



Multi-scale stacked sequential learning

Carlo Gatta^{a,b,*}, Eloi Puertas^a, Oriol Pujol^{a,b}

^a Department of Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08007 Barcelona, Spain

^b Computer Vision Center, Edifici O - UAB, 08193 Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:

Received 15 January 2010

Received in revised form

4 March 2011

Accepted 4 April 2011

Available online 15 April 2011

Keywords:

Stacked sequential learning

Multiscale

Multiresolution

Contextual classification

ABSTRACT

Sequential learning is the discipline of machine learning that deals with dependent data such that neighboring labels exhibit some kind of relationship. The paper main contribution is two-fold: first, we generalize the stacked sequential learning, highlighting the key role of neighboring interactions modeling. Second, we propose an effective and efficient way of capturing and exploiting sequential correlations that takes into account long-range interactions. We tested the method on two tasks: text lines classification and image pixel classification. Results on these tasks clearly show that our approach outperforms the standard stacked sequential learning as well as state-of-the-art conditional random fields.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

As the machine learning community matures, problems it addresses become more challenging. One of the most widely used assumptions in supervised learning is that data is independent and identically distributed (iid). However, there are many real world applications in which this assumption does not necessarily hold. Consider the case of a laughter detection application from voice records. Laugh has a clear pattern alternating voice and non-voice segments. Thus, discriminant information comes from the alternating pattern, and not just by the samples on their own. Another case is part-of-speech tagging in which each example describes a word that is categorized as noun, verb, adjective, etc. In this case it is very unlikely that patterns such as (verb, verb, adjective, verb) occur without verifying the labels of the adjacent samples. All these applications present a common feature: the sequence/context of the labels matters.

Sequential learning [14] breaks the iid assumption and assumes that samples are not independently drawn from a joint distribution of the data samples \mathbf{X} and their labels \mathbf{Y} . In sequential learning the training data consists of sequences of pairs (\mathbf{x}, \mathbf{y}) , so that neighboring examples exhibit some kind of correlation. Usually, sequential learning applications consider one-dimensional relationship support, but this kind of relationships appear very frequently in other domains, such as images, or video. Consider the case of object recognition in image understanding.

It is clear that if one pixel belongs to a certain object category, it is very likely that neighboring pixels also belong to the same object (with the exception of its borders).

Sequential learning is often confused with a very related application, time series prediction. The main difference between both problems lays in the fact that sequential learning has access to the whole data set before any prediction is made and the full set of labels is to be provided at the same time. On the other hand, time series prediction has access to real labels up to the current time t and the goal is to predict the label at $t+1$.

In literature, sequential learning has been addressed from different perspectives; from the point of view of meta-learning by means of sliding window techniques, recurrent sliding windows [14] or stacked sequential learning [27,10]; the method is formulated as combination of classifiers. From the point of view of graphical models, using hidden Markov models [2], partially ordered Markov models [12], Markov random fields [9], conditional random fields [21] to infer the joint or conditional probability of the sequence. Finally, graph transformer networks [7] consider the input and output as a graph and looks for the transformation that minimizes a loss function of the training data using a neural network.

Independently of the specific method, there are still fundamental issues in sequential supervised learning that require the attention of the community. In [14] the authors acknowledge the following ones: (a) How to capture and exploit sequential correlations; (b) how to represent and incorporate complex loss functions; (c) how to identify long-distance interactions; (d) how to make sequential learning computationally efficient.

In this work, we are concerned with meta-learning strategies. Recently, Cohen et al. [10] showed that stacked sequential

* Corresponding author at: Department of Matemàtica Aplicada i Anàlisi, Universitat de Barcelona, Gran Via 585, 08007 Barcelona, Spain.
E-mail address: carlo.gatta@cvc.uab.es (C. Gatta).

learning (SSL from now on) performed better than CRF and HMM on a subset of problems called “sequential partitioning problems”. These problems are characterized by long runs of identical labels. Moreover, SSL is computationally very efficient since it only needs to train two classifiers a constant number of times. Considering these benefits, we decide to explore in depth sequential learning using SSL and generalize that architecture to deal with a wider variety of problems as well as giving answers to most of the open problems described in [14].

In this paper, we argue that a fundamental and overlooked step in SSL is the way the extended set is created. We first provide a general framework in which this extension step is clearly identified and then propose a new aggregation method capable of capturing long-distance interactions efficiently. The proposed step is based on a multi-scale decomposition [8] of the predicted data labels. As a result, we provide answers to the different open issues raised in [14], obtaining a method that (a) captures and exploits sequential correlations (b) since the method is a meta-learning strategy the loss function dependency is delegated to the second step classifier; (c) it efficiently captures long-distance label interactions by means of an explicit neighborhood modeling of the label field at different scales; and (d) it is fast, because it relies on training a few general learners. The benefits of the new method are shown in a one-dimensional support problem in a FAQ structure detection data set. Moreover, along with one-dimensional sequence examples, we provide results and discussion in the image domain using 2D support—note that image processing and understanding is a good example of *sequential partitioning problems*. For the 2D domain, we use the Weizmann horse database [6].

2. State of the art and motivation

While the contribution of this paper can appear limited into the machine learning area, it is actually of interest for the computer vision community too. A large part of the computer vision community is recently devoting efforts to exploit contextual information to improve classification performance in object/class recognition and segmentation. For these reasons, the relevant state of the art comes from both computer vision and machine learning communities.

The use of contextual information is potentially able to cope with ambiguous cases in classification. Moreover, the contextual information can increase a machine learning system performance both in terms of accuracy and precision, thus helping in reducing both false positive and false negatives. However, the methods presented in the previous section suffer for different disadvantages.

CRF [21] (and other graphical models), while introducing a general and powerful framework for combining features and contextual information, have no practical application when the clique is not reduced to a few nodes (usually a 4-neighborhood, i.e. the pixels at north, west, south and east of the center pixel). This is because the computational cost of both training and inference are very high and both proportional to the exponential of the clique cardinality. In fact, successful CRF models are always applied to groups of pixels [26,20] using a clique of size 2 on a 4-neighborhood. Finally, the only method proposing a multi-scale CRF [24] has been devised without rigorously defining the scale-space, i.e. without a clear relationship with the well-known multi-scale theory.

Other methods of the literature exploit contextual information by identifying super-pixels using segmentation algorithms tuned to perform over-segmentation [11,16,17]. In [17], for example, the set of super-pixels is clustered forming a vocabulary of possible

local contexts. Finally, the super-pixels are considered as the context for classification by considering the spatial relationship between the pixel (or area) being classified and the neighborhood super-pixels. In [11,16] the super-pixels are used to form the puzzle that better fits the object, using also contextual information, and geometric coherence, among different puzzles. All these methods assume that an over-segmentation is possible, and hopefully, different super-pixels can cluster together in a semantically meaningful way.

Other contextual methods extract a global representation of the context, and use it to influence the classification step. In [25], the context is modeled globally. Thus, the method does not locally compute the context and cannot relate labels (or objects) spatially (or temporally) by means of the local context.

3. Multi-scale stacked sequential learning (MS-SSL)

In order to clearly define the multi-scale stacked sequential learning, we first propose a generalized stacked sequential learning schema that emphasize important features that differentiate the MS-SSL from the classical SSL. Based upon the proposed generalization, next subsections explain two different possible implementations of the multi-scale paradigm, giving details on appropriate sampling schemes for each of the implementations. The section ends with two discussions on important features of the proposed method: (1) pro and drawbacks of the proposed multi-scale implementations; (2) the trade-off between long-range interaction and resolution.

Table 1 summarizes the mathematical notation used in the paper.

3.1. Generalized stacked sequential learning

The framework for generalizing the stacked sequential learning algorithm includes a new block in the pipeline of the basic SSL. Fig. 1 shows the generalized stacked sequential learning process. A classifier $h_1(\mathbf{x})$ is trained with the input data set (\mathbf{x}, y) by means of cross-validation and the predicted labels are obtained.

Table 1
Mathematical notation.

