Classification of Agricultural Fields in Satellite Images Using Two-Dimensional Hidden Markov Models

J. Baumgartner[1], J. Gimenez[2], J. Pucheta[1] and A. G. Flesia[2,3] ⋆

[1] FCEFyN - Universidad Nacional de Córdoba, Vélez Sarsfield 1611, X5016GCA Córdoba - Argentina. {jbaumgartner,jpucheta}@efn.uncor.edu
[2] FAMAF - Universidad Nacional de Córdoba, Medina Allende s/n , Ciudad Universitaria, X5000HUA Córdoba, Argentina.{jgimenez,flesia}@famaf.unc.edu.ar
[3] Conicet at Universidad Tecnólogica Nacional

Abstract. Image segmentation is a key competence for many real life applications such as precision agriculture. In this work we present an approach to classify agricultural fields in noisy satellite images. We start with the Markovian neighborhood hypothesis from where on we derive a general two-dimensional hidden Markov model (2D-HMM). To make the 2D-HMM feasible we apply the Path-Constrained Variable-State Viterbi Algorithm (PCVSVA) which allows us to approximate the optimal hidden state map. We evaluate the PCVSVA for a Landsat image of the province of Córdoba, Argentina and a synthetic satellite image. In both cases we use Cohen's $\widehat{\kappa}$ coefficient to compare the PCVSVA and the solution obtained by maximum likelihood (ML) to show the effectiveness of 2D-HMM of solving image segmentation tasks.

Keywords: Satellite farming, pattern recognition, image segmentation, hidden Markov models, Viterbi training

1 Introduction

The observation of agricultural regions via satellite images is a standard tool of Precision agriculture. One of the most important tasks of the so called satellite farming is to find intra-field variations in order to optimize yields [1]. The first step of this and many other applications [2], [3] is to segment and classify a given satellite image. In this work we study the capability of two-dimensional hidden Markov models (2D-HMM) to classify satellite images with a low signal-to-noise ratio.

Recently hidden Markov models (HMM) have gained the attention of researchers from different fields. Still the classical HMM are generally limited to those areas where the observed data has only one dimension such as speech recognition [4] or the analysis of genome data [5]. There have been early attempts to use HMM for higher-dimensional tasks like image segmentation but

⋆ Secyt-UNC,PICT 2008 00291.

only for the price of previously converting the two-dimensional data of an image into a single vector by lining up the rows or the columns of the image [6]. The drawback of such an ordering is clearly the loss of information because adjacent pixels in the original image are torn apart.

In the last years efforts were made to extend the classical one-dimensional HMM to higher dimensions [7]. The problem hereby is, that the standard method of parameter estimation for one-dimensional HMM, the Baum-Welch Algorithm [8], is not feasible for higher dimensions. Hence, the main issue is to reduce the computational complexity in order to keep the n-dimensional HMM feasible.

The firsts attempts to extend the HMM framework to two dimensions were the so called pseudo 2D-HMM. Pseudo 2D-HMM use superstates to represent the rows of an image, while the columns are connected by a simple Markov Chain [9]. Later the Viterbi Algorithm was applied to obtain a feasible version of a true 2D-HMM [10, 11].

In this work we compare the *Path-Constrained Variable-State Viterbi Algorithm* (PCVSVA) as presented by [12] with the solution obtained from ML. Note, that the classification of an image with ML does not consider any contextual prior. A detailed description of ML is given in algorithm 1.

Algorithm 1 Maximum Likelihood Classification

Unsupervised Maximum Likelihood classification of image I: To assign the label given by

$$s_{ij} = arg \max_{\ell \in \mathcal{L}} p(I_{ij}|\ell)$$

to pixel (i, j), with parameters given by the Expectation Maximization algorithm for Gaussian Mixtures, with random initializations.

The PCVSVA is a well established 2D-HMM algorithm for image segmentation, still it has never been used to solve to agricultural classification problems. Therefore the main contributions of this work are a complete description of the algorithm including all formulas and an experimental study if the PCVSVA can be applied for precision agriculture tasks. Like most 2D-HMM approaches the PCVSVA has shown good results for noisy images that can be considered a Gaussian mixture Markov random field [10]. For that reason we use two real world satellite images with a low signal-to-noise ratio and a synthetic image to evaluate the PCVSVA.

This paper is organized as follows: In section 2 we present the mathematical background of a 2D-HMM and explain why further assumptions are necessary to make 2D-HMM feasible. Thereafter we present a feasible approximation of a complete 2D-HMM in section 3, namely the PCVSVA. In section 4 we evaluate the results of PCVSVA and ML for the test images using Cohen's Kappa coefficient [13]. Finally we discuss the results in section 5.

