac{1}{T}\sum_{t=1}^T \log p_t(s_t), & \text{if } \beta = 0; \end{cases} \\ R_\infty(s^T;\beta) &= \begin{cases} -\frac{1}{T}\frac{p(s^T)^\beta - 1}{\beta}, & \text{if } \beta \neq 0; \\ -\frac{1}{T}\log p(s^T), & \text{if } \beta = 0. \end{cases} \end{aligned} \tag{55}$$

Thus, the family of risk minimization problems given in (56) below

$$\min_{s^T} \Big[C_1 R_1(s^T \mid x^T;\beta_1) + C_2 R_\infty(s^T \mid x^T;\beta_2) + C_3 R_1(s^T;\beta_3) + C_4 R_\infty(s^T;\beta_4) \Big], \tag{56}$$

$C_i \geq 0$ and $\sum_{i=1}^4 C_i > 0$ unifies and generalizes problem (18) ($\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$) and problem (26) ($\beta_1 = \beta_3 = 1$, $\beta_2 = \beta_4 = 0$). Clearly, the dynamic programming approach of Theorem 4 immediately applies to any member of the above family (56) with $\beta_2 = \beta_4 = 0$. Also, computations of multiple decoders from this family (at least with $\beta_2 = \beta_4 = 0$) are readily parallelizable.

Next, Theorem 6 and Corollaries 7 and 8 obviously generalize to higher order Markov chains as can be seen from the following Proposition.

Proposition 14 *Let p represent a Markov chain of order m, $1 \leq m \leq T$, on S^T. Then for any $s^T \in S^T$ and for any $k \in \{m, m+1, \ldots\}$, we have*

$$\bar{R}_k(s^T) = \bar{R}_m(s^T) + (k-m)\bar{R}_\infty(s^T).$$

Proof This is a straightforward extension of the proof of Theorem 6. ■

The present risk-based discussion of HMM path inference also naturally extends to the problem of optimal *labeling* or *annotation* (already mentioned in Subsection 1.2). Namely, the state space S can be partitioned into subsets S_1, S_2, ..., S_Λ, for some $\Lambda \leq K$, in which case $\lambda(s)$ assigns label λ to every state $s \in S_\lambda$. The fact that the PMAP problem is as easily solved over the label space Λ^T as it is over S^T has already been used in practice. Indeed, Käll et al. (2005), who also add the constraint of admissibility with respect to the prior distribution, in effect average $p_t(s_t \mid x^T)$'s, for each t, within the label classes and then use recursions (15) to obtain the *optimal accuracy labeling* of *a priori* admissible state paths. This clearly corresponds to using the point loss $l(s, s') = \mathbb{I}_{\{\lambda(s) \neq \lambda(s')\}}$ in (4) when solving $\min_{s^T : p(s^T) > 0} R_1(s^T \mid x^T)$ (14). With our definition of admissibility (that is, positivity of the posterior path probability), the same approach (that is, replacing $p_t(s_t \mid x^T)$'s by their within class average $\bar{p}_t(s_t \mid x^T)$) extends to solve $\min_{s^T : p(s^T|x^T) > 0} R_1(s^T \mid x^T)$ (7) under the same loss $l(s, s') = \mathbb{I}_{\{\lambda(s) \neq \lambda(s')\}}$. Clearly, the generalized problem (56) also immediately incorporates the above pointwise label-level loss in either the prior $R_1(\cdot; \beta_3)$ or posterior risk $R_1(\cdot; \beta_1)$, or both. Since computationally these problems are essentially as light as recursion (24), (25), and since Käll et al. (2005) report their special case to be successful in practice, we believe that the above generalizations offer yet more possibilities that are potentially useful in practice.

