Semantic Image Segmentation with a Multi-dimensional Hidden Markov Model

Joakim Jiten, Bernard Merialdo

Institut EURECOM, BP 193,06904 Sophia Antipolis, France
{jiten,merialdo}@eurecom.fr

Abstract. Segmenting an image into semantically meaningful parts is a fundamental and challenging task in image analysis and scene understanding problems. These systems are of key importance for the new content based applications like object-based image and video compression. Semantic segmentation can be said to emulate the cognitive task performed by the human visual system (HVS) to decide what one "sees", and relies on a priori assumptions. In this paper, we investigate how this prior information can be modeled by learning the local and global context in images by using a multidimensional hidden Markov model. We describe the theory of the model and present experiments conducted on a set of annotated news videos.

Keywords: Image Segmentation, Hidden Markov Model, 2D HMM, Block-based.

1 Introduction

Hidden Markov Models (HMM) have become increasingly popular in such diverse applications as speech recognition [1], language modeling, language analysis, and image recognition [3,9,12]. The reason for this is that they have a rich mathematical structure and therefore provide a theoretical basis for many domains. A second reason is the discovery of the Baum-Welch's training algorithm [2] which allows estimating the numerical values of the model parameters from training data.
Most of the current applications involve uni-dimensional data. In theory, HMMs can be applied as well to multi-dimensional data. However, the complexity of the algorithms grows exponentially in higher dimensions, so that, even in dimension 2, the usage of plain HMM becomes prohibitive in practice [4].
For this reason we have proposed an efficient sub-type of multi-dimensional hidden Markov model; the Dependency-Tree Hidden Markov Model [5] (DT-HMM) which preserves a reasonable computational feasibility and therefore enables us to apply it to multidimensional problems such as image segmentation.
In this paper, we explore the intrinsic ability of the DT-HMM to automatically associate pixels (or blocks of pixels) to semantic sub-classes which are represented by the states of the Markov model. To this end we enforce restrictions to the states during

training, by having the training set labeled on pixel level. The performance of the model is demonstrated on a subset of the TrecVideo archive [16] which consists of 60 hours of annotated news broadcast.
The remainder of this paper is organized as follows: section 3 outlines our motivation and presents the theory of DT-HMM. We show how the training and decoding algorithms for DT-HMM keep the same linear complexity as in one dimension. Section 4 will describe the experimental setup conducted on TrecVideo 2003 data and in section 5 we conclude and suggest future work.

2 Related Work

A number of researches have introduced systems for mapping users' perception of semantic concepts to low-level feature values [8,10]. The probabilistic framework of multijects (multi-objects) and multinets by Naphade and Huang [10] maps high level concepts to low level audiovisual features by integrating multiple modalities and infer unobservable concepts based on observable by a probabilistic network (multinet). The Stanford SIMPLIcity system [13] uses a scalable method for indexing and retrieving images based on region segmentation. A statistical classification is done to group images into rough categories, which potentially enhances retrieval by permitting semantically adaptive search methods and by narrowing down the searching range in a database.
Motivated by the desire to incorporate contextual information, Li and Gray [3] proposed a 2D-HMM for image classification based on a block-based classification algorithm using a path constrained Viterbi. An attempt in associating semantics with image features was done by Barnard and Forsyth at University of California at Berkeley [14]. Using region segmentation in a pre-processing step to produce a lower number of color categories, image feature search becomes a text search. The data is modeled as being generated by a fixed hierarchy of nodes organized as a tree. The work has achieved some success for certain categories of images. But, as pointed out by the authors, one serious difficulty is that the algorithm relies on semantically meaningful segmentation which is, in general, not available to image databases.
In recent work by Kumar and Hebert at Carnegie Mellon University [15], a hierarchical framework is presented to exploit contextual information at several levels. The authors claim that the system encodes both short- and long-range dependencies among pixels respectively regions, and that it is general enough to be applied to different domains of labeling and object detection.

3 DT-HMM: Dependency-Tree HMM

For most images with reasonable resolution, pixels have spatial dependencies which should be enforced during the classification. The HMM considers observations (e.g.

feature vectors representing blocks of pixels) statistically dependent on neighboring observations through transition probabilities organized in a Markov mesh, giving a dependency in two dimensions.