2 Theory of two-dimensional Hidden Markov Models

Two-dimensional data – like the pixels of an image – can be handled by a 2D-HMM if we assume the data to be a Markov Random Field. This means, that, given the image, the hidden state of pixel (i, j) is conditionally independent of the pixels outside a certain neighborhood. For pixel (i, j) we define $(i^{'}, j^{'}) \prec (i, j)$ if $i^{'} < i$ or $i^{'} = i$ and $j' < j$. It can be shown that this definition leads to a 2nd order Markov Mesh which specifies for state $s_{i,j}$:

$$P(s_{i,j}|s_{i',j'} : (i^{'}, j^{'}) \prec (i, j)) = P(s_{i,j}|s_{i,j-1}, s_{i-1,j}).$$

In figure 1 the relevant pixels of the 2nd order Markov Mesh are shown. The two pixels $(i, j-1)$ and $(i-1, j)$ can be understood as the "past" of pixel (i, j). In other words we are moving from the top-left pixel to the bottom-right pixel. Hence, the initial probabilities for the 2D-HMM depend only on the first state $s_{0,0}$ and we can write

$$\pi_l = P(s_{0,0} = l) \qquad \forall l \in \mathcal{S}.$$

Next, we assume that the observed pixel intensities of one class are normally distributed. For the sake of completeness let us consider multispectral images where each pixel is a vector from R^k so we can calculate the emission probabilities of state $l \in \mathcal{S}$ with mean μ_l and covariance matrix Σ_l:

$$\begin{aligned} b_l(x) &= P(x|s_{i,j} = l) \\ &= \frac{1}{(2\pi)^{k/2}|\Sigma_l|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_l)^T \Sigma_l^{-1}(x-\mu_l)\right\} \end{aligned}$$

Besides that, we consider $P(s_{i,j}|s_{i,j-1}, s_{i-1,j})$ to be independent of the current pixel so we can gather the transition probabilities in a matrix A where

$$a_{m,n,l} = P(s_{i,j} = l|s_{i,j-1} = m, s_{i-1,j} = n).$$

These assumptions are the basis on which we try to find the optimal hidden state map s^*

$$s^* = \arg\max_s P(s|O, \theta). \tag{1}$$

where O are the observations and s is any admissible hidden state map. In θ we gather all the parameters of a 2D-HMM which are: the means and standard deviations of each hidden state and the transition probabilities A.

The exact formulas for the parameters are:

$$\mu_m = \frac{\sum_{i,j} L_m(i,j)u_{i,j}}{\sum_{i,j} L_m(i,j)}; \tag{2}$$

$$\Sigma_m = \frac{\sum_{i,j} L_m(i,j)\left(u_{i,j} - \mu_m\right)\left(u_{i,j} - \mu_m\right)^T}{\sum_{i,j} L_m(i,j)} \tag{3}$$

$$a_{m,n,l} = \frac{\sum_{i,j} H_{m,n,l}(i,j)}{\sum_{l'=1}^{M} \sum_{i,j} H_{m,n,l'}(i,j)} \tag{4}$$

where L and H are defined as

$$\mathbf{L^{(p)}_{m}(i,j)} = \sum_{\mathbf{s}} I(m = s_{i,j}) \frac{1}{\alpha} I(C(s) = c) \times \prod_{(i',j')\in\mathbb{N}} a^{(p)}_{s_{i'-1,j'},s_{i',j'-1},s_{i',j'}} \times \prod_{(i',j')\in\mathbb{N}} P\left(u_{i',j'}|\mu^{(p)}_{s_{i',j'}}, \Sigma^{(p)}_{s_{i',j'}}\right)$$

$$\mathbf{H^{(p)}_{m,n,l}(i,j)} = \sum_{\mathbf{s}} I\left(m = s_{i-1,j}, n = s_{i,j-1}, l = s_{i,j}\right) \times \frac{1}{\alpha} I\left(C(s) = c\right) \times \prod_{(i',j')\in\mathbb{N}} a^{(p)}_{s_{i'-1,j'},s_{i',j'-1},s_{i',j'}} \times \prod_{(i',j')\in\mathbb{N}} P\left(u_{i',j'}|\mu^{(p)}_{s_{i',j'}}, \Sigma^{(p)}_{s_{i',j'}}\right)$$

Note, that L and H are sums over all possible state maps. Thus, for an image of size $(w \times z)$ with M hidden states $\mathcal{S} = \{1, 2, \ldots, M\}$ there are M^{w*z} possible hidden state maps. This huge number of state maps, even for small images, leads to infeasibility because L and H can not be calculated.

To solve the problem of computational complexity several algorithms were proposed in recent years [10], [11]. Almost all of them apply the Viterbi Algorithm in some form [14]. In the next section we describe an advanced version of the Viterbi Algorithm, the *Path-Constrained Variable-State Viterbi Algorithm* [12].

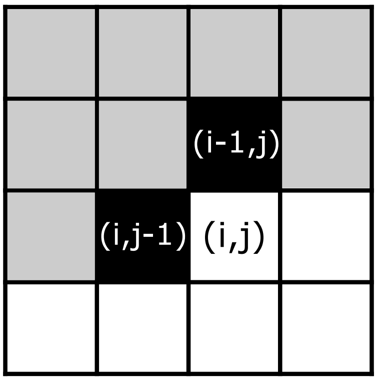


Fig. 1. Transitions among states in a 2nd order Markov Mesh. The gray and the black pixels fulfill $(i^{'}, j^{'}) \prec (i,j)$ but the two black pixels are sufficient statistics for pixel (i,j) under the Markov assumption.

