3

Theory of Segmentation

Jüri Lember[1], Kristi Kuljus[2] and Alexey Koloydenko[3]
[1]*University of Tartu*
[2]*Swedish University of Agricultural Sciences*
[3]*Royal Holloway, University of London*
[1]*Estonia*
[2]*Sweden*
[3]*UK*

1. Introduction

1.1 Preliminaries

In this chapter we focus on what Rabiner in his popular tutorial (Rabiner, 1989) calls "uncovering the hidden part of the model" or "Problem 2", that is, *hidden path inference*. We consider a hidden Markov model $(X, Y) = \{(X_t, Y_t)\}_{t \in \mathbb{Z}}$, where $Y = \{Y_t\}_{t \in \mathbb{Z}}$ is an unobservable, or *hidden*, homogeneous Markov chain with a finite state space $S = \{1, \ldots, K\}$, transition matrix $\mathbb{P} = (p_{i,j})_{i,j \in S}$ and, whenever relevant, the initial probabilities $\pi_s = \mathbf{P}(Y_1 = s)$, $s \in S$. A reader interested in extensions to the continuous case is referred to (Cappé et al., 2005; Chigansky & Ritov, 2010). The Markov chain will be further assumed to be of the first order, bearing in mind that a higher order chain can always be converted to a first order one by expanding the state space. To simplify the mathematics, we assume that the Markov chain Y is stationary and ergodic. This assumption is needed for the asymptotic results in Section 3, but not for the finite time-horizon in Section 2. In fact, Section 2 does not even require the assumption of homogeneity. The second component $X = \{X_t\}_{t \in \mathbb{Z}}$ is an observable process with X_t taking values in $\mathcal{X}$ that is typically a subspace of the Euclidean space, i.e. $\mathcal{X} \subset \mathbb{R}^d$. The process X can be thought of as a noisy version of Y. In order for (X, Y) to be a hidden Markov model, the following properties need to be satisfied:

1) given $\{Y_t\}$, the random variables $\{X_t\}$ are conditionally independent,

2) the distribution of X_t depends on $\{Y_t\}$ only through Y_t.

The process X is sometimes called a *hidden Markov process*. It is well known that the ergodicity of Y implies that of X (see, e.g. (Ephraim & Merhav, 2002; Genon-Catalot et al., 2000; Leroux, 1992)). The conditional distributions $P_s = \mathbf{P}(X_1 \in \cdot | Y_1 = s)$ are called *emission distributions*. Without loss of generality, we will assume that the emission distributions P_s all have densities f_s with respect to a common reference measure μ.

We often restrict the general process defined above to time interval I, where I is either $\{1, \ldots, n\}$ for some $n \geq 1$ (Section 2), or $I = \mathbb{N}$ (Section 3). Thus $\{(X_t, Y_t)\}_{t \geq 1}$ is a restriction of the doubly-infinite HMM to the positive integers and clearly, this process is ergodic as well. Since our study is mainly motivated by statistical and machine learning, our notation reverses the notation used in the mainstream HMM literature, e.g. (Cappé et al., 2005), where the hidden Markov chain is denoted by X and the observed process by Y.

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 usually be suppressed, e.g. $a_1^n \in \mathcal{A}^n$ will be written as a^n. With a slight abuse of notation, we will denote the conditional probability $\mathbf{P}(Y_k^l = y_k^l | X^n = x^n)$ by $p(y_k^l | x^n)$. We will often use the so-called *smoothing probabilities* $p_t(s|x^n) := \mathbf{P}(Y_t = s | X^n = x^n)$, where $s \in S$ and $1 \leq t \leq n$. We will denote the probability of x^n by $p(x^n)$ and, for any $s^n \in S^n$, we will write $p(s^n) = \mathbf{P}(Y^n = s^n)$ for the probability of Y^n having the outcome s^n; the distinction should be clear from the context.

1.2 The segmentation problem in the framework of statistical learning

By *segmentation* we refer to estimation of the unobserved realization $y^n = (y_1, \ldots, y_n)$ of the underlying Markov chain Y, given the observations $x^n = (x_1, \ldots, x_n)$ of X^n. In communications literature segmentation is also known as *decoding* (Bahl et al., 1974; Viterbi, 1967) or *state sequence detection* (Hayes et al., 1982). Segmentation is often the primary interest of the HMM-based inference, but it can also be an intermediate step of a larger problem such as estimation of the model parameters (Lember & Koloydenko, 2008; Rabiner, 1989), which will be discussed in Subsection 4.2. Despite its importance in the HMM-based methodology, a systematic study of different segmentation methods and their properties has been overdue (Lember & Koloydenko, 2010b). Here we present a unified approach to the segmentation problem based on statistical learning, and describe the commonly used as well as recently proposed solutions.

Formally we seek a mapping $g : \mathcal{X}^n \to S^n$ called a *classifier*, that maps every sequence of observations to a state sequence or *path*, which is sometimes also referred to as an *alignment* (Lember & Koloydenko, 2008)[1]. In order to assess the overall quality of g, it is natural to first measure the quality of each individual path $s^n \in S^n$ via a function known as *risk*. Thus, for a given x^n, let us denote the risk of s^n by $R(s^n|x^n)$. A natural approach to solving the segmentation problem is then to compute a state sequence with the minimum risk. In the framework of statistical decision and pattern recognition theory (Bishop, 2006) the risk is usually specified via a more basic entity known as a *loss function* $L : S^n \times S^n \to [0, \infty]$, where $L(a^n, b^n)$ is the loss incurred by estimating the actual state sequence a^n to be b^n. Then for any state sequence $s^n \in S^n$ the risk $R(s^n|x^n)$ is the conditional expectation of the loss $L(Y^n, s^n)$ given that $X^n = x^n$, i.e. $R(s^n|x^n) := E[L(Y^n, s^n)|X^n = x^n]$.

One popular loss function is the *zero-one* loss defined as

$$L_\infty(a^n, b^n) = \begin{cases} 1, \text{ if } a^n \neq b^n; \\ 0, \text{ if } a^n = b^n. \end{cases}$$

The minimizer of the risk $R_\infty(s^n|x^n)$ based on L_∞ is a sequence with maximum posterior probability $p(s^n|x^n)$, hence it is called the *maximum a posteriori (MAP) path*. The MAP-path is also called the *Viterbi path* after the *Viterbi algorithm* (Forney, 1973; Rabiner, 1989; Viterbi, 1967) used for its efficient computation.

Another popular approach is based on pointwise loss functions of the form

$$L_1(a^n, b^n) = \frac{1}{n} \sum_{t=1}^{n} l(a_t, b_t), \tag{1}$$

[1] This usage of the term "alignment" is broader than that of the HMM-based "multiple sequence alignment" in the bioinformatics context .

where $l(a_t, b_t) \geq 0$ is the loss of classifying the t^{th} symbol as b_t when the truth is a_t. Most commonly $l(s, s') = I_{\{s \neq s'\}}$, where I_A is the indicator function of a set A. Then the corresponding risk function $R_1(s^n|x^n)$ is simply the expected misclassification rate given the data x^n. Hence, the minimizer of this risk is a sequence with the lowest expected number of misclassifications. We refer to such sequences as *pointwise maximum a posteriori (PMAP)* alignments (Lember & Koloydenko, 2010b). The name refers to the fact that given x^n, the PMAP-alignment maximizes $\sum_{t=1}^{n} p_t(s_t|x^n)$ that obviously can be done pointwise. Note that the PMAP-alignment equivalently maximizes the product $\prod_{t=1}^{n} p_t(s_t|x^n)$, and therefore minimizes the *pointwise log* risk

$$\bar{R}_1(s^n|x^n) := -\frac{1}{n}\sum_{t=1}^{n} \log p_t(s_t|x^n). \tag{2}$$

Since the purpose is to maximize the expected number of correctly classified states, this is also known as the *optimal accuracy* alignment (Holmes & Durbin, 1998). In statistics, this type of estimation is known as *marginal posterior mode* (Winkler, 2003) or *maximum posterior marginals* (Rue, 1995) (MPM) estimation. In computational biology, this is also known as the *posterior decoding* (PD) (Brejová et al., 2008). 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 & Lawrence, 2008). In communications applications of HMMs, largely influenced by (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 to refer to this method.

Note that the introduced risk-based formalism does not impose any special conditions on Y. In particular, in this and in the next Section the chain Y need not be homogeneous and the conditional distribution of X_t given Y_t can, in principle, vary with t.

2. Hybrid classifiers

2.1 The problem

The Viterbi classifier has several drawbacks. First, the obtained alignment is not optimal and it can actually be quite poor in terms of accuracy as measured by the number of correctly classified states. Related to this is the reluctance of this decoder to switch states as can be seen from the following simple example taken from (Koloydenko & Lember, 2010).

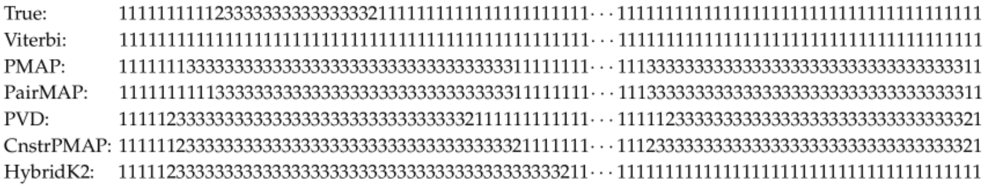
Example 1. A long sequence has been simulated from an HMM with the following parameters:

$$\mathbb{P} = \begin{pmatrix} 0.99 & 0.01 & 0 \\ 0.3 & 0.3 & 0.4 \\ 0 & 0.02 & 0.98 \end{pmatrix}, \qquad \pi = \begin{pmatrix} 0.5882 \\ 0.0196 \\ 0.3922 \end{pmatrix}, \qquad \pi^{\text{t}} = \pi^{\text{t}}\mathbb{P},$$

and the emission distributions

$$p_1 = (0.3, 0.2, 0.2, 0.3), \quad p_2 = (0.1, 0.3, 0.3, 0.3), \quad p_3 = (1/6, 1/6, 1/6, 1/2).$$

The sequence is then decoded with several classifiers including the Viterbi and PMAP ones. Figure 1 gives two fragments of the ground truth and decoded outputs; the complete output can be found in (Koloydenko & Lember, 2010). The example illustrates the typical tendency of the Viterbi classifier to get stuck in a state of sizable probability and therefore systematically



```
True:      1111111111233333333333333321111111111111111111111···11111111111111111111111111111111111111
Viterbi:   1111111111111111111111111111111111111111111111111···11111111111111111111111111111111111111
PMAP:      1111111333333333333333333333333333333333311111111···11133333333333333333333333333333333311
PairMAP:   1111111111333333333333333333333333333333311111111···11133333333333333333333333333333333311
PVD:       1111123333333333333333333333333333332111111111111···11111233333333333333333333333333333321
CnstrPMAP: 1111112333333333333333333333333333333333321111111···11123333333333333333333333333333333321
HybridK2:  1111123333333333333333333333333333333333333333211···11111111111111111111111111111111111111
```

Fig. 1. Decoding performance of different classifiers.

misclassify less frequent states. In its extreme form, this behavior would predict a constant sequence of *Heads* even when the *Heads* side of the coin is only slightly heavier than the *Tails* one. In HMMs, the inaccuracy problem can be alleviated by exploiting the PMAP-classifier. However, as can be noted from the forbidden transitions between states 1 and 3 along the PMAP-path in Figure 1, the PMAP-decoding allows alignments of low or even zero probability. In computational biology, for example, such paths would violate biological constraints and hence be *inadmissible*. We will return to this example when considering alternative classifiers.

As it has been shown in (Käll et al., 2005), it is possible to explicitly constrain the PMAP-classifier to avoid inadmissible paths. The constrained PMAP-alignment is the solution of the following optimization problem

$$\min_{s^n:\, p(s^n|x^n)>0} R_1(s^n|x^n) \quad \Leftrightarrow \quad \max_{s^n:\, p(s^n|x^n)>0} \sum_{t=1}^{n} p_t(s_t|x^n). \tag{3}$$

In our example the solution of (3) is seen in Figure 1 under the name CnstrPMAP. The solution is indeed admissible, i.e. the forbidden zero-probability transitions are not any more present. Observe that in the presence of path constraints, minimizing the R_1-risk is not any more equivalent to minimizing the $\bar{R}_1$-risk, therefore (3) is not equivalent to the optimization problem

$$\min_{s^n:\, p(s^n|x^n)>0} \bar{R}_1(s^n|x^n) \quad \Leftrightarrow \quad \max_{s^n:\, p(s^n|x^n)>0} \sum_{t=1}^{n} \log p_t(s_t|x^n). \tag{4}$$

The solution of (4) was recently used in (Fariselli et al., 2005) under the name *posterior Viterbi decoding* (PVD). On their tasks of predicting membrane proteins, PVD has shown results superior to the Viterbi and PMAP-classifiers. In our example, the solution of (4) is seen in Figure 1 under the name PVD. Note that it differs from the constrained PMAP but it is still admissible.

To summarize: since the overall error rate is not the only measure of accuracy and since PMAP may fail 100% in detecting an infrequent state, such as state 2 in our example, constraining the PMAP-classifier is not an ultimate answer. Also, in some applications such as gene-finding, region-based measures of accuracy are no less important than the pointwise ones, but direct minimization of the corresponding risks generally does not lead to efficient decoding algorithms.

2.2 Further issues and alternative solutions

Another serious drawback of the Viterbi decoding is that "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). In fact, Viterbi paths need not be representative of the overall posterior probability distribution (Carvalho & Lawrence, 2008) and can be significantly atypical (Lember & Koloydenko, 2010a). Indeed, imagine having to estimate the transition probabilities from the Viterbi path in Figure 1. This second problem of the Viterbi segmentation has been addressed by moving from single path inference towards envelops (Holmes & Durbin, 1998) and centroids (Carvalho & Lawrence, 2008). The most common approach here is 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* (Brejová et al., 2008; Fariselli et al., 2005; Käll et al., 2005; Krogh, 1997). However, this leads to the problem of *multiple paths*, which in practically important HMMs renders the dynamic programming approach of the Viterbi algorithm NP-hard (Brejová et al., 2007; Brown & Truszkowski, 2010). Unlike the Viterbi classifier, PMAP handles annotations as easily as it does state paths, including the enforcement of the positivity constraint (Käll et al., 2005). A number of heuristic approaches are also known to alleviate these problems, but none appears to be fully satisfactory (Brejová et al., 2008). Note that mapping optimal state paths to the corresponding annotations need not lead to optimal annotations and can give poor results (Brejová et al., 2007).

Although the Viterbi and PMAP-classifiers have been by far the most popular segmentation methods in practice, their aforementioned drawbacks have invited debates on the trade-off between the path accuracy and probability, and alternative approaches have demonstrated significantly higher performance in, for example, predicting various biological features.

In Subsection 2.3 below, which is based on (Lember & Koloydenko, 2010b), we show how several relevant risks can be combined within a very general penalized risk minimization problem with a small number of penalty terms. Tuning one of the penalty parameters allows us to "interpolate" between the PMAP- and Viterbi classifiers in a natural way, whereas the other terms give further interesting extensions. The minimization problem can then be solved by a dynamic programming algorithm similar to the Viterbi algorithm. We would like to remark that the idea of interpolation between the Viterbi and PMAP-estimators was already hinted at in the seminal tutorial (Rabiner, 1989) and then considered in (Brushe et al., 1998). In spite of those, no general systematic study of hybridization of the Viterbi and PMAP-classifiers has been published before.