Instead of using the same arithmetic averages $\bar{p}_t(s_t \mid x^T)$'s (or $\bar{p}_t(s_t)$'s) for the R_1 risks in (56) regardless of β, we can gain additional flexibility by replacing $\bar{p}_t(s_t)^\beta$ and $\log \bar{p}_t(s_t)$ in (55) ($\beta \neq 0$ and $\beta = 0$ respectively) with

$$\bar{p}_t(s; \beta) \quad \propto \quad \begin{cases} \left(\dfrac{\sum\limits_{s' \in S_{\lambda(s)}} p_t(s')}{|S_{\lambda(s)}|} \right)^\beta, & \text{if } \beta \neq 0; \\ \left(\prod\limits_{s' \in S_{\lambda(s)}} p_t(s') \right)^{\frac{1}{|S_{\lambda(s)}|}}, & \text{if } \beta = 0. \end{cases}$$

Certainly, the choice of the basic loss functions, inflection parameters β_i and weights C_i of the respective risks is application dependent, and can be tuned with the help of labeled data, using, for example, cross-validation.

Finally, these generalizations are presented for the standard HMM setting, and therefore extensions to more complex and practically more useful HMM-based settings (e.g., semi-Markov, autoregressive, coupled, etc.) could also be interesting.

Since the transform based approach, especially the newly proposed power-transform hybridization, has also generated some interest, it would be interesting to evaluate performance of the power-transform hybrids together with the risk-based families on multiple real applications and using various domain specific performance measures.

Acknowledgments

The first author has been supported by the Estonian Science Foundation Grant nr. 9288 and by targeted financing project SF0180015s12, which has also supported a research visit of the second author to Tartu University. The second author has also been supported by UK NIHR Grant i4i II-AR-0209-10012. The authors are also grateful to anonymous reviewers as well as to the action editor for their thorough reviews of this work, additional references, and comments and suggestions on improving this manuscript. The authors are also very thankful to Dr Dario Gasbarra and Dr Kristi Kuljus for reviewing earlier versions of the manuscript and pointing out two subtle mistakes, as well as to Ufuk Mat for pointing out some typing errors.

Appendix A. An Example of an Inadmissible Path of Positive Prior Probability

$$\pi = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}/9, \qquad P = \begin{pmatrix} 5 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 4 & 0 & 5 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 2 & 1 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 3 & 3 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 3 & 3 & 0 & 0 & 0 & 3 & 0 \end{pmatrix}/9.$$

To simplify the verifications, consider an emission alphabet with only four symbols, although the idea of constructing this example readily extends to larger alphabets (in particular, to more practically relevant situations where the emission alphabet is larger than the hidden state space, or the emission distributions are continuous altogether). Then take the following emission distributions:

$$\begin{pmatrix} P_1 & P_2 & P_3 & P_4 \\ 1/25 & 1/20 & 0 & 91/100 \\ 0 & 0 & 1/5 & 4/5 \\ 1/20 & 1/25 & 0 & 91/100 \\ 0 & 0 & 1/5 & 4/5 \\ 1/10 & 0 & 1/5 & 7/10 \\ 0 & 0 & 1/5 & 4/5 \\ 1/15 & 1/15 & 0 & 13/15 \\ 0 & 0 & 1/5 & 4/5 \\ 1/15 & 1/15 & 0 & 13/15 \end{pmatrix}.$$

Suppose now that a sequence $x^3 = (1, 2, 3)$ has been observed. It can then be verified that the (unconstrained) PMAP decoder returns any of the following paths $(5, 1, 5)$, $(5, 3, 5)$, $(5, 7, 5)$, or $(5, 9, 5)$, all of which having zero prior (and posterior) probabilities.

When the decoder is subject to the positivity constraint on the prior probabilities, it would return any of the following paths $(5, 2, 5)$, $(5, 4, 5)$, $(5, 5, 5)$, $(5, 6, 5)$, $(5, 8, 5)$, which, despite being of positive prior probabilities, all have zero posterior probabilities.

Finally, if the decoder is constrained to produce paths of positive posterior probability, it would then return any of the following paths $(5, 7, 2)$, $(5, 7, 6)$, $(3, 3, 5)$, $(9, 3, 5)$.