3 Path-Constrained Variable-State Viterbi Algorithm

In this section we introduce a feasible version of the general 2D-HMM, namely the PCVSVA. Note, that due to the assumptions made, the PCVSVA only approximates the optimal hidden state map as defined by equation 1.

Based on the theoretical foundations from section 2 the PCVSVA can be derived as follows. First of all remember our notion of "past" as shown by figure 1. We now consider each diagonal of the image as one step in time, starting with the top-left pixel. Thus, the diagonals $T_0, T_1, T_2 \ldots$ are

$$T_0 = (s_{0,0}); \quad T_1 = (s_{1,0}, s_{0,1}); \quad T_2 = (s_{2,0}, s_{1,1}, s_{0,2}); \quad \ldots$$

Because we are dealing with a 2nd order Markov Mesh we can make the Markov assumption and get

$$\begin{aligned} P(s) &= P(T_0)P(T_1|T_0)\ldots P(T_{z+w-2}|T_{z+w-3},\ldots,T_0) \\ &= P(T_0)P(T_1|T_0)\ldots P(T_{z+w-2}|T_{z+w-3}). \end{aligned} \tag{5}$$

Note in equation 5, that each diagonal operates as an "isolating" element between neighboring diagonals. Hence, we have transformed the complex two-dimensional model to a pseudo one-dimensional HMM. The problem we are facing here, is that each diagonal consists of up to $\min(w, z)$ states: $T_0 \in \mathcal{S},\ T_1 \in \mathcal{S}^2,\ T_2 \in \mathcal{S}^3,\ \ldots,\ T_{z+w-2} \in \mathcal{S}$.

From now on we denote each combination of states on one diagonal a sequence. Keep in mind that a diagonal can have up to $M^{\min(w,z)}$ sequences – a number generally too high to be feasible.

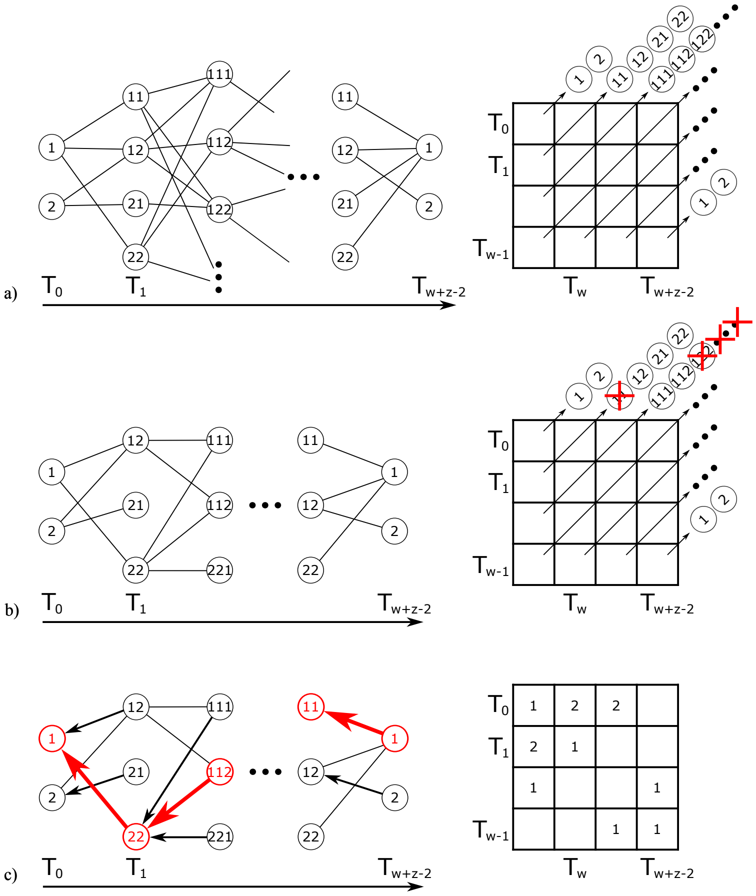
The first step to simplify the computation of the 2D-HMM is to reduce the number of sequences on each diagonal to N. If we set N to a value much smaller than $M^{\min(w,z)}$ we have drastically reduced the computational burden, but the question arises: How do we select the N sequences? For the moment we assume that we can evaluate the posterior of a given diagonal state sequence by simply multiplying the posteriors of each pixel without considering statistical dependencies between pixels, i.e.

$$P(s_{i,j} = l|O_{i,j}, \theta) \propto P(O_{i,j}|s_{i,j} = l, \theta)P(s_{i,j} = l|\theta).$$

By doing so it is computationally easy to classify the possible sequences as more or less probable. Once we have evaluated the posteriors of all sequences of one diagonal we keep the most likely N sequences and forget about the rest. This is clearly a significant simplification but even though we run the risk of throwing away the sequence that belongs to the optimal hidden state map s^* we expect to keep at least some sequences that are close to the optimal one.