2.3 Generalized risk-based hybrid classifiers

Although the constrained PMAP-classifier and PVD guarantee admissible paths, as can also be noted from Figure 1, the (posterior) probability of such paths can still be very low. Hence, it seems natural to consider instead of (3) the following penalized optimization problem:

$$\max_{s^n}\Big[\sum_{t=1}^{n} p_t(s_t|x^n)+\log p(s^n)\Big] \quad\Leftrightarrow\quad \min_{s^n}\Big[R_1(s^n|x^n)+\bar{R}_\infty(s^n)\Big], \tag{5}$$

where

$$\bar{R}_\infty(s^n) := -\frac{1}{n}\log p(s^n)$$

is the prior log risk which does not depend on the data. The logic behind (5) is clear: we aim to look for the alignment that simultaneously minimizes the R_1-risk and maximizes the path probability. A more general problem can be written in the form

$$\min_{s^n}\Big[R_1(s^n|x^n)+Ch(s^n)\Big], \tag{6}$$

where $C \geq 0$ and h is some penalty function. Taking $Ch(s^n) = \infty \times (1 - \operatorname{sign}(p(s^n))$ reduces problem (6) to problem (3).

Similarly, instead of (4), the following problem can be considered:

$$\max_{s^n} \Big[\sum_{t=1}^{n} \log p_t(s_t|x^n) + \log p(s^n) \Big] \quad \Leftrightarrow \quad \min_{s^n} \Big[\bar{R}_1(s^n|x^n) + \bar{R}_\infty(s^n) \Big].$$

Again, the problem above can be generalized as

$$\min_{s^n} \Big[\bar{R}_1(s^n|x^n) + Ch(s^n) \Big]. \tag{7}$$

Taking $Ch(s^n) = \infty \times (1 - \operatorname{sign}(p(s^n))$, reduces problem (7) to problem (4).

2.3.1 A general family of classifiers

Motivated by the previous argument, we consider the following yet more general problem:

$$\min_{s^n} \Big[C_1 \bar{R}_1(s^n|x^n) + C_2 \bar{R}_\infty(s^n|x^n) + C_3 \bar{R}_1(s^n) + C_4 \bar{R}_\infty(s^n) \Big], \tag{8}$$

where $C_1, \ldots, C_4 \geq 0$, $C_1 + \cdots + C_4 > 0$, and

$$\bar{R}_1(s^n|x^n) = -\frac{1}{n} \sum_{t=1}^{n} \log p_t(s_t|x^n), \text{ as defined in equation (2) above,}$$

$$\bar{R}_\infty(s^n|x^n) := \bar{R}_\infty(s^n; x^n) + \frac{1}{n} \log p(x^n),$$

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

$$\bar{R}_1(s^n) := -\frac{1}{n} \sum_{t=1}^{n} \log \mathbf{P}(Y_t = s_t),$$

$$\bar{R}_\infty(s^n) = -\frac{1}{n} \Big[\log \pi_{s_1} + \sum_{t=1}^{n-1} \log p_{s_t s_{t+1}} \Big] = -\frac{1}{n} \log p(s^n).$$

The newly introduced risk $\bar{R}_1(s^n)$ is the prior pointwise log risk. Evidently, the combination $C_1 = C_3 = C_4 = 0$ gives the Viterbi alignment, the combination $C_2 = C_3 = C_4 = 0$ yields the PMAP-alignment, whereas the combinations $C_1 = C_2 = C_3 = 0$ and $C_1 = C_2 = C_4 = 0$ give the *maximum prior probability* decoding and *marginal prior mode* decoding, respectively. The case $C_2 = C_3 = 0$ subsumes (7), and the case $C_1 = C_3 = 0$ is the problem

$$\min_{s^n} \Big[\bar{R}_\infty(s^n|x^n) + C \bar{R}_\infty(s^n) \Big]. \tag{9}$$

Thus, a solution to (9) is a generalization of the Viterbi decoding which allows for suppressed ($C > 0$) contribution of the data. It is important to note that with $C_2 > 0$ every solution of (8) is admissible.

Similarly to the generalized risk minimization problem in (8), a relevant generalization of (6) emerges as follows:

$$\min_{s^n} \Big[C_1 R_1(s^n|x^n) + C_2 \bar{R}_\infty(s^n|x^n) + C_3 R_1(s^n) + C_4 \bar{R}_\infty(s^n) \Big], \tag{10}$$

where

$$R_1(s^n) := \frac{1}{n}\sum_{t=1}^{n} \mathbf{P}(Y_t \neq s_t)$$

is the error rate when the data are ignored.

2.3.2 Solving (8) and (10)

The problems (8) and (10) would only be of theoretical interest if there were not an effective way to solve them. We now present a dynamical programming algorithm (similar to the Viterbi algorithm) for solving these problems. The algorithm requires the smoothing probabilities $p_t(j|x^n)$ for $t = 1,\ldots,n$ and $j \in S$, which can be computed by the usual forward-backward algorithm (Rabiner, 1989). For every $t = 1,\ldots,n$ and $s \in S$, let

$$g_t(s) := C_1 \log p_t(s|x^n) + C_2 \log f_s(x_t) + C_3 \log \mathbf{P}(Y_t = s).$$

Note that the function g_t depends on all the data x^n. For every $j \in S$ and for every $t = 1,2,\ldots,n-1$, define the scores

$$\delta_1(j) := C_1 \log p_1(j|x^n) + (C_2 + C_3 + C_4)\log \pi_j + C_2 \log f_j(x_1), \tag{11}$$

$$\delta_{t+1}(j) := \max_i \big(\delta_t(i) + (C_2 + C_4)\log p_{ij}\big) + g_{t+1}(j). \tag{12}$$

Using the scores $\delta_t(j)$, let for every $t = 1,\ldots,n$,

$$i_t(j) := \begin{cases} \arg\max_{i\in S}[\delta_t(i) + (C_2 + C_4)\log p_{ij}], & \text{when } t = 1,\ldots,n-1, \\ \arg\max_{i\in S}\delta_n(i), & \text{when } t = n; \end{cases} \tag{13}$$

$$\hat{s}^t(j) := \begin{cases} i_1(j), & \text{when } t = 1, \\ \big(\hat{s}^{t-1}(i_{t-1}(j)), j\big), & \text{when } t = 2,\ldots,n. \end{cases}$$

It is now not hard to see (see Th. 3.1 in (Lember & Koloydenko, 2010b)) that recursions (11)-(12) solve (8), meaning that any solution of (8) is in the form $\hat{s}^n(i_n)$, provided the ties in (13) are broken accordingly.

By a similar argument, problem (10) can be solved by the following recursions:

$$\delta_1(j) := C_1 p_1(j|x^n) + (C_2 + C_4)\log \pi_j + C_2 \log f_j(x_1) + C_3 \pi_j,$$

$$\delta_{t+1}(j) := \max_i \big(\delta_t(i) + (C_2 + C_4)\log p_{ij}\big) + g_{t+1}(j),$$

where now

$$g_t(s) = C_1 p_t(s|x^n) + C_2 \log f_s(x_t) + C_3 \mathbf{P}(Y_t = j).$$

2.4 *k*-block PMAP-alignment

As an idea for interpolating between the PMAP- and Viterbi classifiers, Rabiner mentions in his seminal tutorial (Rabiner, 1989) the possibility of maximizing the expected number of correctly estimated pairs or triples of (adjacent) states rather than the expected number of correct single states. With k being the length of the block ($k = 2,3,\ldots$) this entails minimizing the conditional risk

$$R_k(s^n|x^n) := 1 - \frac{1}{n-k+1}\sum_{t=1}^{n-k+1} p(s_t^{t+k-1}|x^n) \tag{14}$$

based on the following loss function:

$$L_k(y^n, s^n) := \frac{1}{n-k+1} \sum_{t=1}^{n-k+1} I_{\{s_t^{t+k-1} \neq y_t^{t+k-1}\}}.$$

Obviously, for $k = 1$ this gives the usual R_1-minimizer – the PMAP-alignment – which is known to allow inadmissible paths. It is natural to think that the minimizer of $R_k(s^n|x^n)$ evolves towards the Viterbi alignment "monotonically" as k increases to n. Indeed, when $k = n$, minimization of $R_k(s^n|x^n)$ in (14) is equivalent to minimization of $\bar{R}_\infty(s^n|x^n)$, which is achieved by the Viterbi alignment. In Figure 1 the minimizer of (14) for $k = 2$ appears under the name PairMAP, and, as the example shows, it still has zero probability. This is a major drawback of using the loss L_k.

We now show that this drawback can be overcome when the sum in (14) is replaced by the product. This is not an equivalent problem, but with the product the k-block idea works well – the longer the block, the bigger the probability and the solution is guaranteed to be admissible even for $k = 2$. Moreover, this gives another interpretation to the risk $\bar{R}_1(s^n|x^n) + C\bar{R}_\infty(s^n|x^n)$. Let $k \in \mathbb{N}$. We define

$$\bar{U}_k(s^n|x^n) := \prod_{j=1-k}^{n-1} p(s_{(j+1)\vee 1}^{(j+k)\wedge n}|x^n), \quad \bar{R}_k(s^n|x^n) := -\frac{1}{n}\log \bar{U}_k(s^n|x^n).$$

Thus $\bar{U}_k(s^n|x^n) = U_1^k \cdot U_2^k \cdot U_3^k$, where

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

Clearly, for $k = 1$, $\bar{R}_k$ equals $\bar{R}_1(s^n|x^n)$ defined in (2), so it is a natural generalization of $\bar{R}_1$. The meaning of the risk $\bar{R}_k$ will be transparent from the following equality proved in (Lember & Koloydenko, 2010b): for every s^n,

$$\bar{R}_k(s^n|x^n) = (k-1)\bar{R}_\infty(s^n|x^n) + \bar{R}_1(s^n|x^n).$$

Thus, the minimizer of $\bar{R}_k(s^n|x^n)$ is a solution of (8) with $C_1 = 1$, $C_2 = k-1$, $C_3 = C_4 = 0$. Note that the solution is admissible for every $k > 1$. It is easy to see that increasing the block length k increases the posterior probability as well as the $\bar{R}_1$-risk of the solution. Hence, this provides a natural interpolation between the Viterbi and the PMAP-alignments. In Figure 1 the minimizer of $\bar{R}_2(s^n|x^n)$ for $k = 2$ is shown under the name HybridK2. The difference between the HybridK2-alignment and the PairMAP-alignment (the minimizer of the R_2-risk) is clearly visible; in particular, the HybridK2-alignment is of positive probability. From the figure it is also evident that the HybridK2-alignment possesses the properties of both the Viterbi and PMAP-alignments.

3. Infinite segmentation

In the previous section, several alignments for segmenting the observations $x^n = (x_1, \ldots, x_n)$ were defined. The next question one can ask is the following: what are the long-run properties of these different classifiers? This question is not easy to answer, since in general there is no obvious notion of infinite (asymptotic) alignment. Indeed, if $(g_1,\ldots,g_n)=g(x_1,\ldots,x_n)$ is an alignment, then adding one more observation x_{n+1} can in principle change the whole alignment so that $g_t(x^n)\neq g_t(x^{n+1})$ for every t, where $g_t(x^n)$ stands for the t^{th} element of $g(x^n)$. On the other hand, it is intuitively expected that such a situation is rather atypical and that a few, say k, first elements of g will be fixed *almost surely* as n goes to infinity. If this is the case, then an infinite classifier can be defined as follows.

Def. For every $n\in\mathbb{N}$, let $g^n:\mathcal{X}^n\to S^n$ be a classifier. We say that the sequence $\{g^n\}$ of classifiers can be *extended to infinity*, if there exists a function $g:\mathcal{X}^\infty\to S^\infty$ such that for almost every realization $x^\infty\in\mathcal{X}^\infty$ the following holds: for every $k\in\mathbb{N}$ there exists $m\geq k$ (depending on x^∞) such that for every $n\geq m$ the first k elements of $g^n(x^n)$ are the same as the first k elements of $g(x^\infty)$, i.e. $g^n(x^n)_i=g(x^\infty)_i$, $i=1,\ldots,k$. The function g will be referred to as an *infinite classifier*. If an infinite classifier exists, then applying it to the observations x^∞ gives us an *infinite alignment* $g(x^\infty)$, and applying it to the process X^∞ gives us a random S-valued process $g(X^\infty)$ that is called the *alignment process*. ◇

Hence, to study the asymptotic properties of various classifiers, the existence of an infinite alignment is the first problem to be addressed. It is also desirable that various SLLN-type results hold for the alignment process. This is guaranteed if the alignment process is regenerative or ergodic. Despite the unified risk-based representation of the different classifiers presented here, proving the existence of the infinite alignment for them requires different mathematical tools.

3.1 Infinite Viterbi alignment and Viterbi process

Justified or not, the Viterbi classifier is the most popular one in practice. In (Lember & Koloydenko, 2008; 2010a), under rather general assumptions on the HMM, a constructive proof of the existence of the infinite Viterbi classifier was given. We shall now explain the basic ideas behind the construction.

3.1.1 The idea of piecewise alignments

The proof is based on the existence of the so-called *barriers*. We believe that the following oversimplified but insightful example will help the reader to understand this concept. Suppose there is a state, say 1, and a set of observations $A\subset\mathcal{X}$ such that $P_1(A)>0$ while $P_l(A)=0$ for $l=2,\ldots,K$. Thus, at time u any observation $x_u\in A$ is almost surely generated under $Y_u=1$, and we say that x_u *indicates its state*. Consider n to be the terminal time and note that any positive probability path, including the MAP/Viterbi ones, has to go through state 1 at time u. This allows us to split the Viterbi alignment $v(x^n)$ into v_1^u and v_{u+1}^n, an alignment from time 1 through time u, and a conditional alignment from time $u+1$ through time n, respectively. Moreover, it is clear that the first piece v_1^u maximizes $p(s^u|x_1^u)$ over all paths from time 1 through time u, $v_u=1$, and the second piece v_{u+1}^n maximizes $\mathbf{P}(Y_{u+1}^n=s^{n-u}|X_{u+1}^n=x_{u+1}^n,Y_u=1)$. Clearly, any additional observations x_{n+1}^m do not change the fact that x_u indicates its state. Hence, for any extension of x^n the first part of the alignment is always v_1^u. Thus, any observation that indicates its state also fixes the beginning of the alignment for every $n>u$. Suppose now that x_t, $u<t<n$, is another observation that indicates its state, say, also 1. By the same argument, the piece v_{u+1}^n can be split into pieces v_{u+1}^t and v_{t+1}^n, where v_{u+1}^t maximizes $\mathbf{P}(Y_{u+1}^t=s^{t-u}|X_{u+1}^t=x_{u+1}^t,Y_u=1)$ and terminates in 1, i.e. $v_t=1$. Again, increasing n does not change the fact that x_t indicates its state, so that v_u^t is independent of all the observations before u and after t. Therefore, the Viterbi alignment up to t can be constructed independently of the observations x_{t+1}^n by concatenating the pieces

v_1^u and v_{u+1}^t. Since our HMM is now a stationary process that has a positive probability to generate state-indicating observations, there will be infinitely many such observations *almost surely*. Since the Viterbi alignment between two such observations x_u and x_t can be be found as the maximizer of $p(\cdot|x_u^t)$, the infinite alignment can be constructed by concatenating the corresponding pieces. We say that the alignment can be constructed *piecewise*.