Appendix B. Proof of Remark 3

Proof Assume $C_3 = C_4 = 0$. For each $C_1, C_2 > 0$, let $\widehat{y^T}_{C_1,C_2} \in S^T$ be a solution to (18), and let $\widehat{y^T}_{PVD}$ be the output of PVD. Thus, we have

$$C_1\bar{R}_1(\widehat{y^T}_{C_1,C_2} \mid x^T) + C_2\bar{R}_\infty(\widehat{y^T}_{C_1,C_2} \mid x^T) \quad \leq \quad C_1\bar{R}_1(\widehat{y^T}_{PVD} \mid x^T) + C_2\bar{R}_\infty(\widehat{y^T}_{PVD} \mid x^T).$$

Then

$$0 \leq C_1(\bar{R}_1(\widehat{y^T}_{C_1,C_2} \mid x^T) - \bar{R}_1(\widehat{y^T}_{PVD} \mid x^T)) \leq C_2(\bar{R}_\infty(\widehat{y^T}_{PVD} \mid x^T) - \bar{R}_\infty(\widehat{y^T}_{C_1,C_2} \mid x^T))$$

holds for any $C_1, C_2 > 0$. Since $\bar{R}_\infty(\widehat{y^T}_{PVD} \mid x^T) - \bar{R}_\infty(\widehat{y^T}_{C_1,C_2} \mid x^T)$ is clearly bounded (and S^T is finite), we obtain $\bar{R}_1(\widehat{y^T}_{C_1,C_2} \mid x^T) = \bar{R}_1(\widehat{y^T}_{PVD} \mid x^T)$ for some sufficiently small C_2. Since $C_2 > 0$, all $\widehat{y^T}_{C_1,C_2}$ are admissible (Remark 1 above), therefore for such sufficiently small C_2, $\widehat{y^T}_{C_1,C_2}$ is also a solution to the PVD Problem (9).

The second statement is proved similarly, recalling Proposition 2 to establish admissibility of $\widehat{y^T}_{C_1,C_4}$ *almost surely.* ■

Appendix C. Supplementary Results on the Trade-Off between $\bar{R}_1$ and $\bar{R}_\infty$ Risks in Problem (18), and between R_1 and $\bar{R}_\infty$ Risks in Problem (26).

Corollary 15 *1. Let $\hat{y}$ and $\hat{y}'$ be solutions to Problem (18) with $C_1 \in [0, 1]$ and $C_2 = 1 - C_1$, $C_3 = C_4 = 0$ and $C'_1 \in [0, 1]$ and $C'_2 = 1 - C'_1$, $C'_3 = C'_4 = 0$, respectively. Assume $C_1 \leq C'_1$. Then $\bar{R}_1(\hat{y} \mid x^T) \geq \bar{R}_1(\hat{y}' \mid x^T)$ and $\bar{R}_\infty(\hat{y} \mid x^T) \leq \bar{R}_\infty(\hat{y}' \mid x^T)$.*

2. Let $\hat{y}$ and $\hat{y}'$ be solutions to Problem (18) with $C_3 \in [0, 1]$ and $C_4 = 1-C_3$, $C_1 = C_2 = 0$ and $C'_3 \in [0, 1]$ and $C'_4 = 1 - C'_3$, $C'_1 = C'_2 = 0$, respectively. Assume $C_3 \leq C'_3$. Then $\bar{R}_1(\hat{y}) \geq \bar{R}_1(\hat{y}')$ and $\bar{R}_\infty(\hat{y}) \leq \bar{R}_\infty(\hat{y}')$.

3. Let $\hat{y}$ and $\hat{y}'$ be solutions to Problem (26) with $C_1 \in [0, 1]$ and $C_2 = 1-C_1$, $C_3 = C_4 = 0$ and $C'_1 \in [0, 1]$ and $C'_2 = 1 - C'_1$, $C'_3 = C'_4 = 0$, respectively. Assume $C_1 \leq C'_1$. Then $R_1(\hat{y} \mid x^T) \geq R_1(\hat{y}' \mid x^T)$ and $\bar{R}_\infty(\hat{y} \mid x^T) \leq \bar{R}_\infty(\hat{y}' \mid x^T)$.

4. Let $\hat{y}$ and $\hat{y}'$ be solutions to Problem (26) with $C_3 \in [0, 1]$ and $C_4 = 1-C_3$, $C_1 = C_2 = 0$ and $C'_3 \in [0, 1]$ and $C'_4 = 1-C'_3$, $C'_1 = C'_2 = 0$. Assume $C_3 \leq C'_3$. Then $R_1(\hat{y}) \geq R_1(\hat{y}')$ and $\bar{R}_\infty(\hat{y}) \leq \bar{R}_\infty(\hat{y}')$.