Input data	
$D \in \mathbb{N}$	Spatial/temporal dimensionality of the data
$\mathcal{D} \subset \mathbb{N}^D$	Compact support of the data domain
$\vec{q}, \vec{p} \in \mathcal{D}$	A spatial/temporal vector location
\mathbf{x}	Input feature vector
C_i	The i th class out of a total of K
$y \in \{C_1, C_2, \dots, C_K\}$	Ground truth labels
Multi-scale SSL	
$h_1(\cdot), h_2(\cdot)$	Two classifiers working on the iid hypothesis
$Y(\vec{q})$	Predicted label value at location \vec{q} by $h_1(\mathbf{x})$
$J(Y, \vec{q}; \rho, \theta)$	A functional that models the labels context
$\mathbf{z} \in \mathbb{R}^W$	The contextual feature vector produced by J
$W \in \mathbb{N}$	The length of the contextual feature vector \mathbf{z}
\mathbf{x}^{ext}	The extended feature vector combining \mathbf{x} and \mathbf{z}
y''	Final MSSL prediction produced by $h_2(\mathbf{x}^{\text{ext}})$
Multi-scale decomposition	
$s \in \{1, 2, \dots, S\}$	Index of the S scales
$G(\mu, \sigma)$	Multi-dimensional Gaussian distribution
$\Phi(\vec{q}; s)$	Multi-resolution decomposition
$\Psi(\vec{q}; s)$	Pyramidal decomposition
Sampling pattern	
ρ	Set of displacement vectors
M	Cardinality of ρ
$\vec{\rho}_m \in \rho$	A generic displacement vector

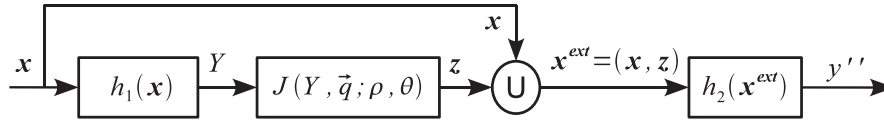


Fig. 1. Block diagram of the generalized stacked sequential learning.

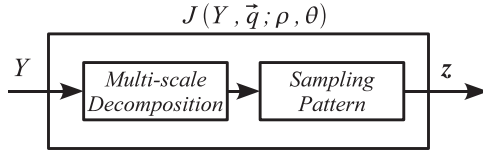


Fig. 2. Design of $J(Y, \tilde{q}; \rho, \theta)$ in two stages: a multi-scale decomposition followed by an appropriate sampling pattern.

The first classifier $h_1(\mathbf{x})$ assumes that the data is iid so that it does not need any spatial information regarding the samples. However, it is important that the predicted labels are arranged to their original spatial/temporal location, so that we define $\tilde{q} \in \mathcal{D} \subset \mathbb{N}^D$ as the spatial position in the compact support \mathcal{D} of \mathbb{N}^D , where N is the spatial dimensionality of the data (e.g. $D=2$ for images). In this way, $Y(\tilde{q})$ represents the value of the predicted label at location \tilde{q} and, in general, Y represents all the predicted labels values in \mathcal{D} .

The second block defines the policy for creating the neighborhood model of the predicted labels. $\mathbf{z} = J(Y, \tilde{q}; \rho, \theta) : \mathbb{R} \times \mathcal{D} \rightarrow \mathbb{R}^W$ is a functional that captures the data interaction with a model parameterized by θ in a neighborhood ρ . The result of this functional is a W -dimensional value, where W is the number of elements in the support lattice of the neighborhood ρ . In the case of defining the neighborhood by means of a window, W is the number of elements in the window. Then, the output of $J(Y, \tilde{q}; \rho, \theta)$ is joined with the original training data creating the extended training set $(\mathbf{x}^{\text{ext}}, y) = ((\mathbf{x}, \mathbf{z}), y)$. This new set is used to train a second classifier $h_2(\mathbf{x}^{\text{ext}})$ with the goal of producing the final prediction y'' . Observe, that the system will be able to deal with neighboring relations depending on how well one is able to characterize them in $J(Y, \tilde{q}; \rho, \theta)$.

In this paper we propose to design $J(Y, \tilde{q}; \rho, \theta)$ in a two stage way: (1) we decompose the output of the classifier $h_1(\mathbf{x})$ according to a multi-scale decomposition and (2) the resulting decomposition is appropriately sampled to create the extended set \mathbf{z} . Fig. 2 shows the two stages composing J .

In next subsections we explain how to obtain the multi-scale decomposition of the label field by means of a *multi-resolution decomposition* and a *pyramidal decomposition*. Then, appropriate sampling patterns are presented for the two types of multi-scale decompositions. Finally, we discuss advantages and disadvantages of each decomposition method. A discussion on how the sampling schema influences the long-range interaction concludes the section.

3.2. Multi-scale decomposition

We propose two ways to decompose the label field result of the outputs of the first classifier $h_1(\mathbf{x})$. A standard multi-resolution (MR-SSL) decomposition and a pyramidal decomposition (Pyr-SSL). To clarify the method, Fig. 3 shows an example in which a label field, resulting from an image classification algorithm, is decomposed and sampled.

3.2.1. Multi-resolution decomposition

The multi-resolution decomposition directly derives from classical multi-resolution theory in image processing and analysis. Given $Y_{C_i}(\tilde{q})$, the probability, likelihood, or the margin of class C_i at position

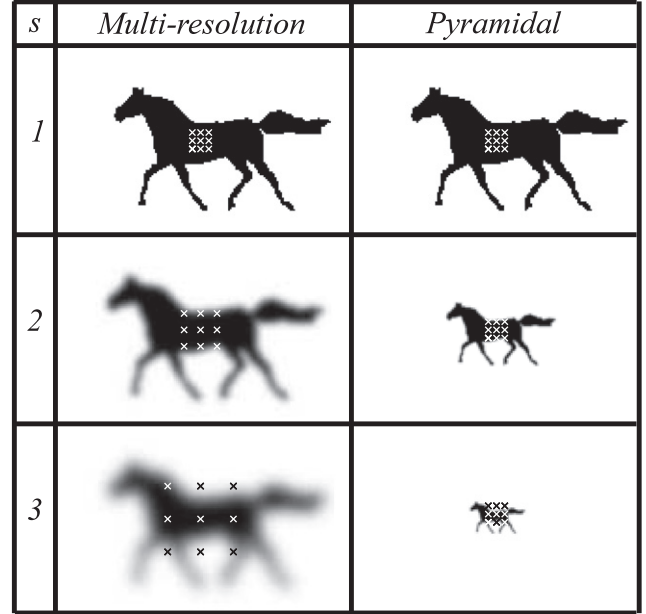


Fig. 3. Two examples of multi-scale decomposition and a possible sampling pattern for both. White and black crosses denote the sampling positions.

\tilde{q} ; we define the multi-resolution decomposition Φ , as follows:

$$\Phi_{C_i}(\tilde{q}; s) = Y_{C_i}(\tilde{q}) * G(0, \gamma^{s-1}), \quad (1)$$

where $s \in \{1, 2, \dots, S\}$ represents the scale; $*$ is the convolution operator, G is a multidimensional Gaussian function with zero mean and $\sigma = \gamma^{s-1}$. Here γ is the “step” of the multi-resolution decomposition (typically $\gamma = 2$). As it can be seen in Fig. 3, this methodology, applied on label fields coming from image pixel classification, is mimicking exactly the well-known multi-scale methodology used in image processing and analysis techniques. However, here the images represent the probability, likelihood or margin of a certain class. As a result, the multi-resolution decomposition provides information regarding the spatial homogeneity and regularity of the label field at different scales. It is easy to understand that, for example, a noisy classification at scale 1 does not importantly influence the results of scale 3. In this way, the coarser scale robustly represents the label field in presence of noisy classification (reaching the limit of an almost homogeneous label field) and, at the same time, intermediate scales give different levels of details of the initial label field.

3.2.2. Pyramidal decomposition

An alternative is provided by the pyramidal decomposition [3]. The pyramidal decomposition is substantially similar to the multi-resolution decomposition with the exception that the resulting pyramid codes more efficiently the multi-scale information. However, it has an important drawback that will be discussed in next subsections.

Starting from the above mentioned multi-resolution decomposition, the pyramidal decomposition Ψ can be obtained as follows:

$$\Psi_{C_i}(\tilde{q}; s) = \Phi_{C_i}(\lfloor k_s s \tilde{q} \rfloor; s), \quad (2)$$

where $\lfloor \cdot \rfloor$ is the floor function, $\tilde{q} \in \mathbb{N}^D$ is a position vector, D is the dimensionality of the data. Here $\tilde{q}_j \in [1, (X_j/\gamma^{s-1})]$, where X_j is the

integer size of every dimension j (for an image, $D=2$, X_1 and X_2 are, respectively, the width and height of the image). Here, k_s is the sampling step and depends on γ , $k_s = \gamma^s/2$. Actually, the pyramidal decomposition theoretically samples the multi-resolution without loss of information, since at coarser scales, the high frequency content has been progressively filtered out.