3.1.2 Nodes

The example above is rather exceptional and we next define nodes to generalize the idea of state-indicating observations. Recall that the Viterbi algorithm is a special case of the general dynamic programming algorithm introduced in Subsection 2.3 with $C_2 = 1$ and $C_1 = C_3 = C_4 = 0$. In particular, the basic recursion for obtaining the scores (11) and (12) for the Viterbi algorithm is as follows: for every $j \in S$ and $t = 1, \ldots, n-1$,

$$\begin{aligned}\delta_1(j) &= \log \pi_j + \log f_j(x_1),\\ \delta_{t+1}(j) &= \max_i \left(\delta_t(i) + \log p_{ij}\right) + \log f_j(x_{t+1}).\end{aligned}$$

The Viterbi alignment $v(x^n)$ is given by $v^n(i_n)$, where for every $j \in S$, the paths $v^t(j)$, $t = 1, \ldots, n$, are obtained recursively:

$$v^t(j) = \begin{cases} i_1(j), & \text{when } t = 1,\\ \left(v^{t-1}(i_{t-1}(j)), j\right), & \text{when } t = 2, \ldots, n;\end{cases}$$

with $i_t(j)$ being (recall (13))

$$i_t(j) = \begin{cases} \arg\max_{i \in S}[\delta_t(i) + \log p_{ij}], & \text{when } t = 1, \ldots, n-1,\\ \arg\max_{i \in S} \delta_n(i), & \text{when } t = n.\end{cases}$$

Def. Given the first u observations, the observation x_u is said to be an *l-node* (of order zero) if

$$\delta_u(l) + \log p_{lj} \geq \delta_u(i) + \log p_{ij}, \quad \forall i, j \in S. \tag{15}$$

We also say that x_u is a *node* if it is an l-node for some $l \in S$. We say that x_u is a strong node if the inequalities in (15) are strict for every $i, j \in S$, $i \neq l$. ◇

In other words, x_u is an l-node if for appropriate tie-breaking $i_u(j) = l$ for every $j \in S$ (see also Figure 2). This obviously implies that (under the same tie-breaking) the first u elements of the Viterbi alignment are fixed independently of the observations x_{u+1}^n. If the node is strong, then all the Viterbi alignments must coalesce at u. Thus, the concept of strong nodes circumvents the inconveniences caused by non-uniqueness: no matter how the ties are broken, every alignment is forced into l at u, and any tie-breaking rule would suffice for the purpose of obtaining the fixed alignments. However tempting, strong nodes unlike the general ones are quite restrictive. Indeed, suppose that the observation x_u indicates its state, say 1. Then $f_1(x_u) > 0$ and $f_i(x_u) = 0$ for $i \neq 1$. Hence $\delta_u(1) > -\infty$ and $\delta_u(i) = -\infty$ for every $i \in S$, $i \neq 1$. Thus (15) holds and x_u is a 1-node. In other words, every observation that indicates its state is a node. If in addition $p_{1j} > 0$ for every $j \in S$, then for every $i, j \in S$, $i \neq 1$, the right-hand side of (15) is $-\infty$, whereas the left-hand side is finite, making x_u a strong node. If, however, there is j such that $p_{1j} = 0$, which can easily happen if $K > 2$, then for such j both sides are $-\infty$ and x_u is not strong anymore.

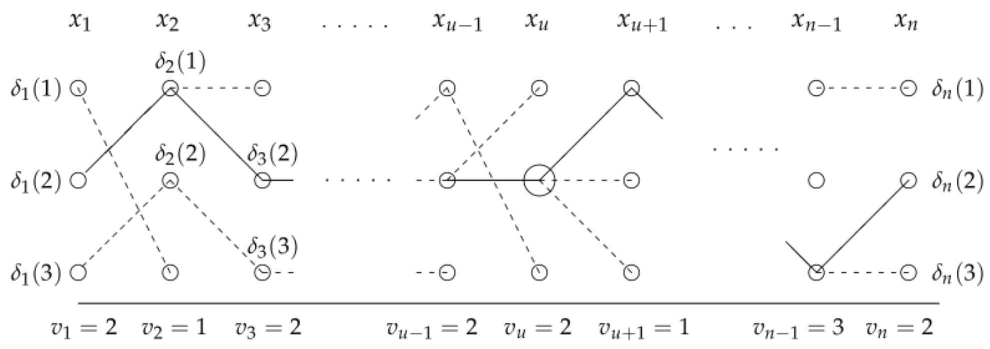


Fig. 2. An example of the Viterbi algorithm in action. The solid line corresponds to the final alignment $v(x^n)$. The dashed links are of the form $(t, i_t(j)) - (t+1, j)$ and are not part of the final alignment. E.g., $(1,3) - (2,2)$ is because $3 = i_1(2)$ and $2 = i_2(3)$. The observation x_u is a 2-node since we have $2 = i_u(j)$ for every $j \in S$.

3.1.3 Higher order nodes

We next extend the notion of nodes to account for the fact that a general ergodic $\mathbb{P}$ can have a zero in every row, in which case nodes of order zero need not exist. Indeed, suppose x^u is such that $\delta_u(i) > -\infty$ for every i. In this case, (15) implies $p_{lj} > 0$ for every $j \in S$, i.e. the l^{th} row of $\mathbb{P}$ must be positive, and (15) is equivalent to

$$\delta_u(l) \geq \max_i [\max_j (\log p_{ij} - \log p_{lj}) + \delta_u(i)].$$

First, we introduce $p_{ij}^{(r)}(u)$, the maximum probability over the paths connecting states i and j at times u and $u+r+1$, respectively. For each $u \geq 1$ and $r \geq 1$, let

$$p_{ij}^{(r)}(u) := \max_{q^r \in S^r} p_{iq_1} f_{q_1}(x_{u+1}) p_{q_1 q_2} f_{q_2}(x_{u+2}) p_{q_2 q_3} \ldots p_{q_{r-1} q_r} f_{q_r}(x_{u+r}) p_{q_r j}.$$

Note that for $r \geq 1$, $p_{ij}^{(r)}(u)$ depends on the observations x_{u+1}^{u+r}. By defining

$$i_t^{(r)}(j) := \arg\max_{i \in S} [\delta_t(i) + \log p_{ij}^{(r)}],$$

we get that for every $t = 1, \ldots, n$ and $j \in S$ it holds that the $(t-r-1)^{\text{th}}$ element of $v^t(j)$ equals $i_{t-r-1}^{(r)}(j)$, i.e.

$$v_{t-r-1}^t(j) = i_{t-r-1}^{(r)}(j). \tag{16}$$

Def. Given the first $u+r$ observations, the observation x_u is said to be an *l-node of order r* if

$$\delta_u(l) + \log p_{lj}^{(r)}(u) \geq \delta_u(i) + \log p_{ij}^{(r)}(u), \quad \forall i, j \in S. \tag{17}$$

The observation x_u is said to be an r^{th} order node if it is an r^{th}-order l-node for some $l \in S$. The node is said to be *strong* if the inequalities in (17) are strict for every $i, j \in S, i \neq l$. ◇

Note that any r^{th}-order node is also a node of order r' for any integer $r \leq r' < n$, and thus by the order of a node we will mean the minimal such r. Note also that for $K = 2$, a node of any order is a node of order zero. Hence, positive order nodes emerge for $K \geq 3$ only.

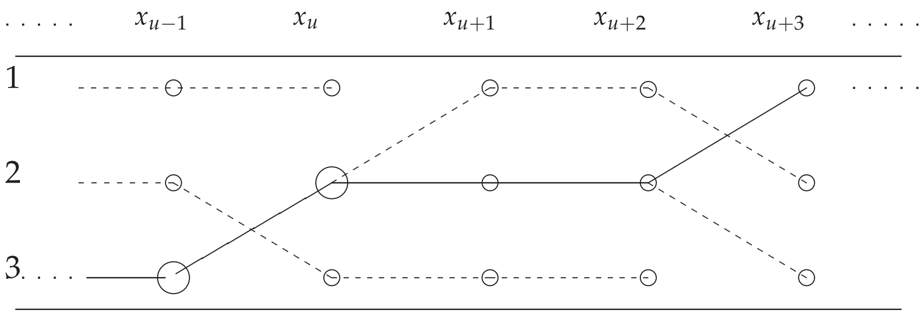


Fig. 3. Suppose $u < n$, and x_u is a 2^{nd} order 2-node, and x_{u-1} is a 3^{rd} order 3-node. Therefore, any alignment $v(x^n)$ has $v(x^n)_u = 2$.

This definition implies that x_u is an l-node of order r if and only if (under suitable tie breaking) $i_u^{(r)}(j) = l$ for every $j \in S$. By (16), this means that $v_u^{u+r+1}(j) = l$ for every $j \in S$, implying that the first u elements of the Viterbi alignment are fixed independently of the observations x_{u+r+1}^n (see Figure 3). This means that the role of higher order nodes is similar to the role of nodes. Suppose now that the realization x^∞ contains infinitely many r^{th} order l-nodes $u_1 < u_2 < \ldots$. Then, as explained in Subsection 3.1.1, the infinite Viterbi alignment $v(x^\infty)$ can be constructed piecewise, i.e. the observations x^{u_1+r} fix the piece v^{u_1}, then the observations $x_{u_1+1}^{u_2+r}$ fix the piece $v_{u_1+1}^{u_2}$, and so on. In the absence of ties, the resulting piecewise infinite alignment is unique. In the presence of ties we require that the ties be broken consistently, and always so that the alignment goes through l at times u_i. Then the resulting infinite Viterbi alignment is called *proper* (see (Lember & Koloydenko, 2008) for details).

3.1.4 Barriers

Recall that nodes of order r at time u are defined relative to the entire realization x^{u+r}. Thus, whether x_u is a node or not depends, in principle, on all observations up to x_{u+r}. On the other hand, the observation that indicates its state is a node independently of the observations before or after it. This property is generalized by the concept of *barrier*. Informally speaking, a barrier is a block of observations that is guaranteed to contain a (probably higher order) node independently of the observations before and after this block. The formal definition is as follows.

Def. Given $l \in S$, a block of observations $z^M \in \mathcal{X}^M$ is called a (strong) *l-barrier* of order $r \geq 0$ and length $M \geq 1$ if for any realization x^n, $M \leq n \leq \infty$, such that $x_{t-M+1}^t = z^M$ for some t, $M \leq t \leq n$, the observation x_{t-r} is a (strong) l-node of order r. $\diamond$

According to this definition, any observation that indicates its state is a barrier of length one. Usually a set $A \subset \mathcal{X}$ can be found such that any observation from A indicates its state. More typically, however, there is a set $B \subset \mathcal{X}^M$ such that any sequence from B is a barrier. Hence barriers can be detected by a sliding window of length M. More importantly, when such a subset B is constructed, then by ergodicity of X almost every realization x^∞ contains infinitely many barriers provided that the set B has a positive probability to occur. As already explained, having infinitely many barriers guarantees infinitely many (usually higher order) nodes, and based on these nodes, the infinite Viterbi alignment can be constructed piecewise.

Thus, everything boils down to the construction of the barrier set B. We are able to do this under some mild assumptions on the HMM. Next we state and discuss briefly these assumptions.

Recall that f_s, $s \in S$, are the densities of $P_s := \mathbf{P}(X_1 \in \cdot | Y_1 = s)$ with respect to some reference measure μ. For each $s \in S$, let $G_s := \{x \in \mathcal{X} : f_s(x) > 0\}$.

Def. We call a non-empty subset $C \subset S$ *a cluster* if the following conditions are satisfied:

$$\min_{j \in C} P_j(\cap_{s \in C} G_s) > 0 \quad \text{and} \quad \text{either } C = S \quad \text{or} \quad \max_{j \notin C} P_j(\cap_{s \in C} G_s) = 0. \qquad \diamond$$

Therefore, a cluster is a maximal subset of states such that $G_C = \cap_{s \in C} G_s$, the intersection of the supports of the corresponding emission distributions, is "detectable". There always exists at least one cluster; distinct clusters need not be disjoint, and a cluster can consist of a single state. In this latter case such a state is not hidden, since it is exposed by any observation it emits. If $K = 2$, then S is the only cluster possible, because otherwise the underlying Markov chain would cease to be hidden. Our first assumption is the following.

A1 (cluster-assumption): There exists a cluster $C \subset S$ such that the sub-stochastic matrix $R = (p_{ij})_{i,j \in C}$ is primitive, i.e. there is a positive integer r such that the r^{th} power of R is strictly positive.

The cluster assumption **A1** is often met in practice. It is clearly satisfied if all elements of the matrix $\mathbb{P}$ are positive. Since any irreducible aperiodic matrix is primitive, the assumption **A1** is also satisfied in this case if the densities f_s satisfy the following condition: for every $x \in \mathcal{X}$, $\min_{s \in S} f_s(x) > 0$, i.e. for all $s \in S$, $G_s = \mathcal{X}$. Thus, **A1** is more general than the *strong mixing condition* (Assumption 4.2.21 in (Cappé et al., 2005)) and also weaker than Assumption 4.3.29 in (Cappé et al., 2005). Note that **A1** implies aperiodicity of Y, but not vice versa.

Our second assumption is the following.

A2: For each state $l \in S$,

$$P_l\Big(\big\{x \in \mathcal{X} : f_l(x) p_l^* > \max_{s, s \neq l} f_s(x) p_s^*\big\}\Big) > 0, \quad \text{where} \quad p_l^* = \max_j p_{jl}, \quad \forall l \in S. \qquad (18)$$

The assumption **A2** is more technical in nature. In (Koloydenko & Lember, 2008) it was shown that for a two-state HMM, (18) always holds for one state, and this is sufficient for the infinite Viterbi alignment to exist. Hence, for the case $K = 2$, **A2** can be relaxed. Other possibilities for relaxing **A2** are discussed in (Lember & Koloydenko, 2008; 2010a). To summarize: we believe that the cluster assumption **A1** is essential for HMMs, while the assumption **A2**, although natural and satisfied for many models, can be relaxed. For more general discussion about these assumptions, see (Koloydenko & Lember, 2008; Lember, 2011; Lember & Koloydenko, 2008; 2010a). The following Lemma is the core of our proof of the existence of the infinite Viterbi alignment.

Lemma 3.1. *Assume* **A1** *and* **A2**. *Then for some integers M and r, $M > r \geq 0$, there exist a set $B = B_1 \times \cdots \times B_M \subset \mathcal{X}^M$, an M-tuple of states $y^M \in S^M$ and a state $l \in S$, such that every $z^M \in B$ is an l-barrier of order r, $y_{M-r} = l$ and $\mathbf{P}\big(X^M \in B, \quad Y^M = y^M\big) > 0$.*

Lemma 3.1 is proved in (Lember & Koloydenko, 2010a), and implies that $\mathbf{P}(X^M \in B) > 0$. Hence almost every realization of X contains infinitely many barriers, which makes the piecewise construction possible.

