APPROXIMATE VITERBI DECODING FOR 2D-HIDDEN MARKOV MODELS

Bernard Merialdo, Stéphane Marchand-Maillet and Benoit Huet

Institut Eurecom
2229 Route des cretes,
06904 Sophia-Antipolis, France.
Bernard.Merialdo@eurecom.fr

ABSTRACT

While one-dimensional Hidden Markov Models have been very successfully applied to numerous problems, their extension to two dimensions has been shown to be exponentially complex, and this has very much restricted their usage for problems such as image analysis.

In this paper we propose a novel algorithm which is able to approximate the search for the best state path (Viterbi decoding) in a 2D HMM. This algorithm makes certain assumptions which lead to tractable computations, at a price of loss in full optimality. We detail our algorithm, its implementation, and present some experiments on handwritten character recognition.

Because the Viterbi algorithm serves as a basis for many applications, and 1D HMMs have shown great flexibility in their usage, our approach has the potential to make 2D HMMs as useful for 2D data as 1D HMMs are for 1D data such as speech.

1. INTRODUCTION

Hidden Markov Models (HMM) have long been used to efficiently model uni-dimensional data (sequences of symbols), in particular in speech recognition systems. While it is tempting to benefit from this efficiency for the modeling of two-dimensional data such as images, efforts to achieve this have been limited by the increase in complexity that occurs when going from one to two dimensions.

In previous work [6], using Hidden Markov Models for image analysis, it was shown that these tools allowed for flexibility in designing image models. However, these models also exhibited a lack of geometrical consistency which generated problems for creating 'semi-rigid' models. This was due to the fact that the two-dimensional structure of the image was taken at two successive levels (i.e. pixel and line levels).

The work of Levin and Pieraccini [5] introduced the idea of planar hidden markov models where the complexity of the problem is reduced by applying image alignment constraints. Miller and Hunt [12] described a sub-optimal 2D Viterbi algorithm for bilevel image reconstruction. Their approach is in fact based on a 1D "column-based" Viterbi that uses decision feedback from previous rows. Recently, Li *et al.* [11] proposed an 2D-HMM algorithm for image classification in which HMM parameters are estimated using the EM algorithm and the 2D Viterbi.

In this paper, we present a new approximation of the 2D Viterbi algorithm that allows to compute an approximation of the best path within a 2D HMM, while avoiding the exponential growth in computation. In section 2, we introduce the concept and theory of the procedure. We present implementation details in section 3. We illustrate the use of this procedure in on some simple real world examples in section 4. Finally, in section 5 we conclude on our findings and present our future research efforts.

2. A NOVEL 2D VITERBI PROCEDURE

One-dimensional Hidden Markov Models (1D HMM) are finite state machines which emit sequences of symbols according to a probabilistic mechanism. The state occupied by the machine at time t is a random variable q_t, and the evolution is controlled by the output probability distributions $P[o_t|q_t = s] = P[o_t|s]$ and the transition probabilities $P[q_t = s_j|q_{t-1} = s_i] = P[s_j|s_i]$.

Two-dimensional Hidden Markov Models (2D HMM) can be defined in a similar way. The output observation is an array of symbols o_{xy}, for example the pixels of an image scanned using a line per line ordering, which are emitted based on the current state q_{xy}. The output probability distributions of the model are now $P[o_{xy}|q_{xy} = s]$. Being two dimensions, we expect the transition probability to depend on horizontal and vertical neighbors, such as $P[q_{xy} = s_k|q_{x-1,y} = s_i, q_{x,y-1} = s_j] = P[s_k|s_i, s_j]$, for local context dependency.

It is easy to see that such a model has similar theoretical properties as a 1D HMM, because a 2D HMM is equivalent to a 1D HMM where the states are $(q_{1,y}, q_{2,y}, ...q_{x-1,y}, q_{x,y-1}, ...q_{X,y-1})$. However, such the equivalent model has a number of states which is now N^X, (if N is the number of states in the model and X is the length of a line), and this leads to an exponential increase in the amount of computation that is needed for the regular Baum-Welch and Viterbi algorithms.

2.1. Notations

The two-dimensional data is represented by the array of symbols o_{xy}, for example the pixels of an image. The 2D HMM has states $(s_1, s_2...s_S)$, and probability distributions $b_i(o) = P(o|s_i)$ and $a_{ijk} = P[s_k|s_i, s_j]$. The random variable used to indicate the state used to emit symbol o_{xy} is q_{xy}.