3.3. Sampling pattern

Once the desired multi-scale representation has been computed, an appropriate sampling pattern should be applied. This pattern can be represented by a set of displacement vectors that defines the neighborhood $\rho = \bigcup_{m=1}^M \delta_m^s$. Once the displacement vectors are defined, the feature vector for the multi-resolution decomposition is obtained by the following formula:

$$\mathbf{z}(\vec{p}) = \underbrace{\{\Phi(\vec{p} + \vec{\delta}_1^s; 1), \Phi(\vec{p} + \vec{\delta}_2^s; 1), \dots, \Phi(\vec{p} + \vec{\delta}_M^s; 1),\}}_{\text{scale } s=1} \underbrace{\{\Phi(\vec{p} + \gamma \vec{\delta}_1^s; 2), \Phi(\vec{p} + \gamma \vec{\delta}_2^s; 2), \dots, \Phi(\vec{p} + \gamma \vec{\delta}_M^s; 2),\}}_{\text{scale } s=2} \dots \underbrace{\{\Phi(\vec{p} + \gamma^{(S-1)} \vec{\delta}_1^s; S), \Phi(\vec{p} + \gamma^{(S-1)} \vec{\delta}_2^s; S), \dots, \Phi(\vec{p} + \gamma^{(S-1)} \vec{\delta}_M^s; S),\}}_{\text{scale } s=S}. \quad (3)$$

This formula shows that the sampling is performed following the displacement vectors at each scale s . However, the displacements at different scales are multiplied by a factor $\gamma^{(s-1)}$ so that, at coarser scales corresponds larger displacement. For the sake of clarity, the sampling in Fig. 3 (left) is obtained with $S=3$, $\gamma=2$, $M=9$ and the following set of displacements:

$$\rho = \{\vec{\delta}_1^s = (-1, -1), \vec{\delta}_2^s = (-1, 0), \vec{\delta}_3^s = (-1, 1), \vec{\delta}_4^s = (0, -1), \vec{\delta}_5^s = (0, 0), \vec{\delta}_6^s = (0, 1), \vec{\delta}_7^s = (1, -1), \vec{\delta}_8^s = (1, 0), \vec{\delta}_9^s = (1, 1)\}. \quad (4)$$

This displacement set can be represented graphically as in Fig. 4.

The feature vector for the pyramidal decomposition can be obtained by the following formula:

$$\mathbf{z}(\vec{p}) = \underbrace{\{\Psi(\vec{p} + \vec{\delta}_1^s; 1), \dots, \Psi(\vec{p} + \vec{\delta}_M^s; 1),\}}_{\text{scale } s=1} \underbrace{\{\Psi(\lfloor \vec{p}/\gamma \rfloor + \vec{\delta}_1^s; 2), \dots, \Psi(\lfloor \vec{p}/\gamma \rfloor + \vec{\delta}_M^s; 2),\}}_{\text{scale } s=2} \dots \underbrace{\{\Psi(\lfloor \vec{p}/\gamma^{(S-1)} \rfloor + \vec{\delta}_1^s; S), \dots, \Psi(\lfloor \vec{p}/\gamma^{(S-1)} \rfloor + \vec{\delta}_M^s; S),\}}_{\text{scale } s=S}, \quad (5)$$

where $\lfloor \cdot \rfloor$ is the floor function. As in the previous case, the sampling is performed over all the displacements and scales. On

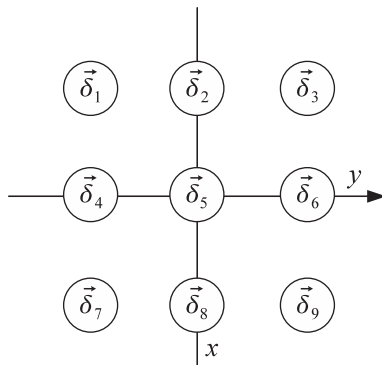


Fig. 4. A graphical representation of the displacements set ρ as defined in formula (4).

the other hand, the position vector \vec{p} is divided by the quantity $\gamma^{(s-1)}$ to adequately re-scale the coordinates to the resized images at coarser scales. The floor function is needed to obtain an integer vector that lay in the image lattice. The displacement pattern ρ is not modified as the images are progressively smaller at coarser scales. Fig. 3 (right) shows an example of this sampling with $S=3$, $\gamma=2$, $M=9$ and the same displacement set ρ as in formula (4).

3.4. Pros and disadvantages of multi-resolution and pyramidal decompositions

The multi-resolution approach is the most appropriate in terms of signal processing theory. However, the pyramidal decomposition actually contains the same information as the multi-resolution one while coding it in a more compact way. Unfortunately, as it can be noticed in Eq. (5), the sampling at large scales is prone to produce blocking artifacts. This is due to the fact that during the pyramidal decomposition process, each scale summarizes the information of the above area in a block that is γ^D times smaller. Obviously, at coarser scales this reflects into sharp transitions from one value to another in the feature vector. This does not happen using the multi-resolution decomposition, where the Gaussian filtering assures smooth transitions at every scale.

Summarizing, if the amount of input data is sufficiently small, the use of the multi-resolution decomposition is highly recommended. However, if the amount of input data is large, the pyramidal decomposition can help to save memory at the cost of possible blocking artifacts. To avoid blocking artifacts, an interpolation technique could be used.

3.5. The coverage-resolution trade-off

If we look carefully at the design of J , we can observe that for a fixed size of the extended set, the sampling policy defines whether we focus on nearby or far away samples. Note that the higher the number of scales is, the longer the range of interactions is considered. This feature allows to capture long-distance interactions with a very small set of features while keeping a relatively good short distance resolution. In order to quantify this effect we define the coverage of the method as the maximum effective range in which two samples affect each other. Similarly we can define the detail as the average detail size considering all the scales in the multi-resolution sampling scheme.

If we restrict to grid square samplings, the sampling scheme can be expressed as the set of displacements obtained by $\vec{\delta} = (k \cdot i, k \cdot j)$ where $i, j \in \{1, 0, -1\}$ and $k = 1, 2, \dots, r$. This defines a square grid of size $2r+1$ centered at the sample of interest as the one shown in Fig. 4. The value of r plays an important role in the scheme since it allows to govern the average detail of the approximation. In this setup the coverage can be computed as $c = \gamma^{(S-1)}(2r+1)^d$, the number of features generated by the proposed approach is $f = S(2r+1)^d$ and detail is computed as the average value of the relative distance between adjacent points at scale i , given by γ^{i-1} . Thus, $d = (1/S) \sum_{i=0}^{S-1} \gamma^i = (\gamma^S - 1)/(\gamma - 1)$.

Fig. 5 shows the relationship among detail, coverage and the number of features. Fig. 5(a) plots the number of features needed for observing a certain number of predicted labels (coverage). Different curves show the effect of altering the size of the support window r . Thus, in a two-dimensional sequential domain if the support window has a size of 3, $r=1$, we need seven scales and a total of 63 features to capture information from about 600 labels. Parameter r governs the trade-off between resolution and coverage. Observe in Fig. 5(b) that average detail is coarser as the coverage increase. Thus, small patterns in long-distance label interactions are lost. As r increases, the number of features also