After cutting the number of state sequences on each diagonal down to N we are ready to run the Viterbi Algorithm. We call each diagonal state sequence $\mathbf{s}_{d,k}$ where d is the index for the diagonal with $d = 0, 1, \ldots, z + w - 2$ and $k = 1, 2, \ldots, N$ indicates the state sequence. The initial state probabilities $\tilde{\pi}_k$ for pixel $(0, 0)$ are

$$\tilde{\pi}_k = P(T_0 = \mathbf{s}_{0,k}).$$



T0	1	2	2	
T1	2	1		
	1			1
Tw-1			1	1
		Tw		Tw+z-2

Fig. 2. Example of Path-Constrained Variable-State Viterbi Algorithm for two possible states and $N = 3$. a) Complete Viterbi Algorithm. b) Viterbi Algorithm constrained to 3 paths per diagonal. c) Finding the most probable combination of diagonals.

We denote $\delta_d(l)$ the maximum joint probability of the observations $\mathbf{O}_0, \ldots, \mathbf{O}_d$ and sequences from T_0 to T_d, where l is a certain sequence on diagonal d . Given the parameters of the 2D-HMM we can write

$$\begin{aligned}\delta_d(l) = &\max_{k_0,\ldots,k_{d-1}} P(\mathbf{s}_{0,k_0}, \ldots, \mathbf{s}_{d-1,k_{d-1}}, \mathbf{s}_{d,l}, \mathbf{O}_0, \ldots, \mathbf{O}_d|\theta), \\ &for \quad d = 0, \ldots, z+w-2; \quad l = 1, \ldots, N.\end{aligned} \tag{6}$$

Furthermore we collect the pixels on diagonal d in a variable $\varDelta(d)$ and define

$$b_{\mathbf{s}_{d,k}}(\mathbf{O}_d) = \prod_{(i,j)\in\varDelta(d)} b_{\mathbf{s}_{d,k}(i,j)}(O_{i,j}) \tag{7}$$

where $b_{\mathbf{s}_{d,k}}(\mathbf{O}_d)$ is the emission probability of sequence k on diagonal d under the assumption that each pixel is statistically independent from its neighbors. Finally we can calculate the transition probability from sequence k on diagonal d to sequence l on diagonal $d+1$:

$$\begin{aligned}\tilde{a}_{d,k,l} &= P(T_{d+1} = \mathbf{s}_{d+1,l}|T_d = \mathbf{s}_{d,k}, \theta) \\ &= \prod_{(i,j)\in\varDelta(d+1)} a_{\mathbf{s}_{d,k}(i-1,j),\mathbf{s}_{d,k}(i,j-1),\mathbf{s}_{d+1,l}(i,j)} \\ &\quad for \quad d = 0, \ldots, z+w-3; \quad k, l = 1, \ldots, N.\end{aligned} \tag{8}$$

Now we are ready to initialize the *Viterbi Algorithm* with the values

$$\begin{aligned}&\delta_0(k) = P(T_0 = \mathbf{s}_{0,k}), \quad b_{\mathbf{s}_{0,k}}(\mathbf{O}_0) = \tilde{\pi}_j b_{\mathbf{s}_{0,j}}(O_{0,0}) \\ &\quad \forall k = 1, 2, \ldots, N.\end{aligned}$$

Then we start the recursion using equations 6, 7 and 8

$$\begin{aligned}&\delta_{d+1}(l) = \left[\max_{1\le k\le N} \delta_d(k)\tilde{a}_{d,k,l}\right] b_{\mathbf{s}_{d+1,l}}(\mathbf{O}_{d+1}) \\ &\qquad \forall d = 0, 1, \ldots, z+w-3 \quad \forall l = 1, 2, \ldots, N.\end{aligned}$$

After each step we save the index of the most probable sequence on diagonal d that leads to sequence l on diagonal $d+1$ in a variable called φ:

$$\begin{aligned}&\varphi_{d+1}(l) = \arg\max_{1\le k\le N}\{\delta_d(k)\tilde{a}_{d,k,l}\} \\ &\qquad \forall d = 0, 1, \ldots, z+w-3 \quad \forall l = 1, 2, \ldots, N\end{aligned}$$

When the algorithm reaches the last diagonal we use the values saved in φ to track back the most probable path through the image starting with the bottom-right pixel

$$s^*_{z+w-2} = \arg\max_{1\le k\le N} \delta_{z+w-2}(k)$$

$$s^*_d = \varphi_{d+1}(s^*_{d+1}) \qquad \forall d = z+w-3, z+w-4, \ldots, 1$$

The final result s^* contains the optimal path through the N sequences at each diagonal. Note that this is equal to knowing the complete hidden state map for the whole image. In figure 2 an example of the PCVSVA is shown.