3.1 2D-HMM

In this section, we briefly recall the basics of 2D HMM and describe our proposed DT-HMM [5]. 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 $S = \{s_{ij}, i=1,\ldots m, j=1,\ldots,n\}$ the state assignment of the HMM, where the HMM is assumed to be in state s_{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 S given λ can be computed as:

$$P(O,S|\lambda) = P(O|S,\lambda)P(S|\lambda) = \prod_{ij} p\left(o_{ij}|s_{ij},\lambda\right)p\left(s_{ij}|s_{i-1,j},s_{i,j-1},\lambda\right) \quad \textbf{(1)}$$

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). We would like to point out that there are two ways of modeling the spatial dependencies between the neighbor state variables; by a causal or non-causal Markov random field (MRF). The former is referred to as Markov mesh and has the advantage that it reduces the complexity of likelihood functions for image classification [6]. The causality also enables the derivation of an analytic iterative algorithm to estimate states with the maximum a posteriori probability, due to that the total observation is progressively built from smaller parts. The state process of DT-HMM is defined by the Markov mesh.

3.2 DT-HMM

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

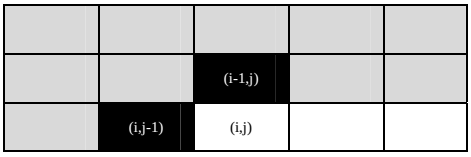

Fig. 1. 2D Neighbors

Our idea is to assume that $s_{i,j}$ 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:

$$t(i,j) = \begin{cases} (i-1,j) & \text{with prob } 0.5 \\ (i,j-1) & \text{with prob } 0.5 \end{cases} \quad (2)$$

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:

$$p(s_{i,j} | s_{i-1,j}, s_{i,j-1}, t) = \begin{cases} p_V(s_{i,j} | s_{i-1,j}) & if\ t(i,j) = (i-1,j) \\ p_H(s_{i,j} | s_{i,j-1}) & if\ t(i,j) = (i,j-1) \end{cases} \quad (3)$$

If we further define a "direction" function:

$$D(t) = \begin{cases} V & \text{if } t = (i-1,j) \\ H & \text{if } t = (i,j-1) \end{cases} \quad (4)$$

then we have the simpler formulation:

$$p(s_{i,j} | s_{i-1,j}, s_{i,j-1}, t) = p_{D(t(i,j))}(s_{i,j} | s_{t(i,j)}) \quad (5)$$

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.

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. We will see that it is also efficient in terms of computation. 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.

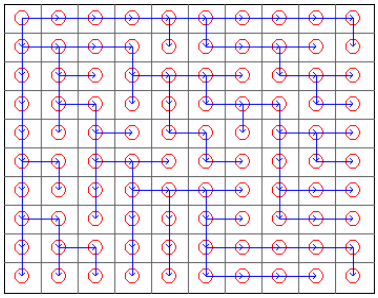

Fig. 2. Example of Random Dependency Tree

4 Application to Image Segmentation

4.1 Viterbi Algorithm

The Viterbi algorithm finds the most probable sequence of states which generates a given observation O:

$$\hat{S} = \underset{S}{\text{Argmax}}\ P(O, S|t) \tag{6}$$

The details of the algorithm for DT-HMM are given in [5][18].The algorithm is used for training the model, by iteratively reestimating the output and transition probabilities with the relative frequencies computed on the Viterbi sequences of states on the training images. It is also used for image segmentation on the test data, where each region is composed of the blocks which are covered by a given state in the Viterbi sequence.

4.2 States with semantic labels

We illustrate the use of DT-HMM for semantic segmentation on the example of segmenting *beach* images (class) into semantic regions (sub-classes). In principle, we should define one state of the model for each semantic region, however, to account for the variability of the visual appearance of semantic region, each semantic region (sub-class) is assigned a range of states. This potentially allows a sub-class such as *sky* to be represented by different states with dominant color blue, white, gray or yellow. The table below lists the sub-classes and their associated number of states.

Table 1. The number of states for each sub-class

Sub Class	No. states
Un-annotated	3
Sky	7
Sea	5
Sand	6
Mountain	3
Vegetation	3
Person	4
Building	3
Boat	2
8 sub-classes	36 states

One special class, called "*un-annotated*", is used for areas that are ambiguous or contain video graphics etc... Ambiguous areas are patches which contain several sub-classes or which are difficult to interpret.

4.3 Model Training

The training was conducted on the TrecVideo archive [16], from which we selected a wide within-class variance of 130 images depicting "Beach" (see Figure 3).

Fig. 3. Example of training images

Each image is split into blocks of 16x16 pixels, and the observation vector for each block is computed as the average and variance of the LUV (CIE LUV color space) coding $\{L_\mu,U_\mu,V_\mu, L_\sigma,U_\sigma,V_\sigma\}$ combined with six quantified DCT coefficients (Discrete Cosine Transform). Thus each block is represented by a 12 dimensional vector. Those images have been manually segmented and annotated, so that every feature vector is annotated with a sub-class.