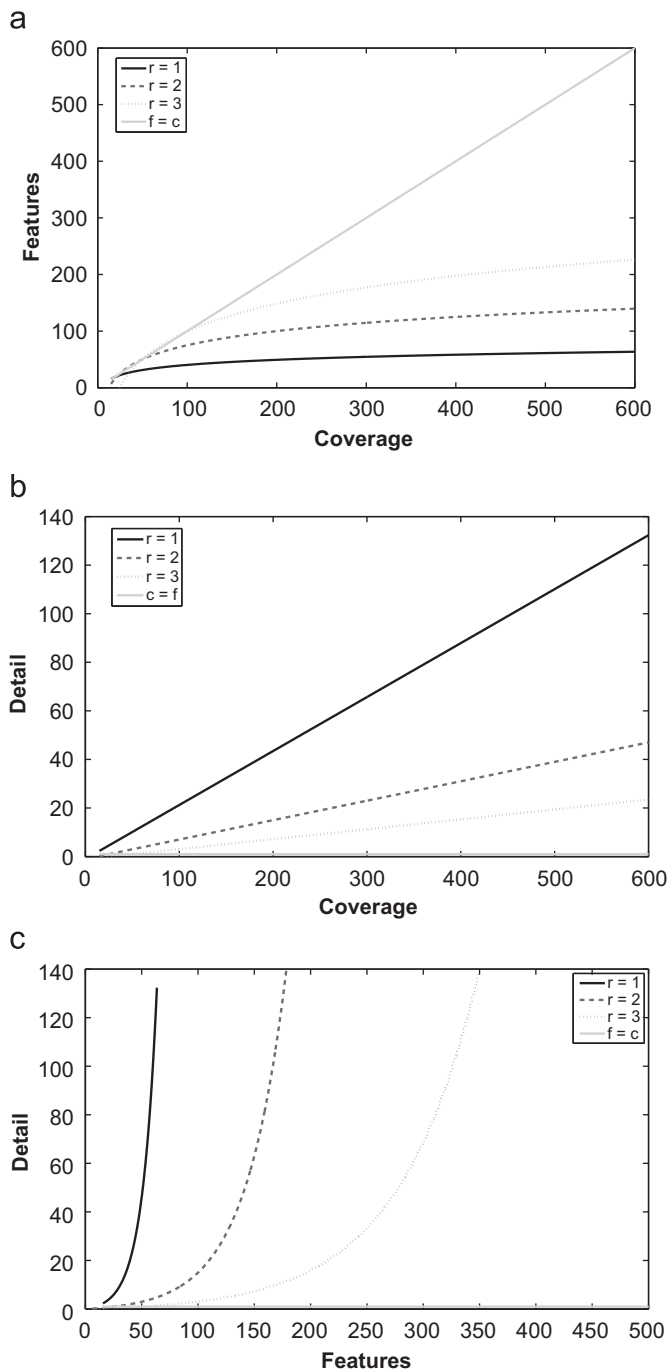


Fig. 5. (a) number of features needed for covering a certain number of predicted values for different window sizes r , (b) detail with respect to the coverage, (c) number of features needed to consider a certain detail value.

increases (Fig. 5(c)), but more complex and detailed label patterns can be captured. On the contrary, maximum interaction details are observed at the cost of using the same number of features as the coverage value. This trade-off allows the practitioner to consider different strategies according to the degree of sequential correlation expected in the sequence.

4. Experiments and results

In order to validate the proposed techniques, MR-SSL and Pyr-SSL are applied and compared to state-of-the-art strategies in two

different scenarios. The first scenario considers one-dimensional correlations in the label field in a text categorization task. The second experiment concerns the image domain, where correlations are found on a two-dimensional support lattice.

4.1. Categorization of FAQ documents

- **Data set:** The FAQ categorization task has been frequently used in literature as a benchmark for sequential learners [10,15]. In this data set, three different computer science FAQ groups pages are used (ai-neural-nets, ai-general, aix). Each FAQ group consists of 5 to 7 long sequences of lines; each sequence corresponding to a single FAQ document. Each line is characterized using McCallun et al. features [23], with 24 attributes that describe line characteristics with the respective class label. In total, each FAQ group contains between 8965 and 12 757 labeled lines. This data set is multi-class, with four possible classes; in our experiments, for each of the three groups we split the multi-class problem into two binary problems considering the following labels “answer” vs “not answer”, and “tail” vs “not tail”, yielding a total of six different problems.
- **Methods:** We compare the pyramidal and multiresolution approaches with standard Adaboost with decision stumps, conditional random fields, and the original stacked sequential learning strategy.
- **Experimental and parameters settings:** The base classifier for SSL, Pyr-SSL and MR-SSL is Adaboost with a maximum of 100 decision stumps. All stacked learning techniques use an inner 5-fold cross-validation on the training set for the first step of the sequential learning schema.
- **Evaluation metrics:** Due to the fact that each sequence must be evaluated as a whole set, and that there is a small amount of sequences per problem, one of the fairest ways for comparing the results is to average the accuracy using a leave-one-sequence-out cross-validation scheme – one sequence is used as testing and the rest of the sequences are joined into one training sequence – for each problem. Different configurations according to the (γ, S, ρ) parameterization are compared. The average rank for each method is also provided.¹
- **Statistical analysis:** In order to guarantee that the results convey statistically relevant information, a statistical analysis is performed for each experiment. First, an Iman’s and Davenport correction of the Friedman’s test is performed to ensure that the differences in the results are not due randomness with respect to the average performance rank. The statistic for this test is distributed according to a F-distribution with $k-1$ and $(N-1)(k-1)$ degrees of freedom, where k is the number of methods compared and N the number of data sets. The statistic is computed as follows:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2},$$

where χ_F^2 is the Friedman’s statistic given by

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j - \frac{k(k+1)^2}{4} \right],$$

where R_j is the average rank of each method.

If the null hypothesis is rejected we can ensure that the resulting ranks convey significantly relevant information. Then, a post-hoc test using Nemenyi’s test can be performed

¹ The average rank accounts for the sum of the ranking position of each method for each database.

Table 2

Average percentage error and methods ranking for different FAQ data sets, different methods; and different parameterization of SSL, Pyr-SSL and MR-SSL. For the sake of table compactness, the following definitions should be considered: $\rho_3 = \{-3, -2, \dots, 2, 3\}$, $\rho_6 = \{-6, -5, \dots, 5, 6\}$, $\rho_1 = \{-1, 0, 1\}$.

	AdaBoost	CRF	SSL		Pyr-SSL		MR-SSL	
(γ, S)	–	–	(–,1)	(–,1)	(2,2)	(4,4)	(2,2)	(4,4)
ρ	–	–	ρ_3	ρ_6	ρ_1	ρ_1	ρ_1	ρ_1
Features	–	–	7	13	6	12	6	12
Coverage	–	–	7	13	12	192	12	192
Neural-netsA	7.0675	7.5812	5.9855	5.8103	6.1495	4.5257	5.4504	3.4667
Neural-netsT	1.8067	2.0831	1.4826	0.7825	1.3146	0.5199	0.2834	0.4065
Ai-GeneralA	8.2764	9.3902	9.2944	10.3183	9.2636	10.3834	9.8923	9.1097
Ai-GeneralT	1.8916	2.4267	1.6275	0.9392	1.7031	1.1964	1.4219	0.0001
Ai-AixA	9.7971	12.5310	9.3519	9.9028	9.3307	9.5689	8.9304	9.7452
Ai-AixT	1.2553	1.5741	0.8966	0.7493	0.9233	0.2662	0.4257	0.0001
Rank	5.67	7.34	4.67	4.67	4.34	3.67	3	2

Table 3

Average percentage error for different configurations of Pyr-SSL and MR-SSL. The last two rows show the average rank for each parameterization as well as the average rank for each of the multi-scale families.

	Pyr-SSL				MR-SSL			
(γ, S)	(2,2)	(2,3)	(2,4)	(4,4)	(2,2)	(2,3)	(2,4)	(4,4)
ρ	ρ_1	ρ_1	ρ_1	ρ_1	ρ_1	ρ_1	ρ_1	ρ_1
Features	6	9	12	12	6	9	12	12
Coverage	12	24	48	192	12	24	48	192
Neural-netsA	6.1495	5.624	4.9195	4.5257	5.4504	4.6303	4.0018	3.4667
Neural-netsT	1.3146	0.7159	0.5257	0.5199	0.2834	0.2499	0.2210	0.4065
Ai-GeneralA	9.2636	9.1998	9.0374	10.3834	9.8923	9.3698	8.8874	9.1097
Ai-GeneralT	1.7031	1.2716	0.6582	1.1964	1.4219	0.7136	0.6407	0.0001
Ai-AixA	9.3307	9.3412	9.1311	9.5689	8.9304	9.1276	9.2907	9.7452
Ai-AixT	0.9233	0.4754	0.2888	0.2662	0.4257	0.0001	0.0806	0.0001
Rank	7	6.17	4	5.33	5	3.25	2.16	3.08
Avg rank			5.63				3.37	

in order to single out methods or groups of methods. Using this statistical test, two sets are statistically different if the difference of ranks is higher than a given critical value computed as follows:

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}},$$

where q_α is based on the studentized base statistic divided by $\sqrt{2}$.

Table 2 shows the results obtained for the FAQ experiments comparing the base algorithm Adaboost with decision stumps, conditional random fields, two configurations of SSL, two configurations of Pyr-SSL and two of MR-SSL. In order to ensure a fair comparison, the configurations have similar number of features in the extended training set. The average performance rank of each method is displayed in the last row of the table. The best performance for each data set is highlighted in bold font.