In the intermediate steps of the algorithm, we are interested by the emission of the block of pixels $O_{xy} = \{o_{uv}\ ,\ 1 \leq u \leq x\ ,\ 1 \leq v \leq y\}$ corresponding to the state block $Q_{xy} = \{q_{uv}\ ,\ 1 \leq u \leq x\ ,\ 1 \leq v \leq y\}$. Note that computing $P[O_{xy}|Q_{xy}]$ is straightforward because the states are known (after a state has been chosen as local context for the border pixels).

We will compute the best path for block O_{xy} from the combination of previously obtained values. For that purpose, we need to introduce the following intermediate notations:

- $p = (x, y)$, $q = q_{xy}$, $o = o_{xy}$,
- $p' = (x-1, y-1)$, $q' = q_{x-1,y-1}$, $o' = o_{x-1,y-1}$,
- $p_1 = (x, y-1)$, $q_1 = q_{x,y-1}$, $o_1 = o_{x,y-1}$, $Q_1 = \{q_{uv}\ ,\ 1 \leq u \leq x\ ,\ 1 \leq v \leq y-1\}$,
- $p_2 = (x-1, y)$, $q_2 = q_{x-1,y}$, $o_2 = o_{x-1,y}$, $Q_2 = \{q_{uv}\ ,\ 1 \leq u \leq x-1\ ,\ 1 \leq v \leq y\}$,

We will also need the following:

- $R_{p'}$ is one state configuration at positions $\{(u, y)\ ;\ 1 \leq u < x-1\}$ (i.e. along the row containing $p' = (x-1, y-1)$, until the position $(x-2, y-1)$).
- $C_{p'}$ is one state configuration at positions $\{(x, v)\ ;\ 1 \leq v < y-1\}$ (i.e. along the column containing $p' = (x-1, y-1)$, until the position $(x-1, y-2)$).

These notations are graphically illustrated in figure 1.

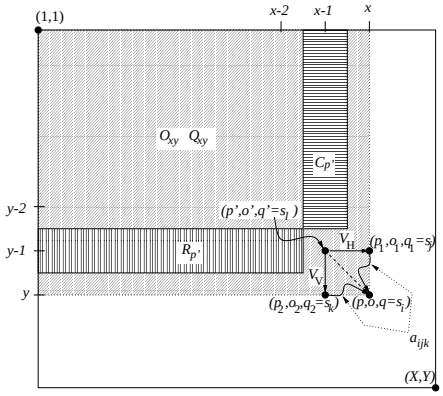

Figure 1: Notation for the 2D Viterbi procedure.

2.2. Approximation

Let us define $V_{xy}(k)$ to be the maximum probability of the emission of output block O_{xy} over all possible state blocks Q_{xy} with $q_{xy} = s_k$. We can decompose

$$V_{xy}(k) = \max_{i,j \leq N} V_{2\mathrm{D}}(x, y, i, j) a_{ijk} b_k(o_{xy}),$$

where $V_{2\mathrm{D}}(x, y, i, j)$ is the maximum probability over all state blocks $Q_{xy} - \{q_{xy}\}$ with $q_{x,y-1} = s_i$ and $q_{x-1,y} = s_j$ corresponding to the emission of pixels in $O_{xy} - \{o_{xy}\}$. In order words, we first produce the output block except the last pixel o_{xy} with states s_i s_j as neighbors for the last pixel, then move in state s_k in position x, y and emit the last pixel o_{xy}.

In turn, $V_{2\mathrm{D}}(x, y, i, j)$ can be decomposed as

$$\begin{aligned} V_{2\mathrm{D}}(x, y, i, j) \quad = \quad & \max_{l \leq N, R_{p'}, C_{p'}} \Big[V_{\mathrm{A}}(p', l, R_{p'}, C_{p'}) \times \\ & T_{2\mathrm{D}}(q_1 = s_i | p', l, R_{p'}, C_{p'}) \times \\ & T_{2\mathrm{D}}(q_2 = s_j | p', l, R_{p'}, C_{p'}) \Big] \end{aligned} \tag{1}$$

where

- $V_{\mathrm{A}}(p', l, R_{p'}, C_{p'})$ is the maximum probability over all state blocks $Q_{x-1,y-1}$ corresponding to the emission of $O_{x-1,y-1}$ with state s_l at position $p' = (x-1, y-1)$ and containing the configurations of states $R_{p'}$ and $C_{p'}$ on its borders.
- $T_{2\mathrm{D}}(q_1 = s_i | p', l, R_{p'}, C_{p'})$ is the maximum probability of over all state blocks $Q_1 = Q_{x,y-1}$ corresponding to $O_{x,y-1}$ with $q_{x,y-1} = s_i$ given the values of s_l at position p', and given that $Q_1 = Q_{x,y-1}$ should comprise the state configurations $R_{p'}$ and $C_{p'}$.

