PROBABILISTIC IMAGE MODELING WITH DEPENDENCY-TREE HIDDEN MARKOV MODELS

Joakim Jiten, Bernard Merialdo
Institut Eurécom, Sophia Antipolis, France
{jiten,merialdo}@eurecom.fr

ABSTRACT

In this paper, we investigate some properties of a new type of 2D Hidden Markov Model, based on the notion of Dependency Tree. DT-HMMs avoid the complexity of regular 2D HMMs by changing the double horizontal and vertical spatial dependencies into a random uni-directional dependency, either horizontal or vertical. We explore various issues about the effect of this random choice. This type of probabilistic model can be useful in many applications for image and video segmentation, classification, and others.

1. INTRODUCTION

Statistical modeling methods are paramount in today's large-scale image analysis. It is critical for almost all image processing problems, such as estimation, compression and classification.

In image modeling there is a fairly wide-spread agreement that objects should be presented as collections of features which appear in a given mutual position or shape (e.g. sun in the sky, sky above landscape or boat in the water etc) [1,2]. This is also applicable on a lower level; consider analyzing local features in a small region; it is sometimes difficult even for a human to tell what the image is about.

For most images, pixels have spatial dependencies which should be enforced during the classification. 2D Hidden Markov Models consider observations (e.g. feature vectors representing blocks of pixels) statistically dependent on neighboring observations through transitions probabilities organized in a Markov mesh [11], giving a dependency in two dimensions. The state process defined by this mesh is a special case of the Markov random field.

The standard algorithms associated with HMMs; Baum-Welch and Viterbi, are efficient in estimating the model parameters in 1D. Unfortunately their computational complexity grows exponentially in higher dimensions, so they are therefore not useful in 2D problems.

Many approaches have been proposed to overcome the complexity of 2D-HMMs [3]. Among the first ones is [4] which uses a 1D HMM to model horizontal bands of face images. A more elaborate idea is to extract 1D features out of the image or video, and model these features with one or more 1D models [5].

Another approach is to use a two-level model, called Embedded HMM or Hierarchical HMM, where a first high level model contains super-states associated to a low level HMM, which models the lines of the observed image [6]. The main disadvantage of these approaches is that they greatly reduce the vertical dependencies between states, as it is only achieved through a single super-state.

Finally several attempts have been done to heuristically reduce the complexity of the HMM algorithms by making simplifying assumptions which approximate the real algorithms:

- select a subset of state configurations only [7],
- ignore correlation of distant states [8],
- approximate probabilities by turbo-decoding [9].

The main disadvantage of these approaches is that they only provide approximate computations, so that the probabilistic model is no longer theoretically sound.

We have recently proposed [10] a novel multi-dimensional Hidden Markov Model based on the idea of a dependency tree between positions. This simplification leads to an efficient implementation of the re-estimation algorithms, while keeping a mix of horizontal and vertical dependencies between positions. The remainder of this paper is organized as follows; section 2 provides the theoretical basis for DT-HMM, section 3 present experimental results and in section 4 we give concluding remarks and suggest future directions.

2. DT-HMM: DEPENDENCY-TREE HMM

The reader is expected to be familiar with 1D-HMM. We denote by $O=\{o_{ij}, i=1,\ldots m, j=1,\ldots,n\}$ the observation, for example each o_{ij} may be the feature vector of a block (i,j) in the image. We denote by $Q = \{q_{ij}, i=1,\ldots m, j=1,\ldots,n\}$ the state assignment of the HMM, where the HMM is assumed to be in state q_{ij} at position (i,j) and produce the observation vector o_{ij}. If we denote by λ the parameters of the HMM, then, under the Markov assumptions, the joint likelihood of O and Q given λ can be computed as:

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q|\lambda) = \prod_{ij} p(o_{ij}|q_{ij},\lambda)p(q_{ij}|q_{i-1,j},q_{i,j-1},\lambda)$$

If the set of states of the HMM is $\{s_1, \ldots s_N\}$, then the parameters λ are:

- the output probability distributions $p(o \mid s_i)$
- the transition probability distributions $p(s_i \mid s_j,s_k)$.