Observe that sequential learning approaches generally reduce the error rate percentage except for the AI-GeneralA data set. Moreover, as the number of features increases the accuracy improves. Comparing SSL with Pyr-SSL and MR-SSL approaches, it is worth noting that using a similar number of features, the multi-scale counterparts achieve much better results. This seems to suggest that the data sets include some structure information that can only be captured at large scales. This idea is reinforced by the fact that for some of the data sets, the larger the coverage, the lower the error rate achieved. Finally, when the multi-scale step γ is doubled – thus quadrupling the coverage by sacrificing resolution in the data sequence – we can see that applying Pyr-SSL

accuracy improves on half of the data sets. On the other hand, when applying MR-SSL accuracy improves in all except for one data set.

Observing the ranks for all proposed methods, it can be noticed that Pyr-SSL and MR-SSL strategies perform better than Adaboost, CRF and SSL. Considering the different configurations of the MSSL technique, Pyr-SSL performs poorer than MR-SSL. A detailed analysis of this effect is shown in Table 3.

A statistical analysis of the results shows that, in our case, Iman's and Davenport correction of the Friedman's test yields $F_F=2.9329$ and the critical value for this test is $F(7,35)=2.249$, so we reject the null hypothesis that the results may be due to randomness with respect to the average rank. Thus we can safely say that the analysis of ranks convey statistically relevant information.

A post-hoc analysis of the data using Nemenyi's test² singles out MR-SSL (4,4) from SSL, CRF and AdaBoost with a 95% confidence. Additionally, all proposed strategies are statistically different than CRF and Adaboost at 90% confidence. On the other hand, SSL shows statistically significant differences with respect to CRF, as reported in the original SSL paper but fails to display any statistically significant difference with respect to Adaboost.

Table 3 shows a detailed comparison between Pyr-SSL and MR-SSL using different parameter configurations. Average ranks for each configuration are provided. Additionally, a single average rank for Pyr-MSSL and MR-SSL is given. The parameters are

² Critical difference using Nemenyi's test at 0.05 is 2.44, and at 0.10 is 2.05.

chosen such that the number of features and coverages are comparable.

Using Iman's and Davenport statistic $F_F=6.0689$, with a critical value for $F(7,35)=2.249$, so we reject the null hypothesis that the results may be due to randomness with respect to the average rank. A post-hoc test using Nemenyi's test critical difference with 95% of confidence shows that MR-SSL always statistically outperforms Pyr-SSL when we compare the same parameter configurations.

4.2. Weizmann horse database

- **Data set:** The Weizmann horse database is composed of 328 side-view color images of horses and their respective manual segmentations. This database has been proposed to evaluate the performance of a segmentation algorithm [5]. In the database, the horses exhibit a sufficiently regular structure; i.e. all are standing and towards left. They vary significantly in color and size.
- **Methods:** We compared the proposed methods against AdaBoost, CRF and standard SSL (using a window of size 7×7). For the following experiments, the CRF implementation is a modified version of the one in [26]. We incorporated AdaBoost to generate the unary potential. In this way, the base classifier for all the SSL strategies is the same as the one used by CRF. For the learning phase we used the stochastic gradient descent as it proved to be one of the fastest ways to train a CRF [26]. Finally, inference is performed by means of the belief propagation method.
- **Experimental and parameters settings:** Two different experiments are performed using this data set:
 - First, in order to perform a fair comparison to the method in [5], we resize all the images to 40×30 pixel size. This problem will also serve to establish proper comparisons with computationally intensive methods such as CRF. A good way to show the capabilities of the proposed method, is to reduce the number of features that the first classifier can use to discriminate the horse class. In this way, the behavior of the proposed method will be easier to describe. We actually reduce the feature vector to pixel-wise color. The images are transformed from sRGB to CIELAB color space to highlight lightness and chromatic components. The first classifier ($h_1(\mathbf{x})$) only has access to the three CIELAB color coordinates for each pixel in the image. Observe that this classification can be prone to several errors due to the similarity between horses color and background color. Moreover, since the classification is performed pixel-wise, the first classifier is prone to isolated misclassifications. As in the previous experiment, the base classifier is an AdaBoost with decision stumps reaching a maximum of 100 iterations. Regarding the MR-SSL, to have an effective long-range label interaction, we set the number of scales $S=5$, $\gamma=2$ and ρ as in formula (4), so that the maximum displacement during sampling is of $\pm \gamma^{(S-1)} = \pm 2^4 = \pm 16$ pixel, thus covering great part of the image size.
 - In order to show that the proposed method also works well with bigger images, the other experiment uses full size images. In this case results are compared with segmentation state-of-the-art methods. To deal with full size images, we use the SIFT descriptor for the first classifier (using the CIELAB color space) because texture is an important feature when the images have a sufficient resolution. Moreover, we set the number of scales $S=7$, $\gamma=2$ and ρ as in formula (4), obtaining a coverage value of ± 64 pixels.

- **Evaluation metrics:** Due to the significative amount of data a 5-fold cross-validation technique is used for obtaining the average evaluation metrics. As done in [5], we propose to use the Jaccard index [18] as a quality measure of the overlapping o between the automatic segmentation and the labeled ground truth; being A and M , respectively, the automatic and manual (ground truth) segmentations, the overlapping is defined as $o = |A \cap M| / |A \cup M|$. This measure is equivalent to the ratio of the true positive over the sum of true positive, false positive and false negatives.
- **Statistical analysis:** The amount of data in these experiments enable the possibility of using powerful statistical tests that were not available in the former experiment. In this case, we test for statistical significance of the results using Wilcoxon signed rank test and comparing the p -values obtained across the different methods, as suggested in [13].

4.2.1. Results on the resized Weizmann horse database

Fig. 6 shows comparisons in terms of accuracy, precision and the overlapping measure o . In the first row, the plots represent, respectively, the accuracy, precision and overlapping of AdaBoost and the proposed MR-SSL method. Each dot in the plot represents the performance obtained on a specific image for both algorithms. In the plot, a distribution that is mainly above the diagonal means that MR-SSL performs better than AdaBoost. The same applies for the second row, where MR-SSL is compared to CRF. It is clear that MR-SSL outperforms both AdaBoost and CRF. Last row shows a comparison between MR-SSL and Pyr-SSL, confirming that the former performs better than the latter. Table 4 shows the average accuracy, precision and overlapping values for all the tested configurations, with the respective standard deviation. Observe that both Pyr-SSL and MR-SSL achieve much better results than standard SSL, CRF and AdaBoost. Moreover, MR-SSL shows improvements over Pyr-SSL.

Tables 5 and 6 show the p -values applying the Wilcoxon signed rank test on the values of accuracy and overlapping for all the images of the horse data set. Values followed by \circ display those results that are not statistically significant. Values followed by \bullet are statistically significant at 10%. In Table 5 we observe that the proposed methodologies as well as the standard SSL show results statistically significant with respect to Adaboost and CRF in terms of accuracy. Furthermore, the multi-scale strategies proposed statistically differ from SSL. And, in particular, MR-SSL is the most statistically different from all techniques. In Table 6 we see that MR-SSL is again the most statistically different from the rest in terms of overlapping. In general, the conclusions are very similar to those obtained in the accuracy table. It is worth noting that SSL and Pyr-SSL are not statistically significant. And, Pyr-SSL only achieve statistically significant results at 10% when compared with Adaboost.

It is also interesting to compare the training and inference time of both MR-SSL and CRF. The CRF implementation used in this work is the one in [26], written in Matlab but with many optimized functions in C. The MR-SSL has been implemented in Matlab without specific optimization. To train one fold (using about 260 images), the CRF requires 52' 6" while the MR-SSL only 1' 10". Regarding inference, for each image CRF requires and average time of 0.4846 s, while the MR-SSL just 0.0882 s. We also tested the training time of the MR-SSL with full size images (using $S=7$), resulting in a training time of 14' 35" for one fold. The same test performed with CRF is unfeasible since training the method on each fold requires several days.