3.1.5 Viterbi process and its regenerativity.

If **A1** and **A2** hold, then by Lemma 3.1 and the piecewise construction there exists an infinite Viterbi classifier $v : \mathcal{X}^\infty \to S^\infty$. By applying v to HMM we obtain the alignment process $V = v(X)$. We shall call the process $V = \{V_t\}_{t=1}^\infty$ the *Viterbi process*. The existence of the Viterbi process follows from the existence of infinitely many barriers. Now recall that Lemma 3.1 states more than merely the existence of infinitely many barriers. Namely, the Lemma actually also states that *almost every* realization of a two-dimensional process (X, Y) contains infinitely many barriers from B synchronized with y^M. In other words, almost every realization of (X, Y) contains infinitely many pairs (z^M, y^M) such that $z^M \in B$. Let τ_i be the random time of the r^{th} order l-node in the i^{th} such pair. Thus $X_{\tau_1}, X_{\tau_2}, \ldots$ are r^{th} order l-nodes. By the assumptions on y^M we also know that for every i,

$$Y_{\tau_i+r-M+1}^{\tau_i+r} = y^M \quad \text{and} \quad Y_{\tau_i} = l.$$

Hence the two-dimensional process (X, Y) is clearly regenerative with respect to the random times $\{\tau_i\}_{i=1}^\infty$. Moreover, the proper piecewise construction ensures that the Viterbi process V is also regenerative with respect to $\{\tau_i\}$, see (Lember & Koloydenko, 2008). The random variables $\tau_1, \tau_2 - \tau_1, \tau_3 - \tau_2, \ldots$ are independent and $\tau_2 - \tau_1, \tau_3 - \tau_2, \ldots$ are i.i.d. Thus, defining $S_i := \tau_{i+1}$, $i = 0, 1, \ldots$, we obtain that the three-dimensional process $Z = (X, Y, V)$ is regenerative with respect to the delayed renewal process $\{S_t\}_{t=0}^\infty$. Let $\tilde{V}^n := v^n(X^n)$, where v^n is a finite Viterbi alignment. The discussion above can be summarized as the following theorem (see (Kuljus & Lember, 2010) for details).

Theorem 3.1. *Let $(X, Y) = \{(X_t, Y_t)\}_{t=1}^\infty$ be an ergodic HMM satisfying* **A1** *and* **A2**. *Then there exists an infinite Viterbi alignment $v : \mathcal{X}^\infty \to S^\infty$. Moreover, the finite Viterbi alignments $v^n : \mathcal{X}^n \to S^n$ can be chosen so that the following conditions are satisfied:*

R1 *the process $Z := (X, Y, V)$, where $V := \{V_t\}_{t=1}^\infty$ is the alignment process, is a positive recurrent aperiodic regenerative process with respect to some renewal process $\{S_t\}_{t=0}^\infty$;*

R2 *there exists a nonnegative integer $m < \infty$ such that for every $j \geq 0$, $\tilde{V}_t^n = V_t$ for all $n \geq S_j + m$ and $t \leq S_j$.*

We actually know that m relates to r, the order of the barriers in Lemma 3.1, as $m = r + 1$. Aperiodicity of Z follows from aperiodicity of Y, the latter being a consequence of **A1**. Obviously, the choice of v^n becomes an issue only if the finite Viterbi alignment is not unique. In what follows, we always assume that the finite Viterbi alignments $v^n : \mathcal{X}^n \to S^n$ are chosen according to Theorem 3.1. With such choices, the process $\tilde{Z}^n := \{(\tilde{V}_t^n, X_t, Y_t)\}_{t=1}^n$ satisfies by **R2** the following property: $\tilde{Z}_t^n = Z_t$ for every $t = 1, \ldots, S_{k(n)}$, where $k(n) = \max\{k \geq 0 : S_k + m \leq n\}$.

Regenerativity of Z makes it possible to obtain without any remarkable effort the SLLN for $\tilde{Z}^n$. To be more precise, let g_p be a measurable function for some $p \in \mathbb{N}$ and let $n \geq p$, and consider the following random variables

$$\tilde{U}_i^n := g_p(\tilde{Z}_{i-p+1}^n, \ldots, \tilde{Z}_i^n), \quad i = p, \ldots, n.$$

Note that if $i \leq S_{k(n)}$, then $\tilde{U}_i^n = U_i := g_p(Z_{i-p+1}, \ldots, Z_i)$. Let

$$M_k := \max_{S_k < i \leq S_{k+1}} |\tilde{U}_{S_k+1}^i + \cdots + \tilde{U}_i^i|.$$

The random variables $M_p, M_{p+1}, \ldots$ are identically distributed, but for $p > 1$ not necessarily independent. The following Theorem in a sense generalizes Th. VI.3.1 in (Asmussen, 2003), and is an important tool for the applications in Subsection 4.1. The process Z^* appearing in the theorem is a stationary version of Z. For the proof and details see (Kuljus & Lember, 2010).

Theorem 3.2. *Let g_p be such that $EM_p < \infty$ and $E|g_p(Z_1^*, \ldots, Z_p^*)| < \infty$. Then we have*

$$\frac{1}{n-p+1} \sum_{i=p}^{n} \tilde{U}_i^n \underset{n\to\infty}{\to} EU_p = Eg_p(Z_1^*, \ldots, Z_p^*) \quad \text{a.s. and in } L_1.$$

3.1.6 Doubly-infinite HMMs

Recall that $\{(X_t, Y_t)\}_{t\geq 1}$ is a restriction of the doubly-infinite HMM $\{X_t, Y_t\}_{t\in\mathbb{Z}}$ to the positive integers. A great advantage of the barrier-based approach is that it allows us to construct a piecewise infinite Viterbi alignment also for the doubly-infinite HMM. Thus, there exists a doubly-infinite Viterbi alignment $v : \mathcal{X}^{\mathbb{Z}} \to S^{\mathbb{Z}}$ that is an extension of finite Viterbi alignments. For the formal definition of a doubly-infinite alignment see (Kuljus & Lember, 2010). An important feature of the doubly-infinite Viterbi alignment is that the decoding process v is stationary, i.e. shifting the realization $x_{-\infty}^{\infty}$ by one time-unit (Bernoulli shift) entails the same shift of the decoded sequence $v(x_{-\infty}^{\infty})$. Hence, applying v to an ergodic doubly-infinite process X gives us an ergodic doubly-infinite Viterbi process $v(X)$. The following theorem (Th. 2.2 in (Kuljus & Lember, 2010)) is a doubly-infinite counterpart of Theorem 3.1.

Theorem 3.3. *Let $(X, Y) = \{(X_t, Y_t)\}_{t\in\mathbb{Z}}$ be a doubly-infinite ergodic HMM satisfying* **A1** *and* **A2**. *Then there exists an infinite Viterbi alignment $v : \mathcal{X}^{\mathbb{Z}} \to S^{\mathbb{Z}}$. Moreover, the finite Viterbi alignments $v_{z_1}^{z_2}$ can be chosen so that the following conditions are satisfied:*

RD1 *the process (X, Y, V), where $V := \{V_t\}_{t\in\mathbb{Z}}$ is the alignment process, is a positively recurrent aperiodic regenerative process with respect to some renewal process $\{S_t\}_{t\in\mathbb{Z}}$;*

RD2 *there exists a nonnegative integer $m < \infty$ such that for every $j \geq 0$, $\tilde{V}_t^n = V_t$ for all $n \geq S_j + m$ and $S_0 \leq t \leq S_j$;*

RD3 *the mapping v is stationary, i.e. $v(\theta(X)) = \theta v(X)$, where θ is the usual shift operator, i.e. $\theta(\ldots, x_{-1}, x_0, x_1, \ldots) = (\ldots, x_0, x_1, x_2, \ldots)$.*

Note the difference between **R2** and **RD2**. Also, as explained above, property **RD3** is important because it guarantees that the doubly-infinite alignment process $V = \{V_t\}_{t\in\mathbb{Z}}$ as well as $Z = \{(X_t, Y_t, V_t)\}_{t\in\mathbb{Z}}$ is ergodic. Hence, by Birkhoff's ergodic theorem it holds that for any integrable function f,

$$\frac{1}{n} \sum_{t=1}^{n} f(\ldots, Z_{t-1}, Z_t, Z_{t+1}, \ldots) \underset{n\to\infty}{\to} E[f(\ldots, Z_{-1}, Z_0, Z_1, \ldots)] \quad \text{a.s. and in } L_1. \tag{19}$$

The convergence (19) is an important tool in proving limit theorems. Let Z^* denote the restriction of $\{(X_t, Y_t, V_t)\}_{t=-\infty}^{\infty}$ to the positive integers, i.e. $Z^* = \{(X_t, Y_t, V_t)\}_{t=1}^{\infty}$. By **RD2**, Z^* is a stationary version of Z as in **R1**. Thus $(X_0, Y_0, V_0) \overset{D}{=} (X_1^*, Y_1^*, V_1^*) := Z_1^*$. Note that the singly-infinite Viterbi process V in **R1** is not defined at time zero so that the random variable V_0 always refers to the doubly-infinite, and hence stationary, case.

3.1.7 Two-state HMM

The two-state HMM is special in many ways. First, since the underlying Markov chain is aperiodic and irreducible, all entries of $\mathbb{P}^2$ are positive. This implies that the cluster-assumption **A1** is always satisfied. Clearly the models of interest have only one cluster consisting of both states. Positiveness of the transition probabilities also suggests that there is no real need to consider either higher order nodes (and therefore higher order barriers) or nodes that are not strong. That all makes the analysis for the case $K = 2$ significantly simpler. The two-state case was first considered in (Caliebe, 2006; Caliebe & Rösler, 2002), where the existence of infinite Viterbi alignments and regenerativity of the Viterbi process were proved under several additional assumptions. The proof is based on the central limit theorem and cannot be extended beyond the two-state case (see (Koloydenko & Lember, 2008; Lember & Koloydenko, 2008) for a detailed discussion). The barrier-based construction for two-state HMMs was considered in detail in (Koloydenko & Lember, 2008). The main result of this paper states that for $K = 2$ also the assumption **A2** can be removed. The only assumption that remains is the natural assumption that the emission measures P_1 and P_2 are different. The main theorem of (Koloydenko & Lember, 2008) states that under this assumption almost every realization of X has infinitely many strong barriers. This result significantly generalizes those in (Caliebe & Rösler, 2002).

3.2 Exponential forgetting and infinite PMAP-alignment

The existence of an infinite PMAP-classifier follows from the convergence of the so-called *smoothing probabilities* as detailed below: for every $s \in S$, $t, z \in \mathbb{Z}$ such that $t \geq z$, we have

$$\mathbf{P}(Y_t = s|X_z, \ldots, X_n) \underset{n\to\infty}{\to} \mathbf{P}(Y_t = s|X_z, X_{z+1} \ldots) =: \mathbf{P}(Y_t = s|X_z^\infty) \quad \text{a.s.} \tag{20}$$

The convergence (20) in its turn follows from Levy's martingale convergence Theorem. When the model is such that for every t there exists s' satisfying

$$\mathbf{P}(Y_t = s'|X^\infty) > \mathbf{P}(Y_t = s|X^\infty), \quad \forall s \neq s' \quad \text{a.s.}, \tag{21}$$

then the existence of infinite PMAP-alignment follows from (20) with $z = 1$, because then

$$\arg\max_s \mathbf{P}(Y_t = s|X^n) = \arg\max_s \mathbf{P}(Y_t = s|X^\infty) \quad \text{eventually,} \quad \text{a.s.}$$

Condition (21) guarantees that $\arg\max_s \mathbf{P}(Y_t = s|X^\infty)$ is almost surely unique. The drawback of the easy construction of the infinite PMAP-alignment (given (21) holds) is that the ergodic properties of the PMAP-process still need to be established. In particular, an analogue of Theorem 3.2 has not yet been established, although we conjecture that under some assumptions it holds.

At the same time, employing Levy's martingale convergence Theorem again, we have

$$\lim_{z\to-\infty} \mathbf{P}(Y_t = s|X_z^\infty) = \mathbf{P}(Y_t = s|\cdots, X_{-1}, X_0, X_1, \ldots) =: \mathbf{P}(Y_t = s|X_{-\infty}^\infty) \quad \text{a.s.}$$

In (Lember, 2011), the rates of the above convergences are studied. In particular, the following *exponential forgetting* Theorem (Th. 2.1 in (Lember, 2011)) is proved. In this Theorem, for every z_1, z_2 such that $-\infty \leq z_1 < z_2 \leq \infty$, $\mathbf{P}(Y_t \in \cdot|X_{z_1}^{z_2})$ denotes the K-dimensional vector of probabilities and $\|\cdot\|$ stands for the total variation distance.

Theorem 3.4. *Assume* **A1**. *Then there exists a finite random variable C and $\rho \in (0,1)$ such that for every z, t, n satisfying $z \leq t \leq n$,*

$$\|\mathbf{P}(Y_t \in \cdot | X_z^n) - \mathbf{P}(Y_t \in \cdot | X_{-\infty}^{\infty})\| \leq C(\rho^{t-z} + \rho^{n-t}) \quad \text{a.s.} \tag{22}$$

The proof of Theorem 3.4 is based on an approach developed in (Cappé et al., 2005). The approach is based on the fact that given x^n, the conditional distribution of the underlying chain Y is still Markov (albeit generally inhomogeneous). Using this, the difference between the smoothing probabilities can be bounded by the Dobrushin coefficient of the product of the (data-dependent) transition matrices. Condition **A1** allows us to bound the Dobrushin coefficient also in the case when the strong mixing condition fails. This is why Theorem 3.4 is more general than the previous similar results where the transition matrix was assumed to have only positive entries or the emission densities f_i were assumed to be all positive (Cappé et al., 2005; Gerencser & Molnar-Saska, 2002; Gland & Mevel, 2000). It is important to note that although the technique used in proving the exponential forgetting inequality differs completely from the one used in proving the infinite Viterbi alignment, the same assumption **A1** appears in both the situations. This gives us a reason to believe that **A1** is indeed essential for HMMs.

4. Applications of infinite segmentation

4.1 Asymptotic risks

Recall (Subsection 1.2) that the quantity $R(g, x^n) := R(g(x^n)|x^n)$ measures the quality of classifier g when it is applied to observations x^n. We are interested in the random variable $R(g, X^n)$. In particular, we ask whether there exists a constant R such that $R(g, X^n) \underset{n\to\infty}{\to} R$ *almost surely*. This constant, when it exists, will be called *asymptotic risk* and for a given risk function, its asymptotic risk depends only on the model and the classifier. Therefore, asymptotic risks can be used to characterize the long-run properties of different classifiers for a given HMM. They provide a tool for comparing how well different segmentation methods work for a particular model. In the following, we present some risk convergence results that were originally proved in (Kuljus & Lember, 2010; Lember, 2011). We also give the main ideas behind the proofs. It should be noted that although the risk-based approach allows us to consider several segmentation methods in a unified framework, we are not aware of any unified method for proving the convergence of the corresponding risks. Therefore, every specific risk as well as any particular classifier requires individual treatment. In the following, we will denote the Viterbi alignment by v and the PMAP-alignment by u.