Proof A straightforward application of Lemma 16 given below. ■

Lemma 16 *Let F and G be functions from a set A to the extended reals $\bar{\mathbb{R}} = \mathbb{R} \cup \{\pm\infty\}$. Let $\alpha_1, \alpha_2 \in [0,1]$ be such that $\alpha_1 \le \alpha_2$. Suppose $a_1, a_2 \in A$ are such that*

$$\alpha_i F(a_i) + (1-\alpha_i)G(a_i) \quad \le \quad \alpha_i F(x) + (1-\alpha_i)G(x), \quad i = 1,2, \quad \textit{for all } x \in A.$$

Then $F(a_1) \ge F(a_2)$ and $G(a_1) \le G(a_2)$.

Although the result is obvious, below we state its proof for completeness.

Proof Write a, b, c, and d for $F(a_1)$, $G(a_1)$, $F(a_2)$, and $G(a_2)$, respectively. Then we have

$$\begin{aligned} \alpha_1(a-c) &\le (1-\alpha_1)(d-b), \\ \alpha_2(a-c) &\ge (1-\alpha_2)(d-b), \end{aligned}$$

and therefore

$$\begin{aligned} \alpha_2\alpha_1(a-c) &\le \alpha_2(1-\alpha_1)(d-b), \\ \alpha_1\alpha_2(a-c) &\ge \alpha_1(1-\alpha_2)(d-b), \end{aligned}$$

which gives $\alpha_1(1-\alpha_2)(d-b) \le \alpha_2(1-\alpha_1)(d-b)$. Since $\alpha_1(1-\alpha_2) \le \alpha_2(1-\alpha_1)$, it follows that $d \ge b$, that is, $G(a_2) \ge G(a_1)$. The fact that $F(a_1) \ge F(a_2)$ is obtained similarly. ■

Appendix D. Pseudo-Code for Computing the Hybrid Decoders (49) Using the Power-Transform with Scaling (52), (53).

Finally, to output the decoded sequence $\widehat{y^T}(\mu)$, a simple tie-breaking rule may be as follows:

for $t = 1, 2, \ldots, T$ **do**
  $\hat{y}_t(\mu) \leftarrow \min \arg\max\{\tilde{\alpha}_t(i;\mu)\tilde{\beta}_t(i;\mu)\}$,
end for

whereas more elaborate rules may involve ordering of the entire state space S^T, or simply outputting all of the winning sequences. (Computations of the transformed and scaled α and β variables are summarized in Algorithms 1 and 2 respectively.)

Algorithm 1 The forward pass to compute $\tilde{\alpha}_t(i;\mu)$ and the scaling constants $c_t(\mu)$.

for $t = 1, 2, \ldots, T$ **do**
  $c_t(\mu) \leftarrow 0$
end for
for $i = 1, 2, \ldots, K$ **do**
  $\alpha_1(i) \leftarrow \pi_i f_i(x_1)$
  $c_1(\mu) \leftarrow c_1(\mu) + \pi_i f_i(x_1)$
end for
for $i = 1, 2, \ldots, K$ **do**
  $\tilde{\alpha}_1(i;\mu) \leftarrow \alpha_1(i)/c_1(\mu)$
end for
if $\mu = 0$ **then**
  for $t = 2, \ldots, T$ **do**
    for $i = 1, 2, \ldots, K$ **do**
      $S_t(i) \leftarrow \{j \in S : \ \tilde{\alpha}_{t-1}(j;\mu)p_{ji} > 0\}$
      $K_t(i) \leftarrow |S_t(i)|$
      $\tilde{\alpha}_t(i;\mu) \leftarrow \left[\prod_{j \in S_t(i)} \tilde{\alpha}_{t-1}(j;\mu)p_{ji}\right]^{\frac{1}{K_t(i)}} f_i(x_t)$
      $c_t(\mu) \leftarrow c_t(\mu) + \tilde{\alpha}_t(i;\mu)$
    end for
    for $i = 1, 2, \ldots, K$ **do**
      $\tilde{\alpha}_t(i;\mu) \leftarrow \tilde{\alpha}_t(i;\mu)/c_t(\mu)$
    end for
  end for
else
  for $t = 2, \ldots, T$ **do**
    for $i = 1, 2, \ldots, K$ **do**
      $\tilde{\alpha}_t(i;\mu) \leftarrow \left[\sum_{j=1}^{K} (\tilde{\alpha}_{t-1}(j;\mu)p_{ji})^{\mu}\right]^{\frac{1}{\mu}} f_i(x_t)$
      $c_t(\mu) \leftarrow c_t(\mu) + \tilde{\alpha}_t(i;\mu)$
    end for
    for $i = 1, 2, \ldots, K$ **do**
      $\tilde{\alpha}_t(i;\mu) \leftarrow \tilde{\alpha}_t(i;\mu)/c_t(\mu)$
    end for
  end for
end if