Finally, it is interesting to note that in Ref. [5] they achieve an average $o=0.71$ using a sophisticated segmentation algorithm on 40×30 image size, requiring about 40 second per image to

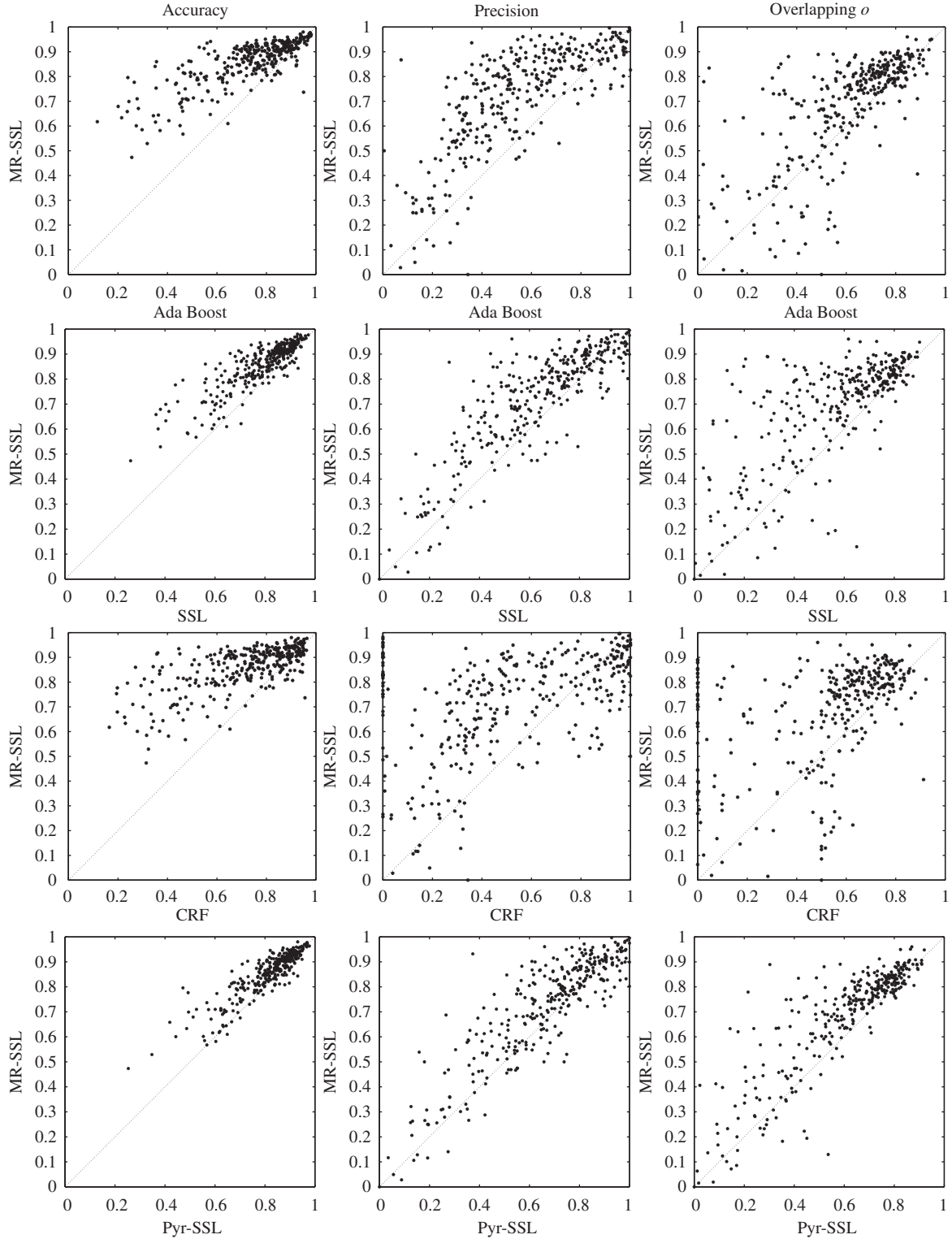


Fig. 6. Comparison of the proposed MR-SSL method to AdaBoost (first row), SSL using a window of size 7×7 (second row), CRF (third row) and Pyr-SSL (last row).

perform the segmentation, using a leave-one-out testing. Using the proposed MR-SSL, and just with color features, we achieved an average $o=0.682$.

However, one can argue that the advantage of the proposed method is relevant only if the first classifier performs poorly. To controvert this hypothesis, we performed another test, in which the feature vector \mathbf{x} is augmented with 27 Haar-like features. As it can be seen in Table 7, the performance of the base AdaBoost

classifier is increased (+0.033 accuracy, +0.017 precision, +0.034 overlapping); but at the same time the MR-SSL performs better (+0.012 accuracy, +0.018 precision, +0.033 overlapping). The Haar-like features helped the first classifier and this improvement has reflected entirely on the MR-SSL performance; this shows that the improvement due to the MR-SSL schema over the base classifier is still significant (+0.11 accuracy, +0.16 precision, +0.07 overlapping). Finally, it is worth to note that

Table 4

The average performance of AdaBoost, SSL (7×7 window size), MR-SSL, CRF and Pyr-SSL in terms of accuracy, precision and overlapping. Standard deviations are in brackets.

	Accuracy	Precision	Overlapping ϕ
AdaBoost	0.7322 (0.1808)	0.5555 (0.2398)	0.6112 (0.2064)
CRF	0.7195 (0.1946)	0.5174 (0.3257)	0.5137 (0.2641)
SSL 7×7	0.7915 (0.1285)	0.6327 (0.2308)	0.5584 (0.2264)
Pyr-SSL	0.8196 (0.1158)	0.6664 (0.2250)	0.6030 (0.2195)
MR-SSL	0.8592 (0.0903)	0.7191 (0.2091)	0.6819 (0.2109)

Table 5

Wilcoxon paired signed rank test p -values for the results of the accuracy measure.

Accuracy p -val	ADA	SSL	Pyr-SSL	MRSSL	CRF
ADA	–	0.0001	0.0000	0.0000	0.6847 \circ
SSL	0.0001	–	0.0032	0.0000	0.0001
PyrSSL	0.0000	0.0032	–	0.0000	0.0000
MRSSL	0.0000	0.0000	0.0000	–	0.0000
CRF	0.6847 \circ	0.0001	0.0000	0.0000	–

Table 6

Wilcoxon paired signed rank test p -values for the results of the overlapping measure.

Overlapping p -val	ADA	SSL	Pyr-SSL	MRSSL	CRF
ADA	–	0.0048	0.0691 \bullet	0.0000	0.0000
SSL	0.0048	–	0.3082 \circ	0.0000	0.0426
PyrSSL	0.0691 \bullet	0.3082 \circ	–	0.0000	0.0031
MRSSL	0.0000	0.0000	0.0000	–	0.0000
CRF	0.0000	0.0426	0.0031	0.0000	–

Table 7

The average performance of AdaBoost and MR-SSL in terms of accuracy, precision and overlapping; adding 27 Haar-like features to the first feature vector \mathbf{x} . Standard deviations are in brackets.

	Accuracy	Precision	Overlapping ϕ
AdaBoost-Haar-27	0.7651 (0.1468)	0.5727 (0.1960)	0.6451 (0.1699)
MR-SSL-Haar-27	0.8716 (0.0745)	0.7379 (0.1659)	0.7147 (0.1846)

Table 8

The average performance of AdaBoost and MR-SSL in terms of accuracy, precision and overlapping using the color plus SIFT descriptor as the first feature vector \mathbf{x} . Standard deviations are in brackets.

	Accuracy	Precision	Overlapping ϕ
AdaBoost-SIFT	0.8123 (0.1122)	0.6286 (0.1818)	0.7041 (0.1302)
MR-SSL-SIFT	0.9209 (0.03)	0.821 (0.13)	0.7466 (0.16)

the proposed MR-SSL in this case achieves an average overlapping ($\phi=0.7147$) that is comparable with the one reported in [5].

4.2.2. Results on the full size Weizmann horse database

Finally, to compare our method to state-of-the-art *segmentation* methods, we used the MR-SSL together with the SIFT descriptor as above detailed. Table 8 shows the results in term of accuracy, precision and overlapping of the Adaboost-SIFT and the MR-SSL-SIFT on full size images of the Weizmann data set. Several segmentation methods have used the Weizmann data set to evaluate their performances: in [16] authors report an average accuracy of 91.47%, the method in [4] reports 93%, method in [1]

reports 93% on 200 horse images, the method in [22] reports an accuracy of 95%, finally the method in [19] reports an accuracy of 96% and precision of 89%. As depicted in Table 8 our method shows an average accuracy of 92.1% and an average precision of 82.1% using all the images of the data set. Fig. 7 shows the best 30 segmentations on full size images, which overlap ranges from 0.92 to 0.85; first column shows the input image, second column the ground truth, third column the classification result and the last column shows a color coded image with true positives (blue), true negatives (white), false positives (cyan) and false negatives (red).