4.1.1 The R_1-risk

The R_1-risk is based on the pointwise loss function L_1 that was defined in (1). When measuring the goodness of segmentation with the R_1-risk, the quantity of actual interest is the so-called *empirical or true* risk

$$R_1(g, Y^n, X^n) := \frac{1}{n}\sum_{t=1}^{n} l(Y_t, g_t(X^n)),$$

where $g_t(X^n)$ is the t^{th} element of the n-dimensional vector $g(X^n)$. Since Y^n is hidden, the empirical risk $R_1(g, Y^n, X^n)$ cannot be found. If g is the Viterbi classifier, then

$$R_1(v, Y^n, X^n) = \frac{1}{n}\sum_{t=1}^{n} l(Y_t, \tilde{V}_t^n),$$

and from Theorem 3.2 it follows that

$$R_1(v, Y^n, X^n) \underset{n\to\infty}{\to} El(Y_0, V_0) =: R_1 \quad \text{a.s. and in } L_1\,. \tag{23}$$

The risk $R_1(v, X^n)$ is the conditional expectation of the empirical risk, i.e.

$$R_1(v, X^n) = E[R_1(v, Y^n, X^n)|X^n].$$

In (Kuljus & Lember, 2010) it is shown that (23) implies also convergence of the conditional expectations. Let us summarize this as the following Theorem (Th. 5 in (Kuljus & Lember, 2010)).

Theorem 4.1. *Let $\{(Y_t, X_t)\}_{t=1}^{\infty}$ be an ergodic HMM satisfying* **A1** *and* **A2**. *Then there exists a constant $R_1 \geq 0$ such that*

$$\lim_{n\to\infty} R_1(v, Y^n, X^n) = \lim_{n\to\infty} R_1(v, X^n) = R_1 \quad \textit{a.s. and in } L_1.$$

From the convergence in L_1 (or by the bounded convergence Theorem) it obviously follows that the expected risk of the Viterbi alignment converges to R_1 as well: $ER_1(v, X^n) \to R_1$.
Assuming that the asymptotic risk R_1 has been found (by simulations, for example), one could now be interested in a large deviation type upper bound on $\mathbf{P}(R_1(v, Y^n, X^n) - R_1 > \epsilon)$. In (Ghosh et al., 2009) it has been shown that under the same assumptions as in the present paper, the following large deviation principle holds:

$$\lim_{n\to\infty} \frac{1}{n} \log \mathbf{P}(R_1(v, Y^n, X^n) > \epsilon + R_1) = -I(R_1 + \epsilon),$$

where I is a rate function and ϵ is small enough. The authors of (Ghosh et al., 2009) do not state the exact bound on the probability $\mathbf{P}(R_1(v, Y^n, X^n) - R_1 > \epsilon)$, but it could be derived from their proof of the above result. We would like to draw the reader's attention to how this theme is different from supervised learning. In supervised learning (pattern recognition) the model is often unknown, but the variables Y^n are observable, thus the empirical risk $R_1(g, Y^n, X^n)$ for any classifier could be calculated. The main object of interest then is the unknown asymptotic risk and the large deviation inequalities are used to estimate the unknown asymptotic risk by the known empirical risk. In our setting the data Y^n are hidden, but the model, and therefore the asymptotic risk, is known, thus it can be used to estimate the unknown empirical risk.
Consider now the R_1-risk for the PMAP-classifier u, that is the minimizer of this risk. Birkhoff's ergodic theorem together with the exponential smoothing inequality (22) immediately imply the existence of a constant R_1^* such that $R_1(u, X^n) \to R_1^*$ almost surely. Indeed, from (19) it follows that

$$\frac{1}{n}\sum_{t=1}^{n} \min_{s} \Big(\sum_{a\in S} l(a, s) P(Y_t = a|X_{-\infty}^{\infty}) \Big) \underset{n\to\infty}{\to} E \min_{s} \Big(\sum_{a\in S} l(a, s) P(Y_0 = a|X_{-\infty}^{\infty}) \Big) =: R_1^* \quad \text{a.s.}$$

The forgetting bound (22) yields

$$\Big| R_1(u, X^n) - \frac{1}{n}\sum_{t=1}^{n} \min_{s\in S} \Big(\sum_{a\in S} l(a, s) P(Y_t = a|X_{-\infty}^{\infty}) \Big) \Big| \leq \frac{C}{n} \sum_{t=1}^{n} (\rho^{t-1} + \rho^{n-t}) \quad \text{a.s.,} \tag{24}$$

see (Lember, 2011) for details. The right-hand side of (24) converges to zero almost surely as n grows. Thus, the following Theorem holds (Th. 3.1 in (Lember, 2011)).

Theorem 4.2. *Let $\{(Y_t, X_t)\}_{t=1}^{\infty}$ be an ergodic HMM satisfying* **A1**. *Then there exists a constant R_1^* such that $R_1(u, X^n) \underset{n\to\infty}{\to} R_1^*$ a.s. and in L_1.*

The asymptotic risk R_1^* measures in the long run the average loss incurred by classifying one symbol. Since the PMAP-classifier is optimal for the R_1-risk, then clearly $R_1^* \leq R_1$ and their difference indicates how well the Viterbi segmentation performs in the sense of R_1-risk in comparison to the best classifier in the sense of R_1-risk. For example, if the pointwise loss function l is symmetric, then the optimal classifier in the sense of misclassification error makes on average about $R_1^* n$ classification mistakes and no other classifier does better.

4.1.2 The $\bar{R}_1$-risk

Recall (2) which defines the $\bar{R}_1$-risk to be

$$\bar{R}_1(s^n|x^n) = -\frac{1}{n}\sum_{t=1}^{n} \log p_t(s_t|x^n).$$

To show the convergence of $\bar{R}_1(v, X^n)$, we use Theorem 3.3. According to **RD3**, the doubly-infinite alignment process v is stationary. Consider the function $f : \mathcal{X}^{\mathbb{Z}} \to (-\infty, 0]$, where

$$f(x_{-\infty}^{\infty}) := \log p_0\big(v_0(x_{-\infty}^{\infty})|x_{-\infty}^{\infty}\big) = \log \mathbf{P}(Y_0 = V_0|X_{-\infty}^{\infty} = x_{-\infty}^{\infty}).$$

It is not hard to see that $f\big(\theta^{(t)}(x_{-\infty}^{\infty})\big) = \log \mathbf{P}(Y_t = V_t|X_{-\infty}^{\infty} = x_{-\infty}^{\infty})$. Thus, by (19),

$$-\frac{1}{n}\sum_{t=1}^{n} \log \mathbf{P}(Y_t = V_t|X_{-\infty}^{\infty}) \underset{n\to\infty}{\to} -E\big(\log \mathbf{P}(Y_0 = V_0|X_{-\infty}^{\infty})\big) =: \bar{R}_1 \quad \text{a.s. and in } L_1,$$

provided the expectation is finite. This convergence suggests that by suitable approximation the following convergence also holds:

$$\lim_{n\to\infty} \bar{R}_1(v, X^n) = \lim_{n\to\infty} -\frac{1}{n}\sum_{t=1}^{n} \log \mathbf{P}(Y_t = \tilde{V}_t^n|X^n) = \bar{R}_1 \quad \text{a.s.} \tag{25}$$

The difficulties with proving the convergence of $\bar{R}_1(v, X^n)$ are caused mainly by the fact that the exponential forgetting inequality in (22) does not necessarily hold for the logarithms. This inequality would hold if the probability $\mathbf{P}(Y_0 = V_0|X_{-\infty}^{\infty})$ were bounded below, i.e. if

$$\mathbf{P}(Y_0 = V_0|X_{-\infty}^{\infty}) > \epsilon \quad \text{a.s.} \tag{26}$$

held for some $\epsilon > 0$. Then by (22) it would hold that almost surely $\mathbf{P}(Y_t = V_t|X^{\infty}) > \frac{\epsilon}{2}$ eventually, and the inequality $|\log a - \log b| \leq \frac{1}{\min\{a,b\}}|a-b|$ together with (22) would imply

$$-\frac{1}{n}\sum_{t=1}^{n} \log \mathbf{P}(Y_t = V_t|X^n) \underset{n\to\infty}{\to} \bar{R}_1 \quad \text{a.s.}$$

Then, by an argument similar to the one in the proof of Theorem 3.2, the convergence (25) would follow. Unfortunately, (26) need not necessarily hold. In (Kuljus & Lember, 2010) the condition (26) is replaced by the following weaker condition: there exists $\alpha > 0$ such that

$$E\Big(\frac{1}{\mathbf{P}(Y_0 = V_0|X_{-\infty}^{\infty})}\Big)^{\alpha} < \infty\,. \tag{27}$$

It can be shown (Prop. 4.1 and Lemma 3 in (Kuljus & Lember, 2010)) that under **A1** the inequality (27) holds. The condition in (27) turns out to be sufficient to prove the convergence of $\bar{R}_1(v, X^n)$. The discussion above can be summarized in the following theorem (Th. 4.1 in (Kuljus & Lember, 2010)).

Theorem 4.3. *Let* $\{(Y_t, X_t)\}_{t=1}^{\infty}$ *be an ergodic HMM satisfying* **A1** *and* **A2**. *Then*

$$\lim_{n\to\infty} \bar{R}_1(v, X^n) = \bar{R}_1 \quad \textit{a.s. and in } L_1 .$$

From the preceding argument it is clear that the convergence of the $\bar{R}_1$-risk is rather easy to prove when instead of the Viterbi alignment the PMAP-alignment is used. Indeed, by (19),

$$-\frac{1}{n}\sum_{t=1}^{n} \max_{s\in S} \log \mathbf{P}(Y_t = s|X_{-\infty}^{\infty}) \underset{n\to\infty}{\to} E[\max_{s\in S} \log \mathbf{P}(Y_0 = s|X_{-\infty}^{\infty})] =: \bar{R}_1^* \quad \text{a.s. and in } L_1.$$

Since $\max_{s\in S} \mathbf{P}(Y_t = s|X^n) \geq K^{-1}$, for the PMAP-alignment the condition (26) is trivially satisfied. From the exponential forgetting inequality (22) it then follows (Cor. 4.2 in (Kuljus & Lember, 2010)) that

$$\bar{R}_1(u, X^n) = -\frac{1}{n}\sum_{t=1}^{n} \max_{s\in S} \log \mathbf{P}(Y_t = s|X^n) \underset{n\to\infty}{\to} \bar{R}_1^* \quad \text{a.s. and in } L_1 .$$

Again, since the $\bar{R}_1$-risk is minimized by the PMAP-classifier, it holds that $\bar{R}_1^* \leq \bar{R}_1$.

4.1.3 The $\bar{R}_\infty$-risk

Recall (Subsection 2.3.1) that the $\bar{R}_\infty$-risk is defined as the negative log-posterior probability given observations x^n, i.e. $\bar{R}_\infty(s^n|x^n) = -\frac{1}{n}\log p(s^n|x^n)$. Let $p(x^n)$ denote the likelihood of x^n. Then

$$p(\tilde{V}^n|X^n) = \mathbf{P}(Y^n = \tilde{V}^n|X^n) = \frac{p(X^n|\tilde{V}^n)\mathbf{P}(Y^n = \tilde{V}^n)}{p(X^n)},$$

therefore

$$\bar{R}_\infty(v, X^n) = -\frac{1}{n}\Big(\log p(X^n|\tilde{V}^n) + \log \mathbf{P}(Y^n = \tilde{V}^n) - \log p(X^n)\Big).$$

By Theorem 3.2, the following convergences hold (see (Kuljus & Lember, 2010) for details):

$$\frac{1}{n}\log p(X^n|\tilde{V}^n) \underset{n\to\infty}{\to} \sum_{s\in S} E\big(\log f_s(X_0) I_s(V_0)\big) \quad \text{a.s.,} \qquad \frac{1}{n}\log \mathbf{P}(Y^n = \tilde{V}^n) \underset{n\to\infty}{\to} E(\log p_{V_0 V_1}) \quad \text{a.s.}$$

The last convergence $-\frac{1}{n}\log p(X^n) \to H_X$, where H_X is the entropy rate of X, follows from the Shannon-McMillan-Breiman Theorem. The ideas above are formalized in the following Theorem (Th. 5.1 in (Kuljus & Lember, 2010)).

Theorem 4.4. *Let for every* $s \in S$ *the function* $\log f_s$ *be* P_s*-integrable. Then there exists a constant* $\bar{R}_\infty$ *such that*

$$\bar{R}_\infty(v, X^n) \underset{n\to\infty}{\to} \bar{R}_\infty \quad \textit{a.s. and in } L_1.$$

By the same argument, there exists another constant $\bar{R}^Y_\infty$ such that

$$-\frac{1}{n}\log \mathbf{P}(Y^n|X^n) \underset{n\to\infty}{\to} \bar{R}^Y_\infty \quad \text{a.s. and in } L_1.$$

Since $E[\log \mathbf{P}(Y^n|X^n)] = -H(Y^n|X^n)$, where $H(Y^n|X^n)$ stands for the conditional entropy of Y^n given X^n, the limit $\bar{R}^Y_\infty$ could be interpreted as the conditional entropy rate of Y given X, it is not the entropy rate of Y. Clearly, $\bar{R}_\infty \leq \bar{R}^Y_\infty$, and the difference of these two numbers shows how much the Viterbi alignment inflates the posterior probability.

4.2 Adjusted Viterbi training

So far we have assumed that the model is known, i.e. both the transition matrix as well as the emission distributions P_s are given. Often the model is given up to parametrization and then parameter estimation becomes of interest. Hence, in this subsection we assume that all emission densities are of the form $f_s(x;\theta_s)$, where $\theta_s \in \Theta_s$ is the emission parameter to be estimated. In practice, e.g. in speech recognition, the transition matrix is often assumed to be known and the emission parameters are the only parameters to be estimated, sometimes however the transition matrix $\mathbb{P} = (p_{ij})$ is to be estimated as well. Thus, in general, the set of unknown parameters is $\psi = (\mathbb{P}, \theta)$, where $\theta = (\theta_1, \theta_2, \ldots, \theta_K)$. (We ignore π, the initial distribution, since in the stationary regime π is determined by $\mathbb{P}$, whereas otherwise its estimation would require multiple samples x^n.)

The classical algorithm for finding the maximum likelihood estimators of HMM-parameters is the so-called *EM-training* (see, e.g. (Cappé et al., 2005; Ephraim & Merhav, 2002; Rabiner, 1989)). Although theoretically justified, the EM-training might be very slow and computationally expensive. Therefore, in practice, the EM-training is sometimes replaced by the much quicker *Viterbi training* (VT), where the expectation over all alignments (E-step) is replaced by the maximum a posteriori alignment. In other words, in the k^{th} iteration the Viterbi alignment is performed using $\psi^{(k)}$, the current estimate of the parameters. According to this alignment, the observations x^n are divided into K subsamples, where the s^{th} subsample consists of those observations that are aligned with the state s. In each subsample the maximum likelihood estimator $\hat{\mu}_s$ of θ_s is found. The estimate of the transition probability $\hat{p}_{ij}$ is the proportion of states i followed by the state j in the Viterbi alignment. The formal algorithm of VT estimation is as follows.