Algorithm 2 The backward pass to compute $\tilde{\beta}_t(i;\mu)$.

for $i = 1, 2, \ldots, K$ **do**
  $\tilde{\beta}_T(i;\mu) \leftarrow 1$
end for
if $\mu = 0$ **then**
  for $t = T-1, T-2, \ldots, 1$ **do**
    for $i = 1, 2, \ldots, K$ **do**
      $S_t^*(i) \leftarrow \{j \in S : \ f_j(x_{t+1}) p_{ij} \tilde{\beta}_{t+1}(j;\mu) > 0\}$
      $K_t^*(i) \leftarrow |S_t^*(i)|$
      $$\tilde{\beta}_t(i;\mu) \leftarrow \left[\prod_{j \in S_t^*(i)} f_j(x_{t+1}) p_{ij} \tilde{\beta}_{t+1}(j;\mu) \right]^{\frac{1}{K_t^*(i)}} / c_{t+1}(\mu)$$
    end for
  end for
else
  for $t = T-1, T-2, \ldots, 1$ **do**
    for $i = 1, 2, \ldots, K$ **do**
      $$\tilde{\beta}_t(i;\mu) \leftarrow \left[\sum_{j=1}^{K} \left(f_j(x_{t+1}) p_{ij} \tilde{\beta}_{t+1}(j;\mu) \right)^{\mu} \right]^{\frac{1}{\mu}} / c_{t+1}(\mu)$$
    end for
  end for
end if

Appendix E. Further Details of the Experiments from Section 5

Below are the estimates of the HMM parameters obtained from the entire data set as described in Section 5.

$$\hat{\pi} = \begin{pmatrix} 0.0016 & 0.0041 & 0.9929 & 0.0014 & 0.0000 & 0.0000 \end{pmatrix},$$

$$\widehat{\mathbb{P}} = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{pmatrix} 0.8359 & 0.0034 & 0.1606 & 0 & 0 & 0 \\ 0.0022 & 0.8282 & 0.1668 & 0.0028 & 0 & 0 \\ 0.0175 & 0.0763 & 0.8607 & 0.0455 & 0 & 0 \\ 0 & 0 & 0 & 0.7500 & 0.2271 & 0.0229 \\ 0 & 0 & 0 & 0 & 0.8450 & 0.1550 \\ 0 & 0.0018 & 0.2481 & 0 & 0 & 0.7501 \end{pmatrix},$$

$$\hat{\pi}_{inv} = \begin{pmatrix} 0.0511 & 0.2029 & 0.4527 & 0.0847 & 0.1240 & 0.0847 \end{pmatrix},$$