The previous formula relates the construction of the best state block for output block O_{xy} from state blocks corresponding to smaller output blocks $O_{x-1,y-1}$. However, an exact usage of Equation 1 is difficult to implement in practice, since it requires to store all possible configurations for q', $R_{p'}$ and $C_{p'}$.

We therefore introduce the following approximations

$$\begin{aligned} V_{2\mathrm{D}}(x, y, i, j) \quad \simeq \quad & \max_{k \leq N} \Bigg(\left[\max_{R_{p'}, C_{p'}} V_{\mathrm{A}}(p', k, R_{p'}, C_{p'}) \right] \times \\ & \left[\max_{R_{p'}, C_{p'}} T_{2\mathrm{D}}(q_1 = s_i | p', k, R_{p'}, C_{p'}) \right] \times \\ & \left[\max_{R_{p'}, C_{p'}} T_{2\mathrm{D}}(q_2 = s_j | p', k, R_{p'}, C_{p'}) \right] \Bigg) \end{aligned}$$

Let us define $V_{\mathrm{H}}(q_1 = s_i | q' = s_k)$ as the maximum probability over all state blocks Q_1 containing the state $q' = s_k$ at position $p' = (x-1, y-1)$ and the state $q_1 = s_i$ at position $p_1 = (x, y-1)$, while emitting $O_{x,y-1}$.

$$\begin{aligned} V_{\mathrm{H}}(q_1 = s_i | q' = s_k) \quad \simeq \quad & \max_{R_{p'}, C_{p'}} V_{\mathrm{A}}(p', k, R_{p'}, C_{p'}) \times \\ & \max_{R_{p'}, C_{p'}} T_{2\mathrm{D}}(q_1 = s_i | p', k, R_{p'}, C_{p'}) \end{aligned} \tag{2}$$

Now, by definition,

$$\max_{R_{p'}, C_{p'}} V_{\mathrm{A}}(p', k, R_{p'}, C_{p'}) = V_{p'}(k) = V_{x-1,y-1}(k)$$

so that,

$$\max_{R_{p'}, C_{p'}} T_{2\mathrm{D}}(q_1 = s_i | p', k, R_{p'}, C_{p'}) \simeq \frac{V_{\mathrm{H}}(q_1 = s_i | q' = s_k)}{V_{p'}(k)}$$

When detailing the case of $p_2 = (x-1, y)$, we define in the same manner $V_{\mathrm{V}}(q_2 = s_i | q' = s_k)$ as the maximum probability over all state blocks Q_2 containing the state $q' = s_k$ at position $p' = (x-1, y-1)$ and the state $q_2 = s_i$

at position $p_1 = (x-1, y)$, while emitting $O_{x-1,y}$. In this context,

$$\max_{R_{p'}, C_{p'}} T_{2D}(q_2 = s_i | p', k, R_{p'}, C_{p'}) \simeq \frac{V_V(q_2 = s_i | q' = s_k)}{V_{p'}(k)}$$

We therefore obtain,

$$V_{2D}(x,y,i,j) = \max_{k \leq N} \left[\frac{V_H(q_{x,y-1} = s_i | q_{x-1,y-1} = s_k)}{V_{x-1,y-1}(k)} \times \frac{V_V(q_{x-1,y} = s_j | q_{x-1,y-1} = s_k)}{V_{x-1,y-1}(k)} \right]$$

Finally, combining all equations, we obtain the 2D Viterbi recursive formula,

$$V_{xy}(i) = \max_{j,k,l \leq N} \left[\frac{V_H(q_1 = s_j | q' = s_l) V_V(q_2 = s_k | q' = s_l)}{V_{x-1,y-1}(l)} \times P[q_{xy} = s_i | q_{x,y-1} = s_j, q_{x-1,y} = s_k]] P[o_{xy} | q_{xy} = s_i] \right]$$

which can be equivalently rewritten as

$$V_{xy}(i) = \max_{l \leq N} \left[\max_{j,k \leq N} \Psi_{xy}(i,j,k,l) \right] b_i(o_{xy}), \qquad (3)$$

where,

$$\Psi_{xy}(i,j,k,l) = a_{ijk} \times \left[\frac{V_H(q_{x,y-1} = s_j | q_{x-1,y-1} = s_l)}{V_{x-1,y-1}(l)} \times \frac{V_V(q_{x-1,y} = s_k | q_{x-1,y-1} = s_l)}{V_{x-1,y-1}(l)} \right] \qquad (4)$$