Viterbi training (VT)

1. Choose initial values for the parameters $\psi^{(k)} = (\mathbb{P}^{(k)}, \theta^{(k)})$, $k = 0$.
2. Given the current parameters $\psi^{(k)}$, obtain the Viterbi alignment $v^{(k)} = v(x^n; \psi^{(k)})$.
3. Update the regime parameters $\mathbb{P}^{(k+1)} := \left(\hat{p}^n_{ij}\right)$, $i, j \in S$, as given below:

$$\hat{p}^n_{ij} := \begin{cases} \dfrac{\sum\limits_{m=1}^{n-1} I_{\{i\}}(v^{(k)}_m) I_{\{j\}}(v^{(k)}_{m+1})}{\sum\limits_{m=1}^{n-1} I_{\{i\}}(v^{(k)}_m)}, & \text{if } \sum\limits_{m=1}^{n-1} I_{\{i\}}(v^{(k)}_m) > 0\,, \\ \mathbb{P}^{(k)}_{ij}, & \text{otherwise.} \end{cases}$$

4. Assign x_m, $m = 1, 2, \ldots, n$, to the class $v^{(k)}_m$. Equivalently, define empirical measures

$$\hat{P}^n_s(A; \psi^{(k)}, x^n) := \frac{\sum_{m=1}^n I_{A\times\{s\}}(x_m, v^{(k)}_m)}{\sum_{m=1}^n I_{\{s\}}(v^{(k)}_m)}, \quad A \in \mathcal{B}, \quad s \in S.$$

5. For each class $s \in S$, obtain the ML estimator $\hat{\mu}_s^n(\psi^{(k)}, x^n)$ of θ_s, given by:

$$\hat{\mu}_s^n(\psi^{(k)}, x^n) := \arg\max_{\theta_s' \in \Theta_s} \int \log f_s(x; \theta_s') \hat{P}_s^n(dx; \psi^{(k)}, x^n),$$

$$\text{and for all } s \in S \text{ let } \theta_s^{(k+1)} := \begin{cases} \hat{\mu}_s^n(\psi^{(k)}, x^n), & \text{if } \sum_{m=1}^{n} I_{\{s\}}(v_m^{(k)}) > 0\,, \\ \theta_s^{(k)}, & \text{otherwise.} \end{cases}$$

For better interpretation of VT, suppose that at some step k, $\psi^{(k)} = \psi$, thus $v^{(k)}$ is obtained using the true parameters. Let y^n be the actual hidden realization of Y^n. The training pretends that the alignment $v^{(k)}$ is perfect, i.e. $v^{(k)} = y^n$. If the alignment were perfect, the empirical measures $\hat{P}_s^n$, $s \in S$, would be obtained from the i.i.d. samples generated from the true emission measures P_s and the ML estimators $\hat{\mu}_s^n$ would be natural estimators to use. Under these assumptions $\hat{P}_s^n \Rightarrow P_s$ *almost surely*, and provided that $\{f_s(\cdot; \theta_s) : \theta_s \in \Theta_s\}$ is a P_s-Glivenko-Cantelli class and Θ_s is equipped with a suitable metric, we would have $\lim_{n\to\infty} \hat{\mu}_s^n = \theta_s$ *almost surely*. Hence, if n is sufficiently large, then $\hat{P}_s^n \approx P_s$ and $\theta_s^{(k+1)} = \hat{\mu}_s^n \approx \theta_s = \theta_s^{(k)}$ for every $s \in S$. Similarly, if the alignment were perfect, then $\lim_{n\to\infty} \hat{p}_{ij}^n = \mathbf{P}(Y_2 = j | Y_1 = i) = p_{ij}$ *almost surely*. Thus, for the perfect alignment

$$\psi^{(k+1)} = (\mathbb{P}^{(k+1)}, \theta^{(k+1)}) \approx (\mathbb{P}^{(k)}, \theta^{(k)}) = \psi^{(k)} = \psi\,,$$

i.e. ψ would be approximately a fixed point of the training algorithm.

Certainly the Viterbi alignment in general is not perfect even when it is computed with the true parameters. The empirical measures $\hat{P}_s^n$ can be rather far from those based on the i.i.d. samples from the true emission measures P_s even when the Viterbi alignment is performed with the true parameters. Hence we have no reason to expect that $\lim_{n\to\infty} \hat{\mu}_s^n(\psi, X^n) = \theta_s$ and $\lim_{n\to\infty} \hat{p}_{ij}^n(\psi, X^n) = p_{ij}$ *almost surely*. Moreover, we do not even know whether the sequences of empirical measures $\hat{P}_s^n(\psi, X^n)$ or the ML estimators $\hat{\mu}_s^n(\psi, X^n)$ and $\hat{p}_{ij}^n(\psi, X^n)$ converge *almost surely* at all. Here again Theorem 3.2 answers the question. From Theorem 3.2 it follows that for any measurable set A, $\hat{P}_s^n(A) \to \mathbf{P}(X_0 \in A | V_0 = s) =: Q_s(A)$ a.s., where $\hat{P}_s^n = \hat{P}_s^n(\psi, X^n)$. This implies that the empirical measures $\hat{P}_s^n$ converge weakly to the measures Q_s *almost surely*, i.e. for every $s \in S$,

$$\hat{P}_s^n \Rightarrow Q_s \quad \text{a.s.} \tag{28}$$

Convergence (28) is the main statement of Theorem 4.1 in (Lember & Koloydenko, 2008). In (Koloydenko et al., 2007) it has been shown that if $f_s(x; \theta_s)$ satisfy some general conditions and if Θ_s are closed subsets of $\mathbb{R}^d$, then convergence (28) implies convergence of $\hat{\mu}_s^n(\psi, X^n)$, i.e.

$$\hat{\mu}_s^n(\psi, X^n) \underset{n\to\infty}{\to} \mu_s \text{ a.s.,} \quad \text{where } \mu_s(\psi) := \arg\max_{\theta_s' \in \Theta_s} \int \log f_s(x; \theta_s') Q_s(dx). \tag{29}$$

Since in general $Q_s \neq P_s(\theta_s)$, clearly μ_s need not equal $\theta_s = \arg\max_{\theta_s'} \int \log f_s(x; \theta_s') P_s(dx)$.

Similarly, Theorem 3.2 also implies that

$$\hat{p}_{ij}^n(\psi; X^n) \underset{n\to\infty}{\to} \mathbf{P}(V_1 = j | V_0 = i) =: q_{ij} \quad \text{a.s.} \tag{30}$$

Again, in general $p_{ij} \neq q_{ij}$. In order to reduce the biases $\theta_s - \mu_s$ and $p_{ij} - q_{ij}$, we have proposed the *adjusted Viterbi training*. We know that convergences (29) and (30) hold for any parameter ψ given that **A1** and **A2** hold. Since the limits μ_s and q_{ij} depend on the true parameters, we can consider the mappings

$$\psi \mapsto \mu_s(\psi), \quad \psi \mapsto q_{ij}(\psi), \quad s,i,j = 1,\ldots,K. \tag{31}$$

These mappings do not depend on the observations x^n, hence the following corrections are well-defined:

$$\Delta_s(\psi) := \theta_s - \mu_s(\psi), \quad R_{ij}(\psi) := p_{ij} - q_{ij}(\psi), \quad s,i,j = 1,\ldots,K. \tag{32}$$

Based on (32), the *adjusted Viterbi training* can be defined as follows.

Adjusted Viterbi training (VA)

1. Choose initial values for the parameters $\psi^{(k)} = (\mathbb{P}^{(k)}, \theta^{(k)})$, $k = 0$.
2. Given the current parameters $\psi^{(k)}$, obtain the Viterbi alignment $v^{(k)} = v(x^n; \psi^{(k)})$.
3. Update the regime parameters $\mathbb{P}^{(k+1)} := (p_{ij}^{(k+1)})$ as follows:

   $$p_{ij}^{(k+1)} := \hat{p}_{ij}^n + R_{ij}(\psi^{(k)}),$$

   where $\hat{p}_{ij}^n$ is defined as in VT.
4. Based on $v^{(k)}$, define empirical measures $\hat{P}_s^n$, $s \in S$, as in VT.
5. Update the emission parameters as follows:

   $$\theta_s^{(k+1)} := \Delta_s(\psi^{(k)}) + \begin{cases} \hat{\mu}_s^n(\psi^{(k)}, x^n), & \text{if } \sum_{m=1}^n I_{\{s\}}(v_m^{(k)}) > 0, \\ \theta_s^{(k)}, & \text{otherwise.} \end{cases}$$

   Here $\hat{\mu}_s^n(\psi^{(k)}, x^n)$ is as in VT.

Provided n is sufficiently large, VA has approximately the true parameters ψ as its fixed point as desired. Indeed, suppose $\psi^{(k)} = \psi$. From (29) we obtain that for every $s \in S$,

$$\hat{\mu}_s^n(\psi^{(k)}, x^n) = \hat{\mu}_s^n(\psi, x^n) \approx \mu_s(\psi) = \mu_s(\psi^{(k)}).$$

Similarly, (30) gives that for all $i, j \in S$,

$$\hat{p}_{ij}^n(\psi^{(k)}, x^n) = \hat{p}_{ij}^n(\psi, x^n) \approx q_{ij}(\psi) = q_{ij}(\psi^{(k)}).$$

Thus, for every $s, i, j \in S$,

$$\begin{aligned} \theta_s^{(k+1)} &= \hat{\mu}_s^n(\psi, x^n) + \Delta_s(\psi) \approx \mu_s(\psi) + \Delta_s(\psi) = \theta_s = \theta_s^{(k)}, \\ p_{ij}^{(k+1)} &= \hat{p}_{ij}^n(\psi, x^n) + R_{ij}(\psi) \approx q_{ij}(\psi) + R_{ij}(\psi) = p_{ij} = p_{ij}^{(k)}. \end{aligned}$$

Hence, $\psi^{(k+1)} = (\mathbb{P}^{(k+1)}, \theta^{(k+1)}) \approx (\mathbb{P}^{(k)}, \theta^{(k)}) = \psi^{(k)}$. The simulations in (Koloydenko et al., 2007; Lember & Koloydenko, 2007), presented in part in Example 2 below, show that the asymptotic fixed point property does make a difference. Namely, unlike the VT estimates, the VA ones are nearly as accurate (and can even be more accurate than) as the ones obtained by the EM-training. At the same time, VA is comparable to VT in terms of the computational cost, and therefore may be preferred to EM.

4.2.1 Example 2

The following simulation study is adapted from (Koloydenko et al., 2007). We consider a two-state HMM with the transition matrix

$$\mathbb{P} = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}, \quad \epsilon \in (0, 0.5],$$

and with the emission distributions $P_1 = \mathcal{N}(\theta_1, 1)$ and $P_2 = \mathcal{N}(\theta_2, 1)$. Thus, there are two emission parameters θ_1 and θ_2 and one regime parameter ϵ in this model. Assume without loss of generality that $\theta_1 < \theta_2$ and let $a = 0.5(\theta_2 - \theta_1)$. With $\epsilon = 0.5$ this model reduces to the i.i.d. (mixture) model. The correction function $\Delta(a, \epsilon)$ was estimated off-line by simulations and achieves its maximum at $\epsilon = 0.5$, i.e. in the i.i.d. case. Using the obtained Δ-function, we apply the adjusted Viterbi training and compare it with the VT- and EM-algorithms. Tables 1-2 present simulation results obtained from samples of size 10^6 and focus on estimation of the emission parameters. The iterations were initialized by setting $\theta_1^{(0)}$ and $\theta_2^{(0)}$ to the first and third quartiles of $x_1, x_2, \ldots, x_n$, respectively, and stopped as soon as the L_∞-distance between successive estimates fell below 0.01. From Tables 1-2 it can be seen that the Viterbi training is quickest to converge, but its estimates are evidently biased. Accuracy of the adjusted Viterbi training is comparable to that of the EM-algorithm, while VA converges somewhat more rapidly than EM. Each step of EM requires significantly more intensive computations, so that one should expect the overall run-time of VA to be notably shorter than that of EM.

Using the same stopping rule as before, we also test the three algorithms for the fixed point property. From Tables 3-4 it is evident that both EM and VA do approximately satisfy this property, whereas VT moves the true parameters to a notably different location.

	EM	VT	VA
Step 0	(-0.689,0.687)	(-0.689,0.687)	(-0.689,0.687)
Step 1	(-0.477,0.475)	(-0.537,0.536)	(-0.460,0.459)
Step 2	(-0.385,0.384)	(-0.474,0.474)	(-0.359,0.358)
Step 3	(-0.335,0.333)	(-0.445,0.445)	(-0.305,0.307)
Step 4	(-0.303,0.301)	(-0.429,0.430)	(-0.273,0.274)
Step 5	(-0.281,0.279)	(-0.420,0.422)	(-0.252,0.254)
Step 6	(-0.265,0.264)		(-0.239,0.241)
Step 7	(-0.253,0.252)		(-0.229,0.232)
Step 8	(-0.244,0.243)		
L_1 error	0.087	0.442	0.061
L_2 error	0.061	0.312	0.043
L_∞ error	0.044	0.222	0.032

Table 1. Estimating θ_1 and θ_2, when $\epsilon = 0.2$, $a = 0.2$, $\theta_1 = -0.2$ and $\theta_2 = 0.2$.

	EM	VT	VA
Step 0	(-1.050,1.053)	(-1.050,1.053)	(-1.050,1.053)
Step 1	(-1.013,1.015)	(-1.166,1.169)	(-1.014,1.016)
Step 2	(-1.003,1.005)	(-1.165,1.169)	(-1.004,1.006)
L_1 error	0.008	0.334	0.010
L_2 error	0.006	0.236	0.007
L_∞ error	0.005	0.169	0.006

Table 2. Estimating θ_1 and θ_2, when $\epsilon = 0.5$, $a = 1$, $\theta_1 = -1$ and $\theta_2 = 1$.

	EM	VT	VA
Step 0	(-0.200,0.200)	(-0.200,0.200)	(-0.200,0.200)
Step 1	(-0.198,0.202)	(-0.252,0.254)	(-0.198,0.200)
Step 2		(-0.298,0.302)	
Step 3		(-0.333,0.339)	
Step 4		(-0.357,0.367)	
Step 5		(-0.373,0.386)	
Step 6		(-0.383,0.399)	
Step 7		(-0.387,0.408)	
L_1 error	0.003	0.396	0.002
L_2 error	0.002	0.280	0.002
L_∞ error	0.002	0.208	0.002

Table 3. Comparison of algorithms for $\epsilon = 0.2$ and $a = 0.2$, and $\theta_1^{(0)} = \theta_1$ and $\theta_2^{(0)} = \theta_2$.

	EM	VT	VA
Step 0	(-1.000,1.000)	(-1.000,1.000)	(-1.000,1.000)
Step 1	(-0.998,1.000)	(-1.165,1.167)	(-0.998,1.000)
Step 2		(-1.165,1.167)	
L_1 error	0.002	0.332	0.002
L_2 error	0.002	0.235	0.002
L_∞ error	0.002	0.167	0.002

Table 4. Comparison of algorithms for $\epsilon = 0.5$ and $a = 1$, and $\theta_1^{(0)} = \theta_1$ and $\theta_2^{(0)} = \theta_2$.

4.3 Generalizations, other training ideas and implementation

4.3.1 Segmentation-based training

As we have seen above, the drawback of VT stems from the fact that the Viterbi process differs systematically from the underlying chain, so that the empirical measures obtained by the Viterbi segmentation can differ significantly from the true emission distributions P_s even when the parameters used to obtain the Viterbi alignment were correct and n were arbitrarily large. Hence, using the Viterbi alignment for segmentation in the training procedure is not theoretically justified.