	$\widehat{P_1}$	$\widehat{P_2}$	$\widehat{P_3}$	$\widehat{P_4}$	$\widehat{P_5}$	$\widehat{P_6}$
A	0.1059	0.0636	0.0643	0.1036	0.1230	0.1230
C	0.0107	0.0171	0.0135	0.0081	0.0111	0.0128
D	0.0538	0.0319	0.0775	0.0634	0.0415	0.0345
E	0.0973	0.0477	0.0620	0.1120	0.0852	0.0848
F	0.0436	0.0576	0.0330	0.0371	0.0386	0.0399
G	0.0303	0.0484	0.1133	0.0447	0.0321	0.0229
H	0.0203	0.0227	0.0259	0.0188	0.0197	0.0221
I	0.0564	0.1010	0.0372	0.0557	0.0694	0.0593
K	0.0672	0.0443	0.0574	0.0560	0.0671	0.0810
L	0.1227	0.1068	0.0674	0.0994	0.1279	0.1477
M	0.0240	0.0219	0.0181	0.0214	0.0293	0.0304
N	0.0299	0.0252	0.0561	0.0259	0.0338	0.0336
P	0.0333	0.0208	0.0757	0.0472	0.0067	0.0031
Q	0.0443	0.0270	0.0330	0.0469	0.0497	0.0472
R	0.0594	0.0464	0.0470	0.0522	0.0677	0.0697
S	0.0496	0.0496	0.0744	0.0485	0.0422	0.0491
T	0.0395	0.0641	0.0572	0.0465	0.0412	0.0375
V	0.0591	0.1386	0.0473	0.0685	0.0677	0.0545
W	0.0168	0.0172	0.0111	0.0135	0.0130	0.0124
Y	0.0359	0.0483	0.0286	0.0306	0.0332	0.0344

.

References

Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403 – 410, 1990.

Zafer Aydin, Yucel Altunbasak, and Mark Borodovsky. Protein secondary structure prediction for a single-sequence using hidden semi-Markov models. *BMC Bioinformatics*, 7 (1):178, 2006.

Lalit R. Bahl, John Cocke, Frederick Jelinek, and Josef Raviv. Optimal decoding of linear codes for minimizing symbol error rate (corresp.). *IEEE Transactions on Information Theory*, 20(2):284–287, 1974.

Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 2000.

Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B. Methodological*, 48(3):259–302, 1986.

Julian Besag and Peter J. Green. Spatial statistics and Bayesian computation. *Journal of the Royal Statistical Society. Series B. Methodological*, 55(1):25–37, 1993.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006.

Matthew Brand, Nuria Oliver, and Alex Pentland. Coupled hidden Markov models for complex action recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 994–999, S.Juan, Puerto Rico, 1997.

Broňa Brejová, Daniel G. Brown, and Tomáš Vinař. The most probable annotation problem in hmms and its application to bioinformatics. *Journal of Computer and System Sciences*, 73(7):1060 – 1077, 2007a.

Broňa Brejová, Daniel G. Brown, and Tomáš Vinař. Advances in hidden Markov models for sequence annotation. In Ion I. Măndoiu and Alexander Zelikovski, editors, *Bioinformatics Algorithms: Techniques and Applications*, pages 55–92. John Wiley & Sons, Inc., 2007b.

Gary D. Brushe, Robert E. Mahony, and John B. Moore. A soft output hybrid algorithm for ML/MAP sequence estimation. *IEEE Transactions on Information Theory*, 44(7): 3129–3140, 1998.

Chris Burge and Samuel Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268(1):78 – 94, 1997.

Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*. Springer Series in Statistics. Springer, New York, 2005.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 (2):255–308, 2009.

Luis E. Carvalho and Charles E. Lawrence. Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proceedings of the National Academy of Sciences of the United States of America*, 105(9):3209–3214, 2008.

Christiane Cocozza-Thivent and Abdelkrim Bekkhoucha. Estimation in Pickard random fields and application to image processing. *Pattern Recognition*, 26(5):747–761, 1993.

Richard Durbin, Sean Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

Sean Eddy. What is a hidden Markov model? *Nature Biotechnology*, 22(10):1315 – 1316, 2004.

Yariv Ephraim and Neri Merhav. Hidden Markov processes. *IEEE Transactions on Information Theory*, 48(6):1518–1569, June 2002.

Piero Fariselli, Pier Martelli, and Rita Casadio. A new decoding algorithm for hidden Markov models improves the prediction of the topology of all-beta membrane proteins. *BMC Bioinformatics*, 6(Suppl 4):S12, 2005.

Kuzman Ganchev, João V. Graça, and Ben Taskar. Better alignments = better translations? In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 986–993, Columbus, Ohio, 2008.