To clearly show the contribution of the extended set to the final classification performance, in Fig. 8 we show the weights Adaboost assigned, respectively, to the (a) pixel-wise color features, (b) textural SIFT descriptor and (c) contextual features. The total weight of the pixel-wise color feature is 1.06 (10.2 %), the total weight of SIFT features is 2.4 (23.2 %) and the total weight of contextual features is 6.89 (66.6 %). This experiment confirms two important facts: (1) the original feature vector \mathbf{x} , in this case the color and SIFT features, contributes to the final classification, giving the second classifier the possibility to exploit correlations between appearance features and contextual features; (2) the contextual features are providing relevant information, as confirmed by the fact that Adaboost assigns the 66.6 % of the weights to these features. Further comments must be devoted to the plot in Fig. 8(c). The second classifier gives a large weight (23.5%) to the central location at the second scale and a very low weight (0.024 %) at the central location at the first scale. This means that the second classifier does not “trust” pixel-wise classifications of the first stage classifier (first scale), but it is much more prone to “trust” the classification if averaged in a small neighborhood (second scale). Moreover, the central locations at scales 6 and 7 have, respectively, the 7% and 4.6%; this means that, at a certain extent, the classifier consider the information over large areas useful, probably due to the fact that the horses’ bodies tend to cover quite a large area. Finally, at central scales, from 3th to 5th, central locations have very low weights while neighborhoods at almost all directions have more consistent weights. This means that the second classifier correctly uses the mid-range interaction between horse and background classes.

5. Discussion

In this section, we discuss in depth some of the results obtained in the experiments. Without loss of generality, we focus this discussion on the Weizmann database because of the easiness of illustrating classification results in the form of images. However, the same conclusions and observations can be drawn for the FAQ data sets.

First of all, it is interesting to note that CRF works especially well when the base classifier is able to perform a quite good classification. In these cases, CRF is able to distinguish and remove isolated misclassifications. Fig. 9 shows one example in which CRF slightly outperforms the proposed MR-SSL method. It is easy to notice that AdaBoost performed well, and the only contribution of the CRF is to remove some isolated misclassification and refine the silhouette of the horse.

When the classification performance of the base classifier is poorer, CRF is usually not able to improve the classification. On the contrary, the proposed MR-SSL method, thanks to its multi-scale approach, is able to improve the classification helping in discriminating ambiguous cases exploiting the contextual information. Fig. 10 shows several cases in which the base classifier performs poorly.



Fig. 7. The best 30 segmentations on full size images. First column shows the input image, second column shows the ground truth, third column shows the classification result and the last column shows a color coded image with true positives (blue), true negatives (white), false positives (cyan) and false negatives (red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

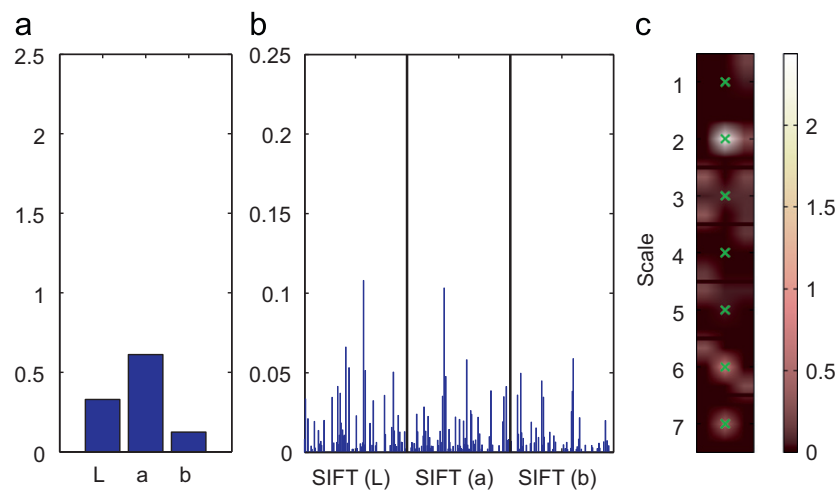


Fig. 8. The weights Adaboost assigned to the (a) pixel-wise color features, (b) textural SIFT descriptor and (c) contextual features. The total weight of the pixel-wise color feature is 1.06 (10.2%), the total weight of SIFT features is 2.4 (23.2%) and the total weight of contextual features is 6.89 (66.6%).







Input	Ground truth	$h_1(x)$	SSL 7×7	CRF	MR-SSL
					

Fig. 9. Input, ground truth and results of different methods on the test image number 142.





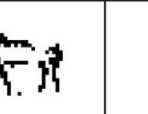































Input	Ground truth	$h_1(x)$	SSL 7×7	CRF	MR-SSL
					
					
					
					
					
					

Fig. 10. Input, ground truth and results of different methods on the test image numbers: 84, 22, 71, 88, 108 and 109, respectively.

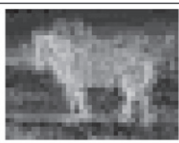



s	1	2	3	4
				

Fig. 11. Label field multi-resolution decomposition classifying image number 84.

The first row shows an example in which the CRF interprets the base classification result (the unary potential) as if it represents a sparse set of misclassifications. Within this interpretation, the best behavior CRF can exhibit is to remove the misclassifications, thus classifying no horse in the picture. Thanks to the multi-scale approach, the label field probability that results from the base classifier is interpreted within the multi-scale paradigm, so that at coarser scales the horse is roughly classified correctly. Fig. 11 shows the multi-resolution decomposition $\Phi_C(\vec{q}; s)$ obtained from the base classifier output, for the scales $s = \{1, 2, 3, 4\}$, when classifying the horse in the first row of Fig. 10.

As it can be noticed, at scale 1, the body of the horse has less probability to be classified correctly than the horse legs and head. However, at coarser scales, the horse outline is roughly visible

and the discrepancy in these probabilities is lower. This “blurred” information gives the necessary contextual information to the second classifier that is consequently able to correctly classify great part of the horse. Obviously, the quality of the classification is not perfect, but compared with the AdaBoost and CRF, the achieved result is clearly superior.

In Fig. 10, the second and third rows show two cases in which the AdaBoost classification is noisy. Here, the CRF removes all the one pixel size false positives. Unfortunately, close to the horse silhouette, it tends to fuse some false classifications (see row 2) or remove thin structures such as the horse legs (see row 3). On the other hand, MR-SSL removes many of the noisy false positives, but not all of them, while preserving small structures in both examples.

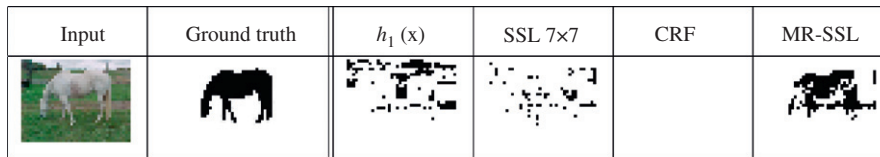


Fig. 12. Input, ground truth and results of different methods on the test image number 41.

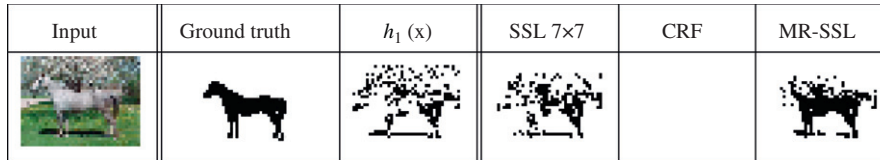


Fig. 13. Input, ground truth and results of different methods on the test image number 153.

Rows 4 and 5 show two examples in which the color of the ground and the fence easily cheats the first AdaBoost classifier, that incorrectly classifies all the ground and the fence as 'horse'. Here, SSL tries to reduce the false positives with limited results. CRF cannot represent the structure of the context and thus converges to blocks that fuse the ground and the horse. Finally, the MR-SSL outperforms the previous methods, removing great part of the false positives, thus segmenting the horse sufficiently well.

The last row of Fig. 10 shows an example in which the result of AdaBoost is already very good; in this case, the proposed method slightly refines the segmentation while the CRF removes the legs of the horse, as they are small structures.

In some of the examples we also found that the contextual information is able to almost invert the classification performed by the first classifier. Fig. 12 shows one of this cases. While the final result is surely not excellent, the ability of the proposed method to understand the context is evident.