Once we know the hidden state of every pixel we can update the parameters of the 2D-HMM. Instead of the exact formulas 2, 3 and 4 we use approximate formulas for iteration step p, where $I(\cdot)$ is the indicator function. One can think of these simplified formulas as "count instead of evaluate":

$$\mu_l^{(p)} = \frac{\sum_{i=0}^{z-1} \sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right) O_{i,j}}{\sum_{i=0}^{z-1} \sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right)} \tag{9}$$

$$\Sigma_l^{(p)} = \frac{\sum_{i=0}^{z-1} \sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right) (O_{i,j} - \mu_l)(O_{i,j} - \mu_l)^T}{\sum_{i=0}^{z-1} \sum_{j=0}^{w-1} I\left(s_{i,j}^{(p-1)} = l\right)} \tag{10}$$

$$a_{n,m,l}^{(p)} = \frac{\sum_{i=1}^{z-1} \sum_{j=1}^{w-1} I\left(s_{i-1,j}^{(p-1)} = n, s_{i,j-1}^{(p-1)} = m, s_{i,j}^{(p-1)} = l\right)}{\sum_{i=1}^{z-1} \sum_{j=1}^{w-1} I\left(s_{i-1,j}^{(p-1)} = n, s_{i,j-1}^{(p-1)} = m\right)} \tag{11}$$

After updating a, μ and Σ we run the PCVSVA with the new parameters and iterate until convergence. In summary we can describe the algorithm of the 2D-HMM as

1. set initial values for μ_l and Σ_l for $l \in \mathcal{S}$
2. assign maximum a posteriori state to each pixel
3. calculate $a_{n,m,l}$ for every $n, m, l \in \mathcal{S}$ using equation 11
4. *Path-Constrained Variable-State Viterbi Algorithm*
5. update $a_{n,m,l}$, μ_l and Σ_l using equations 9, 10 and 11
6. if no convergence of parameters go back to 4.

In this algorithm step 4 will take $\mathcal{O}(w^2 M^2)$ time for an image of size $w \times w$ whereas step 5 requires no remarkable computational effort. In section 5 we compare the complexity of the PCVSVA and other characteristics with a new algorithm that we present in the next section.

4 Experimental results: Image segmentation

In this section we present the experimental results. The proposed algorithms, PCVSVA and ML, are implemented and executed in Matlab. To evaluate the PCVSVA we use Cohen's $\widehat{\kappa}$ coefficient [13] which is defined as

$$\widehat{\kappa} = \frac{P_O - P_E}{1 - P_E}.$$

where $P_O = \sum_{i=1}^{k} p_{ii}$ is the relative observed agreement among segmented image and ground truth and $P_E = \sum_{i=1}^{k} p_{i+}p_{+i}$ is the hypothetical probability of chance agreement.

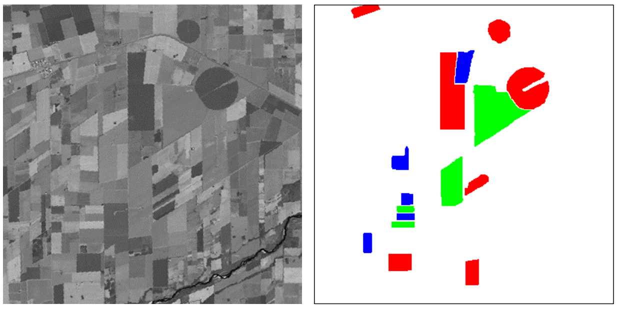


Fig. 3. Part of a Landsat image and the corresponding ground truth.

We propose three different scenarios to evaluate the PCVSVA. First we use the Landsat 7 image $229-082$, that shows a region five kilometers west of Río Primero in the Province of Córdoba. The image shows agricultural fields of different sizes and orientations and two center-pivot irrigations as well as a road. The studied region has a extension of 12 by 12 kilometers.

In this case an agricultural producer from the region provided information about the classes of some of the fields shown in the satellite image. Then we determined the exact borders of the fields running edge detection software in Matlab. As we have no information about the other fields shown in the image we leave them out for the classification as shown in figure 3. The goal is to distinguish the different types of fields in the satellite image. With this information one can compare fields and find irregularities like a reduced or advanced growth in some parts.

Because we have no information of how many classes the image is composed we evaluate the PCVSVA and ML for 3 to 8 classes. The $\widehat{\kappa}$ index of such segmentations are shown in figure 5, higher values (closer to one) indicates better segmentations, the curve indicates that 8 classes have the best segmentation qualification for PCVSA, meanwhile the more classes are added the worse the segmentation when applying ML. The actual segmentations are shown in figure 4.

We also test PCVSVA performance on a synthetic image. We added 5 levels of Gaussian noise to the image, and tested performance of the algorithm computing kappa values while initializing with supervised ML and unsupervised EM-ML.