Stuart Geman and Donald Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

Peter J. Green and Sylvia Richardson. Hidden Markov models and disease mapping. *Journal of the American Statistical Association*, 97(460):1055–1070, 2002.

Jeremiah F. Hayes, Thomas M. Cover, and Juan B. Riera. Optimal sequence detection and optimal symbol-by-symbol detection: similar algorithms. *IEEE Transactions on Communications*, 30(1):152–157, January 1982.

Ian Holmes and Richard Durbin. Dynamic programming alignment accuracy. *Journal of Computational Biology*, 5(3):493–504, 1998.

Xuedong Huang, Yasuo. Ariki, and Mervyn Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, Edinburgh, UK, 1990.

Frederick Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64:532–556, April 1976.

Frederick Jelinek. *Statistical Methods for Speech Recognition*. The MIT Press, Cambridge, Massachusetts, 2001.

Dhiraj Joshi, Jia Li, and James Z. Wang. A computationally efficient approach to the estimation of two- and three-dimensional hidden Markov models. *IEEE Transactions on Image Processing*, 15(7):1871–1886, 2006.

Lukas Käll, Anders Krogh, and Erik L. L. Sonnhammer. An HMM posterior decoder for sequence feature prediction that includes homology information. *Bioinformatics*, 21 (suppl_1):i251–257, 2005.

Alexey A. Koloydenko and Jüri Lember. Infinite Viterbi alignments in the two state hidden Markov models. *Acta et Commentationes Universitatis Tartuensis de Mathematica*, (12): 109–124, 2008.

Timo Koski. *Hidden Markov Models for Bioinformatics*, volume 2 of *Computational Biology Series*. Kluwer Academic Publishers, Dordrecht, 2001.

Anders Krogh. Two methods for improving performance of an HMM and their application for gene finding. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 179–186, Halkidiki, Greece, 1997.

Anders Krogh. An Introduction to Hidden Markov Models for Biological Sequences. In David B.Searls Steven L. Salzberg and Simon Kasif, editors, *Computational Methods in Molecular Biology*. Elsevier Science, first edition, 1998.

Kristi Kuljus and Jüri Lember. Asymptotic risks of Viterbi segmentation. *Stochastic Processes and Their Applications*, 122(9):3312–3341, 2012.

Hans Künsch, Stuart Geman, and Athanasios Kehagias. Hidden Markov random fields. *The Annals of Applied Probability*, 5(3):577–602, 1995.

Steffen L. Lauritzen. *Graphical models*, volume 17 of *Oxford Statistical Science Series*. Oxford University Press, New York, 1996.

Jüri Lember. On approximation of smoothing probabilities for hidden Markov models. *Statistics and Probability Letters*, 81(2):310–316, 2011a.

Jüri Lember. A correction on approximation of smoothing probabilities for hidden Markov models. *Statistics and Probability Letters*, 81(9):1463–1464, September 2011b.

Jüri Lember and Alexey A. Koloydenko. The Adjusted Viterbi training for hidden Markov models. *Bernoulli*, 14(1):180–206, 2008.

Jüri Lember and Alexey A. Koloydenko. A constructive proof of the existence of Viterbi processes. *IEEE Transactions on Information Theory*, 56(4):2017–2033, 2010.

Jüri Lember, Kristi Kuljus, and Alexey A. Koloydenko. Theory of segmentation. In Przemyslaw Dymarski, editor, *Hidden Markov Models, Theory and Applications*, Bioinformatics, pages 51–84. InTech, 2011.

Jia Li, Robert M. Gray, and Richard A. Olshen. Multiresolution image classification by hierarchical modeling with two-dimensional hidden Markov models. *IEEE Transactions on Information Theory*, 46(5):1826–1841, 2000.

Shu Lin and Daniel J. Costello Jr. *Error Control Coding: Fundamental and Applications*. Computer Applications in Electrical Engineering. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

William H. Majoros and Uwe Ohler. Advancing the state of the art in computational gene prediction. In Sorin Istrail, Pavel Pevzner, and Michael Waterman, editors, *Knowledge Discovery and Emergent Complexity in Bioinformatics*, volume 4366 of *Lecture Notes in Computer Science*, pages 81–106. Springer Berlin / Heidelberg, 2007.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, Massachusetts, 1999.