These equations now create a link between the resulting most likely state at position $p' = (x-1, y-1)$ and that at position $p = (x, y)$ via the enumeration of all combination of states at positions $p_1 = (x, y-1)$ and $p_2 = (x-1, y)$. Equation 3 and 4 will form the core of the procedure we propose for retrieving the most likely state block from a given output block O.

3. 2D VITERBI PROCEDURE

3.1. Forward pass

We now detail the practical implementation of our 2D Viterbi procedure. A model Λ is represented by the following parameters:

- N, the number of states in the model.
- $\mathcal{V}$, the vocabulary (i.e. the set of all possible values of an observation). $\mathcal{V}$ can possibly be of infinite size.
- $A = \{a_{ijk} = P[q_{xy} = s_i | q_{x,y-1} = s_j, q_{x-1,y} = s_k]\ ;\ 1 \leq i \leq N,\ 1 \leq j \leq N,\ 1 \leq k \leq N\}$, the 2D-transition probability values.
- $B = \{b_i(v) = P[o_{xy} = v | q_{xy} = s_i]\ ;\ 1 \leq i \leq N\ \forall v \in \mathcal{V}\}$, the set of state output probability values.
- $\Pi = \{\pi_i = P[q_{xy} = s_i | x = 1 \text{ or } y = 1]\ ,\ 1 \leq i \leq N\}$, the set of initial state probability which determine the (upper and left) border constraints.

The following intermediate variables are calculated during the forward process,

- $V_{xy}(i)$ is the maximum probability of obtaining a state block Q_{xy} containing the state $q = s_i$ at position $p = (x, y)$, while emitting O_{xy},
- $V_H(q_{x,y-1} = s_j | q_{x-1,y-1} = s_l)$ is the maximum probability of obtaining a state block Q_1 containing the state $q' = s_l$ at position $p' = (x-1, y-1)$ and the state $q_1 = s_j$ at position $p_1 = (x, y-1)$, while emitting $O_{x,y-1}$,
- $V_V(q_{x-1,y} = s_k | q_{x-1,y-1} = s_l)$ is the maximum probability of obtaining a state block Q_2 containing the state $q' = s_l$ at position $p' = (x-1, y-1)$ and the state $q_2 = s_k$ at position $p_2 = (x-1, y)$, while emitting $O_{x-1,y}$,
- $\Psi_{xy}(i,j,k,l)$ is the probability of obtaining one state block Q containing the states s_i, s_j, s_k and s_l at positions $p = (x,y)$, $p_1 = (x, y-1)$, $p_2 = (x-1, y)$ and $p' = (x-1, y-1)$, respectively, while emitting $O_{xy} - \{o_{xy}\}$.

Similarly to the 1D Viterbi procedure, the values of V, V_H, and V_V are defined recursively and calculated online during the forward pass.

3.2. Backward pass

The retrieval of the best state block Q^* which emits the output O is possible via the storage of states which lead to maximal values during the forward pass. For example, we have

$$P[O|\Lambda] = \max_i V_{XY}(i) \qquad (5)$$

so that the index which achieves the maximum defines the final state q^*_{XY} of the best state block. From the final state, we can recover the previous two neighbors (given the final state), then retrieve backwards all states from the last row and column, finally all internal states (in reverse order, given their successors), through the array of indices

$$\Phi_{xy}(i,j,k) = l^* = \arg\max_l [\Psi_{xy}(i,j,k,l)] \, with \ 1 \leq i \leq N, \ 2 \leq x \leq X-1, 2 \leq y \leq Y-1$$

4. EXPERIMENTS

4.1. Segmentation

We have used our procedure to train 2D HMM on images of handwritten characters. One application is the segmentation of such images. In the following example, we use a 5 state 2D HMM, with initial transition probabilities computed from the transition of the five regions of shown in figure 2. Note that this initialization induces some constraints on the boundaries between regions, as certain transition probabilities will be set to zero and will prevent certain combination of state neighborhoods to occur. After training on character images, the best state block provides a segmentation of training images shown in figure 3.