Since the PMAP-alignment minimizes the error rate, using the PMAP-segmentation in the training procedure could be the lesser of the two evils. The empirical measures obtained by the PMAP-alignment would, of course, also differ from the emission measures even when the parameters are correct and n is arbitrarily large, and in particular the transition probability estimators can easily be biased. However, since the PMAP-alignment has more correctly estimated states in the long run, the emission estimators $\hat{\mu}_n$ obtained by the PMAP-alignment are expected to be closer to the ML estimators that would have been obtained if the underlying state sequence were known. A tandem training that would synergize the Viterbi and PMAP alignments may also be worth considering.

In general, one can speak about *segmentation-based training*, where the observations are divided into K subsamples (empirical measures) according to a segmentation procedure with current parameter estimates. Every subsample is considered to be an i.i.d. sample from P_s and the corresponding MLE is found. The transition probabilities are directly obtained from the segmentation. Once again, the PMAP-training should give more precise estimates of the emission parameters than the Viterbi training, but the PMAP-alignment might produce forbidden transitions (Section 1.2). Thus, even when a transition probability is set to zero initially, or turns zero at some iteration, it would not necessarily remain zero at later iterations of the PMAP training. This is different from VT which cannot turn a zero transition probability positive. This more liberal behavior of the PMAP-training can easily be constrained "by hand", which would be appropriate when, for example, the forbidden transitions are known a priori. More generally, the k-block alignment defined in Subsection 2.4 could be considered in the training procedure which would automatically preserve zero transition probabilities. Preservation of zero transition probabilities is not necessarily a goal in itself as it can prevent the algorithm from detecting rare transitions. However, using the k-block alignment for segmentation based training is worth considering as further optimization may be achieved by varying k.

Recall that the adjusted Viterbi training is largely based on Theorem 3.2, since this is the main theoretical result behind the existence of the adjustments in (32). Although not yet proved, we believe that a counterpart of Theorem 3.2 holds for many alignments other than Viterbi. Now, if the training is based on an alignment for which the above result does hold, the adjusted version of the training can then be defined along the lines of the Viterbi training.

4.3.2 Independent training

The order of the observations $x^n = (x_1, \ldots, x_n)$ provides information about the transition probabilities. When we reorder the observations, we loose all information about the transitions, but the information about the emission distributions remains. Often the emission parameters are the primary interest of the training procedure, the transition matrix could for example be known or considered to be a nuisance parameter. Then it makes sense to estimate the emission parameters by treating the observations $x_1, \ldots, x_n$ as an i.i.d. sample from a mixture density $\sum_s \pi_s f_s(\cdot; \theta_s)$, assuming π to be the invariant distribution of $\mathbb{P}$. This approach is introduced in (Koloydenko et al., 2007; Lember & Koloydenko, 2008) under the name of *independent training*. Besides the EM-training the Viterbi training can also be used for data that is regarded as independent, and in this case it is equivalent to the PMAP-training. As shown in (Koloydenko et al., 2007) (see Subsection 4.2.1), the bias Δ_s is relatively large for the i.i.d. case, which makes the replacement of VT by VA particularly attractive in this case. The advantage of the VA-based independent training over VA is that training is usually significantly easier in the i.i.d. case. In particular, the adjustment terms Δ_s are more likely to be found theoretically. Also, the i.i.d. case is usually computationally much cheaper. Another appealing procedure that could be applied for independent training is VA2 that will be described next. For a more detailed discussion about independent training, see (Koloydenko et al., 2007; Lember & Koloydenko, 2008). For VA in the i.i.d. case, see (Lember & Koloydenko, 2007).

4.3.3 VA2

For the case of i.i.d. data from a mixture density $\sum_s \pi_s f_s(\cdot; \theta_s)$, a slightly modified version of VA was proposed in (Lember & Koloydenko, 2007). To explain the main idea, assume that the weights π_s are known so that the emission parameters $\theta = (\theta_1, \ldots, \theta_K)$ are the only parameters to be estimated. VA2 is based on the observation that for the i.i.d. case the segmentation of the data into subsamples is induced by a partition of the sample space. Indeed, given the current estimates $\theta^{(k)}$, the observation x_t belongs to the subsample corresponding to state 1 if and only if $x_t \in \mathcal{S}_1 := \{x : \pi_1 f_1(x; \theta_1^{(k)}) \geq \pi_s f_s(x; \theta_s^{(k)}),\ \forall s \in S\}$. The sets $\mathcal{S}_1, \ldots, \mathcal{S}_K$ form a partition of X (upto the ties) that depends only on $\theta^{(k)}$. It is intuitively clear, especially in the case of $K = 2$, that many different parameters θ could induce the same partition. In particular, $\theta^{(k)}$ could induce the partition corresponding to the true parameter θ^* even when $\theta^{(k)} \neq \theta^*$. In that case, the ML estimates $\hat{\mu}_s$ would be the same for both $\theta^{(k)}$ and θ^*. However, since the correction term $\Delta(\theta^{(k)})$ does depend on $\theta^{(k)}$, it follows that the adjusted estimate $\theta^{(k+1)}$ need not be close to θ^*. The adjustment in VA2 tries to overcome the mentioned deficiency. In particular, the parameters $\theta^{(k)}$ are taken into account via their induced partition only. Given the partition and the ML estimates $\hat{\mu}_s$, the new adjustment seeks θ that would asymptotically induce the given partition and the given estimates. Now, if the partition corresponded to θ^*, then $\theta^{(k+1)}$ obtained in such a way would be close to θ^*. For details, see (Lember & Koloydenko, 2007).

4.3.4 Implementation

The difficulties in implementing VA are caused by the fact that apart from the i.i.d. case, finding the adjustment functions $\Delta(\psi)$ theoretically is very hard. However, since the adjustments do not depend on the data, they can be found by simulations independently of the data. It is important to point out that even if such simulations require significant effort, they are done *off-line* and can be reused with the same model.

Another, computationally less demanding approach, is the so called *stochastically adjusted Viterbi training* (SVA). Instead of estimating the correction at every point as in the previous approach, SVA estimates the correction by simulations at every iteration and therefore only at the points visited by the algorithm. Clearly, if the number of iterations is relatively small, this method should require less overall computing. On the other hand, if a model is to be used repeatedly, estimating the correction function off-line as in the previous example might still be preferable.

Several implementation ideas for the i.i.d. case, i.e. for estimating mixture parameters, are discussed in (Lember & Koloydenko, 2007). The implementation of VA2 depends on the model. Instead of calculating the correction function Δ, for VA2 a certain inverse function should be found. This might be difficult to do even for simple models, but when it is done, it can be reused again and again.

4.4 Segmentation with partially revealed observations

Consider the situation where some hidden states can be revealed on request, albeit possibly at a very high cost. The purpose of uncovering a number of states is to improve the alignment by reducing the number of incorrectly estimated states. With the additional information a constrained alignment can be obtained, which in general will lower the empirical risk considerably. The decision on how many and which states to reveal is a trade-off between the cost of learning an unknown state and the reduction in the alignment risk.

One way to approach the problem of which states to reveal is to study the conditional misclassification probabilities at every time point $t = 1, \ldots, n$, given the observations $X_1, \ldots, X_n$. One can order the calculated conditional probability $\mathbf{P}(Y_t \neq g_t(X^n)|X^n = x^n)$, $t = 1, \ldots, n$, and ask for the actual states of the points with the largest probability. This approach involves finding the alignment and computing the conditional probabilities for every single realization. In order to use this approach, one needs to know the conditional probability of incorrect segmentation given a certain observation, or a segment of observations, in advance. Let us denote the conditional misclassification probability given an observation x by $P(incorrect|x)$.

4.4.1 Definition of misclassification probability for Viterbi alignment

In this section $l(a_t, b_t)$ stands for the symmetric pointwise loss: $l(a_t, b_t) = 1$ if $a_t \neq b_t$ and 0 otherwise. Thus the R_1-risk measures the expected number of misclassified observations. Recall that (X_0, Y_0, V_0) belongs to the stationary version of (X_t, Y_t, V_t). Define for every measurable A,

$$P^{correct}(A) := \mathbf{P}(X_0 \in A | Y_0 = V_0)\,,$$
$$P^{incorrect}(A) := \mathbf{P}(X_0 \in A | Y_0 \neq V_0)\,.$$

The probability measure $P^{(in)correct}$ can be interpreted as the asymptotic distribution of an observation given the Viterbi alignment at that point is (in)correct. Because of stationarity of X the distribution of every observation is given by

$$\begin{aligned} P(A) &= \mathbf{P}(X_0 \in A) = \mathbf{P}(X_0 \in A|Y_0 = V_0)\mathbf{P}(Y_0 = V_0) + \mathbf{P}(X_0 \in A|Y_0 \neq V_0)\mathbf{P}(Y_0 \neq V_0) \\ &= P^{correct}(A)(1 - R_1) + P^{incorrect}(A)R_1, \end{aligned}$$

where R_1 is the asymptotic risk as defined in Subsection 4.1.1. Thus, the probability $P(incorrect|\cdot)$ can be defined as follows:

$$P(incorrect|A) := \mathbf{P}(Y_0 \neq V_0 | X_0 \in A) = \frac{P^{incorrect}(A)R_1}{P^{correct}(A)(1 - R_1) + P^{incorrect}(A)R_1}\,.$$

The probability distribution of any observation of X can be written as a weighted sum of emission distributions P_s: $P_X = \sum_{s \in S} \pi_s P_s$. Because the emission distributions P_s have densities f_s with respect to some measure μ, from the equality $P(A) = P^{correct}(A)(1 - R_1) + P^{incorrect}(A)R_1$ it follows that $P^{(in)correct}$ have densities with respect to μ, we denote them by $f^{(in)correct}$. The conditional probability that the Viterbi alignment makes a mistake given that the observation is x, can now be defined as

$$P(incorrect|x) := \mathbf{P}(Y_0 \neq V_0 | x) = \frac{f^{incorrect}(x)R_1}{f^{correct}(x)(1 - R_1) + f^{incorrect}(x)R_1}. \tag{33}$$

Observe that $P(incorrect|x)$ depends only on the model. This implies that once the alignment error probability for a given x is estimated, we can use this value whenever working with the same model. It is also important to emphasize that $P(incorrect|x)$ is a function of both $f^{correct}$ and $f^{incorrect}$, thus it takes into account both the proportions of correctly and incorrectly classified states that emit x. For example, if $f^{incorrect}(x_t)$ and $f^{incorrect}(x_u)$ are both large but $f^{correct}(x_t)$ is much smaller than $f^{correct}(x_u)$, then it makes more sense to seek more information on Y at time t.

One way to estimate $P(incorrect|x)$ is from simulations by using empirical measures. If we would know the true underlying states $Y_1, \ldots, Y_n$ for a given sequence of observations $X_1, \ldots, X_n$, we could after performing the Viterbi segmentation calculate the number of correctly and incorrectly classified states. We could also tally (in)correctly classified states with emissions in A. Thus, we can consider empirical measures $P_n^{correct}$ and $P_n^{incorrect}$ defined as follows:

$$P_n^{correct}(A) := \frac{\sum_{t=1}^n I_{A\times\{0\}}(X_t, l(Y_t, v_t(X^n)))}{\sum_{t=1}^n I_{\{0\}}(l(Y_t, v_t(X^n)))} = \frac{\sum_{t=1}^n I_A(X_t) I_{\{Y_t = \tilde{V}_t^n\}}}{\sum_{t=1}^n I_{\{Y_t = \tilde{V}_t^n\}}}\,,$$

$$P_n^{incorrect}(A) := \frac{\sum_{t=1}^n I_{A\times\{1\}}(X_t, l(Y_t, v_t(X^n)))}{\sum_{t=1}^n I_{\{1\}}(l(Y_t, v_t(X^n)))} = \frac{\sum_{t=1}^n I_A(X_t) I_{\{Y_t \neq \tilde{V}_t^n\}}}{\sum_{t=1}^n I_{\{Y_t \neq \tilde{V}_t^n\}}}\,.$$

Similarly, we can define the empirical measure $P_n(incorrect|\cdot)$ that calculates the proportion of classification errors given the observation belongs to $A \in \mathcal{B}$:

$$P_n(incorrect|A) := \frac{\sum_{t=1}^n I_{A\times\{1\}}(X_t, l(Y_t, \tilde{V}_t^n))}{\sum_{t=1}^n I_A(X_t)} = \frac{\sum_{t=1}^n I_A(X_t) I_{\{Y_t \neq \tilde{V}_t^n\}}}{\sum_{t=1}^n I_A(X_t)}.$$

In practice, the empirical measures defined above are unknown. It follows directly from Theorem 3.2 that the empirical measures $P_n^{(in)correct}$ and $P_n(incorrect|\cdot)$ converge almost surely to $P^{(in)correct}$ and $P(incorrect|\cdot)$ respectively, i.e. for every $A \in \mathcal{B}$,

$$P_n^{(in)correct}(A) \to P^{(in)correct}(A), \qquad P_n(incorrect|A) \to P(incorrect|A) \quad a.s.$$

These convergences allow us to estimate the densities $f^{(in)correct}$, R_1, and hence also $P(incorrect|x)$, when it is difficult to find any of these quantities analytically.

Example 3. This example demonstrates estimation of $f^{correct}$, $f^{incorrect}$ and $P(incorrect|x)$ by simulations. A two-state HMM with emission distributions $\mathcal{N}(3, 2^2)$ and $\mathcal{N}(10, 3^2)$ and transition matrix

$$\mathbb{P} = \begin{pmatrix} 0.3 & 0.7 \\ 0.7 & 0.3 \end{pmatrix}$$

was considered. The estimates of the densities $f^{correct}$, $f^{incorrect}$ and $P(incorrect|x)$ for a sample of size $n = 100000$ are presented in Figure 4 graphs (a), (b) and (c), respectively. The R-package 'HiddenMarkov' (Harte, 2010) was used for these simulations.

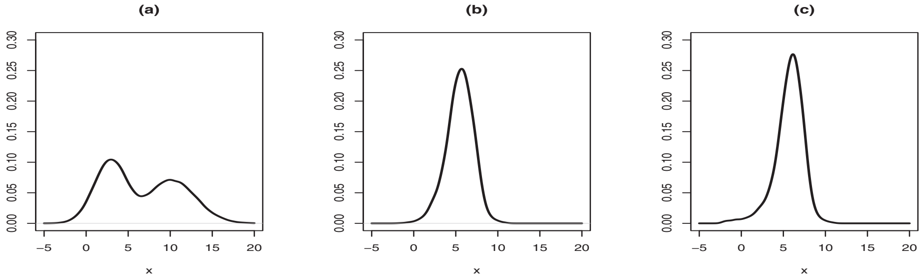


Fig. 4. Estimates of $f^{correct}$, $f^{incorrect}$ and $P(incorrect|x)$.

In (Raag, 2009), the decrease in the number of Viterbi alignment errors when a number of true states are uncovered, is compared for the following three cases: states are revealed randomly, the states with largest conditional point risk are uncovered, the states with the largest misclassification error are revealed. The simulation studies in (Raag, 2009) investigate for example, how the number of mistakes of the constrained alignments depends on the transition probabilities or dependence between the states, and how the decrease in the number of errors is affected by the number of states in the model.