To define the initial output probabilities, a GMM (Gaussian Mixture Model) is trained with the feature vectors corresponding to each sub-class. We allow three GMM components for every state, so the GMM for the sub-class *sky* has 21 components and for *vegetation* (see Table 1). Then we group the components into as many clusters as there are states for this sub-class (using the k-means algorithm). Finally, the GMM model for each state is built by doubling the weight of the components of the corresponding cluster in the GMM of the sub-class. The transition probabilities are initialized uniformly. Then, during training we iterate the following steps:

- We generate a random dependency tree and perform a Viterbi alignment to generate a new labeling of the image. The Viterbi training procedure is modified to consider only states that correspond to the annotated sub-class at each position, thus constraining the possible states for the observations (the manual annotation specifies the sub-class for each feature vector, but not the state).
- We reestimate the output and transition probabilities by relative frequencies (emission of an observation by a state, horizontal and vertical successors of a state) with Lagrange smoothing.

4.4 Experimental Results

During training, we can observe the state assignments at each iteration as an indication of how the model fits the training data. For example, the first ten iterations on the training image to the left in figure 4 provide the following state assignments:

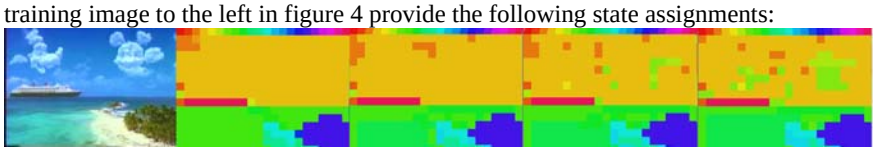

Fig. 4. State segmentation after 0, 2, 6 and 10 iterations

This shows that the model has rapidly adapted each sub-class to a particular set of observations. As such, the Viterbi labeling provides a relevant segmentation of the image. The graph below shows the evolution of likelihood of the training data during the training iterations. We can see that the likelihood for the model given the data has an asymptotic shape after 10 iterations.

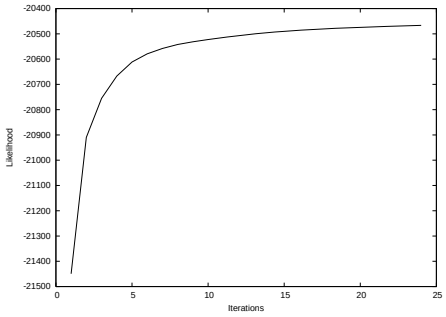

Fig. 5 Likelihood of the training data after N iterations

Once the model is trained, we can apply it on new images. Below is an example of the state assignment for an image in the test set; 70% of the blocks are correctly classified.

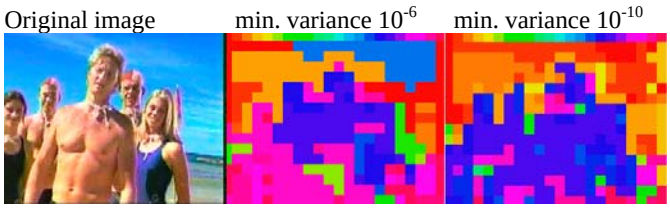

Fig. 6 State segmentations on test image

It should be emphasized that this is not just a simple segmentation of the images, but that each region is also assigned one of the 36 states (which belongs to one of the 8 sub-classes). The definition of those states has been done taking into account all training data simultaneously, and provides a model for the variability of the visual evidence of each sub-class.

During training, we impose a minimum variance for the Gaussian distributions, in order to avoid degeneracy. This minimum has an impact, as we noted that the number of correct labeled blocks in the example above increased to 72% when changing the minimum variance from 10^{-6} to 10^{-10}. An explanation for this is that if the selected minimum variance is too high, some Gaussians will be flattened out and collides with Gaussians from states representing similar observations.

Sometimes the result is degraded because of visually ambiguous regions, as in the examples below (looking through a window, or sky reflection on the sea). Because the output probabilities of model have generally a greater dynamic range than the transition probabilities, they often play the major contribution in the choice of the best state assignment.

Fig. 7 Test images with ambiguous regions

Still, to show the effect of transition probabilities, we used the model to semantically segment 40 test images. We compare the best state assignment obtained by the Viterbi algorithm (this takes into account both output and transition probabilities) with the assignment where each feature vector is assigned the state which has the highest output probability. The average rate of correctly labeled blocks was 38% when taking transition probabilities into account and 32% with only the output probabilities. Figure 8 shows an example, with the original example image, the sub-class assignment without transition probabilities (56% blocks correctly labeled), and the Viterbi assignment (72% correct).