The ground truth of this synthetic image and two implementations - the lowest and the highest noise level - can be seen in figure 6. The segmentation outputs of this experiment are shown in figure 7. Automatic initialization produces a dirty looking segmentation, while supervised ML as initialization helps the algorithm deliver a better segmentation. In figure 8 we show the $\widehat{\kappa}$ values computed over the segmentations in both cases, supervised and unsupervised. Note, that for the synthetic image, as well as for the satellite image, the PCVSVA outperforms the segmentation achieved by ML. The difference between the $\widehat{\kappa}$ values is smaller

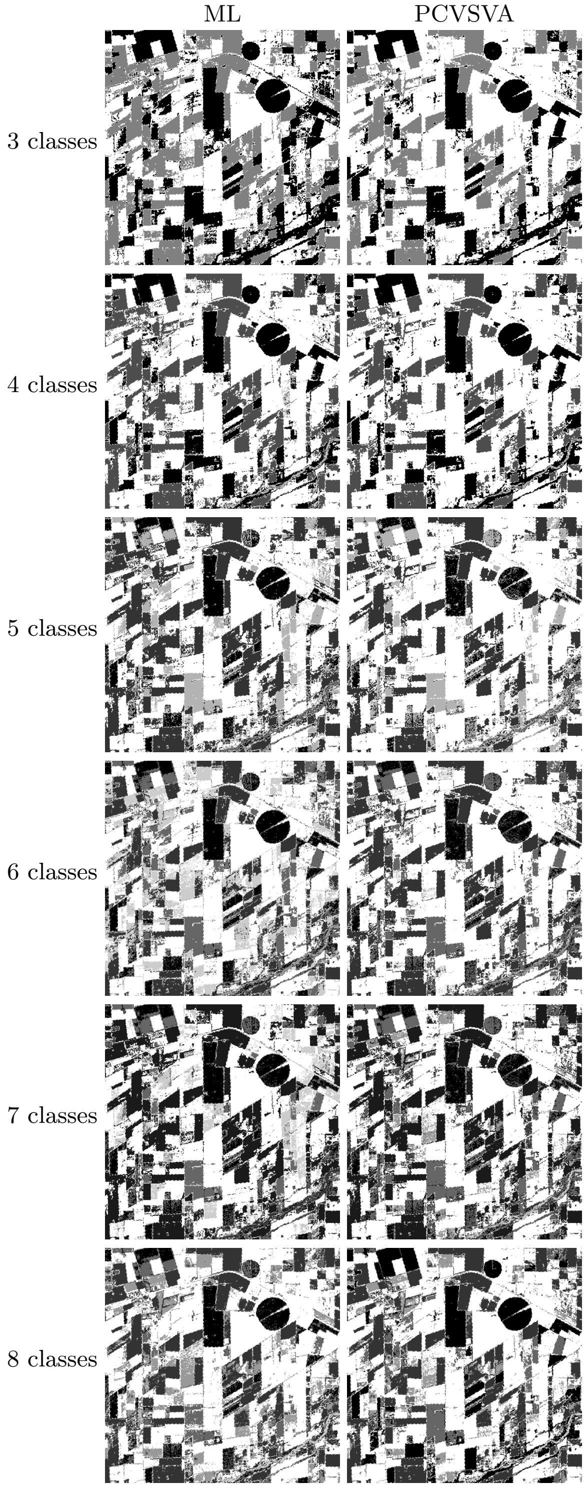


Fig. 4. Segmentation results for three to eight classes of the PCVSVA and ML for the Landsat 7 image $229 - 082$, that shows a region five kilometers west of Río Primero in the Province of Córdoba

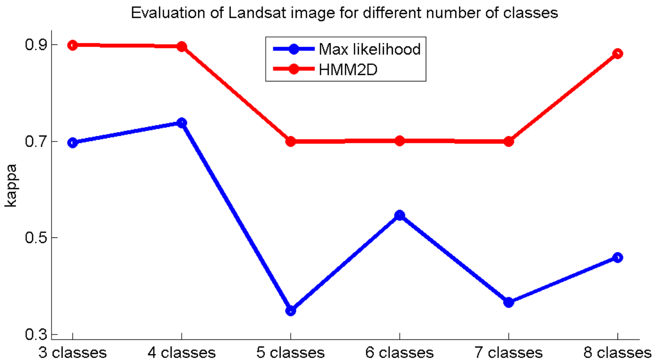


Fig. 5. Results of the segmentation of a Landsat image. The PCVSVA is clearly superior to the ML approach for real life data.

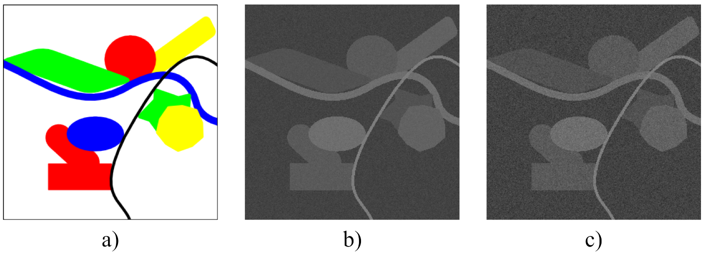


Fig. 6. a) Ground truth of the synthetic satellite image. This image contains different kinds of fields (red, green, yellow), water regions (blue) as well as a street (black). b) Synthetic image observed at a very low noise level. c) Synthetic image observed at a very high noise level.

for the synthetic image, which was designed as a Gaussian mixture. Markovian models have better fit in real imagery, but they should be at least as good as ML under i.i.d conditions.