Jose L. Marroquin, Edgar Arce Santana, and Salvador Botello. Hidden markov measure field models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1380–1387, 2003.

Joshua Mason, Kathryn Watkins, Jason Eisner, and Adam Stubblefield. A natural language approach to automated cryptanalysis of two-time pads. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, pages 235–244, Alexandria, Virginia, 2006.

MATLAB. *Version 7.13.0.564 (R2011b)*. The MathWorks, Inc., Natick, Massachusetts, 2011.

Erik McDermott and Timothy J. Hazen. Minimum classification error training of landmark models for real-time continuous speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Montreal, Quebec, 2004.

Clare A. McGrory, D. Michael Titterington, Robert W. Reeves, and Anthony N. Pettitt. Variational Bayes for estimating the parameters of a hidden Potts model. *Statistics and Computing*, 19(3):329–340, 2009.

Hermann Ney, Volker Steinbiss, Reinhold Haeb-Umbach, B.-H. Tran, and Ute Essen. An overview of the Philips research system for large vocabulary continuous speech recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):33–70, 1994.

Mukund Padmanabhan and Michael A. Picheny. Large-vocabulary speech recognition algorithms. *Computer*, 35(4):42 – 50, 2002.

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

Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., Upper Saddle River, New Jersey, 1993.

Lawrence R. Rabiner, Jay G. Wilpon, and Biing-Hwang Juang. A segmental k-means training procedure for connected word recognition. *AT&T Technical Journal*, 65(3):21–31, 1986.

Patrick Robertson, Emmanuelle Villebrun, and Peter Hoeher. A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain. In *Proceedings of IEEE International Conference on Communications*, volume 2, pages 1009–1013, Seattle, Washington, 1995.

Havard Rue. New loss functions in Bayesian imaging. *Journal of the American Statistical Association*, 90(431):900–908, 1995.

Asaf A. Salamov and Victor V. Solovyev. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignments. *Journal of Molecular Biology*, 247(1):11 – 15, 1995.

Kengo Sato, Michiaki Hamada, Kiyoshi Asai, and Toutai Mituyama. Centroidfold: a web server for RNA secondary structure prediction. *Nucleic Acids Research*, 37(suppl 2): W277–W280, 2009.

Han Shu, I. Lee Hetherington, and James Glass. Baum-Welch training for segment-based speech recognition. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 43–48, St. Thomas, U. S. Virgin Islands, 2003.

Softberry, Inc. SSENVID: Protein secondary structure and environment assignment from atomic coordinates. `http://linux1.softberry.com/berry.phtml?topic=ssenvid&group=help&subgroup=propt`, 2001. Accessed: 15.10.2011.

Volker Steinbiss, Herman Ney, Xavier L. Aubert, Stefan Besling, Christian Dugast, Ute Essen, Daryl Geller, Reinhold Haeb-Umbach, Reinhard Kneser, Humberto G. Meier, Martin Oerder, and B.-H. Tran. The Philips research system for continuous-speech recognition. *Philips Journal of Research*, 49:317–352, 1995.

Nikko Ström, I. Lee Hetherington, Timothy J. Hazen, Eric Sandness, and James Glass. Acoustic modeling improvements in a segment-based speech recognizer. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 139–142, Keystone, Colorado, 1999.

The MathWorks, Inc. *Statistics Toolbox*$^{\text{TM}}$ *User's Guide*. Natick, Massachusetts, R2012a edition, 2012.

Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

Stephan Vogel, Hermann Ney, and Christoph Tillmann. HMM-based word alignment in statistical translation. In *Proceedings of the 16th Conference on Computational Linguistics*, volume 2, pages 836–841, Copenhagen, Denmark, 1996.

Gerhard Winkler. *Image Analysis, Random Fields and Markov chain Monte Carlo Methods*, volume 27 of *Applications of Mathematics (New York)*. Springer-Verlag, Berlin, second edition, 2003.

Christopher Yau and Chris C. Holmes. A decision theoretic approach for segmental classification using Hidden Markov models. *ArXiv e-prints*, 2010. URL `http://arxiv.org/abs/1007.4532`.