4.4.2 Generalization

Thus far, we have defined the misclassification probability conditionally given a single observation on X. Stationarity makes this probability time invariant. Now we are going to generalize this definition and take into account also information from the neighbors of X. We

will consider a $(2k+1)$-tuple of observations $X_{t-k}, \ldots, X_t, \ldots, X_{t+k}$, $k > 0$. In the following, the tuples $X^k_{-k}, Y^k_{-k}, V^k_{-k}$ are from the doubly-infinite and hence stationary process Z.
Let $A_1 \times \ldots \times A_{2k+1} \in \mathcal{B}^{2k+1}$. By analogy with the single observation case, for $(2k+1)$-tuples of observations we can define the following measures:

$$\begin{aligned} P^{incorrect}(A_1 \times \ldots \times A_{2k+1}) &= \mathbf{P}(X_{-k} \in A_1, \ldots, X_k \in A_{2k+1} | Y_0 \neq V_0) \\ P^{correct}(A_1 \times \ldots \times A_{2k+1}) &= \mathbf{P}(X_{-k} \in A_1, \ldots, X_k \in A_{2k+1} | Y_0 = V_0) \\ P(incorrect|A_1 \times \ldots \times A_{2k+1}) &= \mathbf{P}(Y_0 \neq V_0 | X_{-k} \in A_1, \ldots, X_k \in A_{2k+1}) \end{aligned}$$

Clearly, the decomposition

$$\begin{aligned} \mathbf{P}(X_{-k} \in A_1, \ldots, X_k \in A_{2k+1}) = \; & P^{incorrect}(A_1 \times \ldots \times A_{2k+1}) R_1 \\ & + P^{correct}(A_1 \times \ldots \times A_{2k+1})(1 - R_1) \end{aligned} \tag{34}$$

holds. Since the random variables X_i have densities with respect to μ, it follows that the vector X^k_{-k} has the density with respect to the product measure μ^{2k+1}. From (34) it now follows that the measures $P^{(in)correct}$ have densities $f^{(in)correct}$ with respect to μ^{2k+1} as well so that (33) generalizes as follows:

$$P(incorrect|x^{2k+1}) := \mathbf{P}(Y_0 \neq V_0 | X^k_{-k} = x^{2k+1}) = \frac{f^{incorrect}(x^{2k+1}) R_1}{f^{correct}(x^{2k+1})(1 - R_1) + f^{incorrect}(x^{2k+1}) R_1}.$$

The probability $P(incorrect|x^{2k+1})$ is the asymptotic conditional misclassification probability given the neighbors. It is interesting to note that for some neighborhood this probability can be bigger than 0.5, see Figure 5. Obviously, as in the single observation case, the probability $P(incorrect|x^{2k+1})$ could be estimated by simulation. For this, one can define the empirical measures

$$P^{correct}_n(A_1 \times \ldots \times A_{2k+1}) = \frac{\sum_{t=k+1}^{n-k} I_{\{A_1 \times \ldots \times A_{2k+1}\} \times \{0\}}(X_{t-k}, \ldots, X_{t+k}, l(Y_t, \tilde{V}^n_t))}{\sum_{t=k+1}^{n-k} I_{\{0\}}(l(Y_t, \tilde{V}^n_t))},$$

$$P^{incorrect}_n(A_1 \times \ldots \times A_{2k+1}) = \frac{\sum_{t=k+1}^{n-k} I_{\{A_1 \times \ldots \times A_{2k+1}\} \times \{1\}}(X_{t-k}, \ldots, X_{t+k}, l(Y_t, \tilde{V}^n_t))}{\sum_{t=k+1}^{n-k} I_{\{1\}}(l(Y_t, \tilde{V}^n_t))},$$

$$P_n(incorrect|A_1 \times \ldots \times A_{2k+1}) = \frac{\sum_{t=k+1}^{n-k} I_{\{A_1 \times \ldots \times A_{2k+1}\} \times \{1\}}(X_{t-k}, \ldots, X_{t+k}, l(Y_t, \tilde{V}^n_t))}{\sum_{t=k+1}^{n-k} I_{\{A_1 \times \ldots \times A_{2k+1}\}}(X_{t-k}, \ldots, X_{t+k})}.$$

From Theorem 3.2 it follows again that the empirical measures converge to the corresponding theoretical ones at every Borel set almost surely (hence the measures converge weakly almost surely). Therefore, the densities $f^{(in)correct}$ as well as the probabilities $P(incorrect|x^{2k+1})$ could be estimated.

Example 4. This example illustrates how the misclassification error $P(incorrect|x^{2k+1})$, $k = 1$, depends on the transition probabilities. We consider a two-state HMM, where the process X can take on the values $1, 2, 3, 4$. The transition probability matrix and the emission distributions are as follows:

$$\mathbb{P} = \begin{pmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{pmatrix}, \quad P_1 = (1/2, 1/8, 1/8, 1/4), \quad P_2 = (1/5, 1/5, 1/10, 1/2).$$

For each value of ϵ a chain of $n = 10000$ observations was simulated and the misclassification probabilities $P(incorrect|111)$ and $P(incorrect|141)$ were estimated. To estimate the standard deviations of the estimators, the simulations were replicated 100 times. In Figure 5, the estimated probabilities are plotted together with their $+/-$ one standard deviation bands.

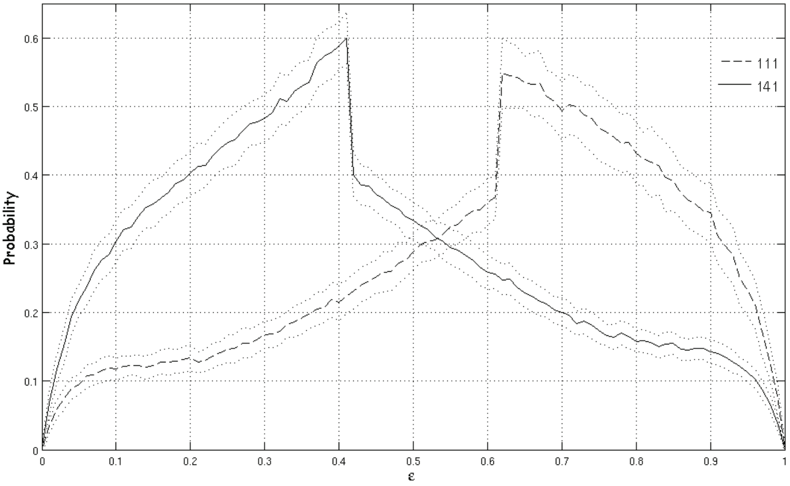


Fig. 5. Estimates of $P(incorrect|111)$ and $P(incorrect|141)$.

Example 5. In the previous example, $P(incorrect|141) \approx 0.6$ for $\epsilon = 0.41$, see Figure 5. Now we consider the HMM of Example 4 for $\epsilon = 0.41$ and study how $P(incorrect|141)$ is affected when we intervene in the Viterbi segmentation process with an increasing intensity. Let m denote the number of occurrences of the word 141 in the simulated process. Then, for example, the intensity 0.2 means that we would intervene in classification the process at $0.2m$ sites. The following four types of interventions were studied:

1) at uniformly distributed random times t, $\tilde{V}_t^n$ was replaced by the opposite state;
2) at uniformly distributed random times t, $\tilde{V}_t^n$ was replaced by the true state Y_t;
3) at the times of occurrence of 141, $\tilde{V}_t^n$ was replaced by the opposite state;
4) at the times of occurrence of 141, $\tilde{V}_t^n$ was replaced by the true state Y_t.

For each thereby constrained Viterbi segmentation, the error rate – the proportion of misclassified states of the constrained alignment – was computed. The results are plotted in Figure 6. The most interesting is of course to see, how the number of Viterbi alignment errors decreases depending on how many true states are revealed.

5. Acknowledgment

J. Lember is supported by Estonian Science Foundation grant 7553.

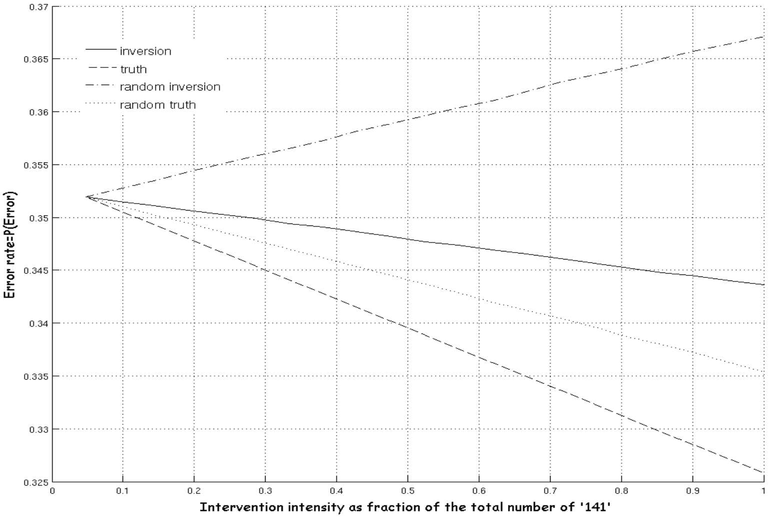


Fig. 6. Misclassification probability as a function of intervention rate.

6. References

Asmussen, S. (2003). *Applied Probability and Queues*, Springer.

Bahl, L., Cocke, J., Jelinek, F. & Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate (Corresp.), *IEEE Trans. Inform. Theory* 20(2): 284–287.

Bishop, C. (2006). *Pattern Recognition and Machine Learning*, Springer, New York.

Brejová, B., Brown, D. & Vinař, T. (2007). The most probable annotation problem in HMMs and its application to bioinformatics, *J. Comput. Syst. Sci.* 73(7): 1060 – 1077.

Brejová, B., Brown, D. & Vinař, T. (2008). Advances in hidden Markov models for sequence annotation, *in* I. Măndoiu & A. Zelikovsky (eds), *Bioinformatics Algorithms: Techniques and Applications*, Wiley, chapter 4, pp. 55–92.

Brown, D. & Truszkowski, J. (2010). New decoding algorithms for Hidden Markov Models using distance measures on labellings, *BMC Bioinformatics* 11(Suppl 1): S40.

Brushe, G., Mahony, R. & Moore, J. (1998). A soft output hybrid algorithm for ML/MAP sequence estimation, *IEEE Trans. Inform. Theory* 44(7): 3129–3134.

Caliebe, A. (2006). Properties of the maximum a posteriori path estimator in hidden Markov models, *IEEE Trans. Inform. Theory* 52(1): 41–51.

Caliebe, A. & Rösler, U. (2002). Convergence of the maximum a posteriori path estimator in hidden Markov models, *IEEE Trans. Inform. Theory* 48(7): 1750–1758.

Cappé, O., Moulines, E. & Rydén, T. (2005). *Inference in Hidden Markov Models*, Springer Series in Statistics, Springer, New York.

Carvalho, L. & Lawrence, C. (2008). Centroid estimation in discrete high-dimensional spaces with applications in biology, *PNAS* 105(9): 3209–3214.
URL: *http://www.pnas.org/content/105/9/3209.abstract*

Chigansky, P. & Ritov, Y. (2010). On the Viterbi process with continuous state space, *Bernoulli*. To appear. URL: *http://arxiv.org/abs/0909.2139v1*

Ephraim, Y. & Merhav, N. (2002). Hidden Markov processes, *IEEE Trans. Inform. Theory* 48(6): 1518–1569. Special issue on Shannon theory: perspective, trends, and applications.

Fariselli, P., Martelli, P. & Casadio, R. (2005). 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.

Forney, G. (1973). The Viterbi algorithm, *Proc. IEEE* 61(3): 268–278.

Genon-Catalot, V., Jeantheau, T. & Larédo, C. (2000). Stochastic volatility models as hidden Markov models and statistical applications, *Bernoulli* 6(6): 1051–1079.

Gerencser, L. & Molnar-Saska, G. (2002). A new method for the analysis of Hidden Markov model estimates, *Proceedings of the 15th IFAC World Congress*, Vol. 15.

Ghosh, A., Kleiman, E. & Roitershtein, A. (2009). Large deviation bounds for functionals of Viterbi paths. Accepted with revision in IEEE Trans. Inform. Theory.
URL: *http://www.public.iastate.edu*

Gland, F. L. & Mevel, L. (2000). Exponential forgetting and geometric ergodicity in hidden Markov models, *Math. Control Signals Systems* 13: 63 – 93.

Harte, D. (2010). *Package 'Hidden Markov'*, Statistics Research Associates, Wellington, New Zealand.
URL: *http://cran.at.r-project.org/web/packages/HiddenMarkov*

Hayes, J., Cover, T. & Riera, J. (1982). Optimal sequence detection and optimal symbol-by-symbol detection: similar algorithms, *IEEE Transactions on Communications* 30(1): 152–157.

Holmes, I. & Durbin, R. (1998). Dynamic programming alignment accuracy, *J. Comput. Biol.* 5(3): 493–504.

Käll, L., Krogh, A. & Sonnhammer, E. (2005). An HMM posterior decoder for sequence feature prediction that includes homology information, *Bioinformatics* 21(Suppl. 1): i251–i257.

Koloydenko, A., Käärik, M. & Lember, J. (2007). On adjusted Viterbi training, *Acta Appl. Math.* 96(1-3): 309–326.

Koloydenko, A. & Lember, J. (2008). Infinite Viterbi alignments in the two state hidden Markov models, *Acta Comment. Univ. Tartu. Math.* (12): 109–124. Proc. 8th Tartu Conf. Multivariate Statist. June 2007.

Koloydenko, A. & Lember, J. (2010). Hidden path inference, *Technical report*, Mathematics Department, Royal Holloway, University of London. http://personal.rhul.ac.uk/utah/113/pfinds/index.html.

Krogh, A. (1997). Two methods for improving performance of an HMM and their application for gene finding, *ISMB-97 Proceedings*, AAAI, pp. 179–186.

Kuljus, K. & Lember, J. (2010). Asymptotic risks of Viterbi segmentation, http://arxiv.org/abs/1002.3509. submitted.

Lember, J. (2011). On approximation of smoothing probabilities for hidden Markov models, *Statistics and Probability Letters* . To appear.

Lember, J. & Koloydenko, A. (2007). Adjusted Viterbi training: A proof of concept, *Probab. Eng. Inf. Sci.* 21(3): 451–475.

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

Lember, J. & Koloydenko, A. (2010a). A constructive proof of the existence of Viterbi processes, *IEEE Trans. Inform. Theory* 56(4): 2017–2033.

Lember, J. & Koloydenko, A. (2010b). A generalized risk-based approach to segmentation based on hidden Markov models, http://arxiv.org/abs/1007.3622v1. submitted. URL: *http://arxiv.org/abs/1007.3622v1*

Leroux, B. (1992). Maximum-likelihood estimation for hidden Markov models, *Stochastic Process. Appl.* 40(1): 127–143.

Raag, M. (2009). Risk and errors of Viterbi alignment, Master Thesis. in Estonian.

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

Robertson, P., Villebrun, E. & Hoeher, P. (1995). A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain, *ICC '95 Seattle, 'Gateway to Globalization', 1995 IEEE International Conference on Communications*, Vol. 2, pp. 1009–1013.

Rue, H. (1995). New loss functions in Bayesian imaging, *J. Am. Stat. Assoc.* 90(431): 900–908. URL: *http://www.jstor.org/stable/2291324*

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, *IEEE Trans. Inform. Theory* 13(2): 260–269.

Winkler, G. (2003). *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods. A Mathematical Introduction*, Vol. 27 of *Applications of Mathematics (New York)*, Springer-Verlag, Berlin.