Depending on the type of output (discrete or continuous) the output probability distribution are discrete or continuous (typically a mixture of Gaussian distribution).

The problem with 2D-HMM is the double dependency of $q_{i,j}$ on its two neighbors, $q_{i-1,j}$ and $q_{i,j-1}$, which does not allow the factorization of computation as in 1D, and makes the computations practically intractable.

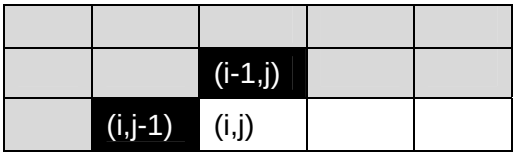

Figure 1. 2D Neighbors

Our idea is to assume that $q_{i,j}$ depends on one neighbor at a time only. But this neighbor may be the horizontal or the vertical one, depending on a random variable t(i,j). More precisely, t(i,j) is a random variable with two possible values:

$$t(i,j) = \begin{cases} (i-1,j) & \textit{with prob } 0.5 \\ (i,j-1) & \textit{with prob } 0.5 \end{cases}$$

For the position on the first row or the first column, t(i,j) has only one value, the one which leads to a valid position inside the domain. t(0,0) is not defined. So, our model assumes the following simplification:

$$p(q_{i,j}|q_{i-1,j},q_{i,j-1},t) = \begin{cases} p_V(q_{i,j}|q_{i-1,j}) & \text{if } t(i,j) = (i-1,j) \\ p_H(q_{i,j}|q_{i,j-1}) & \text{if } t(i,j) = (i,j-1) \end{cases}$$

If we further define a "direction" function:

$$D(t) = \begin{cases} V & \text{if } t = (i-1,j) \\ H & \text{if } t = (i,j-1) \end{cases}$$

then we have the simpler formulation:

$$p(q_{i,j}|q_{i-1,j},q_{i,j-1},t) = p_{D(t(i,j))}(q_{i,j}|q_{t(i,j)})$$

Note that the vector **t** of the values t(i,j) for all (i,j) defines a tree structure over all positions, with (0,0) as the root. Figure 2 shows an example of random Dependency Tree.

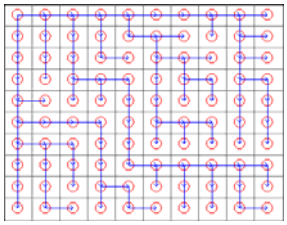

Figure 2. Example of a Random Dependency Tree

The DT-HMM replaces the N^3 transition probabilities of the complete 2D-HMM by $2N^2$ transition probabilities. Therefore it is efficient in terms of storage. But, more importantly the computation also keeps a linear complexity with the number of positions since we only have dependencies in one direction at the time. Position (0,0) has no ancestor. In this paper, we assume for simplicity that the model starts with a predefined initial state s_I in position (0,0). It is straightforward to extend the algorithms to the case where the model starts with an initial probability distribution over all states. A full presentation for the Maximum Likelihood and Viterbi reestimation algorithms can be found in research report [10].

The 2D HMM has numerous interesting applications for image and video understanding, for example:

- The best state assignment (one state for each position) provides a segmentation of the image, or the localization of a specific object (identified by one or more states),
- The probability P(O|t) may be trained for several classes of images, therefore leading to a Maximum Likelihood classifier for unknown images.

3. DT-HMM PROBABILITY ESTIMATION

According to the model, the exact probability of an observation is:

$$P(O) = \sum_t P(O|t)P(t)$$

We may assume that all dependency trees are equally likely, so that the distribution P(t) is uniform. As there are $2^{(m-1)(n-1)}$ different trees for an image of mxn blocks, the complete computation is prohibitive. Therefore, it is important to search for approximations of this value which are easy to compute. In this paper we investigate three different ways of doing this estimation by: unique sampling (P^u), tree average (P^a) and dual tree (P^d). The section below gives a detailed study of their qualities.