Finally, Fig. 13 shows an example in which the MR-SSL algorithm is able to remove an important quantity of misclassifications while, at the same time, increasing the precision. As it can be noticed, the upper part of the picture background is very similar to the horse. For this picture, the overlapping o for AdaBoost is 0.3534 while for the MR-SSL is 0.7298. The precision is 0.4023 for AdaBoost, and 0.6811 for MR-SSL. This tremendous increase in the performance parameters clearly show the potential ability of the MR-SSL method to solve ambiguous classification cases. However, this behavior is not occasional, but it is evident as a general trend in the plots of Fig. 6.

6. Conclusions and future work

In this paper we generalized the stacked sequential learning approach, highlighting the key points of the extended feature vector creation. Within the generalized stacked sequential approach we proposed a multi-scale decomposition of the label field followed by an appropriate sampling schema to form an extended feature vector. The proposed method is able to capture long-range interactions in the label field efficaciously and efficiently. The resulting multi-scale sequential learning approaches are significantly faster than the fastest CRF implementation so far, both in training and inference, and its of easy implementation. The proposed method is highly modular, thus other more powerful classifiers can be used instead of AdaBoost. Finally, a clear and detailed explanation on how the practitioner can define the sampling schema with respect to the desired coverage/resolution trade-off is provided.

The proposed method has been tested on two different data sets, the first on a 1D correlation lattice and the second on a 2D correlation lattice. For the 1D case, the proposed method outperforms the CRF on five data sets over six. In the only case in which CRF outperforms the proposed MR-SSL, the best performance is given by the base AdaBoost classifier. For the 2D case, the proposed method outperforms both the base AdaBoost classifier and the CRF in terms of accuracy, precision and overlapping.

As a future perspective, we desire to extend the method to multi-class problems and to consider non-isotropic and/or non-linear decomposition methods.

Acknowledgments

This work has been supported in part by the projects: La Marató de TV3 082131, TIN2009-14404-C02, and CONSOLIDER-INGENIO CSD 2007-00018. The work of C. Gatta is supported by a Beatriu de Pinos Fellowship.

References

- [1] LOCUS: learning object classes with unsupervised segmentation, vol. 1, 2005.
- [2] K. Aas, L. Eikvil, R.B. Huseby, Applications of hidden Markov chains in image analysis, *Pattern Recognition* 32 (4) (1999) 703–713.
- [3] E.H. Adelson, C.H. Anderson, J.R. Bergen, P.J. Burt, J.M. Ogden, Pyramid methods in image processing, *RCA Engineer* 29 (6) (1984) 33–41.
- [4] E. Borenstein, J. Malik, Shape guided object segmentation, *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, IEEE Computer Society, Washington, DC, USA, 2006, pp. 969–976.
- [5] E. Borenstein, S. Ullman, Class-specific, top-down segmentation, *Proceedings of the European Conference on Computer Vision*, vol. 2351, Copenhagen, Denmark, 2002, pp. 109–124.
- [6] E. Borenstein, S. Ullman, Learning to segment, in: T. Pajdla, J. Matas (Eds.), *European Conference on Computer Vision* (3), Lecture Notes in Computer Science, vol. 3023, Springer, 2004, pp. 315–328.
- [7] L. Bottou, Y. Bengio, Y. Le Cun, Global training of document processing systems using graph transformer networks, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Juan, Puerto Rico, 17–19 June 1997, pp. 489–494.
- [8] P. Burt, E. Adelson, The Laplacian pyramid as a compact image code, *IEEE Trans. Commun.* 31 (4) (1983) 532–540.
- [9] J.H. Cai, Z.Q.A. Liu, Pattern recognition using Markov random field models, *Pattern Recognition* 35 (3) (2002) 725–733.
- [10] W.W. Cohen, V. Rocha de Carvalho, Stacked sequential learning, in: *Proceedings of the International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005, pp. 671–676.
- [11] T. Cour, J. Shi, Recognizing objects by piecing together the segmentation puzzle, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* IEEE Computer Society, Minneapolis, Minnesota, USA, 2007, pp. 1–8.

- [12] J.L. Davidson, N. Cressie, X. Hua, Texture synthesis and pattern recognition for partially ordered Markov models, *Pattern Recognition* 32 (9) (1999) 1475–1505.
- [13] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (2006) 1–30.
- [14] T.G. Dietterich, Machine learning for sequential data: a review, in: *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Springer-Verlag, London, UK, 2002, pp. 15–30.
- [15] T.G. Dietterich, T.D.A. Ashenfelder, Y. Bulatov, Training conditional random fields via gradient tree boosting, in: *Proceedings of the 21th International Conference on Machine Learning*, ACM, Banff, Alberta, Canada, 2004, pp. 217–224.
- [16] L. Gorelick, R. Basri, Shape based detection and top-down delineation using image segments, *Int. J. Comput. Vision* 83 (3) (2009) 211–232.
- [17] G. Heitz, D. Koller, Learning spatial context: using stuff to find things, in: *Proceedings of the 10th European Conference on Computer Vision*, Springer-Verlag, Marseille, France, 2008, pp. 30–43.
- [18] P. Jaccard, Distribution de la flore alpine dans le bassin des drouces et dans quelques régions voisines, *Bulletin del la Société Vaudoises des Sciences Naturelles* 37 (1901) 241–272.
- [19] M. Pawan Kumar, P.H.S. Torr, A. Zisserman, Obj cut, *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, IEEE Computer Society, Washington, DC, USA, 2005, pp. 18–25.
- [20] S. Kumar, M. Hebert, Discriminative random fields: a discriminative framework for contextual interaction in classification, in: *Proceedings of the 2003 IEEE International Conference on Computer Vision (ICCV '03)*, vol. 2, Nice, France, 2003, pp. 1150–1157.
- [21] J.D. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, in: *Proceedings of the International Conference on Machine Learning*, Williamstown, MA, USA, 2001, pp. 282–289.
- [22] A. Levin, Y. Weiss, Learning to combine bottom-up and top-down segmentation, *Int. J. Comput. Vision* 81 (2009) 105–118.
- [23] A. McCallum, D. Freitag, F.C.N. Pereira, Maximum entropy markov models for information extraction and segmentation, in: *Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2000, pp. 591–598.
- [24] X. He Richard, R.S. Zemel, M.Á. Carreira-perpiñán, Multiscale conditional random fields for image labeling, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2004, pp. 695–702.
- [25] A. Torralba, Contextual priming for object detection, *Int. J. Comput. Vision* 53 (2) (2003) 169–191.
- [26] S.V.N. Vishwanathan, N.N. Schraudolph, M.W. Schmidt, K.P. Murphy, Accelerated training of conditional random fields with stochastic gradient methods, in: *Proceedings of the 23rd International Conference on Machine Learning*, ACM, New York, NY, USA, 2006, pp. 969–976.
- [27] D.H. Wolpert, Stacked generalization, *Neural Networks* 5 (1992) 241–259.

Carlo Gatta obtained the degree in Electronic Engineering in 2001 from the Università degli Studi di Brescia (Italy). In 2006 he received the Ph.D. in Computer Science at the Università degli Studi di Milano (Italy) with a thesis on perceptually based color image processing. In September 2007 he joined the Computer Vision Center at Universitat Autònoma de Barcelona (UAB) as a postdoc researcher working mainly on medical imaging. He is member of the Computer Vision Center and the BCN Perceptual Computing Lab. His main research interests are image processing, medical imaging, computer vision and contextual learning.

Eloi Puertas obtained the degree in Informatics Engineering in 2002 from the Universitat Autònoma de Barcelona (UAB). In 2004, he joined the Applied Mathematics and Analysis Department at Universitat de Barcelona (UB). In 2007 he received the M.S. degree in Computer Science and Artificial Intelligence at the UAB on work in Machine Learning in MultiAgent Systems. He is member of the BCN Perceptual Computing Lab. Currently he is working in his Ph.D. thesis about ensemble learning, sequential learning and their application to object recognition and scene understanding.

Oriol Pujol obtained the degree in Telecommunications Engineering in 1998 from the Universitat Politècnica de Catalunya (UPC). The same year, he joined the Computer Vision Center and the Computer Science Department at Universitat Autònoma de Barcelona (UAB). In 2004 he received the Ph.D. in Computer Science at the UAB on work in deformable models, fusion of supervised and unsupervised learning and intravascular ultrasound image analysis. In 2005 he joined the Department of Matemàtica Aplicada i Anàlisi at Universitat de Barcelona (UB) where he became associate professor. He is member of the BCN Perceptual Computing Lab and the Computer Vision Center. He has been since 2004 an active member in the organization of scientific activities related to image analysis, computer vision, machine learning and artificial intelligence.