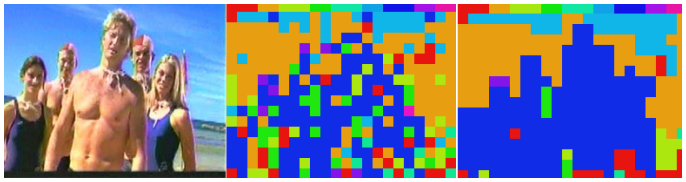

Fig. 8 Sub-class assignment without/with transition probabilities

5 Conclusions and Future Research

The contribution of this paper is to illustrate semantic segmentation of an image by a two dimensional hidden Markov model. We show how the model can be trained on manually segmented data, and used for labeling new test data. In particular, we use a modified version of the Viterbi algorithm that is able to handle the situation a visual sub-class is represented by several states, and only the sub-class annotation (not the state annotation) is available. We investigated several properties of this process. The motivation for this approach is that it can be easily extended to an larger number of classes and sub-classes, provided that training data is available. Allowing several states per sub-class gives the model the flexibility to adapt to sub-classes which may have various visual evidence.

6 Acknowledgements

The research leading to this paper was supported by the Institut Eurecom and by the European Commission under contract FP6-027026, Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content - K-Space.

References

[1] Rabiner, L.R., S.E. Levinson, and M.M. Sondhi, (1983). On the application of vector quantization and hidden Markov models to speaker independent, isolated word recognition. B.S.T.J.62,1075-1105

[2] LE. Baum and T. Petrie, Statistical Inference for Probabilistic Functions of Finite State Markov Chains, Annual Math., Stat., 1966, Vol.37, pp. 1554-1563.

[3] 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.

[4] Levin, E.; Pieraccini, R.; Dynamic planar warping for optical character recognition, IEEE International Conference on Acoustics, Speech, and Signal Processing, , Volume 3, 23-26 March 1992 Page(s):149 - 152

[5] Merialdo, B; Dependency Tree Hidden Markov Models, Research Report RR-05-128, Institut Eurecom, Jan 2005
[6] Kanal, L.N.: Markov mesh models in Image Modeling. New York: Academic, 1980, pp. 239-243
[7] P. F. Felzenszwalb , D. P. Huttenlocher, Image Segmentation Using Local Variation, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p.98, June 23-25, 1998
[8] F. Golshani, Y. Park, S. Panchanathan, "A Model-Based Approach to Semantic-Based Retrieval of Visual Information", SOFSEM 2002: 149-167
[9] O. Agazzi, S. Kuo, E. Levin, and R. Pieraccini. Connected and degraded text recognition using planar hidden Markov models. In Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP), volume 5, pages 113-116, 1993.
[10] M. R. Naphade, and T. S. Huang, "Extracting Semantics from Aduiovisual Content: The Final Frontier in Multimedia Retrieval", IEEE Transactions on Neural Network, Vol. 13, No. 4, 793--810, 2002.
[11] Merialdo, B.; Marchand-Maillet, S.; Huet, B.; 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
[12] Perronnin, F.; Dugelay, J.-L.; Rose, K.; Deformable face mapping for person identification, International Conference on Image Processing, Volume 1, 14-17 Sept. 2003 Page(s):I - 661-4
[13] J.Z. Wang, "Integrated Region-Based Image Retrieval", Dordrecht: Kluwer Academic, 2001
[14] K. Barnard and D. Forsyth, “Learning The Semantics of Words and Pictures,” Proc. Int’l Conf. Computer Vision, vol 2, pp. 408-415, 2001.
[15] S. Kumar and M. Hebert, "A Hierarchical Field Framework for Unified Context-Based Classification," Proc. ICCV, October, 2005.
[16] TREC Video Retrieval Evaluation, http://www-nlpir.nist.gov/projects/TrecVideo/
[17] J. Jiten, B. Mérialdo; “Probabilistic image modeling with dependency-tree hidden Markov models”, WIAMIS 2006, 7th International Workshop on Image Analysis for Multimedia Interactive Services, April 19-21, 2006, Incheon, Korea
[18] J. Jiten, B. Mérialdo, B. Huet;”Multi-dimensional dependency-tree hidden Markov models “, ICASSP 2006, 31st IEEE International Conference on Acoustics, Speech, and Signal Processing, May 14-19, 2006, Toulouse, France