The experiments are conducted using a 36-state model trained on a consistent set of 130 *Beach* images annotated at the pixel level into 8 different sub classes. We divide the images in 22x16 blocks that are represented by color moments and DCT coefficients. Evaluation is done by computing P(O|t) (also called the score), using the Viterbi algorithm, for 100 random images, with different set of trees, to investigate the behavior of the dependency.

We give an example below to illustrate the state alignment in one image for two different trees.

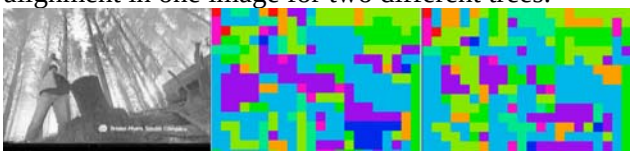

Figure 3. State alignment with two different trees

Estimation by Average

The best approximation (in terms of quality) that we can do is to compute an average over as many trees as possible: consider a set of trees $T_k=\{t_1, t_2, \ldots t_k\}$ and compute the estimate as:

$$P_k^a(O) = \frac{1}{k}\sum_{i=1}^{k} P(O|t_i)$$

As k gets larger, the value of $P_k^a(O)$ should converge towards the optimal value. The graph in figure 4 shows the magnitude of the relative error E(k) computed as the relative difference of this score to the average over the largest number of trees (here chosen to be 50), i.e.:

$$E(k) = \left| \frac{\frac{1}{50}\sum_{i=1}^{50} P(O|t_i) - \frac{1}{k}\sum_{i=1}^{k} P(O|t_i)}{\frac{1}{50}\sum_{i=1}^{50} P(O|t_i)} \right|$$

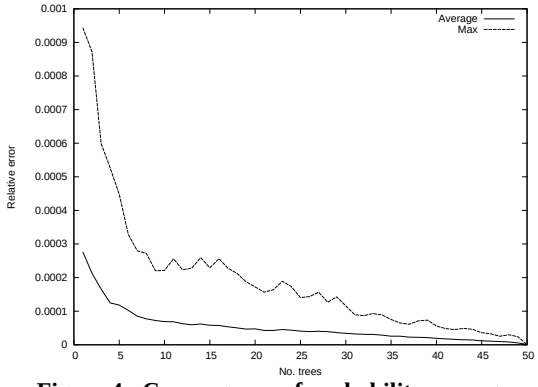

Figure 4. Convergence of probability average

We can see that with 10 trees or more, the average relative error is below 0.01% while the maximum relative error remains below 0.03%.

Estimation by unique sampling

As the computation of the probability with a lot of different trees can be computationally expensive, we would like a faster approximation of this probability. The fastest way is to use a single tree,

$$P^u(O) = P(O|t)$$

To evaluate the quality of the approximation, we compute the relative difference between an estimation and the mean over all 50 trees, and we compute the histograms of these values for different estimations (P^u, P_{15}^a, P^d).

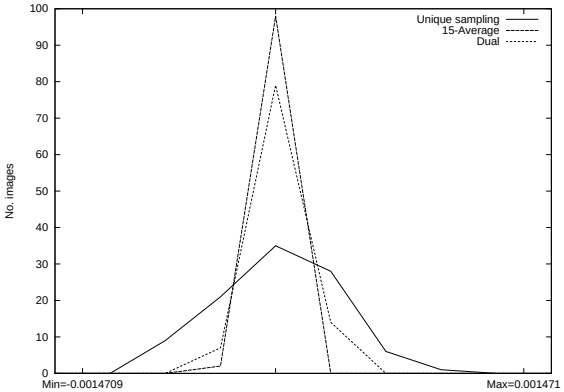

Figure 5. Distribution of scores

P^d is the dual estimation and is explained below. We can see that 90% of the values for P_{15}^a are within a

range of ± 0.015%. By comparison P^u has only 63% of its values with in the same range. For a confidence interval of 90%, P^u has a variation of ± 0.060%.