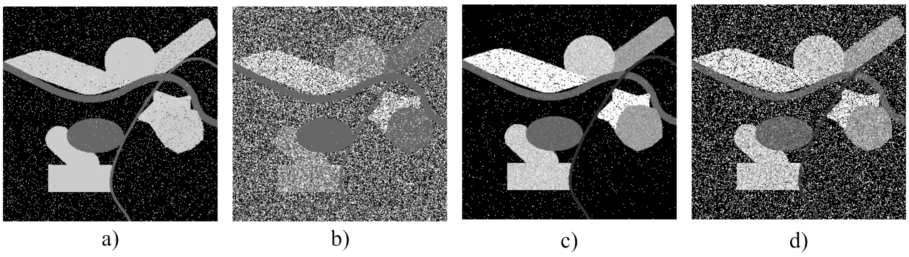


Fig. 7. Results of the segmentation of a synthetic image. a) Low noise level, unsupervised. b) High noise level, unsupervised. c) Low noise level, supervised. d) High noise level, supervised.

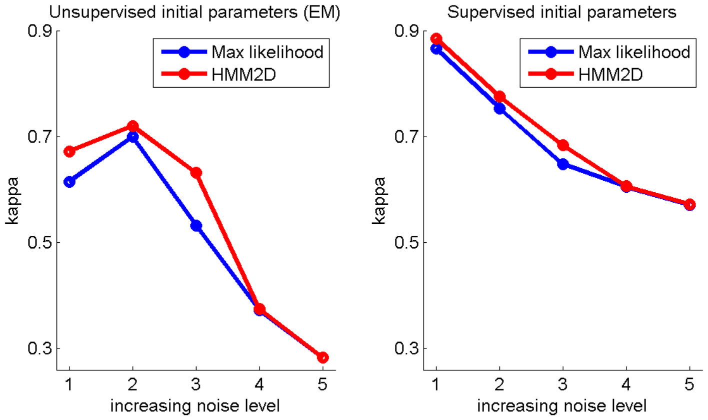


Fig. 8. Results of the segmentation of a synthetic image. On the left the initial parameters are found by running the Expectation Algorithm whereas on the right a training sample is used to initialize the PCVSVA.

Finally we show the results of PCVSVA and ML for a Landsat 7 image taken from a region two kilometers north of Arroyito. The Landsat number of the image is 228 − 082. There is no ground truth available so we just show the results of both algorithms in figure 9. In the next section we discuss the results and draw conclusions about image classification with the PCVSVA.