It is visible that the initial constraints produce a model which can easily fit the shapes on first row, but is less adequate for the shapes in the second row, resulting in a sometimes awkward segmentation of these images.

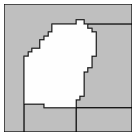

Figure 2: Initial segmentation

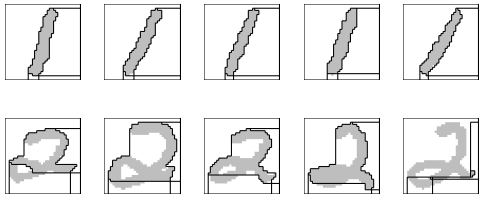

Figure 3: Segmentation using the completed model

4.2. Recognition

In this experiment, we use a portion of the NIST database (see figure 4) to train a generic 10x10 model on a set of instances for each digit from 0 to 9. 10 instances of each figures are taken as training set, to build 10 HMM models Λ_i, $i = 0, \ldots, 9$.

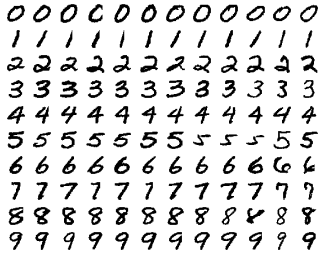

Figure 4: Handwritten character training database

Recognition is performed by searching for the best model which emits an unknown character image

$$n = \arg\max_i P[I|\Lambda_i]$$

As an example, the sequence of handwritten characters shown in figure 5 is recognized as 9 7 7 3 8 6 9 7 2 3.

The system incorrectly identified three of the ten characters. The errors have occured on the first characters "4", "1", "3". When comparing with the database these errors are consistent with the training set (i.e. for example, the first "4" is very similar to some instances of "9").

5. CONCLUSION

We have presented a procedure which approximates the computation of the Viterbi path in a 2D HMM. This opens new possibilities for the modelization of 2D data such as images, using probabilistic procedures which takes into account the local context of each pixel.

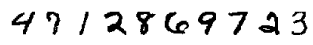

Figure 5: Example of a test figure string

We have illustrated how this technique can be applied to handwritten optical character recognition. In the future, we intend to evaluate the sensitivity and performance of our algorithm on larger problems, and extend other 1D HMM procedures to this 2D HMM framework.

6. REFERENCES

[1] H. Bourlard and N. Morgan. *Connectionist Speech Recognition – A Hybrid Approach*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1994.

[2] A. Kosmala and G. Rigoll. On-line handwritten formula recognition using statistical methods. In *Proceedings of ICPR'98*, pages PR33, 1998.

[3] S.-S. Kuo and O. E. Agazzi. Automatic keyword recognition using Hidden Markov Models. *Journal of Visual Communication and Image Representation*, 5(3):265–272, 1994.

[4] S.-S. Kuo and O. E. Agazzi. Keyword spotting in poorly printed documents using Pseudo 2-D Hidden Markov models. *IEEE PAMI*, 16(8):842–848, 1994.

[5] E. Levin and R Pieraccini. Dynamic planar warping for optical character recognition. In *Proceedings of ICASSP'92*, volume III, pages 149–152, 1992.

[6] S. Marchand-Maillet and B. Merialdo. Stochastic models for face image analysis. In *Proceedings of the 1st European Workshop on Content-Based Multimedia Indexing*, October 1999.

[7] A. P. Pentland, R. W. Picard and S. Sclaroff. Photobook: Content-Based Manipulation of Image Databases. *International Journal of Computer Vision*, 18(3):233-254, June 1996.

[8] R. W Picard. A Society of Models for Video and Image Libraries. *IBM Systems Journal*, MIT Media Lab Special Issue, Vol. 35, Nos. 3 & 4, pp. 292-312, 1996.

[9] L. R. Rabiner. A tutorial on Hidden Markov Models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.

[10] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[11] J. Li, A. Najmi and R. M. Gray. Image Classification by a Two Dimensional Hidden Markov Model. In *Proceedings of ICASSP'99*, volume VI, pages 3313–3316, 1999.

[12] C. L. Miller, B. R. Hunt, M. A. Neifeld, and M. W. Marcellin, Bi-level image reconstructions via 2-D Viterbi search. In *Proceedings of IEEE ICIP'97*, pages 181–184, 1997.