Estimation with dual tree

The dual t^T of a tree t is defined as the tree where the horizontal dependencies are replaced by vertical dependencies and vice versa (except for the border nodes for which the ancestor is unique).

As the couple (t, t^T) contains all the possible vertical and horizontal dependencies, it is interesting to consider the estimation by the average probability between the tree t and its dual t^T.

$$P^d(O) = \frac{1}{2}\left(P(O|t) + P(O|t^T)\right)$$

Where t^T is the dual tree of t, defined by replacing horizontal by vertical dependencies (and vice versa), except for boundary constraints. This formulation introduces both horizontal and vertical dependencies for all neighbor pairs in the observation.

As indicated in figure 5 the dual estimation gives a better approximation than the unique sampling. The dual estimation has 82% of its samples in the previously mentioned interval with limits ± 0.015%. For a confidence interval of 90%, P^d has a variation of ± 0.022%.

This comparison may lead to the choice of the best approximation method, based on the deviation which is considered reasonable for a given application.

4. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have presented a new type of Hidden Markov Models based on dependency trees. This presentation allows efficient algorithms to be used for training the model and computing the probability of new observations. We have investigated the variation of the observation probability with respect to the particular dependency tree that is used. We have found that there are great chances that a random tree will provide a value which is close to the average.

The DT-HMM is quite recent and we have just started exploring its capabilities. We expect that this type of model will be useful in the future for applications such as image and video segmentation and semantic classification.

REFERENCES

[1] Joo-Hwee Lim, Jesse S. Jin, "Semantics Discovery for Image Indexing", ECCV (1) 2004: 270-281.

[2] P. Moreels, M. Maire, P. Perona, "Recognition by Probabilistic Hypothesis Con-struction", ECCV (1) 2004: 55-68.

[3] M. A. Mohamed and P. Gader, "Generalized Hidden Markov Models-Part I: Theoretical Frameworks", IEEE Transaction on Fuzzy Systems, February, 2000, Vol.8, No.1, pp. 67-81.

[4] Samaria, Ferdinando and Fallside, "Frank Face Identification and Feature Extraction Using Hidden Markov Models", Image Processing: Theory and Applications, Elsevier, 1993, pp 295-298.

[5] M. Brand, N. Oliver, and A. Pentland. "Coupled hidden markov models for complex action recognition", In Proceedings, CVPR, pages 994--999. IEEE Press, 1997.

[6] S. Fine, Y. Singer, N. Tishby, "The hierarchical hidden Markov model: Analysis and applications", Machine Learning 32(1998).

[7] J. Li, A. Najmi, and R. M. Gray, "Image classification by a two-dimensional hidden markov model", IEEE Trans. Signal Processing, vol. 48, no. 2, pp. 517–533, 2000.

[8] B. Merialdo, S. Marchand-Maillet, B. Huet, "Approximate Viterbi decoding for 2D-hidden Markov models", IEEE International Conference on , Acoustics, Speech, and Signal Processing, Volume 6, 5-9 June 2000 Page(s):2147 - 2150 vol.4

[9] F. Perronnin, J.-L- Dugelay, K. Rose, "Deformable face mapping for person identification", International Conference on Image Processing, Volume 1, 14-17 Sept. 2003 Page(s):I - 661-4.

[10] B. Merialdo, "Dependency Tree Hidden Markov Models", Research Report RR-05-128, Institut Eurecom, Jan 2005.

[11] L.N. Kanal, “Markov mesh models” in Image Modeling. New York: Academic, 1980, pp. 239-243