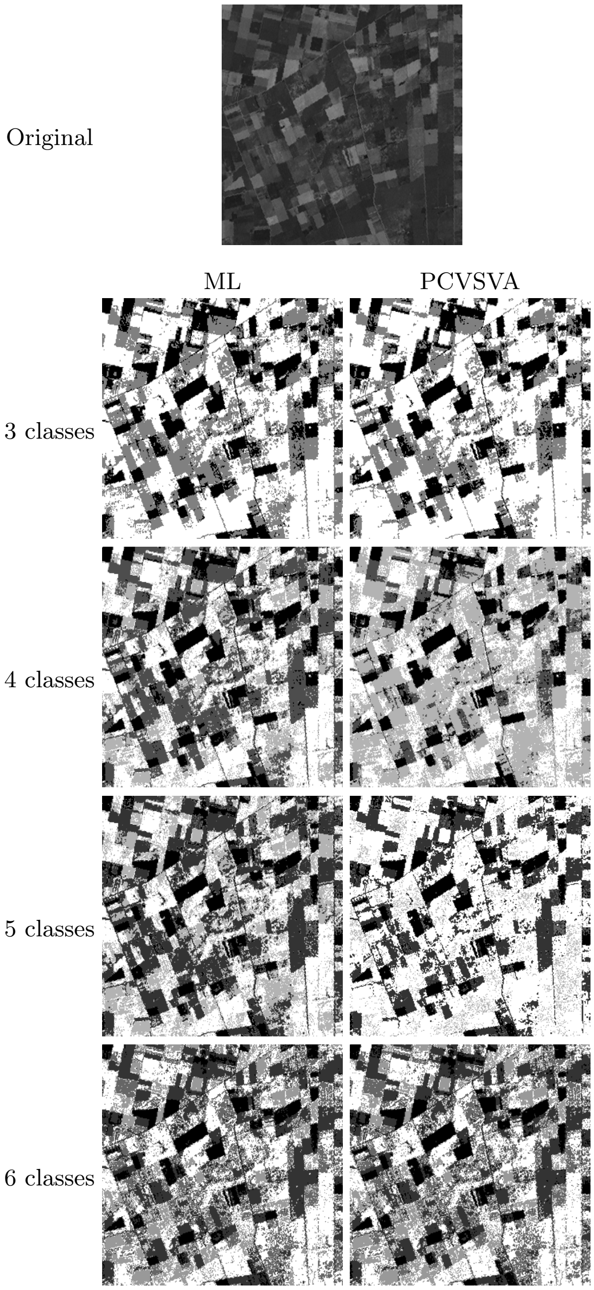


Fig. 9. Segmentation results of PCVSVA and ML methods for a Landsat 7 image taken from a region two kilometers north of Arroyito showing an agricultural region. The Landsat number of the image is $228 - 082$.

5 Conclusions

The classification algorithm presented in this work is based on the theory of Markov random fields. This means, that the state of each pixel depends not only on its gray value, but also on the hidden states in its neighborhood. Therefore it is valid to compare the PCVSVA with the ML solution, where the hidden states depend only on the observed gray values.

The experiments show the superiority of the 2D-HMM approach with respect to ML, especially for real life images. In the case of the synthetic image, the gap between PCVSVA and ML is much smaller – probably due to the design of experiment, which is a true Gaussian mixture. Nevertheless, the percentage of improvement over ML is always positive.

When it comes to computational resources, the PCVSVA reveals its weak point. While ML usually segments an image within seconds, the PCVSVA needed more than one hour for each of the shown images on a PC with i3 processor. Nevertheless, PCVSVA has the advantage of decoding selecting the best state's diagonal from a bag of sequences. A quick glance to the segmentation can be made using the most probable 50 sequences, which reduces running time to 15 minutes. Our code was made from scratch, on Matlab 2013a platform, and allows us to control all the routines involved in decoding. Another important point to add to our code is that it handle multivariate data coming from several classes. To our knowledge, this is the only one Matlab implementation of a Markovian labeling method based on Hidden Markov Models.

Despite the inevitable computational burden that bring 2D-HMM, we can conclude that the PCVSVA is capable of classifying agricultural fields in satellite images. Thereby it is important to mention, that the presented algorithm is robust to variations of the number of classes, as we have shown in the Landsat experiment. Besides that, the PCVSVA does not depend on the geometric shapes of the classified objects. Round objects like pivot irrigations are classified, as well as agricultural fields of any kind and shape. In summary, we have presented an algorithm, that has a consistent mathematical justification and shows good results for real life data. An issue for future works is to reduce the computational burden of the PCVSVA in order to make this algorithm easier to handle for users.

References

1. Carlson, T.N., Perry, E.M., Schmugge, T.J.: Remote estimation of soil moisture availability and fractional vegetation cover for agricultural fields. Agricultural and Forest Meteorology **52(1-2)** (1998) 45 – 69
2. Luoto, M., Toivonen, T., Heikkinen, R.: Prediction of total and rare plant species richness in agricultural landscapes from satellite images and topographic data. In: Landscape Ecology. Volume 17., Kluwer Academic Publishers (2002) 195 – 217
3. Moulin, S., Bondeau, A., Delecolle, R.: Combining agricultural crop models and satellite observations: From field to regional scales. International Journal of Remote Sensing **19(6)** (1998) 1021 – 1036

4. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. In: Proc. of the IEEE **77**. (1989) 257 – 286
5. Lukashin, A.V., Borodovsky, M.: Genemark.hmm: new solutions for gene finding. In: Nucleic Acids Research **26**. (1998) 1107 – 1115
6. Kuo, S.S., Agazzi, O.E.: Automatic keyword recognition using hidden markov models. Journal of Visual Communication and Image Representation **5(3)** (1994) 265 – 272
7. Ma, X., Schonfeld, D., Khokhar, A.: Distributed multi-dimensional hidden markov models for image and trajectory-based video classifications. In: Proc. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing. (2008) 957 – 960
8. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. In: Ann. Math. Stat 1. (1970) 164 – 171
9. Yen, C.C., Kuo, S.S.: Degraded documents recognition using pseudo 2d hidden markov models in gray-scale images. In: Proc. of SPIE 2277. (1994) 180 – 191
10. Li, J., Najmi, A., Gray, R.M.: Image classification by a two dimensional hidden markov model. In: IEEE Transactions on Signal Processing. Volume 48(2). (2000) 517 – 533
11. Ma, X., Schonfeld, D., Khokhar, A.: Image segmentation and classification based on a 2d distributed hidden markov model. In: Proc. SPIE 6822, Visual Communications and Image Processing. (2008)
12. Joshi, D., Li, J., Wang, J.Z.: A computationally efficient approach to the estimation of two- and three-dimensional hidden markov models. In: IEEE Transactions on Image Processing. Volume 15(7). (2006) 1871 – 1886
13. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement **20(1)** (1960) 37 – 46
14. Viterbi, A.J.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory **13(2)** (1967) 260 – 269